

KUKA

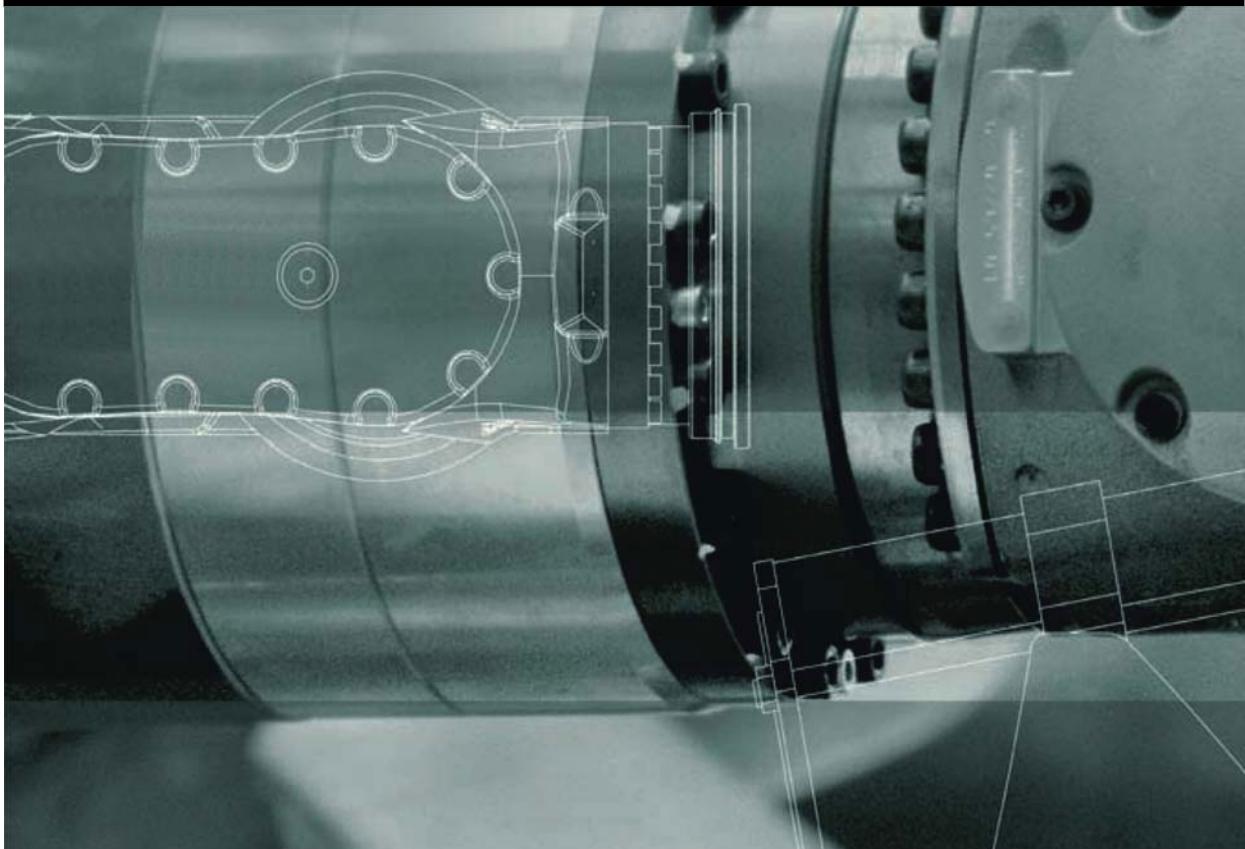
System Software

KUKA Laboratories GmbH

KUKA Sunrise.OS 1.5 KUKA Sunrise.Workbench 1.5

Para LBR iiwa

Instrucciones de manejo y programación



Edición: 25.11.2014

Versión: KUKA Sunrise.OS 1.5 V1

© Copyright 2014

KUKA Laboratories GmbH
Zugspitzstraße 140
D-86165 Augsburg
Alemania

La reproducción de esta documentación – o parte de ella – o su facilitación a terceros solamente está permitida con expresa autorización del KUKA Laboratories GmbH.

Además del volumen descrito en esta documentación, pueden existir funciones en condiciones de funcionamiento. El usuario no adquiere el derecho sobre estas funciones en la entrega de un aparato nuevo, ni en casos de servicio.

Hemos controlado el contenido del presente escrito en cuanto a la concordancia con la descripción del hardware y el software. Aún así, no pueden excluirse totalmente todas las divergencias, de modo tal, que no aceptamos responsabilidades respecto a la concordancia total. Pero el contenido de estos escritos es controlado periódicamente, y en casos de divergencia, éstas son enmendadas y presentadas correctamente en la edición siguiente.

Reservados los derechos a modificaciones técnicas que no tengan influencia en el funcionamiento.

Traducción de la documentación original

KIM-PS5-DOC

Publicación: Pub KUKA Sunrise.OS 1.5 (PDF) es

Estructura de libro: KUKA Sunrise.OS 1.5 V1.1

Versión: KUKA Sunrise.OS 1.5 V1

Índice

1	Introducción	15
1.1	Grupo destinatario	15
1.2	Documentación del robot industrial	15
1.3	Representación de observaciones	15
1.4	Marcas	16
1.5	Terminología utilizada	16
2	Descripción del producto	19
2.1	Vista general del sistema de robot	19
2.2	Vista general de los componentes software	20
2.3	Resumen de KUKA Sunrise.OS	20
2.4	Vista general de KUKA Sunrise.Workbench	21
2.5	Utilización del System Software conforme a los fines previstos	21
3	Seguridad	23
3.1	Condiciones marco legales	23
3.1.1	Observaciones sobre responsabilidades	23
3.1.2	Uso conforme a lo previsto del robot industrial	23
3.1.3	Declaración de conformidad de la CE y declaración de montaje	24
3.2	Funciones de seguridad	24
3.2.1	Términos utilizados	25
3.2.2	Personal	27
3.2.3	Campos y zonas de trabajo, protección y de peligro	28
3.2.4	Funciones destinadas a la seguridad	29
3.2.4.1	Dispositivo de PARADA DE EMERGENCIA	29
3.2.4.2	Dispositivo de validación	30
3.2.4.3	Protección del operario	30
3.2.4.4	Dispositivo externo de PARADA DE EMERGENCIA	31
3.2.4.5	Parada de seguridad externa 1 (sobre la trayectoria)	31
3.2.4.6	Dispositivo de validación externo	31
3.2.4.7	Parada de servicio externa segura	32
3.2.5	Causas de reacciones de parada destinadas a la seguridad	32
3.2.6	Funciones no destinadas a la seguridad	33
3.2.6.1	Selección de modos de servicio	33
3.2.6.2	Interruptor de final de carrera de software	34
3.3	Equipamiento de protección adicional	34
3.3.1	Modo paso a paso	34
3.3.2	Identificaciones en el robot industrial	35
3.3.3	Dispositivos de seguridad externos	35
3.4	Medidas de seguridad	36
3.4.1	Medidas generales de seguridad	36
3.4.2	Transporte	37
3.4.3	Puesta en servicio y reanudación del servicio	37
3.4.4	Modo de servicio manual	39
3.4.5	Modo de servicio automático	40
3.4.6	Mantenimiento y reparación	40
3.4.7	Cese del servicio, almacenamiento y eliminación de residuos	41
3.4.8	Medidas de seguridad para el "Single Point of Control"	41

3.5	Normas y prescripciones aplicadas	42
4	Instalación de KUKA Sunrise.Workbench	45
4.1	Requisitos del sistema PC	45
4.2	Instalar Sunrise.Workbench	45
4.3	Desinstalar Sunrise.Workbench	45
4.4	Instalar paquete de idioma en Sunrise.Workbench	46
5	Manejo de KUKA Sunrise.Workbench	47
5.1	Iniciar Sunrise.Workbench	47
5.2	Vista general de la interfaz de usuario de Sunrise.Workbench	47
5.2.1	Disponer vistas de otro modo	49
5.2.2	Visualizar diferentes perspectivas de la interfaz de usuario	49
5.2.3	Barras de herramientas	50
5.3	Crear proyecto Sunrise con plantilla	51
5.4	Crear una nueva aplicación del robot	54
5.4.1	Crear nuevo paquete de Java	54
5.4.2	Crear aplicación del robot con paquete	54
5.4.3	Crear una aplicación del robot para el paquete existente	55
5.5	Crear una nueva tarea en segundo plano	55
5.5.1	Crear una tarea en segundo plano con paquete	55
5.5.2	Crear una tarea en segundo plano para el paquete existente	56
5.6	Zona de trabajo	56
5.6.1	Crear una zona de trabajo nueva	56
5.6.2	Cambiar a una zona de trabajo existente	56
5.6.3	Cambiar entre los campos de trabajo abiertos por última vez	57
5.6.4	Archivar proyectos	57
5.6.5	Cargar proyectos desde el archivo en la zona de trabajo	57
5.6.6	Cargar proyectos del directorio en la zona de trabajo	57
5.7	Proyectos Sunrise con proyectos Java referenciados	58
5.7.1	Crear proyecto nuevo de Java	58
5.7.1.1	Añadir bibliotecas de clases específicas del robot en el proyecto Java	58
5.7.2	Referenciar proyectos Java	59
5.7.3	Eliminar la referencia para los proyectos Java	59
5.8	Renombrar elementos en el "Explorador de paquetes"	60
5.9	Eliminar elementos en el "Explorador de paquetes"	60
5.9.1	Eliminar elementos de un proyecto	60
5.9.2	Eliminar proyectos	60
5.10	Activar la detección automática de modificaciones	61
6	Manejo de KUKA smartPAD	63
6.1	Dispositivo de mando manual KUKA smartPAD	63
6.1.1	Parte delantera	63
6.1.2	Parte trasera	65
6.2	Conectar/desconectar la unidad de control del robot	66
6.2.1	Conectar la unidad de control del robot e iniciar el System Software	66
6.2.2	Desconectar la unidad de control del robot	66
6.3	Actualización automática del software del smartPAD	66
6.4	Interfaz de usuario KUKA smartHMI	67

6.4.1	Barra de navegación	68
6.4.2	Indicación de estados	69
6.4.3	Teclado	70
6.4.4	Vista de la estación	70
6.4.5	Vista del robot	72
6.5	Activar el menú principal	73
6.6	Cambiar modo de servicio	75
6.7	Sistemas de coordenadas	76
6.8	Desplazar el robot de forma manual	78
6.8.1	Ventana Opciones de desplazamiento manual	78
6.8.2	Ajustar el override manual (HOV)	80
6.8.3	Con las teclas de desplazamiento, desplazar en específicos del eje	80
6.8.4	Con las teclas de desplazamiento, desplazar de forma cartesiana	81
6.8.4.1	Movimiento de espacio cero	82
6.9	Modo de servicio KRF – Desbloquear el robot de forma controlada	82
6.10	Guiar el robot manualmente	83
6.11	Reanudar control de seguridad	83
6.12	Abrir los frenos de retención	84
6.13	Programar por aprendizaje y desplazar manualmente los Frames	85
6.13.1	Visualizar Frames	85
6.13.2	Programar por aprendizaje los Frames	86
6.13.3	Ventana Tipo de desplazamiento	88
6.13.4	Desplazamiento manual hasta Frames	89
6.14	Ejecución del programa	90
6.14.1	Seleccionar la aplicación de robot	90
6.14.2	Seleccionar el modo de ejecución del programa	93
6.14.3	Ajustar el Override manual	94
6.14.4	Arrancar el programa hacia adelante (manual)	95
6.14.5	Arrancar el programa hacia adelante (automático)	95
6.14.6	Retroceder el robot después de abandonar la trayectoria	95
6.15	Activar las teclas de usuario	96
6.16	Funciones de indicación en pantalla	97
6.16.1	Visualizar el Frame de destino del movimiento ejecutado actualmente	97
6.16.2	Visualizar la posición real específica del eje	98
6.16.3	Visualizar la posición real cartesiana	99
6.16.4	Visualizar momentos específicos del eje	100
6.16.5	Visualizar el grupo E/S y modificar el valor de una salida	100
6.16.6	Mostrar la dirección IP y la versión de software	102
6.16.7	Visualizar el tipo de robot y el número de serie	103
7	Puesta en servicio y reanudación del servicio	105
7.1	Ajuste de posición	105
7.1.1	Ajustar ejes	105
7.1.2	Desajustar los ejes de forma manual	105
7.2	Medición	106
7.2.1	Medir la herramienta	106
7.2.1.1	Medir el TCP: Método XYZ 4 puntos	107
7.2.1.2	Definir la orientación: Método ABC 2 puntos	109
7.2.1.3	Definir la orientación: Método ABC mundial	111

7.2.2	Medición base: Método de 3 puntos	112
7.3	Determinar datos de carga de la herramienta	114
8	Prueba de frenos	119
8.1	Resumen de la prueba de frenos	119
8.2	Crear la aplicación de la prueba de frenos a partir de la plantilla	122
8.2.1	Adaptar la aplicación de la prueba de frenos para la prueba de acuerdo con un mínimo momento de frenado de los frenos	124
8.2.2	Modificar la secuencia de movimientos para la determinación del valor del momento	125
8.2.3	Modificar la posición de salida para la prueba de frenos	125
8.3	Interfaz de programación para la prueba de frenos	126
8.3.1	Evaluar momentos generados y determinar el máximo valor absoluto	126
8.3.2	Consultar el resultado de la evaluación de los máximos momentos absolutos ...	127
8.3.3	Crear un objeto para la prueba de frenos	129
8.3.4	Iniciar la ejecución de la prueba de frenos	130
8.3.5	Evaluar la prueba de frenos	131
8.3.5.1	Consultar el resultado de la prueba de frenos	133
8.4	Ejecutar la prueba de frenos	135
8.4.1	Resultado de la evaluación de los máximos momentos absolutos (indicación) ...	136
8.4.2	Consultar el resultado de la prueba de frenos (indicación)	136
9	Gestión del proyecto	139
9.1	Proyecto Sunrise – Resumen	139
9.2	Gestión de Frames	139
9.2.1	Crear un nuevo Frame	140
9.2.2	Marcar Frame como base	140
9.2.3	Desplazar Frames	141
9.2.4	Borrar Frames	142
9.2.5	Visualizar y modificar las propiedades de un Frame	142
9.2.6	Insertar un Frame en la instrucción de movimiento	144
9.3	Gestión de objetos	144
9.3.1	Estructura geométrica de las herramientas	144
9.3.2	Estructura geométrica de las piezas	145
9.3.3	Crear herramienta o pieza	146
9.3.4	Crear Frame para herramienta o pieza	146
9.3.5	Establecer Frame estándar para movimientos	148
9.3.6	Datos de carga	149
9.3.6.1	Introducir los datos de carga	149
9.3.7	Herramienta destinada a la seguridad	150
9.3.7.1	Definir la herramienta destinada a la seguridad	151
9.3.8	Piezas de trabajo destinadas a la seguridad	153
9.3.8.1	Definir piezas de trabajo destinadas a la seguridad	154
9.4	Sincronización de proyectos	155
9.4.1	Transferir un proyecto a la unidad de control del robot	156
9.4.2	Actualizar el proyecto en la unidad de control del robot o en Sunrise.Workbench	157
9.5	Cargar el proyecto desde la unidad de control del robot	158
10	Configuración de la estación e instalación	159
10.1	Abrir la configuración de la estación	159
10.1.1	Configurar parámetros para medición	159

10.2 Instalar el System Software	160
10.2.1 Convertir la configuración de seguridad a una versión nueva de software	161
10.3 Instalar paquete de idioma	161
10.4 Instalar escáner de virus	162
11 Configuración del bus	163
11.1 Configuración y conexión de E/S en WorkVisual – Resumen	163
11.2 Crear nueva configuración de E/S	164
11.3 Abrir una configuración de E/S existente	164
11.4 Crear E/S Sunrise	164
11.4.1 Ventana "Crear señales de E/S"	165
11.4.2 Crear el grupo E/S y las entradas/salidas del grupo	166
11.4.3 Editar grupo E/S	167
11.4.4 Borrar grupo E/S	167
11.4.5 Modificar la entrada/salida de un grupo	167
11.4.6 Borrar la entrada/salida de un grupo	167
11.4.7 Exportar grupo E/S como plantilla	167
11.4.8 Importar grupo E/S desde la plantilla	168
11.5 Conectar bus	169
11.5.1 Ventana Círculo EA	169
11.5.2 Botones de la ventana Círculo EA	170
11.5.3 Conectar E/S Sunrise	171
11.6 Exportar la configuración de E/S en el proyecto Sunrise	171
12 Control externo	173
12.1 Configurar control externo	173
12.1.1 Control externo de entradas	174
12.1.2 Control externo de salidas	174
12.1.3 Configurar el control externo en las propiedades del proyecto	175
12.2 Marcar la aplicación del robot como aplicación por defecto	176
12.3 Establecer salidas de aviso para el proyecto no controlado externamente	177
13 Configuración de seguridad	179
13.1 Resumen de la configuración de seguridad	179
13.2 Concepto de seguridad	180
13.3 Permanent Safety Monitoring	182
13.4 Event-driven Safety Monitoring	184
13.5 Resumen de Atomic Monitoring Functions	185
13.5.1 Standard Atomic Monitoring Functions	185
13.5.2 Atomic Monitoring Functions parametrizables	186
13.5.3 Extended Atomic Monitoring Functions	188
13.6 Configuración de seguridad con KUKA Sunrise.Workbench	188
13.6.1 Cambiar la vista de la configuración de seguridad y activar en el control	189
13.6.2 Abrir la configuración de seguridad	190
13.6.2.1 Evaluación de la configuración de seguridad	190
13.6.2.2 Resumen de la configuración de seguridad de la interfaz de usuario	191
13.6.3 Configurar las funciones de seguridad del mecanismo de PSM	192
13.6.3.1 Abrir la tabla PSM de usuarios	192
13.6.3.2 Crear funciones de seguridad para mecanismos PSM	194
13.6.3.3 Borrar la función de seguridad del mecanismo PSM	194

13.6.3.4	Editar funciones de seguridad existentes del mecanismo PSM	194
13.6.4	Configurar estados seguros del mecanismo ESM	195
13.6.4.1	Añadir nuevo estado ESM	196
13.6.4.2	Abrir la tabla de un estado ESM	196
13.6.4.3	Borrar estado ESM	198
13.6.4.4	Crear una función de seguridad para el estado ESM	198
13.6.4.5	Borrar la función de seguridad de un estado ESM	198
13.6.4.6	Editar una función de seguridad existente de un estado ESM	199
13.6.4.7	Desactivar el mecanismo ESM	199
13.6.4.8	Comutar entre estados ESM	199
13.7	Activar la configuración de seguridad en la unidad de control del robot	200
13.7.1	Desactivar la configuración de seguridad	201
13.7.2	Restaurar la configuración de seguridad	201
13.7.3	Modificar la contraseña para la activación de la configuración de seguridad	201
13.8	Utilización y parametrización de las Atomic Monitoring Functions	202
13.8.1	Evaluación de los dispositivos de seguridad en el KUKA smartPAD	202
13.8.2	Evaluación del modo de servicio	202
13.8.3	Evaluación del movimiento habilitado	203
13.8.4	Evaluación de la referenciación de posición	203
13.8.5	Evaluación de la referenciación del momento	204
13.8.6	Control de entradas seguras	204
13.8.7	Control de dispositivos de validación en equipos manuales de guiado	205
13.8.8	Controles de velocidad	207
13.8.8.1	Definir el control de velocidad específico del eje	207
13.8.8.2	Definir el control de velocidad cartesiana	208
13.8.9	Zonas de control	208
13.8.9.1	Definir campos de trabajo cartesianos	211
13.8.9.2	Definir zonas de protección cartesianas	213
13.8.9.3	Definir las zonas de control específicas del eje	215
13.8.10	Control de la orientación de herramienta	217
13.8.11	Control de parada (parada de servicio segura)	219
13.8.12	Retardo de conexión para las funciones de seguridad	220
13.8.13	Control de fuerzas y momentos	220
13.8.13.1	Control del campo del eje	221
13.8.13.2	Detección de colisión	221
13.8.13.3	Control de fuerza TCP	222
13.9	Ejemplo de una configuración de seguridad	223
13.9.1	Función	224
13.9.2	Requisito	224
13.9.3	Propuesta de solución para la tarea	224
13.10	Referenciación de posición y del momento	227
13.10.1	Referenciación de posición	227
13.10.2	Referenciación del momento	228
13.10.3	Crear la aplicación para la referenciación de posición y del momento	229
13.11	Resumen de la recepción de seguridad	230
13.11.1	Lista de verificación de funciones de seguridad general	231
13.11.2	Listas de verificación de la herramienta orientada a la seguridad	233
13.11.2.1	Datos geométricos de la herramienta destinada a la seguridad	233
13.11.2.2	Datos de carga de la herramienta destinada a la seguridad	234

13.11.3	Liste de verificación de las piezas de trabajo destinadas a la seguridad	235
13.11.4	Lista de verificación de las líneas de la tabla de usuarios PSM	236
13.11.5	Listas de verificación de estados ESM	236
13.11.5.1	Estados ESM utilizados	236
13.11.5.2	Estados ESM no utilizados	238
13.11.6	AMFs utilizadas en listas de verificación	238
13.11.6.1	AMF PARADA DE EMERGENCIA smartPAD	238
13.11.6.2	AMF Validación smartPAD	238
13.11.6.3	AMF Validación pánico smartPAD	239
13.11.6.4	AMF Validación equipo manual de guiado	239
13.11.6.5	AMF Modo de servicio Test	239
13.11.6.6	AMF Modo de servicio Automático	239
13.11.6.7	AMF Modo de servicio con velocidad reducida	239
13.11.6.8	AMF Modo de servicio de alta velocidad	240
13.11.6.9	AMF Señal de entrada	240
13.11.6.10	AMF Control de parada de todos los ejes	240
13.11.6.11	AMF Movimiento habilitado	240
13.11.6.12	AMF Control del campo del eje	240
13.11.6.13	AMF Monitorización de la velocidad del eje	240
13.11.6.14	AMF Referenciación de posición	241
13.11.6.15	AMF Referenciación de momentos	241
13.11.6.16	AMF Control del campo del eje	241
13.11.6.17	AMF Control de velocidad cartesiana	241
13.11.6.18	AMF Control del campo de trabajo cartesiano / Monitorización cartesiana de las zonas de protección	242
13.11.6.19	AMF Identificación de colisión	242
13.11.6.20	AMF Control de fuerza TCP	243
13.11.6.21	AMF Retardo de tiempo	243
13.11.6.22	AMF Orientación de la herramienta	243
13.11.7	Generar informe relativo a la configuración de seguridad	244
14	Principios de la programación de movimiento	247
14.1	Resumen Tipos de movimiento	247
14.2	Tipo de movimiento PTP	247
14.3	Tipo de movimiento LIN	248
14.4	Tipo de movimiento CIRC	248
14.5	Tipo de movimiento SPL	249
14.6	Tipo de movimiento Spline	249
14.6.1	Perfil de velocidad para movimientos Spline	250
14.6.2	Cambios en bloques Spline	252
14.6.3	Transición LIN-SPL-LIN	254
14.7	Tipo de movimiento guiado manual	255
14.8	Posicionamiento aproximado	256
14.9	Control de la orientación LIN, CIRC, SPL	258
14.9.1	CIRC – Sistema de referencia del control de la orientación	260
14.9.2	CIRC – Combinación de sistema de bus y tipo de control de la orientación	261
14.10	Información de redundancia	262
14.10.1	Ángulo de redundancia	263
14.10.2	Estado	263
14.10.3	Turn	264
14.11	Singularidades	265

14.11.1	Singularidades cinemáticas	265
14.11.2	Singularidades condicionadas por el sistema	267
15	Programación	269
15.1	Editor de Java	269
15.1.1	Abrir la aplicación del robot en el editor de Java	269
15.1.2	Estructura de una aplicación del robot	269
15.1.3	Funciones de procesamiento	270
15.1.3.1	Renombrar variables	270
15.1.3.2	Autocompletar	270
15.1.3.3	Plantillas – Entrada rápida para instrucciones Java	271
15.1.3.4	Crear plantillas específicas de usuario	272
15.1.3.5	Extraer métodos	272
15.1.4	Visualizar información Javadoc	273
15.1.4.1	Estructura del navegador Javadoc	275
15.2	Caracteres y tipos de letra	278
15.3	Tipos de datos	278
15.4	Variables	279
15.5	Comunicación de red a través de UDP y TCP/IP	279
15.6	Información sobre la versión de RoboticsAPI	280
15.6.1	Visualizar la versión de RoboticsAPI	280
15.6.2	Estructura del número de versión de RoboticsAPI	280
15.7	Programación de movimiento: PTP, LIN, CIRC	280
15.7.1	Estructura de una instrucción de movimiento (move/moveAsync)	280
15.7.2	PTP	281
15.7.3	LIN	282
15.7.4	CIRC	283
15.7.5	LIN REL	283
15.7.6	MotionBatch	284
15.8	Programación de movimiento: Spline	285
15.8.1	Sugerencias de programación para movimientos Spline	285
15.8.2	Crear un bloque Spline CP	286
15.8.3	Crear un bloque Spline JP	287
15.8.4	Utilizar Spline en una instrucción de movimiento	288
15.9	Programar el guiado manual	288
15.10	Parámetros de movimiento	290
15.10.1	Programar parámetros de movimiento específicos del eje	292
15.11	Utilizar herramientas y piezas en el programa	293
15.11.1	Declarar herramientas y piezas	293
15.11.2	Inicializar herramientas y piezas	294
15.11.3	Conectar herramientas y piezas con el robot	294
15.11.3.1	Conectar la herramienta con la brida del robot	295
15.11.3.2	Conectar la pieza con otros objetos	295
15.11.3.3	Desconectar conexiones	297
15.11.4	Desplazar herramientas y piezas	297
15.11.5	Ordenar cambios de carga en el control de seguridad	298
15.12	Entradas/Salidas	300
15.12.1	Crear el campo de datos para el grupo E/S	302
15.12.2	Inicializar el campo de datos para el grupo E/S	302

15.12.3 Leer entradas/salidas	303
15.12.4 Ajustar salidas	303
15.13 Consultar momentos axiales	304
15.14 Leer fuerzas y momentos cartesianos	305
15.14.1 Consultar datos de fuerza y de momentos calculados	306
15.14.2 Consultar valores de fuerza y de momentos individuales	307
15.14.3 Comprobar la fiabilidad de los valores de fuerza y de momentos calculados	307
15.14.4 Consultar valores individuales de un vector	309
15.15 Consulta de la posición del robot	309
15.15.1 Consultar la posición real o nominal específica del eje	310
15.15.2 Consultar la posición real o nominal cartesiana	311
15.15.3 Consultar la diferencia nominal-real cartesiana	312
15.16 Posición HOME	313
15.16.1 Modificar la posición HOME	313
15.17 Consultar estados del sistema	314
15.17.1 Consultar la posición HOME	314
15.17.2 Consultar el estado de ajuste	315
15.17.3 Consultar la disponibilidad de desplazamiento	315
15.17.3.1 Reaccionar a una modificación de la señal de disponibilidad de desplazamiento	316
15.17.4 Consultar la actividad del robot	316
15.17.5 Consultar y evaluar señales de seguridad	317
15.17.5.1 Consultar el estado de las señales de seguridad	317
15.17.5.2 Reaccionar ante la modificación del estado de las señales de seguridad	319
15.18 Modificar y consultar el modo de ejecución del programa	320
15.19 Modificar y consultar el Override	321
15.19.1 Reaccionar a un cambio de Override	322
15.20 Condiciones	323
15.20.1 Condiciones en la RoboticsAPI	323
15.20.2 Condiciones complejas	324
15.20.3 Condición de los momentos axiales	325
15.20.4 Condición de fuerza	326
15.20.4.1 Condición para la fuerza cartesiana desde todas las direcciones	328
15.20.4.2 Condición para fuerza normal	328
15.20.4.3 Condición para la fuerza de corte	330
15.20.5 Condición de componentes de Fuerza	331
15.20.6 Condición referida a la trayectoria	333
15.20.7 Condición para señales booleanas	335
15.20.8 Condición para el rango de valores de una señal	336
15.21 Condiciones de cancelación para instrucciones de movimiento	336
15.21.1 Establecer la condición de cancelación	336
15.21.2 Evaluar condiciones de cancelación	337
15.21.2.1 Consultar condición de cancelación	338
15.21.2.2 Consultar la posición del robot en el momento de cancelación	339
15.21.2.3 Consultar movimiento interrumpido (bloque Spline, MotionBatch)	340
15.22 Acciones de conmutación referidas a la trayectoria (Trigger)	340
15.22.1 Programar un Trigger	340
15.22.2 Programar acción de conmutación referida a la trayectoria	341
15.22.3 Evaluar información del Trigger	343
15.23 Control de procesos (Monitoring)	344

15.23.1 Listener para el control de condiciones	345
15.23.2 Crear objeto de Listener para el control de una condición	346
15.23.3 Registrar el Listener para la notificación en caso de modificación del estado	347
15.23.4 Activar o desactivar el servicio de notificación para el Listener	348
15.23.5 Ejemplo de programación de control	348
15.24 Espera con bloqueo de la condición	349
15.25 Registrar y evaluar datos	350
15.25.1 Crear objeto para el registro de datos	350
15.25.2 Establecer datos para el registro	352
15.25.3 Iniciar el registro de datos	353
15.25.4 Finalizar el registro de datos	354
15.25.5 Consultar estados del objeto DataRecorder	355
15.25.6 Programa de ejemplo para el registro de datos	355
15.26 Definir teclas de usuario	356
15.26.1 Crear una barra de teclas de usuario	358
15.26.2 Añadir teclas de usuario a la barra	358
15.26.3 Establecer la función de una tecla de usuario	360
15.26.4 Rotulación y diseño gráfico de la barra de teclas de usuario	362
15.26.4.1 Asignar elemento textual	363
15.26.4.2 Asignar un ícono de LED	364
15.26.5 Identificar teclas de usuario críticas para la seguridad	365
15.26.6 Publicar la barra de teclas de usuario	366
15.27 Programación de mensajes	366
15.27.1 Programar mensajes de usuario	366
15.27.2 Programar cuadros de diálogo	367
15.28 Control de la ejecución del programa	369
15.28.1 Pausar la aplicación	369
15.28.2 Pausar la ejecución del movimiento	369
15.28.3 Bucle for	369
15.28.4 Bucle while	370
15.28.5 Bucle do-while	371
15.28.6 Ramificación if-else	372
15.28.7 Ramificación switch	374
15.28.8 Ejemplos de bucles intercalados	375
15.29 Continuar con una aplicación pausada en el modo automático (recovery)	376
15.30 Tratamiento de fallos	378
15.30.1 Tratamiento de instrucciones de movimiento fallidas	378
15.30.2 Tratamiento de instrucciones de desplazamiento sincronizadas y fallidas	378
15.30.3 Tratamiento de instrucciones de desplazamiento asíncronas y fallidas	380
16 Tareas en segundo plano	385
16.1 Utilización de tareas en segundo plano	385
16.2 Tarea en segundo plano cíclica	386
16.3 Tarea en segundo plano no cíclica	387
17 Programación con robot flexible	389
17.1 Sensores y regulación	389
17.2 Resumen de reguladores disponibles	389
17.3 Utilizar el regulador en la aplicación del robot	390

17.3.1	Crear un objeto del regulador	390
17.3.2	Determinar parámetros del regulador	390
17.3.3	Transferir el objeto del regulador como parámetro de movimiento	390
17.4	Regulador de posición	391
17.5	Regulador de impedancia cartesiano	391
17.5.1	Cálculo de las fuerzas según la ley de elasticidad	392
17.5.2	Parametrización del regulador de impedancia	394
17.5.2.1	Representación de los grados de libertad cartesianos	394
17.5.2.2	Determinar parámetros del regulador para grados de libertad individuales	395
17.5.2.3	Parámetros del regulador específicos del grado de libertad	396
17.5.2.4	Parámetros del regulador independientes del grado de libertad	397
17.6	Regulador de impedancia cartesiano con oscilación de fuerza conectada adicionalmente 400	
17.6.1	Conexión de una oscilación de fuerza simple	400
17.6.2	Conexión de oscilaciones de fuerza superiores (figuras de Lissajous)	401
17.6.3	Parametrización del regulador de impedancia con oscilación de fuerza conectada adicionalmente	402
17.6.3.1	Parámetros del regulador específicos del grado de libertad	403
17.6.3.2	Parámetros del regulador independientes del grado de libertad	405
17.7	Métodos estáticos para el regulador de impedancia con oscilación de fuerza superior.	407
17.7.1	Conectar fuerza constante	408
17.7.2	Conectar la oscilación de fuerza simple	408
17.7.3	Conectar curva de Lissajous	409
17.7.4	Conectar la oscilación de fuerza en espiral	410
17.8	Mantener la posición con regulación	412
18	Diagnóstico	413
18.1	Visualizar errores de bus de campo	413
18.1.1	Errores generales de bus de campo	413
18.1.2	Estado de error de E/S y de grupos E/S	413
18.2	Visualizar el protocolo	413
18.2.1	Vista Protocolo	414
18.2.2	Filtrar entradas del protocolo	415
18.3	Indicación de mensajes de error (vista de aplicación)	416
18.4	Recopilar información de diagnóstico para el análisis de errores de KUKA	419
18.4.1	Crear un paquete de diagnóstico mediante smartHMI	419
18.4.2	Crear un paquete de diagnóstico mediante smartPAD	420
18.4.3	Crear un paquete de diagnóstico mediante Sunrise.Workbench	420
18.4.4	Cargar paquetes de diagnóstico existentes desde la unidad de control del robot	420
19	Servicio KUKA	423
19.1	Requerimiento de soporte técnico	423
19.2	KUKA Customer Support	423
Índice		431

1 Introducción

1.1 Grupo destinatario

Esta documentación está destinada al usuario con los siguientes conocimientos:

- Conocimientos avanzados de sistema sobre la unidad de control del robot
- Conocimientos avanzados en programación con Java



Para una utilización óptima de nuestros productos, recomendamos a nuestros clientes que asistan a un curso de formación en el KUKA College. En www.kuka.com puede encontrar información sobre nuestro programa de formación, o directamente en nuestras sucursales.

1.2 Documentación del robot industrial

La documentación del robot industrial consta de las siguientes partes:

- Documentación para el sistema mecánico del robot
- Documentación para la unidad de control del robot
- Instrucciones de servicio y programación para el software de sistema
- Instrucciones para opciones y accesorios
- Catálogo de piezas en el soporte de datos

Cada manual de instrucciones es un documento por sí mismo.

1.3 Representación de observaciones

Seguridad

Estas observaciones son de seguridad y se **deben** tener en cuenta.



PELIGRO Estas observaciones indican que, si no se toman las medidas de precaución, es probable o completamente seguro que **se produzcan** lesiones graves o incluso la muerte.



ADVERTENCIA Estas observaciones indican que, si no se toman las medidas de precaución, **pueden** producirse lesiones graves o incluso la muerte.



ATENCIÓN Estas observaciones indican que, si no se toman las medidas de precaución, **pueden** producirse lesiones leves.



AVISO Estas observaciones indican que, si no se toman las medidas de precaución, **pueden** producirse daños materiales.



Estas observaciones remiten a información relevante para la seguridad o a medidas de seguridad generales.
Estas indicaciones no hacen referencia a peligros o medidas de precaución concretos.

Esta observación llama la atención acerca de procedimientos que sirven para evitar o eliminar casos de emergencia o avería:



Los procedimientos señalados con esta observación **tienen** que respetarse rigurosamente.

Observaciones Estas indicaciones sirven para facilitar el trabajo o contienen remisiones a información que aparece más adelante.



Observación que sirve para facilitar el trabajo o remite a información que aparece más adelante.

1.4 Marcas

Java es una marca de Sun Microsystems (Oracle Corporation).

Windows es una marca de Microsoft Corporation.



es una marca de Beckhoff Automation GmbH.

1.5 Terminología utilizada

Término	Descripción
AMF	Atomic Monitoring Function Unidad más pequeña de una función de control (español: control elemental)
API	Application Programming Interface Interfaz para la programación de aplicaciones
Servidor de aplicaciones	El servidor de aplicaciones gestiona las aplicaciones del robot y las ejecuta en su Java Virtual Machine. Las aplicaciones del robot acceden a Sunrise a través de RoboticsAPI y ordenan de este modo el control en tiempo real. Además, en el servidor de aplicaciones se gestionan servicios a través de los que se puede realizar el enlace del software de manejo smartHMI.
CIB-SR	Cabinet Interface Board Small Robot
ESM	Event-Driven Safety Monitoring Controles de seguridad que se activan a través de incidencias definidas
Exception	Excepción o situación excepcional (en inglés, exception) Una excepción define un procedimiento para transmitir información acerca de determinados estados de programa, normalmente estados de error, a otros niveles de programa para que se puedan seguir procesando.
Frame	Un Frame es un sistema de coordenadas tridimensional que se describe en relación con un sistema de referencia mediante su posición y su orientación. Con la ayuda del Frame se pueden definir puntos en el espacio de forma sencilla. Los Frames a menudo están dispuestos jerárquicamente en una estructura de árbol.
FSoE	Fail Safe over EtherCAT FSoE es un protocolo para la transferencia de datos relevantes para la seguridad a través de EtherCAT empleando un maestro FSoE y un esclavo FSoE.
Javadoc	Javadoc es una documentación generada a partir de comentarios especiales de Java.
JRE	Java Runtime Environment Entorno del tiempo de ejecución del lenguaje de programación Java

Término	Descripción
KRF	<p>Kontrollierte Roboterfahrt (desplazamiento controlado del robot)</p> <p>KRF es un modo de servicio que está disponible si el control de seguridad detiene al robot industrial por alguna de las siguientes causas:</p> <ul style="list-style-type: none"> ■ El robot industrial vulnera un espacio controlado de forma segura. ■ La orientación de la herramienta destinada a la seguridad está fuera del sector controlado de forma segura. ■ El robot industrial vulnera un límite de fuerza o de momento controlado de forma segura. ■ Un sensor de posición no está ajustado con el control cartesiano de velocidad activo. <p>En el modo de servicio KRF el robot puede desplazarse manualmente y devolverse a una posición en la que el control con activación de parada ya no se vea vulnerado.</p>
KUKA RoboticsAPI	<p>Interfaz de programación de Java para robots KUKA</p> <p>RoboticsAPI es una interfaz Java orientada al objeto para el control de robots y dispositivos periféricos.</p>
KUKA smartHMI	<p>KUKA smart Human-Machine Interface</p> <p>Denominación de la interfaz de usuario de la unidad de control del robot</p>
KUKA smartPAD	<p>El smartPAD es el dispositivo de mando manual de la celda del robot (estación). Contiene todas las funciones de control e indicación necesarias para el manejo de la estación.</p>
KUKA Sunrise Cabinet	<p>Hardware de control para el servicio del robot industrial LBR iiwa</p>
KUKA Sunrise.OS	<p>KUKA Sunrise.Operating System</p> <p>System Software para robots industriales que se operan con la unidad de control del robot KUKA Sunrise Cabinet</p>
MRK	<p>Cooperación hombre-robot</p>
PROFINET	<p>PROFINET un bus de campo basado en Ethernet.</p>
PROFIsafe	<p>PROFIsafe es una interfaz de seguridad basada en PROFINET para enlazar un PLC de seguridad con la unidad de control del robot. (PLC = maestro, unidad de control del robot = esclavo)</p>
PSM	<p>Permanent Safety Monitoring</p> <p>Controles de seguridad que están activas de forma permanente</p>
PLC	<p>Controlador con memoria programable</p>
TCP	<p>Tool Center Point (punto central de la herramienta)</p> <p>El TCP es el punto de trabajo de una herramienta. Se pueden definir varios puntos de trabajo para una herramienta.</p>

2 Descripción del producto

2.1 Vista general del sistema de robot

Un sistema de robot ([>>> Fig. 2-1](#)) comprende todos los grupos constructivos de un robot industrial, como el manipulador (mecánica del robot con instalación eléctrica), la unidad de control, los cables de unión, las herramientas y partes del equipamiento.

El robot industrial consta de los siguientes componentes:

- Manipulador
- Unidad de control del robot KUKA Sunrise Cabinet
- Dispositivo de mando manual KUKA smartPAD
- Cables de unión
- Software
- Opciones, accesorios



Fig. 2-1: Vista general del sistema del robot

- 1 Cable de conexión al smartPAD
- 2 Dispositivo de mando manual KUKA smartPAD
- 3 Manipulador
- 4 Cable de unión de la unidad de control del robot KUKA Sunrise Cabinet
- 5 Unidad de control del robot KUKA Sunrise Cabinet

2.2 Vista general de los componentes software

Se instalarán los siguientes componentes software:

- KUKA Sunrise.OS 1.5
- KUKA Sunrise.Workbench 1.5
- WorkVisual 3.0

2.3 Resumen de KUKA Sunrise.OS

Descripción	Con KUKA Sunrise.OS se realiza una separación estricta del manejo y la programación del sistema de robot.
	<ul style="list-style-type: none">■ Una estación se compone actualmente de una unidad de control del robot, un manipulador y otros dispositivo.■ Una estación puede ejecutar varias aplicaciones (tareas).■ Con KUKA Sunrise.Workbench se programan aplicaciones del robot.■ Mediante el dispositivo de mando manual KUKA smartPAD se maneja una celda del robot (estación).

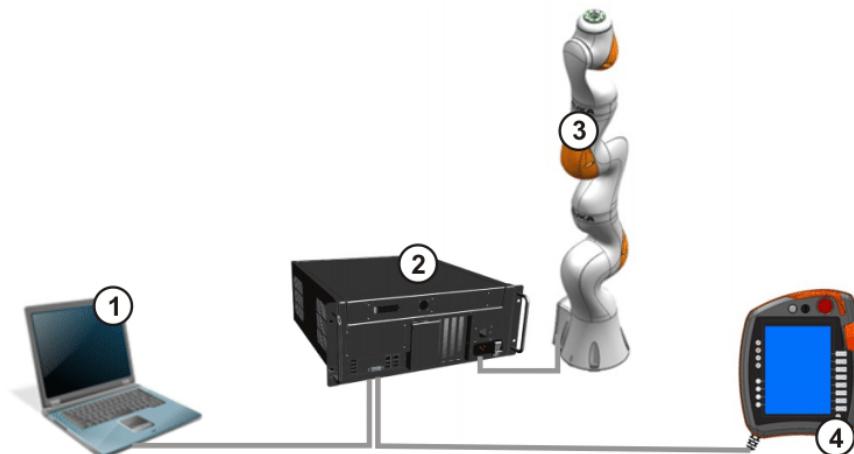


Fig. 2-2: Separación de manejo y programación

- 1 Ordenador de desarrollo con KUKA Sunrise.Workbench
- 2 Unidad de control del robot KUKA Sunrise Cabinet
- 3 Manipulador
- 4 Dispositivo de mando manual KUKA smartPAD

División de tareas	KUKA Sunrise.Workbench es la herramienta para la puesta en servicio de una estación y para el desarrollo de aplicaciones del robot. Para la configuración del bus y la conexión de bus se utiliza WorkVisual.
	<p>El smartPAD solo se necesita en la fase de puesta en servicio para las tareas que no se pueden ejecutar con KUKA Sunrise.Workbench, por motivos prácticos o técnicos de seguridad, p. ej. el ajuste, la medición o la programación por aprendizaje.</p> <p>Después de la puesta en servicio y del desarrollo de aplicaciones, el operario puede ejecutar tareas sencillas de conservación y de manejo a través del smartPAD. El operario no puede modificar la configuración de la estación y de seguridad, ni la programación.</p>

Resumen

Función	WorkVisual	Workbench	smartPAD
Configuración de la estación		✓	
Instalación del software		✓	
Configuración/diagnóstico de bus	✓		
Conexión de bus	✓		
Configurar los ajustes de seguridad		✓	
Activar la configuración de seguridad			✓
Programación		✓	
Depuración		✓	
Gestionar/editar datos del tiempo de ejecución		✓	
Programar por aprendizaje los Frames			✓
Selección de modos de servicio			✓
Desplazamiento manual			✓
Ajuste			✓
Ajustar/consultar salidas			✓
Consultar entradas			✓
Iniciar/detener aplicaciones del robot			✓
Crear un paquete de diagnóstico		✓	✓

2.4 Vista general de KUKA Sunrise.Workbench

KUKA Sunrise.Workbench es el entorno de desarrollo de la celda del robot (estación). Cuenta con las siguientes funcionalidades para la puesta en servicio y el desarrollo de la aplicación:

- | | |
|-----------------------------------|---|
| Puesta en servicio | <ul style="list-style-type: none"> ■ Instalar el System Software ■ Configurar la celda del robot (estación) ■ Editar la configuración de seguridad ■ Crear la configuración de E/S ■ Transmitir el proyecto a la unidad de control del robot |
| Desarrollo de aplicaciones | <ul style="list-style-type: none"> ■ Programar aplicaciones del robot en Java ■ Gestionar proyectos y programas ■ Editar y gestionar datos del tiempo de ejecución ■ Sincronización del proyecto |

2.5 Utilización del System Software conforme a los fines previstos

Utilización El System Software está destinado exclusivamente para operar un KUKA LBR iiwa en combinación con KUKA Sunrise Cabinet.

Cada una de las versiones del System Software solo se podrán utilizar si se cumplen los requisitos del sistema especificados para las versiones.

Uso incorrecto

Todas las aplicaciones que difieren del uso previsto se consideran usos incorrectos y no están permitidas. La empresa KUKA Laboratories GmbH no se responsabiliza por los daños ocasionados como consecuencia de un uso incorrecto. El explotador será el único responsable y asumirá todos los riesgos.

Entre los usos incorrectos se incluyen, por ejemplo:

- Operar una cinemática que no es un robot de estructura liviana de KUKA
- Operar el System Software con otros requisitos del sistema que no sean los especificados

3 Seguridad

3.1 Condiciones marco legales

3.1.1 Observaciones sobre responsabilidades

El equipo descrito en el presente documento es un robot industrial o uno de sus componentes.

Componentes del robot industrial:

- Manipulador
- Unidad de control del robot
- Dispositivo de mando manual
- Cables de unión
- Software
- Opciones, accesorios

El robot industrial se ha construido de conformidad con el estado actual de la técnica y con las normas técnicas reconocidas en materia de seguridad. No obstante, en caso de uso incorrecto puede haber riesgo de lesiones, incluso peligro de muerte, así como riesgo de daños materiales en el robot industrial o en otros.

El robot industrial debe ser utilizado únicamente en perfecto estado técnico y para los fines previstos, respetando las normas de seguridad y con conciencia de los peligros que entraña. La utilización debe realizarse bajo consideración del presente documento y de la declaración de montaje del robot industrial, que se adjunta en el suministro. Cualquier avería que pueda afectar a la seguridad deberá subsanarse de inmediato.

Información sobre la seguridad

Las indicaciones sobre seguridad no pueden ser interpretadas en contra de KUKA Laboratories GmbH. Aún cuando se hayan respetado todas las advertencias de seguridad, no puede garantizarse que el robot industrial no provoque algún tipo de lesión o daño.

Sin la debida autorización de KUKA Laboratories GmbH no deben efectuarse modificaciones en el robot industrial. Es posible integrar componentes adicionales (herramientas, software, etc.) en el robot industrial que no pertenecen al volumen de suministro de KUKA Laboratories GmbH. Si debido a la integración de dichos componentes el robot industrial u otros bienes materiales sufren daños, la responsabilidad es del usuario.

Además del capítulo sobre seguridad, la presente documentación contiene otras advertencias de seguridad que debe respetarse obligatoriamente.

3.1.2 Uso conforme a lo previsto del robot industrial

El robot industrial está diseñado única y exclusivamente para el uso descrito en el capítulo "Uso previsto" de las instrucciones de servicio o de montaje.

Todas las utilizaciones que difieran de los fines previstos se consideran usos incorrectos y no están permitidos. El fabricante no se hace responsable de los posibles daños causados por un uso incorrecto. El explotador será el único responsable y asumirá todos los riesgos.

Se considera también una utilización conforme a los fines previstos del robot industrial, el respetar las instrucciones de montaje y servicio de los componentes individuales, y, sobre todo, el cumplimiento de las condiciones de mantenimiento.

El explotador es responsable de la realización de un análisis de riesgos. De este análisis se obtienen los dispositivos de seguridad adicionales necesarios, de cuya instalación también es responsable el explotador.

Uso incorrecto	Todas las utilizaciones que difieran del uso previsto se consideran usos incorrectos y no están permitidos. Entre ellos se encuentran, p. ej.:
	<ul style="list-style-type: none">■ Transporte de personas o animales■ Utilización como pasarela■ Utilización fuera de los límites de servicio especificados■ Utilización en ambientes con riesgo de explosión■ Utilización sin dispositivos de seguridad adicionales necesarios■ Utilización al aire libre■ Utilización bajo tierra

3.1.3 Declaración de conformidad de la CE y declaración de montaje

El robot industrial se considera una máquina incompleta de conformidad con la Directiva CE relativa a las máquinas. El robot industrial solo puede ponerse en servicio cuando se cumplen los requisitos siguientes:

- El robot industrial está integrado en una instalación.
O bien: el robot industrial conforma una instalación junto con otras máquinas
O bien: el robot industrial se ha completado con todas las funciones de seguridad y dispositivos de protección necesarios para ser considerado una máquina completa de acuerdo con la Directiva CE relativa a las máquinas.
- La instalación cumple con los requisitos de la Directiva CE relativa a las máquinas, lo cual se ha comprobado mediante un proceso de evaluación de conformidad.

Declaración de conformidad	El integrador del sistema debe redactar una declaración de conformidad para toda la instalación de acuerdo con la normativa sobre construcción de máquinas. La declaración de conformidad es fundamental para la concesión de la marca CE para la instalación. El robot industrial debe operarse siempre de conformidad con las leyes, prescripciones y normas específicas del país. La unidad de control del robot cuenta con una certificación CE de conformidad con la Directiva CEM y la Directiva de baja tensión.
-----------------------------------	--

Declaración de montaje	El robot industrial, en calidad de máquina incompleta, se suministra con una declaración de montaje de acuerdo con el anexo II B de la directiva sobre máquinas 2006/42/CE. En la declaración de montaje se incluye un listado con los requisitos básicos cumplidos según el anexo I y las instrucciones de montaje. Mediante la declaración de montaje se declara que está prohibida la puesta en servicio de la máquina incompleta mientras no se monte en una máquina o se integre, con la ayuda de otras piezas, en una máquina que cumpla con las disposiciones de la Directiva CE relativa a las máquinas y con la declaración de conformidad CE según el anexo II A.
-------------------------------	--

3.2 Funciones de seguridad

Las funciones de seguridad se distinguen según los requisitos de seguridad que cumplen:

- Funciones destinadas a la seguridad para la protección de personas
Las funciones destinadas a la seguridad del robot industrial cumplen los siguientes requisitos de seguridad:
 - **Categoría 3 y Performance Level d** según EN ISO 13849-1:2008

■ **SIL 2** según EN 62061

No obstante, los requisitos se satisfacen únicamente en las siguientes condiciones:

- Durante la puesta en marcha y cada 12 meses se comprueba el funcionamiento de todos los componentes electrónicos y mecánicos del robot que sean relevantes para la seguridad, siempre y cuando no se decida otra cosa según la valoración de riesgos en el puesto de trabajo. Aquí se incluyen:
 - Dispositivo de PARADA DE EMERGENCIA en el smartPAD
 - Dispositivo de validación en el smartPAD
 - Dispositivo de validación en la brida de medios Touch (si existe)
 - Interruptor de llave en el smartPAD
 - Salidas seguras de la interfaz de seguridad discreta
- Funciones no destinadas a la seguridad para la protección de máquinas
Las funciones no destinadas a la seguridad del robot industrial no cumplen con ningún requisito específico de seguridad.



El robot industrial puede causar lesiones o daños materiales si las funciones o dispositivos de seguridad necesarios no están en servicio. En caso de que se hayan desmontado o desactivado las funciones y dispositivos de seguridad necesarios, no se debe hacer funcionar el robot industrial.



Durante la fase de planificación de la instalación también se deben planificar y diseñar las funciones de seguridad de toda la instalación.
El robot industrial se debe integrar en este sistema de seguridad de toda la instalación.

3.2.1 Términos utilizados

Término	Descripción
Campo del eje	Zona en grados o milímetros en la que se puede mover cada uno de los ejes. El campo del eje debe definirse para cada eje.
Distancia de parada	Distancia de parada = distancia de reacción + distancia de frenado La distancia de parada forma parte de la zona de peligro.
Zona de trabajo	La zona de trabajo es aquella en la que se puede mover el manipulador. La zona de trabajo se obtiene a partir de la suma de cada uno de los campos del eje.
Automático (AUT)	Modo de servicio para el modo de programación. El manipulador se desplaza con la velocidad programada.
Explotador (usuario)	El explotador de un robot industrial puede ser el empresario, el contratante o una persona delegada responsable de la utilización del robot industrial.
Zona de peligro	La zona de peligro está compuesta por la zona de trabajo y las distancias de parada.
Vida útil	La vida útil de un componente relevante para la seguridad comienza en el momento del suministro de la pieza al cliente. La vida útil no se ve afectada por la utilización o no de la pieza en una unidad de control del robot o en otra parte, ya que los componentes relevantes para la seguridad también envejecen durante el almacenamiento.

Término	Descripción
KRF	<p>Kontrollierte Roboterfahrt (desplazamiento controlado del robot)</p> <p>KRF es un modo de servicio que está disponible si el control de seguridad detiene al robot industrial por alguna de las siguientes causas:</p> <ul style="list-style-type: none"> ■ El robot industrial vulnera un espacio controlado de forma segura. ■ La orientación de la herramienta destinada a la seguridad está fuera del sector controlado de forma segura. ■ El robot industrial vulnera un límite de fuerza o de momento controlado de forma segura. ■ Un sensor de posición no está ajustado con el control cartesiano de velocidad activo. <p>En el modo de servicio KRF el robot puede desplazarse manualmente y devolverse a una posición en la que el control con activación de parada ya no se vea vulnerado.</p>
KUKA smartPAD	El smartPAD es el dispositivo de mando manual de la celda del robot (estación). El smartPAD contiene todas las funciones de control e indicación necesarias para el manejo.
Manipulador	El sistema mecánico del robot y la instalación eléctrica pertinente
Zona de seguridad	La zona de seguridad es aquella en la que no se puede mover el manipulador. La zona de seguridad es la zona situada fuera de la zona de peligro.
Parada de seguridad	<p>La parada de seguridad se activa a través del control de seguridad, interrumpe el proceso de trabajo y provoca la parada de todos los movimientos del robot. En caso de parada de seguridad, los datos del programa se conservan y el programa puede continuarse en el punto donde ha quedado interrumpido.</p> <p>La parada de seguridad puede efectuarse como categoría de parada 0, categoría de parada 1 o categoría de parada 1 (sobre la trayectoria).</p> <p>Indicación: Una parada de seguridad de la categoría de parada 0 se denomina en el documento como parada de seguridad 0, una parada de seguridad de la categoría de parada 1 como parada de seguridad 1 y una parada de seguridad de la categoría de parada 1 (sobre la trayectoria) como parada de seguridad 1 (sobre la trayectoria).</p>
Categoría de parada 0	Los accionamientos se desconectan de inmediato y se activan los frenos.
Categoría de parada 1	<p>El manipulador no frena sobre la trayectoria. El manipulador se convierte en parada con los accionamientos. En cuanto el eje está parado, se desconecta el accionamiento y se activa el freno.</p> <p>El proceso de frenado se controla de forma destinada a la seguridad por el sistema electrónico de accionamiento interno del robot. En caso de error, se ejecuta una categoría de parada 0.</p>
Categoría de parada 1 (sobre la trayectoria)	<p>El manipulador frena sobre la trayectoria. Con la parada se desconectan los accionamientos y se activan los frenos.</p> <p>Si se activa una categoría de parada 1 (sobre la trayectoria) desde el control de seguridad, se controla el proceso de frenado por el control de seguridad. Cuando ha transcurrido como máximo 1 s, se desconectan los accionamientos y se activan los frenos. En caso de error, se ejecuta una categoría de parada 0.</p>
Integrador de sistemas (Integrador de la instalación)	Los integradores de sistema son las personas responsables de integrar el robot industrial de forma segura en una instalación y de ponerlo en servicio.

Término	Descripción
T1	Modo de servicio de Prueba velocidad reducida manualmente (<= 250 mm/s)
T2	Modo de servicio Test Manual velocidad alta (> 250 mm/s admisible)

3.2.2 Personal

Para el uso del robot industrial se definen las personas o grupos de personas siguientes:

- Explotador
- Personal



Todas las personas que trabajan con el robot industrial, deben haber leído y entendido la documentación con el capítulo sobre seguridad del robot industrial.

Explotador

El operario debe respetar las normas legales de seguridad en el trabajo. Entre ellas, las siguientes:

- El operario debe cumplir sus obligaciones de vigilancia.
- El operador debe asistir periódicamente a cursos de formación.

Personal

Antes de comenzar a trabajar con la garra se deberá informar al personal implicado sobre la naturaleza y el alcance de los trabajos que se realizarán, así como sobre los posibles peligros. Periódicamente se deberán realizar cursos informativos. También será necesario organizar cursos informativos después de que hayan tenido lugar determinados sucesos o tras haber realizado modificaciones técnicas.

Se consideran miembros del personal:

- El integrador del sistema
- Los usuarios, que se dividen en:
 - Personal encargado de la puesta en servicio, el mantenimiento y el servicio técnico
 - Operario
 - Personal de limpieza



El montaje, reemplazo, ajuste, operación, mantenimiento y reparación sólo deben ser realizados atendiendo a las prescripciones del manual de servicio o montaje del correspondiente componente del robot industrial, y por personal especialmente entrenado para ello.

Integrador del sistema

El integrador del sistema es el encargado de integrar el robot industrial en la instalación respetando todas las medidas de seguridad pertinentes.

El integrador de sistema es responsable de las siguientes tareas:

- Emplazamiento del robot industrial
- Conexión del robot industrial
- Evaluación de riesgos
- Instalación de las funciones de seguridad y de protección necesarias
- Emisión de la declaración de conformidad
- Colocación de la marca CE
- Elaboración de las instrucciones de servicio de la instalación

Usuario

El usuario debe cumplir las siguientes condiciones:

- El usuario deberá haber recibido la debida formación para desempeñar los trabajos que va a realizar.
- Los trabajos a ejecutar en el robot industrial sólo deben ser realizados por personal cualificado. Por personal cualificado entendemos aquellas personas que, de acuerdo a su formación, conocimientos y experiencia, y en conocimiento de las normas vigentes, son capaces de evaluar los trabajos que se han de llevar a cabo y de detectar posibles peligros.



Los trabajos en la parte eléctrica y mecánica del manipulador solo deben ser ejecutados por KUKA Laboratories GmbH.

3.2.3 Campos y zonas de trabajo, protección y de peligro

Las zonas de trabajo deben limitarse a la medida mínima necesaria para descartar el peligro para personas y objetos. Pueden configurarse las limitaciones seguras del campo del eje necesarias para la protección personal.



Para más información acerca de la configuración de las limitaciones seguras del campo del eje, consultar las instrucciones de manejo y programación en el capítulo "Configuración de seguridad".
(>>> 13 "Configuración de seguridad" Página 179)

La zona de peligro está compuesta por la zona de trabajo y las distancias de parada del manipulador. En caso de parada, el manipulador frena y se detiene en la zona de peligro. La zona de seguridad es la zona situada fuera de la zona de peligro.

La zona de peligro debe asegurarse mediante dispositivos separadores de protección, p. ej. mediante barreras de luz, cortinas de luz o vallas. En caso de que no se disponga de dispositivos separadores de protección, deberán cumplirse los requisitos para el funcionamiento cooperativo conforme a la norma EN ISO 10218. En las zonas de carga o transferencia de materiales no debe haber ningún punto con riesgo de sufrir cortes o magulladuras.

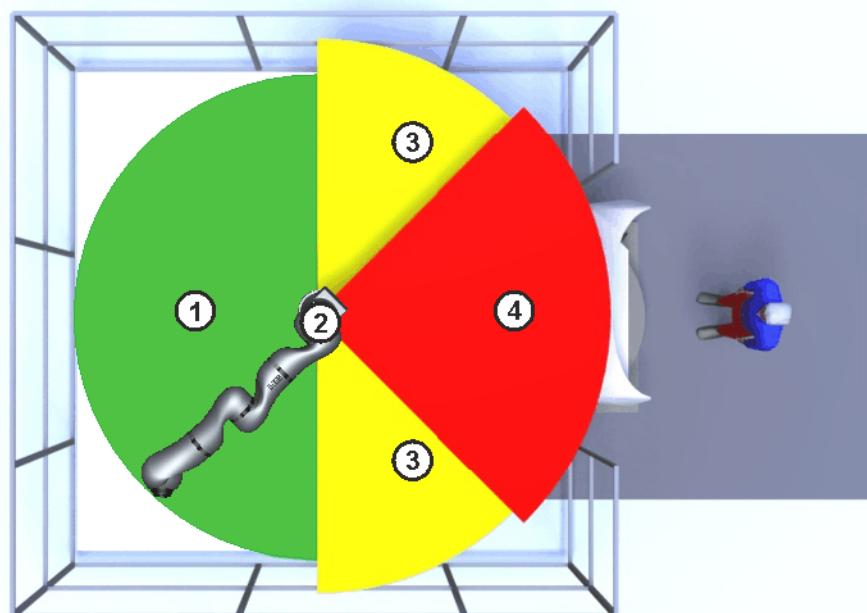


Fig. 3-1: Ejemplo campo del eje 1

1	Zona de trabajo	3	Distancia de parada
2	Manipulador	4	Zona de seguridad

3.2.4 Funciones destinadas a la seguridad

Las siguientes funciones destinadas a la seguridad se encuentran en el robot industrial y están definidas de forma fija:

- Dispositivo de PARADA DE EMERGENCIA
- Dispositivo de validación
- Bloqueo del modo de servicio (a través de un interruptor de llave)

Las siguientes funciones destinadas a la seguridad están preconfiguradas y pueden integrarse en la instalación a través de la interfaz de seguridad de la unidad de control del robot:

- Protección del operario (= conexión para el bloqueo de dispositivos separadores de protección)
- Dispositivo externo de PARADA DE EMERGENCIA
- Parada de seguridad externa 1 (sobre la trayectoria)

Pueden configurarse otras funciones destinadas a la seguridad que no se encuentren disponibles por defecto, p. ej.:

- Dispositivo de validación externo
- Parada de servicio segura externa
- Control del campo de trabajo específico del eje
- Control del campo de trabajo cartesiano
- Control de las zonas de protección cartesiano
- Control de velocidad
- Control de parada
- Control del campo del eje
- Detección de colisión



Para más información acerca de la configuración de las funciones de seguridad, consultar las instrucciones de manejo y programación en el capítulo "Configuración de seguridad". ([>>> 13 "Configuración de seguridad"](#) Página 179)

En los siguientes apartados acerca de la seguridad se describe la configuración por defecto de las funciones de seguridad preconfiguradas.

3.2.4.1 Dispositivo de PARADA DE EMERGENCIA

El dispositivo de PARADA DE EMERGENCIA del robot industrial es el dispositivo de PARADA DE EMERGENCIA del smartPAD. El dispositivo debe pulsarse en situaciones de peligro o en caso de emergencia.

Reacción del robot industrial al pulsarse el dispositivo de PARADA DE EMERGENCIA:

- El manipulador se detiene con una parada de seguridad 1 (sobre la trayectoria).

Para poder seguir con el servicio, debe desenclavarse el pulsador de PARADA DE EMERGENCIA girándolo.

ADVERTENCIA

Las herramientas y otras dispositivos unidos al manipulador que puedan suponer algún peligro deben estar conectados desde la instalación al circuito de PARADA DE EMERGENCIA.
Si no se respeta esta advertencia, pueden ocurrir importantes daños materiales, lesiones graves e incluso la muerte.

Si se utiliza un soporte para el smartPAD que cubra el dispositivo de PARADA DE EMERGENCIA, es necesario instalar un dispositivo de PARADA DE EMERGENCIA externo accesible en cualquier momento.

(>>> 3.2.4.4 "Dispositivo externo de PARADA DE EMERGENCIA" Página 31)

3.2.4.2 Dispositivo de validación

El dispositivo de validación del robot industrial son los pulsadores de validación del smartPAD.

En el smartPAD se encuentran instalados 3 pulsadores de validación. Los pulsadores de validación tienen 3 posiciones:

- No pulsado
- Posición intermedia
- Pulsado a fondo (posición de pánico)

En los modos de servicio de test y KRF, el manipulador únicamente puede desplazarse si el pulsador de validación se mantiene en la posición intermedia.

- Al soltar el pulsador de validación se produce una parada de seguridad 1 (sobre la trayectoria).
- Al presionar el pulsador de validación se activa una parada de seguridad 1.
- Se pueden mantener pulsados al mismo tiempo 2 pulsadores de validación durante unos segundos en la posición intermedia. Esto permite agarrar de un pulsador de validación a otro. Si 2 pulsadores de validación se mantienen pulsados a la vez en la posición intermedia durante más de 15 segundos, se activa una parada de seguridad 1.

Si el pulsador de validación (p. ej., bornes en posición intermedia) funciona incorrectamente, el robot industrial puede detenerse con los métodos siguientes:

- Presionar a fondo el pulsador de validación.
- Accionar el dispositivo de PARADA DE EMERGENCIA.
- Soltar la tecla de arranque.

ADVERTENCIA

Los pulsadores de validación no deben sujetarse con cintas adhesivas o similares ni ser manipulados de cualquier otro modo.
Pueden producirse daños materiales, lesiones graves e incluso la muerte.

3.2.4.3 Protección del operario

La señal "Protección del operario" sirve para bloquear distintos dispositivos separadores de protección, p. ej. puertas de protección. En la configuración por defecto, el servicio automático no es posible sin esta señal. Por lo demás, deben cumplirse los requisitos para el funcionamiento cooperativo conforme a la norma EN ISO 10218.

Reacción del robot industrial en caso de pérdida de señal durante el modo automático, p. ej. Puerta de protección se abre (configuración por defecto):

- El manipulador se detiene con una parada de seguridad 1 (sobre la trayectoria).

En los modos de servicio Manual velocidad reducida (T1) y KRF, la protección del operario no está activada por defecto, es decir, la señal no se evalúa. En el modo de servicio Manual velocidad alta (T2), la protección del operario está activa.

ADVERTENCIA

Tras una pérdida de señal, el modo automático no debe reanudarse únicamente cerrando el dispositivo de seguridad, sino cuando otro dispositivo adicional establezca la señal de protección del operario, p. ej. un pulsador de confirmación. El integrador de sistemas debe encargarse de ello. El objetivo de este paso es evitar una reanudación del modo automático no intencionada hallándose personas dentro de la zona de peligro como, por ej., en caso de una puerta de protección cerrada equivocadamente.

- Este dispositivo adicional se debe implementar de forma que primero se pueda comprobar realmente la zona de peligro. No está permitido ningún dispositivo que no permita esta comprobación (p. ej., confirmación que se produce automáticamente después de cerrar el dispositivo de seguridad).
- Si no se respeta esta medida, pueden ocurrir importantes daños materiales, lesiones graves e incluso la muerte.

3.2.4.4 Dispositivo externo de PARADA DE EMERGENCIA

En todas las estaciones de operación que puedan accionar un movimiento del robot o crear una situación susceptible de ser peligrosa, se debe disponer de dispositivos de PARADA DE EMERGENCIA. El integrador de sistemas debe velar por ello.

Reacción del robot industrial al pulsarse el dispositivo de PARADA DE EMERGENCIA externo (configuración por defecto):

- El manipulador se detiene con una parada de seguridad 1 (sobre la trayectoria).

Pueden conectarse varios dispositivos de PARADA DE EMERGENCIA externos mediante la interfaz de seguridad de la unidad de control del robot. Los dispositivos externos de PARADA DE EMERGENCIA no se incluyen en el volumen de suministro del robot industrial.

3.2.4.5 Parada de seguridad externa 1 (sobre la trayectoria)

La parada de seguridad 1 externa (sobre la trayectoria) puede activarse a través de una entrada en la interfaz de seguridad (configuración por defecto). El estado se mantiene mientras la señal externa permanezca en FALSE. Cuando la señal externa cambia a TRUE, se puede volver a desplazar el manipulador. No es necesaria ninguna confirmación.

3.2.4.6 Dispositivo de validación externo

Los dispositivos de validación externos son necesarios cuando deban situarse varias personas en la zona de peligro del robot industrial.

Pueden conectarse varios dispositivos de validación externos mediante la interfaz de seguridad de la unidad de control del robot. Los dispositivos externos de validación no pertenecen al volumen de suministro del robot industrial.

3.2.4.7 Parada de servicio externa segura

La parada de servicio segura es un control de parada. No detiene el movimiento del robot, sino que controla si los ejes del robot se detienen.

La parada de servicio segura también se puede accionar a través de una entrada de la interfaz de seguridad. El estado se mantiene mientras la señal externa permanezca en FALSE. Cuando la señal externa cambia a TRUE, se puede volver a desplazar el manipulador. No es necesario confirmar.

3.2.5 Causas de reacciones de parada destinadas a la seguridad

Se ejecutan reacciones de parada debido a operaciones realizadas o como reacción ante controles y errores. Las siguientes tablas muestran las reacciones de parada en función del modo de servicio seleccionado.

Resumen

En KUKA Sunrise se distinguen las siguientes causas:

- Causas de definición fija
Las causas de definición fija para las reacciones de parada y la correspondiente categoría de parada están predeterminadas por sistema y no pueden modificarse. Sin embargo, en el marco de la configuración de seguridad específica del usuario, es posible que la reacción de parada utilizada se intensifique.
- Causas específicas del usuario
Además de las causas de definición fija, el usuario puede configurar otras causas para las reacciones de paradas, incluida la categoría de parada correspondiente.



Para más información acerca de la configuración de las funciones de seguridad, consultar las instrucciones de manejo y programación en el capítulo "Configuración de seguridad". ([>>> 13 "Configuración de seguridad"](#) Página 179)

Causas de definición fija

Las siguientes causas de reacciones de parada están definidas de forma fija:

Causa	T1, T2, KRF	AUT
Cambiar el modo de servicio durante el servicio	Parada de seguridad 1 (sobre la trayectoria)	
Soltar el pulsador de validación	Parada de seguridad 1 (sobre la trayectoria)	-
Accionar pulsador de validación (posición de pánico)	Parada de seguridad 1	-
Accionar PARADA DE EMERGENCIA local	Parada de seguridad 1 (sobre la trayectoria)	
Error en el control de seguridad	Parada de seguridad 1	

Causas específicas del usuario

La unidad de control del robot se suministra con una configuración de seguridad que se activa durante la primera puesta en servicio. Dicha configuración contiene las causas específicas del usuario preconfiguradas por KUKA para las reacciones de parada (además de las causas de definición fija).

Causa	T1, KRF	T2, AUT
Abrir la puerta de protección (protección del operario)	-	Parada de seguridad 1 (sobre la trayectoria)

Al crear un nuevo proyecto Sunrise se genera de forma automática una configuración de seguridad específica del proyecto. Dicha configuración contiene las causas específicas del usuario preconfiguradas por KUKA para las reacciones de parada (además de las causas de definición fija).



Al transferir el proyecto Sunrise a la unidad de control del robot, la configuración de seguridad establecida de fábrica se sobrescribe por la configuración de seguridad específica del proyecto. De este modo, es necesaria una activación de la configuración de seguridad.
Para más información acerca de la activación de la función de seguridad, consultar las instrucciones de manejo y programación en el capítulo "Configuración de seguridad". ([>>> 13 "Configuración de seguridad" Página 179](#))

Causa	T1, KRF	T2, AUT
Abrir la puerta de protección (protección del operario)	-	Parada de seguridad 1 (sobre la trayectoria)
Activar PARADA DE EMERGENCIA externa	Parada de seguridad 1 (sobre la trayectoria)	
Parada de seguridad externa	Parada de seguridad 1 (sobre la trayectoria)	

3.2.6 Funciones no destinadas a la seguridad

3.2.6.1 Selección de modos de servicio

El robot industrial puede utilizarse en los siguientes modos de servicio:

- Manual Velocidad reducida (T1)
- Manual Velocidad alta (T2)
- Automático (AUT)
- Desplazamiento controlado del robot (KRF)

Modo de servicio	Utilización	Velocidades
T1	Programación, programación por aprendizaje y prueba de programas	<ul style="list-style-type: none"> ■ Verificación del programa: velocidad reducida programada, máximo 250 mm/s ■ Modo de servicio manual: velocidad de desplazamiento manual, máximo 250 mm/s
T2	Prueba de programas Solo es posible con la puerta de protección cerrada	<ul style="list-style-type: none"> ■ Verificación del programa: velocidad programada ■ Modo de servicio manual: no es posible

Modo de servicio	Utilización	Velocidades
AUT	Ejecución automática de programas Para robots industriales con y sin unidad de control superior	<ul style="list-style-type: none"> ■ Servicio con programa: velocidad programada ■ Modo de servicio manual: no es posible
KRF	<ul style="list-style-type: none"> ■ Extraer el robot industrial de una zona vulnerable cartesiana o específica de eje. ■ Extraer el robot industrial de una zona vulnerable de la orientación de herramienta ■ Liberar el robot industrial de situaciones de bloqueo en caso de vulneración de límites de fuerza o de momento ■ Desplazar el robot industrial si con el control de velocidad cartesiano activo, existe una pérdida de ajuste para al menos un sensor de posición <p>KRF es un modo de servicio que está disponible si el control de seguridad detiene al robot industrial por alguna de las siguientes causas:</p> <ul style="list-style-type: none"> ■ El robot industrial vulnera un espacio controlado de forma segura. ■ La orientación de la herramienta destinada a la seguridad está fuera del sector controlado de forma segura. ■ El robot industrial vulnera un límite de fuerza o de momento controlado de forma segura. ■ Un sensor de posición no está ajustado con el control cartesiano de velocidad activo. 	<ul style="list-style-type: none"> ■ Servicio con programa: velocidad reducida programada, máximo 250 mm/s ■ Modo de servicio manual: velocidad de desplazamiento manual, máximo 250 mm/s

3.2.6.2 Interruptor de final de carrera de software

Los campos de todos los ejes del manipulador se encuentran limitados por medio de interruptores de final de carrera de software no destinados a la seguridad. Estos interruptores de final de carrera de software solo sirven como protección de la máquina y están preajustados de forma que el manipulador se detenga de forma regulada al sobrepasar el límite del eje y el sistema mecánico no resulte dañado.

3.3 Equipamiento de protección adicional

3.3.1 Modo paso a paso

En los modos de servicio Manual velocidad reducida (T1), Manual velocidad alta (T2) y KRF, la unidad de control del robot únicamente puede ejecutar un programa en el modo tecleado. Esto significa: para ejecutar un programa, deben mantenerse pulsados un interruptor de validación y la tecla de arranque.

- Al soltar el pulsador de validación del smartPAD se produce una parada de seguridad. (>>> 3.2.5 "Causas de reacciones de parada destinadas a la seguridad" Página 32)
- Al pulsar a fondo el pulsador de validación del smartPAD se produce una parada de seguridad 1.
- Al soltar la tecla de arranque, se produce una parada de la categoría 1 (sobre la trayectoria).

3.3.2 Identificaciones en el robot industrial

Todas las placas, indicaciones, símbolos y marcas son piezas integrantes del robot industrial relevantes para la seguridad. No deben modificarse ni quitarse en ningún caso.

Placas de identificación en el robot industrial son:

- Placas características
- Indicaciones de advertencia
- Símbolos de seguridad
- Rótulos
- Identificación de cables
- Placas de características



Puede encontrar más información en los datos técnicos de las instrucciones de servicio o de montaje de los componentes del robot industrial.

3.3.3 Dispositivos de seguridad externos

Los dispositivos de seguridad se encargan de impedir el acceso de personas a la zona de peligro del robot industrial. Por lo demás, deben cumplirse los requisitos para el funcionamiento cooperativo conforme a la norma EN ISO 10218. El integrador de sistemas debe velar por ello.

Los dispositivos separadores de protección deben cumplir los siguientes requisitos:

- Cumplen los requisitos de la norma EN 953.
- Impiden el acceso de personas a la zona de peligro y no pueden sortearse fácilmente.
- Están bien fijados y resisten las fuerzas mecánicas previsibles provenientes del servicio y del entorno.
- No suponen ni causan un peligro por sí mismos.
- Respetar la distancia mínima prescrita a la zona de peligro.

Las puertas de protección (puertas de mantenimiento) deben cumplir los siguientes requisitos:

- El número de puertas se limita al mínimo necesario.
- Los enclavamientos (p. ej. los interruptores de las puertas) deben estar conectados con las entradas configuradas de protección del operario de la unidad de control del robot.
- Los dispositivos de conmutación, los interruptores y el tipo de circuito cumplen los requisitos del nivel de eficiencia d y la categoría 3 de la norma EN 13849-1.
- En función del peligro, la puerta de protección además se debe asegurar con un cierre que solo permita abrir la puerta cuando el manipulador esté parado por completo.

- El dispositivo para activar la señal de protección del operario, p. ej., el botón para confirmar la puerta de protección, está instalado fuera de la zona delimitada por los dispositivos de seguridad.



En las correspondientes normas y prescripciones puede encontrarse información adicional. Ésta incluye también la norma EN 953.

Otros dispositivos de protección

Otros dispositivos de protección deben ser integrados a la instalación en concordancia con las correspondientes normas y prescripciones.

3.4 Medidas de seguridad

3.4.1 Medidas generales de seguridad

El robot industrial solo deberá utilizarse para los fines previstos y deberá encontrarse en un estado idóneo desde el punto de vista técnico respetando todas las medidas de seguridad. En caso de realizar alguna acción indebida, pueden provocarse daños personales y materiales.

Aún estando la unidad de control del robot desconectada y asegurada, el robot industrial puede efectuar movimientos inesperados. Por un montaje incorrecto (p. ej. sobrecarga) o defectos mecánicos (p. ej. freno defectuoso), el manipulador puede descender. Si se ha de trabajar con el robot industrial desconectado, el manipulador debe desplazarse previamente a una posición tal que no pueda moverse por sí mismo con o sin influencia de la carga montada. Si esto no fuese posible, debe asegurarse el manipulador de forma adecuada.



PELIGRO El robot industrial puede causar lesiones o daños materiales si las funciones o dispositivos de seguridad no están en servicio. En caso de que se hayan desmontado o desactivado las funciones y dispositivos de seguridad, no se debe hacer funcionar el robot industrial.



ADVERTENCIA Permanecer debajo del sistema mecánico del robot puede causar lesiones graves e incluso la muerte, en especial si con el robot industrial se mueven objetos que puedan soltarse (p. ej. de una garra). Por este motivo está terminantemente prohibido permanecer debajo del sistema mecánico del robot.

smartPAD

El explotador debe asegurarse de que únicamente las personas autorizadas manejen el robot industrial con el smartPAD.

Modificaciones

Si se ha efectuado alguna modificación en el robot industrial, se debe comprobar que quede garantizado el nivel de seguridad necesario. Para esta comprobación se deben tener en cuenta las disposiciones vigentes nacionales y regionales en materia de protección laboral. Además, debe comprobarse también que todas las funciones de seguridad se activan correctamente.

Los programas nuevos o modificados siempre se deben probar primero en el modo de servicio Manual Velocidad reducida (T1).

Tras efectuar alguna modificación en el robot industrial, los programas existentes siempre deben probarse primero en el modo de servicio Manual Velocidad reducida (T1). Esto es válido para todos los componentes del robot industrial y también incluye las modificaciones de software y los ajustes de configuración.

Con la unidad de control en marcha, el robot no puede conectarse y desconectarse.

Averías

En caso de avería en el robot industrial se debe proceder del modo siguiente:

- Desconectar la unidad de control del robot y asegurarla contra una reconnection indebida (p. ej., con un candado).
- Informar sobre la avería mediante un cartel con la indicación correspondiente.
- Llevar un registro de las averías.
- Subsanar la avería y verificar el funcionamiento.

3.4.2 Transporte**Manipulador**

Debe respetarse la posición de transporte prescrita para el manipulador. El transporte debe realizarse conforme a las instrucciones de servicio o las instrucciones de montaje del manipulador.

Durante el transporte, evitar vibraciones o golpes para no dañar el sistema mecánico del robot.

Unidad de control del robot

Debe respetarse la posición de transporte prescrita para la unidad de control del robot. El transporte debe realizarse conforme a las instrucciones de servicio o las instrucciones de montaje de la unidad de control del robot.

Durante el transporte, evitar vibraciones o golpes para no dañar la unidad de control del robot.

3.4.3 Puesta en servicio y reanudación del servicio

Antes de la primera puesta en servicio de una instalación o un dispositivo, debe realizarse una comprobación para asegurarse de que la instalación o el dispositivo estén completos y en condiciones de funcionamiento, que pueden ser operados en condiciones de seguridad y que se pueden detectar posibles daños.

Para esta comprobación se deben tener en cuenta las disposiciones vigentes nacionales y regionales en materia de protección laboral. Además, debe comprobarse también que todas las funciones de seguridad funcionan correctamente.



La contraseña para la activación de la configuración de seguridad ha de modificarse antes de la puesta en servicio. Dicha contraseña solo debe confiarse a técnicos de puesta en servicio con seguridad formados y que estén autorizados para activar la configuración de seguridad.
 (>>> 13.7.3 "Modificar la contraseña para la activación de la configuración de seguridad" Página 201)



PELIGRO La unidad de control del robot se encuentra preconfigurada para el robot industrial correspondiente. En caso de cables intercambiados, el manipulador puede recibir datos erróneos y provocar por ello daños a personas u objetos. Si una instalación se compone de varios manipuladores, conectar siempre los cables de unión al manipulador y a la correspondiente unidad de control del robot.



Cuando se integran componentes adicionales (p. ej. cables) en el sistema del robot industrial que no pertenecen al volumen de suministro de KUKA Laboratories GmbH, el explotador se hace responsable de que dichos componentes no interfieran en las funciones de seguridad del robot o lo pongan fuera de servicio.

AVISO

Cuando la temperatura interior del armario de la unidad de control del robot difiere demasiado de la temperatura ambiente, se puede formar agua de condensación el cual podría causar daños en la parte eléctrica. La unidad de control del robot recién debe ser puesta en servicio cuando la temperatura interior del armario se haya aproximado a la temperatura ambiente.

Prueba de funcionamiento

Antes de la puesta en servicio o de la reanudación del servicio deben realizarse las siguientes comprobaciones:

Prueba general:

Asegurarse de que:

- El robot industrial está correctamente colocado y fijado conforme a las indicaciones incluidas en la documentación.
- Sobre el robot industrial no hay cuerpos extraños, ni piezas sueltas o defectuosas.
- Todos los dispositivos de seguridad necesarios están correctamente instalados y en condiciones de funcionamiento.
- Los valores de conexión del robot industrial coinciden con la tensión y la estructura de la red local.
- El cable de puesta a tierra y el cable equipotencial están bien tendidos y correctamente conectados.
- Los cables de unión están correctamente conectados y los conectores bloqueados.

Comprobación de las funciones de seguridad:

Mediante una prueba de funcionamiento se debe garantizar que todas las funciones destinadas a la seguridad trabajan correctamente.

(>>> 13.11 "Resumen de la recepción de seguridad" Página 230)

Comprobación de los componentes mecánicos y electromecánicos relevantes para la seguridad:

Las siguientes comprobaciones deben realizarse antes de la puesta en servicio y al menos cada 12 meses, siempre y cuando no se haya determinado otra cosa según la valoración de riesgos en el puesto de trabajo.

- Pulsar el dispositivo de PARADA DE EMERGENCIA en el smartPAD. Debe mostrarse un mensaje en el smartPAD que indique que se ha accionado la PARADA DE EMERGENCIA y no debe haber ningún mensaje de error respecto al dispositivo de PARADA DE EMERGENCIA.
- En los 3 pulsadores de validación del smartPAD y en el pulsador de validación de la brida de medios Touch (si existe) Desplazar el robot en el modo de prueba y soltar el pulsador de validación. El movimiento del robot debe detenerse y no se debe mostrar ningún mensaje de error relativo al dispositivo de validación. Si el estado del pulsador de validación está configurado a una salida, la comprobación también se puede realizar a través de la salida.
- En los 3 pulsadores de validación del smartPAD y en el pulsador de validación de la brida de medios Touch (si existe) Desplazar el robot en el modo de prueba y pulsar a fondo el pulsador de validación. El movimiento del robot debe detenerse y no se debe mostrar ningún mensaje de error relativo al dispositivo de validación. Si el estado del pulsador de validación está configurado a una salida, la comprobación también se puede realizar a través de la salida.
- Girar el interruptor de llave del smartPAD hacia la derecha y hacia su posición original. No se debe mostrar ningún mensaje de error en el smartPAD.

- Comprobar la capacidad de desconexión de las salidas seguras a través de la desconexión y la nueva conexión de la unidad de control del robot. Después de la conexión, no se debe indicar ningún mensaje de error relativo a una salida segura.



Si la puesta en servicio de la instalación se lleva a cabo de forma incompleta, será necesario aplicar y documentar medidas compensatorias para reducir el riesgo como p.ej. la colocación de una valla de protección o un cartel de advertencia, el enclavamiento del interruptor principal,etc. La puesta en servicio se lleva a cabo de forma incompleta cuando, por ejemplo, no se han implementado todos los controles de seguridad necesarios o no se ha comprobado que todas las funciones de seguridad funcionan correctamente.

Realización de la prueba de funcionamiento de los frenos:

Para el KUKA LBR iiwa (todas las variantes) es necesario comprobar, mediante una prueba de frenos que se ha de realizar regularmente, si los frenos de todos los ejes aplican un momento de frenado suficiente.

(>>> 8 "Prueba de frenos" Página 119)

La prueba de frenos ha de realizarse durante la puesta en servicio y la reanudación del servicio del robot industrial.

Durante el servicio es necesario realizar la prueba de frenos a diario independientemente del tiempo de servicio y el tipo de aplicación.

3.4.4 Modo de servicio manual

El servicio manual es el modo de servicio indicado para realizar los trabajos de ajuste. Se consideran trabajos de ajuste todos los trabajos que deban llevarse a cabo en el robot industrial para poder ser operado en el modo automático. Son trabajos de ajuste:

- Modo paso a paso
- Programación por aprendizaje
- Verificación del programa

En el modo de servicio manual deben tenerse en cuenta los siguientes aspectos:

- Los programas nuevos o modificados siempre se deben probar primero en el modo de servicio Manual velocidad reducida (T1).
- Ni las herramientas ni el manipulador deben tocar nunca la valla de seguridad o sobresalir de la misma.
- Las piezas de trabajo, herramientas u otros objetos no deben quedar apretados por el desplazamiento del robot industrial, ni tampoco provocar cortocircuitos o caerse.
- Todos los trabajos de ajuste deben realizarse, en la medida de lo posible, fuera del espacio delimitado por los dispositivos de seguridad.

En caso de que los trabajos de ajuste deban realizarse dentro del espacio delimitado con dispositivos de seguridad, se deberán tener en cuenta los siguientes puntos.

En el modo de servicio **Manual velocidad reducida (T1)**:

- Si se puede evitar, no debe hallarse ninguna otra persona dentro de la zona delimitada por los dispositivos de seguridad.

Si es imprescindible que varias personas permanezcan dentro de la zona delimitada por los dispositivos de seguridad, se debe tener en cuenta lo siguiente:

- Todas las personas deben tener a su disposición un dispositivo de validación.
- Todas las personas deben tener un contacto visual sin obstáculos con el robot industrial.
- Debe existir contacto visual entre todas las personas implicadas.
- El operario debe situarse en una posición desde la cual pueda visualizar la zona de peligro y, así, poder evitar posibles peligros.

En el modo de servicio **Manual velocidad alta (T2)**:

- Este modo de servicio solo puede utilizarse cuando se requiera la realización de una prueba con velocidad más alta que la del modo de servicio Manual velocidad reducida.
- Este modo de servicio no permite la programación por aprendizaje.
- Antes de iniciar la prueba, el operario debe asegurarse de que los dispositivos de validación están en condiciones de funcionamiento.
- No debe haber ninguna otra persona dentro de la zona delimitada por los dispositivos de seguridad. El operario debe encargarse de ello.

3.4.5 Modo de servicio automático

El modo de servicio automático solo es posible si se cumplen las siguientes medidas de seguridad:

- Todos los dispositivos de seguridad y protección están debidamente montados y en condiciones de funcionamiento.
- En la instalación no se encuentra ninguna persona. O bien se cumplen los requisitos para el funcionamiento cooperativo conforme a la norma EN ISO 10218.
- Se cumplen los procedimientos definidos para la ejecución de los trabajos.

Cuando el manipulador se detiene sin motivo aparente, se debe acceder a la zona de peligro cuando se haya activado la función de PARADA DE EMERGENCIA.

3.4.6 Mantenimiento y reparación

Tras haber realizado trabajos de mantenimiento o reparación, comprobar si el nivel de seguridad necesario está garantizado. Para esta comprobación se deben tener en cuenta las disposiciones vigentes nacionales y regionales en materia de protección laboral. Además, debe comprobarse también que todas las funciones de seguridad funcionan correctamente.

El mantenimiento y las reparaciones tienen la finalidad de asegurar que se mantiene el estado funcional o que se restablece en caso de avería. La reparación comprende la localización de averías y su subsanación.

Las medidas de seguridad que se deben tomar al realizar trabajos en el robot industrial son:

- Efectuar los trabajos fuera de la zona de peligro. En caso de que se deban efectuar trabajos dentro de la zona de peligro, el operario debe implementar medidas adicionales de seguridad para garantizar la seguridad de las personas.
- Desconectar el robot industrial y asegurarlo contra una reconexión indebida (p. ej., con un candado). En caso de que se deban realizar trabajos con la unidad de control del robot conectada, el explotador debe implementar medidas de seguridad adicionales para garantizar la seguridad de las personas.

- En caso de que los trabajos deban realizarse con la unidad de control del robot conectada, deberán realizarse exclusivamente en el modo de servicio T1.
- Informar por medio de un cartel de que se están realizando trabajos en la instalación. Este cartel deberá mantenerse también si se interrumpen temporalmente los trabajos.
- Los dispositivos de PARADA DE EMERGENCIA deben mantenerse activos. Si para realizar los trabajos de mantenimiento o de reparación es necesario desactivar alguna función o dispositivo de seguridad, deberá restablecerse de inmediato la protección.

 **PELIGRO**

Antes de realizar trabajos en componentes del sistema de robot que estén bajo tensión, debe desconectarse el interruptor principal y asegurarse contra una reconexión. A continuación debe controlarse que los componentes no estén bajo tensión. Antes de realizar trabajos en componentes bajo tensión, no basta con activar una PARADA DE EMERGENCIA/parada de seguridad o con desconectar los accionamientos, ya que el sistema de robot no es desconectado de la red. Hay componentes que continúan estando bajo tensión. Existe peligro de muerte o de sufrir lesiones graves.

Los componentes defectuosos deben sustituirse por componentes nuevos con el mismo número de artículo o por componentes que KUKA Laboratories GmbH considere equivalentes.

Los trabajos de limpieza y mantenimiento deben efectuarse de conformidad con las instrucciones de servicio.

Unidad de control del robot

Aún con la unidad de control del robot desconectada, pueden encontrarse partes bajo tensión conectadas a los dispositivos periféricos. Por consiguiente, las fuentes externas se deben desconectar cuando haya que efectuar trabajos en la unidad de control del robot.

Al efectuar cualquier tarea en los componentes de la unidad de control del robot, se deben respetar las prescripciones relativas a componentes sometidos a riesgos electrostáticos.

Después de desconectar la unidad de control del robot, los distintos componentes pueden contener, durante varios minutos, tensiones superiores a 60 V. Para evitar lesiones con peligro de muerte, durante ese lapso de tiempo no deben efectuarse tareas en el robot industrial.

Debe evitarse la penetración de restos de agua y polvo en la unidad de control del robot.

3.4.7 Cese del servicio, almacenamiento y eliminación de residuos

El cese de servicio, el almacenamiento y la eliminación del robot industrial deberán llevarse a cabo de conformidad con las leyes, prescripciones y normas específicas del país.

3.4.8 Medidas de seguridad para el "Single Point of Control"

Vista general

Cuando el robot industrial utiliza determinados componentes, deben aplicarse medidas de seguridad para poner en práctica por completo el principio del "Single Point of Control" (SPOC).

Componentes:

- Herramientas para configurar los sistemas de bus con función online



Puede que sea necesaria la aplicación de otras medidas de seguridad. Este aspecto debe aclararse en función del caso y es responsabilidad del explotador de la instalación.

Puesto que los estados de seguridad de los actuadores que se encuentran en la periferia de la unidad de control del robot únicamente los conoce el integrador del sistema, es su responsabilidad poner dichos actuadores en estado seguro.

T1, T2, KRF

En los modos de servicio T1, T2 y KRF solo puede activarse un movimiento del robot si en el smartPAD se encuentra pulsado un pulsador de validación.

Herramientas para configurar los sistemas de bus

Si estos componentes disponen de función online, se podrán modificar programas, salidas u otros parámetros de la unidad de control del robot a través de accesos de escritura sin que lo noten las personas que se hallan en la instalación.

- KUKA Sunrise.Workbench
- WorkVisual de KUKA
- Herramientas de otros fabricantes

Medidas de seguridad:

- En los modos de servicio de test los programas, salidas u otros parámetros de la unidad de control del robot no pueden modificarse con estos componentes.

3.5 Normas y prescripciones aplicadas

Nombre	Definición	Edición
2006/42/CE	Directiva relativa a las máquinas: Directiva 2006/42/CE del Parlamento Europeo y de la Comisión, de 17 de mayo de 2006, relativa a las máquinas y por la que se modifica la Directiva 95/16/CE (refundición)	2006
2004/108/CE	Directiva sobre compatibilidad electromagnética: Directiva 2004/108/CE del Parlamento Europeo y de la Comisión, de 15 de diciembre de 2004, relativa a la aproximación de las legislaciones de los Estados miembros en materia de compatibilidad electromagnética y por la que se deroga la Directiva 89/336/CEE	2004
EN ISO 13850	Seguridad de las máquinas: Principios generales de configuración para PARADA DE EMERGENCIA	2008
EN ISO 13849-1	Seguridad de máquinas: Componentes de seguridad de los sistemas de control. Parte 1: Principios generales de configuración	2008
EN ISO 13849-2	Seguridad de máquinas: Componentes de seguridad de los sistemas de control. Parte 2: Validación	2012

EN ISO 12100	Seguridad de las máquinas: Principios generales de configuración, evaluación y reducción del riesgo	2010
EN ISO 10218-1	Robot industrial: Seguridad Indicación: Contenido cumple con ANSI/RIA R.15.06-2012, parte 1	2011
EN 614-1	Seguridad de máquinas: Principios de diseño ergonómico. Parte 1: Terminología y principios generales	2009
EN 61000-6-2	Compatibilidad electromagnética (CEM): Parte 6-2: Normas genéricas. Inmunidad en entornos industriales	2005
EN 61000-6-4 + A1	Compatibilidad electromagnética (CEM): Parte 6-4: Normas genéricas. Norma de emisión en entornos industriales	2011
EN 60204-1 + A1	Seguridad de máquinas: Equipo eléctrico de las máquinas. Parte 1: Requisitos generales	2009

4 Instalación de KUKA Sunrise.Workbench

4.1 Requisitos del sistema PC

Hardware	Requisitos mínimos
	<ul style="list-style-type: none"> ■ PC con procesador Pentium IV, mínimo 1500 MHz ■ 1 GB de memoria de trabajo ■ 1 GB de memoria libre del disco duro ■ Tarjeta gráfica compatible DirectX8 con una resolución de 1024x768 píxeles
Software	Requisito mínimo
	<ul style="list-style-type: none"> ■ Windows 7 <p>Para la configuración de bus se necesita el siguiente software:</p> <ul style="list-style-type: none"> ■ WorkVisual 3.0

4.2 Instalar Sunrise.Workbench

Condición previa	<ul style="list-style-type: none"> ■ Derechos de administrador local
Procedimiento	<ol style="list-style-type: none"> 1. Iniciar la instalación a través del archivo Setup.exe. 2. Seleccionar el idioma y confirmar Ok. Se abre el asistente de instalación Sunrise Workbench Setup. 3. Avanzar a la pantalla siguiente con Continuar >. 4. Aceptar las condiciones de la licencia. Avanzar a la pantalla siguiente con Continuar >. 5. En el campo Directorio: se indica el directorio por defecto para la instalación. Mediante Seleccionar... se puede seleccionar otro directorio de instalación. Avanzar a la pantalla siguiente con Continuar >. 6. Si no debe crearse un acceso directo desde el escritorio para Sunrise.Workbench: Desactivar la opción Escritorio (quitar el símbolo de confirmación). 7. Avanzar a la pantalla siguiente con Continuar > y hacer clic en Instalar. Sunrise.Workbench se instala. 8. Cuando haya finalizado la instalación, hacer clic en Finalizar para cerrar el asistente de instalación. A continuación, se inicia directamente Sunrise.Workbench. Si no se desea que esto ocurra, desactivar la opción Ejecutar Sunrise Workbench (quitar símbolo de confirmación).

4.3 Desinstalar Sunrise.Workbench

 Se recomienda archivar todos los datos antes de desinstalar un software.

- Condición previa**
- Derechos de administrador local
- Procedimiento**
1. En el menú de inicio de Windows seleccionar **Panel de control (> Programas)** > **Programas y funciones**. Se muestra una lista con los programas instalados.
 2. Marcar la entrada **Sunrise Workbench**. Hacer clic en **Desinstalar** y responder con **Sí** a la pregunta de seguridad.
- Procedimiento alternativo**
- Abrir en el menú de inicio de Windows el directorio de instalación indicado durante la instalación y hacer clic en **Uninstall**.

4.4 Instalar paquete de idioma en Sunrise.Workbench

Descripción La interfaz de usuario de Sunrise.Workbench actualmente está disponible en los siguientes idiomas:

Alemán	Italiano
Inglés	Español
Francés	

Los idiomas que solo estén disponibles después del suministro de un software se pueden instalar posteriormente.

- Condición previa**
- Derechos de administrador local
 - El paquete de idiomas que contiene el idioma deseado está disponible.
- Procedimiento**
1. Cerrar Sunrise.Workbench.
 2. Ejecutar archivo **SWB_LanguagePack-<Version>-Setup.exe**. El paquete de idioma se instala.
 3. Volver a abrir Sunrise.Workbench. Sunrise.Workbench se inicia automáticamente en el idioma local ajustado en Windows.

5 Manejo de KUKA Sunrise.Workbench

5.1 Iniciar Sunrise.Workbench

Procedimiento

1. Hacer doble clic sobre el icono de **Sunrise Workbench** situado en el escritorio.

Alternativa:

Abrir el directorio de instalación en el menú de inicio de Windows y hacer doble clic en **Sunrise Workbench**.

Se abre la ventana **Programa de inicio para la zona de trabajo**.

2. En el campo **Zona de trabajo** indicar el directorio para la zona de trabajo en el que se guardan proyectos.

- Se indica un directorio. Mediante **Examinar...** se puede cambiar el directorio.
- Si en el siguiente inicio de Sunrise.Workbench no se debe volver a preguntar la zona de trabajo, activar la opción **Utilizar este valor como valor estándar[...]** (marcar el símbolo de confirmación).

Confirmar ajustes con **OK**.

3. En el primer inicio de Sunrise.Workbench se abre una pantalla de bienvenida. Aquí hay diferentes opciones disponibles.

- Hacer clic en **Zona de trabajo** para abrir la interfaz de usuario de Sunrise.Workbench.
(>>> 5.2 "Vista general de la interfaz de usuario de Sunrise.Workbench" Página 47)
- Hacer clic en **Nuevo proyecto Sunrise** para crear directamente un nuevo proyecto Sunrise. Se abre el asistente para la creación de proyectos.
(>>> 5.3 "Crear proyecto Sunrise con plantilla" Página 51)

5.2 Vista general de la interfaz de usuario de Sunrise.Workbench

La interfaz de usuario de KUKA Sunrise.Workbench se compone de varias vistas. La agrupación de varias vistas se denomina perspectiva. KUKA Sunrise.Workbench ofrece diferentes perspectivas preconfiguradas.

Por defecto se abre la perspectiva **Programación**. También se pueden mostrar otras perspectivas.

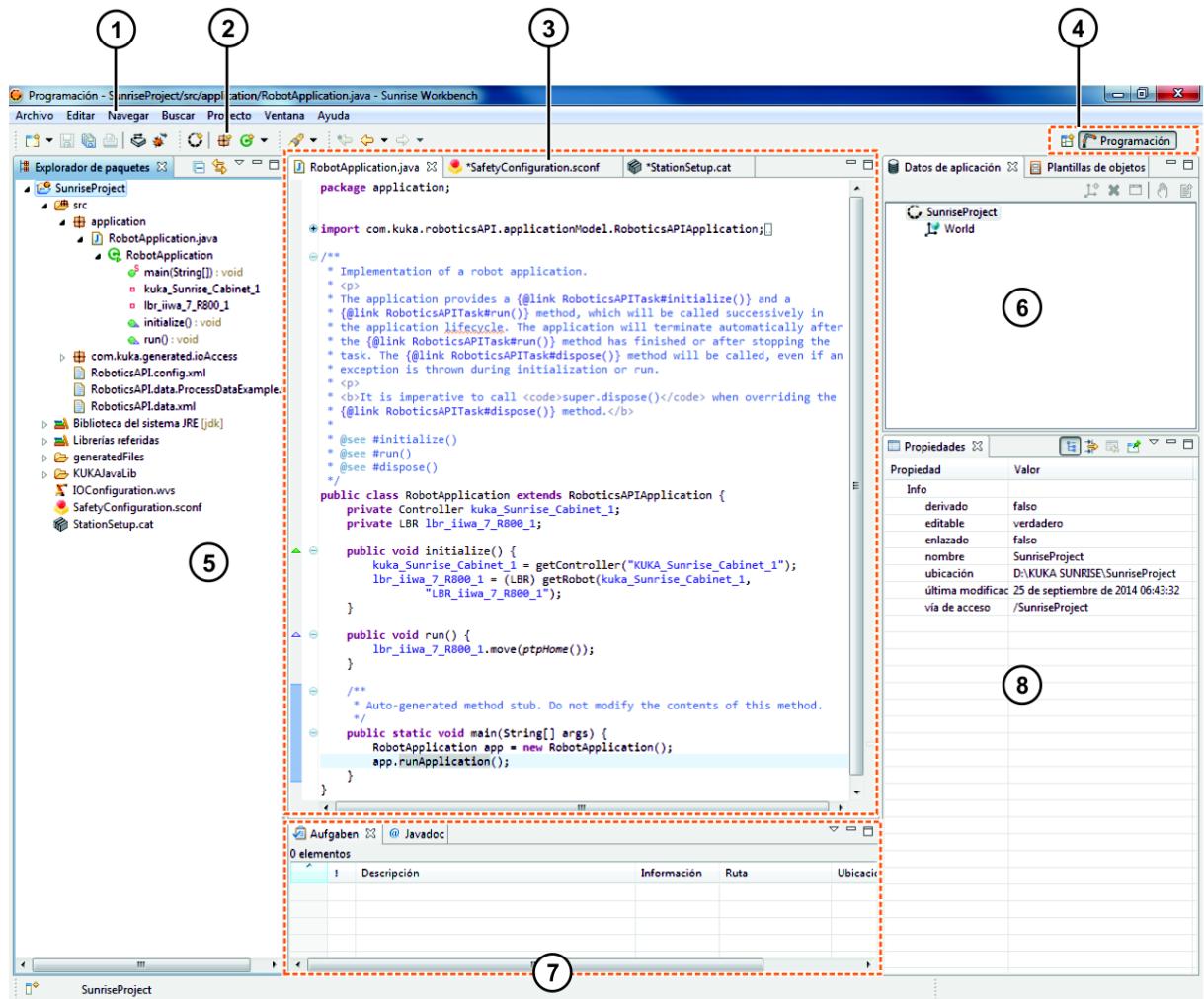


Fig. 5-1: Vista general de la interfaz de usuario - Perspectiva "Programación"

Pos.	Descripción
1	Barra de menús
2	Barras de herramientas (>>> 5.2.3 "Barras de herramientas" Página 50)
3	Zona de los editores Aquí se muestran y se procesan los archivos abiertos, p. ej. aplicaciones del robot.
4	Selección de perspectivas Aquí se puede cambiar entre diferentes perspectivas ya utilizadas, haciendo clic en el nombre de la perspectiva correspondiente o seleccionándolo mediante el símbolo Abrir perspectiva . (>>> 5.2.2 "Visualizar diferentes perspectivas de la interfaz de usuario" Página 49)
5	Vista Explorador de paquetes La vista contiene los proyectos creados y los archivos correspondientes.

Pos.	Descripción
6	Vistas Datos de la aplicación y Plantillas de objetos <ul style="list-style-type: none"> ■ Datos de la aplicación: La vista muestra los Frames creados para un proyecto en una estructura de árbol. ■ Plantillas de objetos: La vista muestra los objetos geométricos, las herramientas y las piezas creadas para un proyecto en una estructura de árbol.
7	Vistas Tareas y Javadoc <ul style="list-style-type: none"> ■ Tareas: Indicación de las tareas que un usuario ha creado ■ Javadoc: Indicación de la información Javadoc para los elementos marcados de un aplicación Java
8	Vista Propiedades Si un objeto, p. ej. un proyecto, un Frame o una herramienta, se selecciona en una vista, se mostrarán sus propiedades.

5.2.1 Disponer vistas de otro modo

- Procedimiento**
1. Sujete la vista de la barra de título con el botón izquierdo del ratón pulsado y arrástrela hasta la interfaz de usuario.
Las posibles posiciones para la vista se muestran mediante un marco gris.
 2. Soltar el botón del ratón cuando esté marcada la posición en la que debe insertarse la vista.

5.2.2 Visualizar diferentes perspectivas de la interfaz de usuario

Descripción La interfaz de usuario se puede visualizar con diferentes perspectivas. Éstas se pueden seleccionar mediante la secuencia de menú **Ventana > Abrir perspectiva** o a través del símbolo **Abrir perspectiva**.

Las perspectivas están ajustadas para diferentes puntos esenciales de trabajo:

Perspectiva	Centro de gravedad
Programación	La perspectiva contiene vistas adecuadas para la edición de proyectos Sunrise. Por ejemplo para la configuración de la estación, la configuración de seguridad y el desarrollo la aplicación.
Depurar	La perspectiva contiene vistas adecuadas para la búsqueda de errores y la eliminación de errores de programación.

Las perspectivas se pueden adaptar a las necesidades del usuario. Ejemplos:

- Crear perspectivas propias
- Visualizar / ocultar vistas
- Visualizar / ocultar menús
- Visualizar / ocultar opciones de menú

Existe la posibilidad de guardar la perspectiva adaptada como ajuste por defecto de la perspectiva o con un nombre propio.

Procedimiento

Mostrar las vistas en la perspectiva actual:

- Seleccionar la secuencia de menú **Ventana > Mostrar vistas** y seleccionar la vista deseada.

A través de la opción de menú **Otras...** se pueden seleccionar otras vistas.

Restablecer la perspectiva actual al ajuste por defecto:

- Seleccionar la secuencia de menú **Ventana > Restablecer perspectiva...** y responder con **Sí** a la pregunta de seguridad.

Guardar la perspectiva definida por el usuario:

1. Seleccionar la secuencia de menú **Ventana > Guardar perspectiva como....**
2. En el campo **Nombre** introducir el nombre para la perspectiva y confirmar con **OK**.

Si ya está seleccionada una perspectiva existente y se sobrescribe, la perspectiva se abrirá en el futuro con estos ajustes.

5.2.3 Barras de herramientas

Los botones que están disponibles por defecto en las barras de símbolos dependen de la perspectiva activa. Aquí se describen los botones de la perspectiva **Programación**.

Programación

Símbolo	Nombre / descripción
	Nuevo Abre el asistente para la creación de nuevos documentos.
	Guardar Guarda el archivo actualmente abierto y seleccionado.
	Guardar todo Guarda todos los archivos y proyectos que han sido procesados desde el último almacenamiento.
	Imprimir Abre el menú para imprimir un archivo.
	Sincronizar proyecto Sincroniza el proyecto seleccionado con la unidad de control del robot.
	Depurar proyecto Establece una conexión remota con la unidad de control del robot para depurar una aplicación durante el servicio en ejecución.
	Proyecto Sunrise Abre el asistente para la creación de un nuevo proyecto Sunrise.
	Nuevo paquete Java Abre el asistente para la creación de un nuevo paquete Java en el proyecto seleccionado.
	Nueva clase Java Abre el asistente para la creación de una nueva clase Java en el proyecto seleccionado.

Símbolo	Nombre / descripción
	Buscar Abre el asistente para buscar palabras o componentes de texto.
	Última posición de procesamiento Conmuta la última posición de procesamiento en el archivo actualmente abierto y seleccionado.
	Volver a ... Retrocede hacia los pasos de procesamiento anteriores.
	Continuar hacia ... Vuelve a avanzar hacia los pasos de procesamiento siguientes.

5.3 Crear proyecto Sunrise con plantilla

Procedimiento

1. Seleccionar la secuencia del menú **Archivo > Nuevo > Proyecto Sunrise**. Se abre el asistente para la creación de proyectos.
2. En el campo **Dirección IP del control:**, introducir la dirección IP de la unidad de control del robot, para la que debe crearse el proyecto.



La unidad de control del robot utiliza por defecto los siguientes rangos de dirección para fines internos. Por esta razón, el usuario no puede asignar las direcciones IP de estos rangos.

- 192.168.0.0 ... 192.168.0.255
- 172.16.0.0 ... 172.16.255.255
- 172.17.0.0 ... 172.17.255.255

3. Mantener el ajuste **Crear nuevo proyecto (offline)**. Avanzar a la pantalla siguiente con **Continuar >**.
4. En el campo **Nombre del proyecto** introducir un nombre para el proyecto.
5. En el campo **Posición** está determinado por defecto el directorio de proyectos.

Se puede seleccionar otro directorio: Para ello, quitar el símbolo de confirmación en **Utilizar la posición estándar** y seleccionar **Examinar....** Avanzar a la pantalla siguiente con **Continuar >**.

6. En la lista **Plantilla de topología**, seleccionar el robot para la configuración de la estación, p. ej. LBR iiwa 7 R800. Avanzar a la pantalla siguiente con **Continuar >**.
7. Para completar la configuración de la estación, seleccionar la brida para medios con la que está equipada el robot seleccionado.



El peso y la altura de la brida para medios seleccionada se tienen en cuenta automáticamente por el System Software.

Para saber si un robot está equipado con una brida para medios, se puede consultar la placa de características que está instalada en el pie del robot. La entrada relativa a la brida para medios se encuentra en la placa de características en **Nombre del transformador**:

Entrada	Descripción
M0	Robot sin brida para medios
MF	Robot con brida para medios Nota: La denominación exacta de la brida para medios montada se debe consultar en el pedido o en el albarán.
MLA	MAM Light Alpha (MLA)
MLB	MAM Light Beta (MLB)

8. Por defecto se encuentra ajustada la dirección de montaje del robot montado en el suelo. ($A=0^\circ$, $B=0^\circ$, $C=0^\circ$)

Para el montaje del robot en el techo o en la pared, introducir la dirección de montaje referida al robot montado en el suelo:

- Giro alrededor del eje Z en ° (ángulo A):** Giro del ángulo A sobre el eje Z del sistema de coordenadas del pie del robot ($-180^\circ \leq A \leq 180^\circ$)
- Giro alrededor del eje Y en ° (ángulo B):** Giro del ángulo B sobre el eje Y ($-90^\circ \leq B \leq 90^\circ$). El giro alrededor del eje Y hace referencia al sistema de coordenadas girado del paso a.
- Giro alrededor del eje X en ° (ángulo C):** Giro del ángulo C sobre el eje X ($-180^\circ \leq C \leq 180^\circ$). El giro alrededor del eje X hace referencia al sistema de coordenadas girado del paso b.

(>>> 6.7 "Sistemas de coordenadas" Página 76)



La dirección de montaje del robot debe haberse introducido de forma correcta. Una dirección de montaje introducida de forma errónea puede tener, por ejemplo, el siguiente efecto:

- Comportamiento inesperado del robot en la regulación de impedancia.
- Posición alterada de Frames ya programados por aprendizaje
- Impedimento del permiso de movimiento debido a la detección de colisiones y el control de fuerza TCP
- Comportamiento inesperado durante el desplazamiento manual en el sistema de coordenadas mundial.

9. Avanzar a la pantalla siguiente con **Continuar >**. Se muestra un resumen con información sobre el proyecto.

Quitar el símbolo de confirmación en **Crear aplicación (inicia un nuevo asistente)** y hacer clic sobre **Finalizar**. El proyecto se crea y se inserta en el **Explorador de paquetes**.

Si el símbolo de confirmación está puesto en **Crear aplicación (inicia un nuevo asistente)**, se abre el asistente para la creación de aplicaciones. Se puede crear directamente la primera aplicación del robot para el proyecto recién creado.

(>>> 5.4.2 "Crear aplicación del robot con paquete" Página 54)

Descripción

La figura muestra la estructura de un nuevo proyecto Sunrise creado en el que aún no se ha creado ninguna aplicación del robot o se ha realizado otra modificación. El robot configurado para el proyecto Sunrise dispone de una brida para medios.

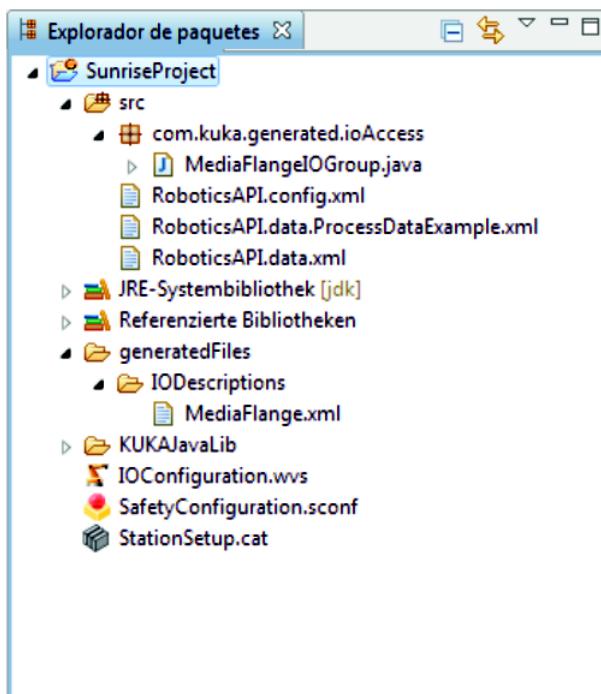


Fig. 5-2: Resumen de la estructura del proyecto

Elemento	Descripción
src	<p>Carpeta de origen del proyecto</p> <p>En la carpeta de origen se almacenan las aplicaciones del robot creadas y las clases Java.</p> <p>El paquete Java com.kuka.generated.ioAccess contiene la clase Java MediaFlangeIOGroup.java. La clase contiene los métodos necesarios para la programación, permitiendo el acceso a las entradas y salidas de la brida para medios.</p> <p>Aquí se puede encontrar más información sobre la utilización de entradas/salidas durante la programación: (>>> 15.12 "Entradas/Salidas" Página 300)</p> <p>Además, la carpeta de origen contiene diferentes archivos XML en los que se guardan datos del tiempo de ejecución junto a los datos de configuración, como por ejemplo los Frames y las herramientas creadas por el usuario.</p> <p>Los archivos XML puede visualizarse pero no editarse.</p>
Biblioteca de sistema JRE	<p>Biblioteca de sistema Java Runtime Environment</p> <p>La biblioteca de sistema contiene las bibliotecas de clases de Java que se pueden utilizar para la programación Java estándar.</p>
Bibliotecas referenciadas	<p>Bibliotecas referenciadas</p> <p>Las bibliotecas referenciadas se pueden utilizar en proyectos. Por defecto se añaden automáticamente las bibliotecas de clases de Java al crear un proyecto Sunrise. El usuario tiene la posibilidad de añadir otras bibliotecas.</p>
generatedFiles	<p>Carpeta con subcarpeta IODescriptions</p> <p>Los datos de las entradas/salidas configuradas para la brida para medios están guardados en un archivo XML.</p> <p>El archivo XML puede visualizarse pero no editarse.</p>

Elemento	Descripción
KUKAJavaLib	Carpeta con bibliotecas especiales que son necesarias para la programación del robot.
IOConfiguration.wvs	Configuración de E/S para la brida para medios La configuración de E/S contiene la estructura de bus completa de la brida para medios, incluida la conexión de E/S. La configuración de E/S puede abrirse, editarse y exportarse de nuevo al proyecto Sunrise en WorkVisual. Nota: La configuración E/S solo se realiza automáticamente para las entradas y salidas existentes en la brida de medios seleccionada. Los restantes dispositivos EtherCat conectados a la brida de medios deben configurarse con WorkVisual. (>>> 11 "Configuración del bus" Página 163)
SafetyConfiguration.sconf	El archivo contiene las funciones de seguridad preconfiguradas por KUKA. La configuración puede mostrarse y editarse. (>>> 13 "Configuración de seguridad" Página 179)
StationSetup.cat	El archivo contiene la configuración de la estación para la estación (unidad de control) seleccionada en la creación del proyecto. La configuración puede mostrarse y editarse. Mediante la configuración de la estación se puede instalar el System Software en la unidad de control del robot. (>>> 10 "Configuración de la estación e instalación" Página 159)



La carpeta **generatedFiles** se utiliza en el sistema y no se podrá utilizar para guardar archivos propios.

5.4 Crear una nueva aplicación del robot

Las aplicaciones del robot son programas Java. Describen tareas que deben ejecutarse en una estación. Se transmiten con el proyecto a la unidad de control del robot y se pueden seleccionar y ejecutar mediante el smartPAD.

Las aplicaciones del robot se agrupan para formar paquetes. Esto hace que la programación sea más sencilla y facilita el uso posterior de un paquete en otros proyectos.

5.4.1 Crear nuevo paquete de Java

Procedimiento

1. Marcar el proyecto en el **Explorador de paquetes**.
2. Seleccionar la secuencia del menú **Archivo > Nuevo > Paquete**. Se abre la ventana **Nuevo paquete de Java**.
3. En el campo **Nombre** introducir un nombre para el paquete.
4. Hacer clic en **Finalizar**. El paquete se crea y se inserta en la carpeta "src" del proyecto.

El paquete aún no contiene archivos. Un paquete vacío se señala mediante un símbolo de paquete blanco. Cuando un paquete contiene archivos, el color del símbolo será marrón.

5.4.2 Crear aplicación del robot con paquete

Procedimiento

1. Marcar el proyecto en el **Explorador de paquetes**.

2. Seleccionar la secuencia de menú **Archivo > Nuevo > Aplicación del robot**. Se abre la ventana **Nueva aplicación del robot**.
 3. En el campo **Paquete** introducir el nombre del paquete en el que debe crearse la aplicación.
 4. En el campo **Nombre** introducir un nombre para la aplicación.
 5. Hacer clic en **Finalizar**. Se crean la aplicación y el paquete y se insertan en el proyecto.
- La aplicación *Nombre.java* se abre en la zona de los editores.

5.4.3 Crear una aplicación del robot para el paquete existente

- | | |
|----------------------|--|
| Procedimiento | <ol style="list-style-type: none"> 1. Marcar el paquete en el Explorador de paquetes. 2. Seleccionar la secuencia de menú Archivo > Nuevo > Aplicación del robot. Se abre la ventana Nueva aplicación del robot. 3. En el campo Nombre introducir un nombre para la aplicación. 4. Hacer clic en Finalizar. La aplicación se crea y se inserta en el paquete. <p>La aplicación <i>Nombre.java</i> se abre en la zona de los editores.</p> |
|----------------------|--|

5.5 Crear una nueva tarea en segundo plano

Las tareas en segundo plano son programas Java que se ejecutan de forma paralela a la aplicación del robot en la unidad de control del robot. Por ejemplo, pueden ejecutar tareas de control para dispositivos periféricos.

La utilización y la programación de las tareas en segundo plano se describen aquí: ([>>> 16 "Tareas en segundo plano"](#) Página 385)

Las siguientes propiedades se determinan al crear la tarea:

- Tipo de inicio de la tarea
 - **Automático**
La tarea se inicia automáticamente después del arranque de la unidad de control del robot. (por defecto)
 - **Manual**
La tarea se debe iniciar manualmente mediante el smartPAD. (La función aún no es soportada por el momento.)
- Plantilla de la tarea
 - **Tarea en segundo plano cíclica**
Plantilla para tareas que deben ejecutarse cíclicamente (por defecto)
 - **Tarea en segundo plano no cíclica**
Plantilla para tareas que deben ejecutarse una vez

5.5.1 Crear una tarea en segundo plano con paquete

- | | |
|----------------------|---|
| Procedimiento | <ol style="list-style-type: none"> 1. Marcar el proyecto en el Explorador de paquetes. 2. Seleccionar la secuencia de menú Archivo > Nuevo > Tarea en segundo plano. Se abre la ventana Nueva tarea en segundo plano. 3. En el campo Paquete introducir el nombre del paquete en el que debe crearse la tarea. 4. En el campo Nombre introducir un nombre para la tarea. 5. Hacer clic en Continuar > y seleccionar el tipo de inicio de la tarea. 6. Hacer clic en Continuar > y seleccionar la plantilla de la tarea. 7. Hacer clic en Finalizar. Se crean la tarea y el paquete y se insertan en el proyecto. |
|----------------------|---|

La tarea *Nombre.java* se abre en la zona de los editores.

5.5.2 Crear una tarea en segundo plano para el paquete existente

Procedimiento

1. Marcar el paquete en el **Explorador de paquetes**.
2. Seleccionar la secuencia de menú **Archivo > Nuevo > Tarea en segundo plano**. Se abre la ventana **Nueva tarea en segundo plano**.
3. En el campo **Nombre** introducir un nombre para la tarea.
4. Hacer clic en **Continuar >** y seleccionar el tipo de inicio de la tarea.
5. Hacer clic en **Continuar >** y seleccionar la plantilla de la tarea.
6. Hacer clic en **Finalizar**. La tarea se crea y se inserta en el paquete.

La tarea *Nombre.java* se abre en la zona de los editores.

5.6 Zona de trabajo

El directorio en el que se guardan los proyectos creados y los ajustes definidos por el usuario de Sunrise.Workbench, se denomina zona de trabajo. El directorio para la zona de trabajo se debe establecer por el usuario en el primer inicio de Sunrise.Workbench. Se pueden crear otras zonas de trabajo en Sunrise.Workbench y se puede conmutar entre ellas.

5.6.1 Crear una zona de trabajo nueva

Procedimiento

1. Seleccionar la secuencia de menú **Archivo > Cambiar zona de trabajo > Otro....** Se abre la ventana **Programa de inicio para la zona de trabajo**.
2. En el campo **Zona de trabajo** introducir la ruta hasta el nuevo directorio del proyecto.
Alternativa:
 - Navegar con **Examinar...** hasta el directorio en el que debe crearse la nueva zona de trabajo.
 - Mediante **Crear nueva carpeta** crear el nuevo directorio del proyecto. Confirmar con **OK**.
La ruta hasta el nuevo directorio del proyecto se acepta en el campo **Zona de trabajo**.
3. Confirmar la nueva zona de trabajo con **OK**. Sunrise.Workbench se reinicia y se abre una pantalla de bienvenida.

5.6.2 Cambiar a una zona de trabajo existente

Condición previa

- Hay más campos de trabajo disponibles.

Procedimiento

1. Seleccionar la secuencia de menú **Archivo > Cambiar zona de trabajo > Otro....** Se abre la ventana **Programa de inicio para la zona de trabajo**.
2. Navegar a través de **Examinar...** hasta el directorio de la zona de trabajo deseada y seleccionarlo.
3. Confirmar con **OK**. La ruta hasta el nuevo directorio del proyecto se acepta en la ventana **Programa de inicio para la zona de trabajo**.
4. Confirmar la zona de trabajo seleccionada con **OK**. Sunrise.Workbench se reinicia y abre la zona de trabajo seleccionada.

5.6.3 Cambiar entre los campos de trabajo abiertos por última vez

- Condición previa** ■ Hay más campos de trabajo disponibles.
- Procedimiento**
1. Seleccionar la secuencia de menú **Archivo > Cambiar zona de trabajo**. Las últimas zonas de trabajo utilizadas se muestran en una lista (como máximo 4).
 2. Seleccionar la zona de trabajo deseada en la lista. Sunrise.Workbench se reinicia y abre la zona de trabajo seleccionada.

5.6.4 Archivar proyectos

- Procedimiento**
1. Seleccionar la secuencia de menú **Archivo > Exportar**. Se abre el asistente para la exportación de archivos.
 2. En la carpeta **General** marcar la opción **Archivo de archivado** y hacer clic en **Continuar >**.
 3. En la zona superior izquierda se muestran en una lista todos los proyectos que se encuentran en la zona de trabajo. Seleccionar los proyectos que deben ser archivados (marcar el símbolo de confirmación en la casilla).
 4. A través de **Examinar...** navegar hasta el lugar de almacenamiento deseado, introducir el nombre para el archivo y hacer clic en **Guardar**.
 5. Hacer clic en **Finalizar**. Se crea el archivo de archivado.

5.6.5 Cargar proyectos desde el archivo en la zona de trabajo

- Condición previa** ■ El archivo de archivado (p. ej. un archivo ZIP) con los proyectos a cargar está disponible.
■ La zona de trabajo no contiene ningún proyecto con el nombre de un proyecto a cargar.
- Procedimiento**
1. Seleccionar la secuencia de menú **Archivo > Importar**. Se abre el asistente para la importación de archivos.
 2. En la carpeta **General** marcar la opción **Objetos existentes en la zona de trabajo** y hacer clic en **Continuar >**.
 3. Activar el botón de radio **Seleccionar archivo de archivado** y navegar a través de **Examinar...** hasta el archivo de archivado deseado y marcarlo.
 4. Hacer clic en **Abrir**. En **Proyectos** se muestran en una lista todos los proyectos que se encuentran en el archivo.
 5. Seleccionar los proyectos que deben cargarse en la zona de trabajo (el símbolo de confirmación en la casilla debe estar marcado).
 6. Hacer clic en **Finalizar**. Los proyectos seleccionados se cargan.

5.6.6 Cargar proyectos del directorio en la zona de trabajo

- Condición previa** ■ Uno o varios proyectos están disponibles en un directorio cualquiera.
■ La zona de trabajo no contiene ningún proyecto con el nombre de un proyecto a cargar.
- Procedimiento**
1. Seleccionar la secuencia de menú **Archivo > Importar**. Se abre el asistente para la importación de archivos.
 2. En la carpeta **General** marcar la opción **Objetos existentes en la zona de trabajo** y hacer clic en **Continuar >**.
 3. Activar el botón de radio **Seleccionar directorio principal** y navegar a través de **Examinar...** hasta el directorio deseado y marcarlo.

4. hacer clic en **OK**. En **Proyectos** se muestran en una lista todos los proyectos que se encuentran en el directorio seleccionado.
5. Seleccionar los proyectos que deben cargarse en la zona de trabajo (el símbolo de confirmación en la casilla debe estar marcado).
6. Hacer clic en **Finalizar**. Los proyectos seleccionados se cargan.

5.7 Proyectos Sunrise con proyectos Java referenciados

En un proyecto Sunrise se pueden referenciar uno o varios proyectos Java. La opción de referenciación de proyectos Java permite utilizarlos en el número deseado de proyectos Sunrise y, de este modo, en diferentes unidades de control del robot.

Los proyectos Java referenciados pueden referenciar a su vez otros proyectos Java. Entre todos los proyectos referenciados entre sí, solo puede haber un solo proyecto Sunrise.



Con la sincronización de proyectos Sunrise también se transmiten proyectos Java referenciados a la unidad de control del robot. Si dentro de un proyecto Sunrise se ha referenciado otro proyecto Sunrise, se cancelará la sincronización con un mensaje de error.

5.7.1 Crear proyecto nuevo de Java

Procedimiento

1. Seleccionar la secuencia del menú **Archivo > Nuevo > Proyecto....** Se abre el asistente para la creación de proyectos.
2. En la carpeta **Java**, marcar la opción **Proyecto de Java** y hacer clic en **Continuar >**.
3. En el campo **Nombre del proyecto**: introducir el nombre del proyecto de Java.
4. En la zona **JRE**, seleccionar la versión de JRE que corresponda a la versión JRE del proyecto de Sunrise. Generalmente es la JavaSE-1.6.
5. Hacer clic en **Continuar >** y seguidamente en **Finalizar**.
6. Cuando se crea por primera vez un proyecto de Java en la zona de trabajo o los proyectos anteriores de Java aún no han memorizado la decisión, aparecerá una consulta sobre si se debe abrir la perspectiva **Java**.
 - Según necesidad, seleccionar **Sí** o **No**.
 - Si la consulta no ha de aparecer en el siguiente proyecto de Java que se cree en la zona de trabajo, activar la opción **Memorizar decisión** (activar el símbolo de confirmación).



En los proyectos de Java, todas las clases que son referenciadas desde fuera deben depositarse en un paquete de Java definido. Si las clases referenciadas se generan en el paquete estándar, pueden no ser encontradas en el proyecto Sunrise.

5.7.1.1 Añadir bibliotecas de clases específicas del robot en el proyecto Java

Descripción

Si se utiliza un proyecto Java para la programación del robot, se deberán añadir en el proyecto las bibliotecas KUKA especiales que son necesarias para ello. Por defecto, estas bibliotecas no están incluidas en un proyecto Java.

Las bibliotecas KUKA se deben copiar de un proyecto Sunrise compatible. Lo ideal es un proyecto Sunrise en el que está referenciado o se va a referenciar el proyecto Java. La condición previa para la compatibilidad de los proyectos referenciados es que coincidan las versiones de RoboticsAPI.

- Condición previa**
- En la zona de trabajo está disponible al menos un proyecto Sunrise compatible.
- Procedimiento**
1. Copiar la carpeta **KUKAJavaLib** de un proyecto Sunrise compatible: En el **Explorador de paquetes** hacer clic con el botón derecho del ratón sobre la carpeta y seleccionar en el menú contextual **Copiar**.
 2. Añadir la carpeta **KUKAJavaLib** en el proyecto Java: En el **Explorador de paquetes** hacer clic con el botón derecho del ratón sobre el proyecto Java deseado y seleccionar en el menú contextual **Añadir**.
 3. Hacer de nuevo clic con el botón derecho del ratón sobre el proyecto Java y seleccionar en el menú contextual **Ruta de creación > Configurar ruta de creación....** Se abre la ventana **Propiedades del proyecto**.
 4. En la **Ruta de creación Java** seleccionar la pestaña **Bibliotecas** y hacer clic sobre el botón **Añadir JAR....** Se abre la ventana **Selección de JAR**.
 5. Se muestran en una lista todos los proyectos que se encuentran en la zona de trabajo. Desplegar el proyecto Java en el que deben añadirse las bibliotecas referenciadas.
 6. Desplegar la carpeta **KUKAJavaLib** y marcar los archivos JAR disponibles.
 7. Confirmar selección con **OK**. Los archivos JAR se aceptan en la pestaña **Bibliotecas** de la ruta de creación.
 8. Cerrar la ventana con **OK**. Las bibliotecas referenciadas se añaden al proyecto Java.

5.7.2 Referenciar proyectos Java

- Condición previa**
- Las clases referenciadas están almacenadas en un paquete Java definido (no en el paquete estándar).
 - Para los proyectos Java que utilizan bibliotecas KUKA referenciadas: En los proyectos referenciados deben coincidir las versiones de RoboticsAPI.
- Procedimiento**
1. En el **Explorador de paquetes** hacer clic con el botón derecho del ratón sobre el proyecto para el que se deben referenciar proyectos Java.
 2. Seleccionar en el menú contextual **Ruta de creación Java > Configurar ruta de creación....** Se abre la ventana **Propiedades del proyecto**.
 3. En la **Ruta de creación Java** seleccionar la pestaña **Proyectos** y hacer clic sobre el botón **Añadir** Se abre la ventana **Selección de proyecto necesario**.
 4. Se muestran en una lista todos los proyectos que se encuentran en la zona de trabajo. Seleccionar los proyectos Java que deben ser referenciados (marcar el símbolo de confirmación en la casilla).
 5. Confirmar selección con **OK**. Los proyectos seleccionados se aceptan en la pestaña **Proyectos** de la ruta de creación.
 6. Cerrar la ventana con **OK**.

5.7.3 Eliminar la referenciación para los proyectos Java

- Descripción**
- Las referenciaciones para los proyectos añadidos de forma incorrecta o que (ya) no son necesarios, se pueden eliminar.
- Procedimiento**
1. En el **Explorador de paquetes** hacer clic con el botón derecho del ratón sobre el proyecto del que se deben eliminar los proyectos referenciados.
 2. Seleccionar **Propiedades** en el menú contextual. Se abre la ventana **Propiedades del proyecto**.
 3. En la **Ruta de creación Java** seleccionar la pestaña **Proyectos**.
 4. Marcar los proyectos que no sean necesarios y hacer clic en **Eliminar**.

5. Cerrar la ventana con **OK**.

5.8 Renombrar elementos en el "Explorador de paquetes"

Descripción	En la vista Explorador de paquetes se pueden modificar los nombres de los elementos insertados, p. ej. los nombres de proyectos Sunrise, paquetes Java o aplicaciones del robot.
Procedimiento	<ol style="list-style-type: none">1. Hacer clic con el botón derecho del ratón sobre el elemento. Seleccionar en el menú contextual Refactoring > Renombrar. Se abre la ventana ...renombrar. (El nombre exacto de la ventana depende del tipo de elemento seleccionado.)2. En el campo Nuevo nombre: introducir el nombre deseado. Hacer clic en Finalizar.3. Antes de la finalización, se hace referencia a los posibles conflictos. Si los conflictos se han tenido en cuenta y se han comprobado, hacer clic de nuevo en Finalizar.

5.9 Eliminar elementos en el "Explorador de paquetes"

En la vista **Explorador de paquetes** se pueden volver a eliminar los elementos añadidos, p. ej. proyectos completos o paquetes Java individuales y aplicaciones de un proyecto.

5.9.1 Eliminar elementos de un proyecto

Descripción	Si se eliminan elementos de un proyecto en el Explorador de paquetes , se eliminarán de forma permanente. Es decir, se borran también de la carpeta del proyecto en el soporte de datos y no se pueden restaurar.
Procedimiento	<ol style="list-style-type: none">1. Hacer clic con el botón derecho del ratón sobre el elemento y seleccionar Borrar en el menú contextual.2. Confirmar la pregunta de seguridad con OK. Se borra el elemento.

5.9.2 Eliminar proyectos

Descripción	Para eliminar un proyecto en el Explorador de paquetes se encuentran disponibles las siguientes opciones: <ul style="list-style-type: none">■ El proyecto solo se elimina en el Explorador de paquetes y se conserva en la carpeta del proyecto en el soporte de datos. (Ajuste por defecto)■ El proyecto también se borra de la carpeta del proyecto en el soporte de datos y no se puede restablecer de nuevo.
Procedimiento	<ol style="list-style-type: none">1. Hacer clic con el botón derecho del ratón sobre el proyecto y seleccionar Borrar en el menú contextual.2. Se muestra una pregunta de seguridad sobre si realmente se debe borrar el proyecto de la zona de trabajo.<ul style="list-style-type: none">■ Si el proyecto se debe borrar del soporte de datos, activar la opción Borrar el contenido del proyecto en el soporte de datos ... (marcar el símbolo de confirmación).■ Si el proyecto se debe conservar en el soporte de datos, no activar la opción. (Ajuste por defecto)3. Confirmar la pregunta de seguridad con OK. Se borra el elemento.

5.10 Activar la detección automática de modificaciones

Descripción	La detección automática de modificaciones de Sunrise.Workbench está activada por defecto. Si se ha desactivado, esto provoca por ejemplo, que al exportar una configuración E/S desde WorkVisual no se creen en el proyecto Sunrise las clases Java y los archivos necesarios para utilizar las señales.
Procedimiento	<ol style="list-style-type: none">1. Seleccionar la secuencia de menú Ventana > Especificaciones de usuario. Se abre la ventana Especificaciones de usuario.2. En la estructura del directorio, seleccionar en la zona izquierda de la ventana General > Zona de trabajo.3. Poner el símbolo de confirmación en Actualizar a través de native Hooks o Polling para activar la detección automática de modificaciones.

6 Manejo de KUKA smartPAD

6.1 Dispositivo de mando manual KUKA smartPAD

6.1.1 Parte delantera

Función El smartPAD es el dispositivo de mando manual del robot industrial. El smartPAD contiene todas las funciones de control e indicación necesarias para el manejo.

El smartPAD dispone de una pantalla táctil: el smartHMI se puede manejar con el dedo o un lápiz óptico. No es necesario utilizar un ratón o un teclado externo.

Resumen

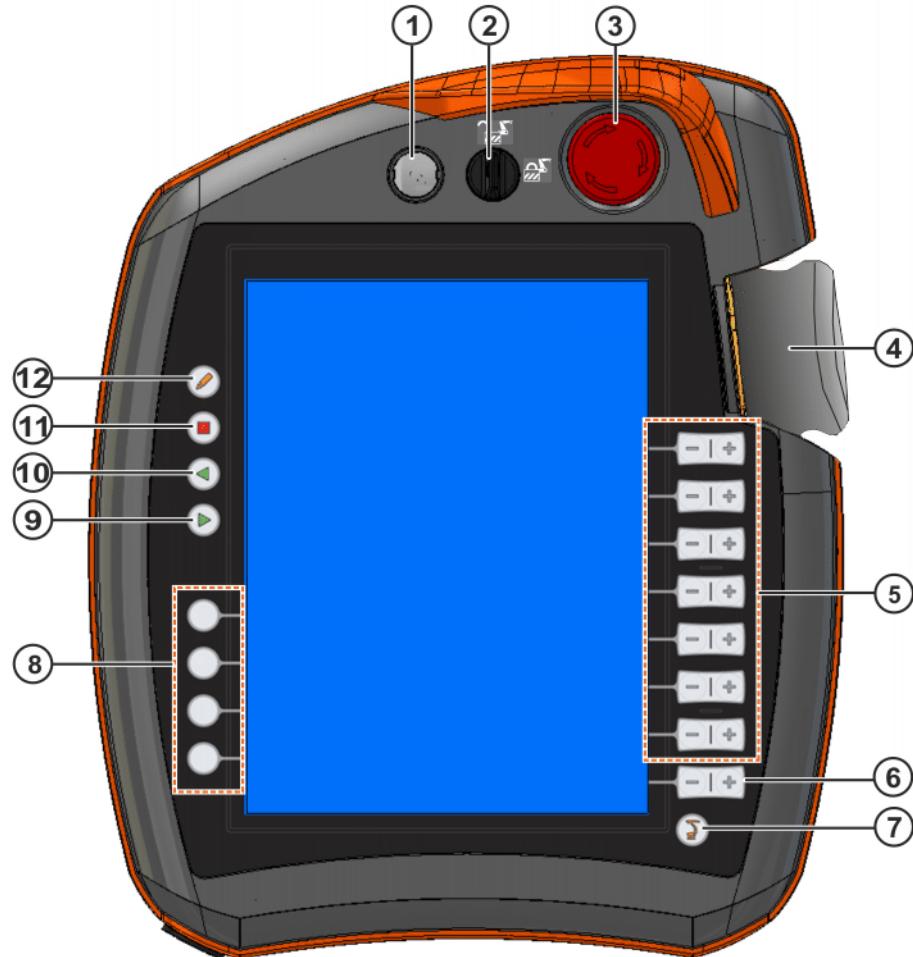


Fig. 6-1: KUKA smartPAD, lado delantero

Pos.	Descripción
1	Botón para desenchufar el smartPAD actualmente inhabilitada.
2	Interruptor de llave Con el interruptor de llave se activa el gestor de conexiones. El gestor de conexiones permite cambiar el modo de servicio. (>>> 6.6 "Cambiar modo de servicio" Página 75)

Pos.	Descripción
3	Dispositivo de PARADA DE EMERGENCIA Con el dispositivo de PARADA DE EMERGENCIA se detiene el robot en situaciones de peligro. El dispositivo de PARADA DE EMERGENCIA queda bloqueado cuando se pulsa.
4	Space Mouse actualmente inhabilitada.
5	Teclas de desplazamiento Con las teclas de desplazamiento se desplaza el robot manualmente. (>>> 6.8 "Desplazar el robot de forma manual" Página 78)
6	Tecla para ajustar el override manual.
7	Tecla de menú principal La tecla de menú principal muestra y oculta el menú principal en la smartHMI.
8	Teclas de usuario La función de las teclas de usuario es libremente programable. Las teclas de usuario se pueden utilizar, por ejemplo, para controlar dispositivos periféricos o para activar acciones específicas para aplicaciones. (>>> 15.26 "Definir teclas de usuario" Página 356)
9	Tecla de arranque Con la tecla de Inicio se inicia un programa. La tecla de arranque también se utiliza para avanzar manualmente a los frames y desplazar el robot de vuelta a la trayectoria. (>>> 6.14 "Ejecución del programa" Página 90) (>>> 6.13.4 "Desplazamiento manual hasta Frames" Página 89) (>>> 6.14.6 "Retroceder el robot después de abandonar la trayectoria" Página 95)
10	Tecla de arranque-hacia atrás actualmente inhabilitada.
11	Tecla de PARADA Con la tecla de PARADA se detiene una aplicación que esté en ejecución.
12	Tecla del teclado actualmente inhabilitada.



Para las teclas de desplazamiento, las teclas de usuario y la tecla de arranque, la tecla de arranque-hacia atrás y tecla de PARADA se aplica lo siguiente:

- La función actual se muestra junto a la tecla en la smartHMI.
- Si no hay ninguna indicación disponible, la tecla actual carece de función.

6.1.2 Parte trasera

Resumen



Fig. 6-2: KUKA smartPAD, parte posterior

- | | | | |
|---|-------------------------|---|--------------------------|
| 1 | Pulsador de validación | 4 | Conexión USB |
| 2 | Tecla de inicio (verde) | 5 | Pulsador de validación |
| 3 | Pulsador de validación | 6 | Placa de características |

Descripción

Elemento	Descripción
Placa de carac- terísticas	Placa de características
Tecla de arran- que	Con la tecla de Inicio se inicia un programa. La tecla de arranque también se utiliza para avanzar manualmente a los frames y desplazar el robot de vuelta a la trayectoria.

Elemento	Descripción
Pulsador de validación	<p>El pulsador de validación tiene 3 posiciones:</p> <ul style="list-style-type: none"> ■ No pulsado ■ Posición intermedia ■ Pulsado a fondo <p>En los modos de servicio T1, T2 y KRF, el pulsador de validación debe mantenerse en la posición intermedia para poder efectuar movimientos con el manipulador.</p> <p>En el modo de servicio Automático, el pulsador de validación carece de función por defecto.</p>
Conexión USB	<p>La conexión USB se utiliza, p. ej., para el archivado.</p> <p>Únicamente para memorias USB con formato FAT32.</p>

6.2 Conectar/desconectar la unidad de control del robot

6.2.1 Conectar la unidad de control del robot e iniciar el System Software

Procedimiento

- Colocar el interruptor principal de la unidad de control del robot en posición "I".
- El software de sistema se inicia automáticamente.

La unidad de control del robot está lista para el servicio cuando la indicación de estado que indica el estado de arranque de la unidad de control del robot se ilumina en verde (vista de la estación / cuadro **KUKA_Sunrise_Cabinet**).

6.2.2 Desconectar la unidad de control del robot

Procedimiento

- Colocar el interruptor principal de la unidad de control del robot en posición "0".

AVISO

Si al desconectar la unidad de control del robot aún se está ejecutando una aplicación, se detendrán los movimientos que se estén ejecutando. Esto puede provocar daños en el robot. Por ello, la unidad de control del robot solo se puede desconectar cuando no se esté ejecutando ninguna aplicación y el robot esté parado.

6.3 Actualización automática del software del smartPAD

Con el reinicio de la unidad de control del robot y al conectar el smartPAD a una unidad de control del robot en ejecución, se ejecuta automáticamente una comprobación de la versión del software del smartPAD. En caso de que se produzcan conflictos entre el software del smartPAD y el System Software en la unidad de control del robot, será necesaria la actualización del software del smartPAD.

Propiedades de la actualización del software del smartPAD:

- En los modos de servicio T1, T2 y KRF, la actualización se ejecuta automáticamente.
- En el modo de servicio automático no se puede realizar ninguna actualización automática del software del smartPAD.

Si al conectar el smartPAD se detecta un conflicto de versión en el modo de servicio automático, se bloqueará el smartPAD para la introducción de datos de usuario. En este caso, cambiar al modo de servicio T1 o T2 para permitir la actualización automática.

- Durante la actualización, el smartPAD está bloqueado para la introducción de datos de usuario.

AVISO La actualización no se debe cancelar, ya que de lo contrario puede resultar dañado el smartPAD. Se debe prestar atención para que el smartPAD no sea desconectado de la unidad de control del robot durante la actualización o que la unidad de control del robot no se desconecte de la alimentación de corriente.

- Despues de la actualización, el smartPAD se inicia automáticamente. La zona superior de la smartHMI está tapado por una barra.

i Despues de la actualización automática, se deberá reiniciar la unidad de control del robot para poder visualizar la smartHMI completamente y para poder utilizar el sistema.

6.4 Interfaz de usuario KUKA smartHMI



Fig. 6-3: Interfaz de usuario KUKA smartHMI

Pos.	Descripción
1	Barra de navegación: Menú principal e indicaciones de estado (>>> 6.4.1 "Barra de navegación" Página 68)
2	Zona de indicación Indicación del nivel seleccionado en la barra de navegación, aquí la vista de la estación

Pos.	Descripción
3	<p>Botón Opciones de desplazamiento manual</p> <p>Muestra el sistema de coordenadas actual para el desplazamiento manual con las teclas de desplazamiento. Al tocar este botón se abre la ventana Opciones de desplazamiento manual en la que se pueden ajustar el sistema de coordenadas de referencia y otros parámetros para el desplazamiento manual.</p> <p>(>>> 6.8.1 "Ventana Opciones de desplazamiento manual" Página 78)</p>
4	<p>Indicador de las teclas de desplazamiento</p> <p>Si se ha seleccionado el desplazamiento específico del eje, se mostrarán aquí los números de los ejes (A1, A2, etc.). Si se ha seleccionado el desplazamiento cartesiano, se mostrarán aquí las direcciones del sistema de coordenadas (X, Y, Z, A, B, C), así como el ángulo del codo (R) para la ejecución de un movimiento de espacio cero.</p> <p>(>>> 6.8 "Desplazar el robot de forma manual" Página 78)</p>
5	<p>Botón HOV</p> <p>Muestra el override manual actual. Al tocar este botón se abre la ventana Velocidad de desplazamiento manual, en la que se puede ajustar el Override manual.</p> <p>(>>> 6.8.2 "Ajustar el override manual (HOV)" Página 80)</p>
6	<p>Indicador de señal de funcionamiento</p> <p>Una señal de funcionamiento intermitente de manera uniforme indica que la smartHMI está activa.</p>
7	<p>Botón Selección del idioma</p> <p>Al tocar este botón se abre el menú Selección del idioma, en el que se puede modificar el idioma de indicación de la smartHMI.</p>
8	<p>Botón Selección de teclas de usuario</p> <p>Al tocar este botón se abre la ventana Selección de teclas de usuario, en la que se pueden seleccionar las barras de teclas de usuario actualmente disponibles.</p> <p>(>>> 6.15 "Activar las teclas de usuario" Página 96)</p>
9	<p>Botón Reloj</p> <p>El reloj muestra la hora del sistema. Al tocar este botón se visualiza la hora del sistema en formato digital y la fecha actual.</p>
10	<p>Botón Tipo de desplazamiento</p> <p>Muestra el modo ajustado actualmente de la tecla de arranque. Al tocar este botón se abre la ventana Tipo de desplazamiento, en la que se puede modificar el modo.</p> <p>(>>> 6.13.3 "Ventana Tipo de desplazamiento" Página 88)</p>
11	<p>Botón Atrás</p> <p>Con este botón se puede retroceder a la vista anterior.</p>

6.4.1 Barra de navegación

La barra de navegación es el menú principal de la interfaz de usuario y se divide en 4 niveles. Sirve para la navegación entre los distintos niveles.

Algunos niveles se dividen en dos:

- Lista de selección abajo: Abre una lista en la que se puede seleccionar una aplicación, un robot o un grupo E/S, en función del nivel.
- Botón arriba: Si se ha realizado una selección a través de la lista, mediante este botón se podrá visualizar la correspondiente vista de la aplicación, la vista del robot o el grupo E/S.

Alternativamente se puede activar el menú principal mediante la tecla de menú principal en la smartPAD. Éste contiene otros menús que no se pueden seleccionar a través de la guía de menú de la lista de navegación.

(>>> 6.5 "Activar el menú principal" Página 73)

Resumen

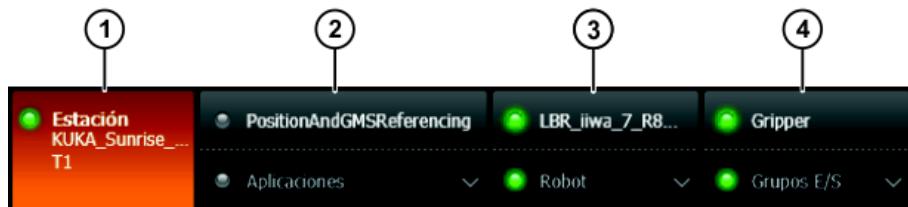


Fig. 6-4: Barra de navegación de KUKA smartHMI

Pos.	Descripción
1	Plano Estación Visualización del nombre del control y del modo de servicio seleccionado (>>> 6.4.4 "Vista de la estación" Página 70)
2	Plano Aplicaciones Visualización de la aplicación seleccionada (>>> 6.14.1 "Seleccionar la aplicación de robot" Página 90)
3	Plano Robot Indicación del robot seleccionado (>>> 6.4.5 "Vista del robot" Página 72)
4	Plano Grupos E/S Visualización del grupo E/S seleccionado (>>> 6.16.5 "Visualizar el grupo E/S y modificar el valor de una salida" Página 100)

6.4.2 Indicación de estados

El estado de los componentes del sistema está señalizado en la smartHMI mediante círculos de colores.

En la barra de navegación (>>> Fig. 6-4) se muestra el "Estado colectivo" en la parte inferior de barra. En la parte superior se muestra respectivamente el estado de los componentes seleccionados. P. ej., se puede ejecutar una aplicación y al mismo tiempo otra aplicación puede estar en estado de error.

Estado	Descripción
	Error grave El componente del sistema no se puede utilizar. El motivo puede ser un error de manejo o un error del componente del sistema.
	Advertencia Para el componente del sistema existe una indicación de advertencia. La disposición para el servicio del componente puede estar limitada. Por ello se recomienda subsanar la causa. En las aplicaciones, la indicación de estado amarilla significa que la aplicación está pausada.
	Estado OK Para el componente del sistema no existen advertencias ni averías.
	Estado desconocido El estado del componente del sistema no ofrece ninguna información.

6.4.3 Teclado

Para la introducción de letras y cifras se encuentra disponible un teclado en la smartHMI. La smartHMI detecta cuándo es necesaria la introducción de letras o cifras y visualiza el teclado adecuado.

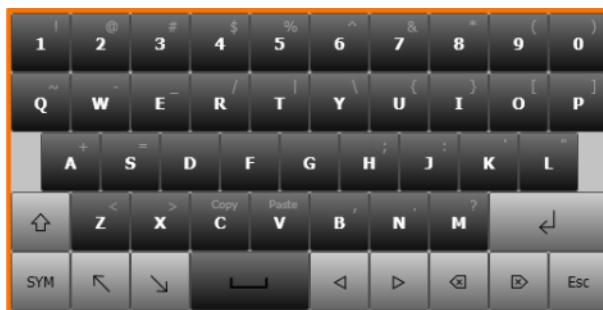


Fig. 6-5: Ejemplo de teclado



SYM debe pulsarse para poder utilizar la segunda asignación de las teclas, p. ej., el carácter "=" sobre la tecla "S". La tecla tiene función de autorretención pulsándola una vez. No es necesario, por tanto, mantenerla pulsada.

6.4.4 Vista de la estación

La vista de la estación permite el acceso a la información y a las funcionalidades que afectan a la estación completa.



Fig. 6-6: Vista de la estación

Pos.	Descripción
1	Cuadro Datos de proceso Aquí se muestra información sobre los datos de proceso. La configuración de los datos de proceso actualmente no es posible.
2	Cuadro Seguridad Abre la vista Seguridad y muestra el estado de seguridad de la estación. (>>> 13.7 "Activar la configuración de seguridad en la unidad de control del robot" Página 200)
3	Cuadro Frames Abre la vista de Frames. La vista contiene los Frames del proyecto Sunrise creados para la estación. (>>> 6.13 "Programar por aprendizaje y desplazar manualmente los Frames" Página 85)
4	Cuadro KUKA_Sunrise_Cabinet Muestra el estado de la unidad de control del robot y abre una vista que contiene los cuadros Estado de arranque y Buses de campo . El cuadro Estado de arranque muestra el estado de arranque de la unidad de control del robot. El cuadro Buses de campo muestra el estado de los buses de campo. Solo se muestra si están creados grupos E/S para el proyecto Sunrise y están conectadas las señales correspondientes con WorkVisual.
5	Cuadro Estado HMI Muestra el estado de conexión entre la smartHMI y el server de la aplicación.

Pos.	Descripción
6	Cuadro Información Muestra información del sistema, p. ej. la dirección IP de la unidad de control del robot.
7	Cuadro Protocolo Abre la vista Protocolo y muestra las incidencias protocolizadas y las modificaciones de estado del sistema. La indicación se puede filtrar mediante diferentes criterios. (>>> 18.2 "Visualizar el protocolo" Página 413)

6.4.5 Vista del robot

La vista del robot permite acceso a información y funcionalidades que afectan al robot seleccionado.

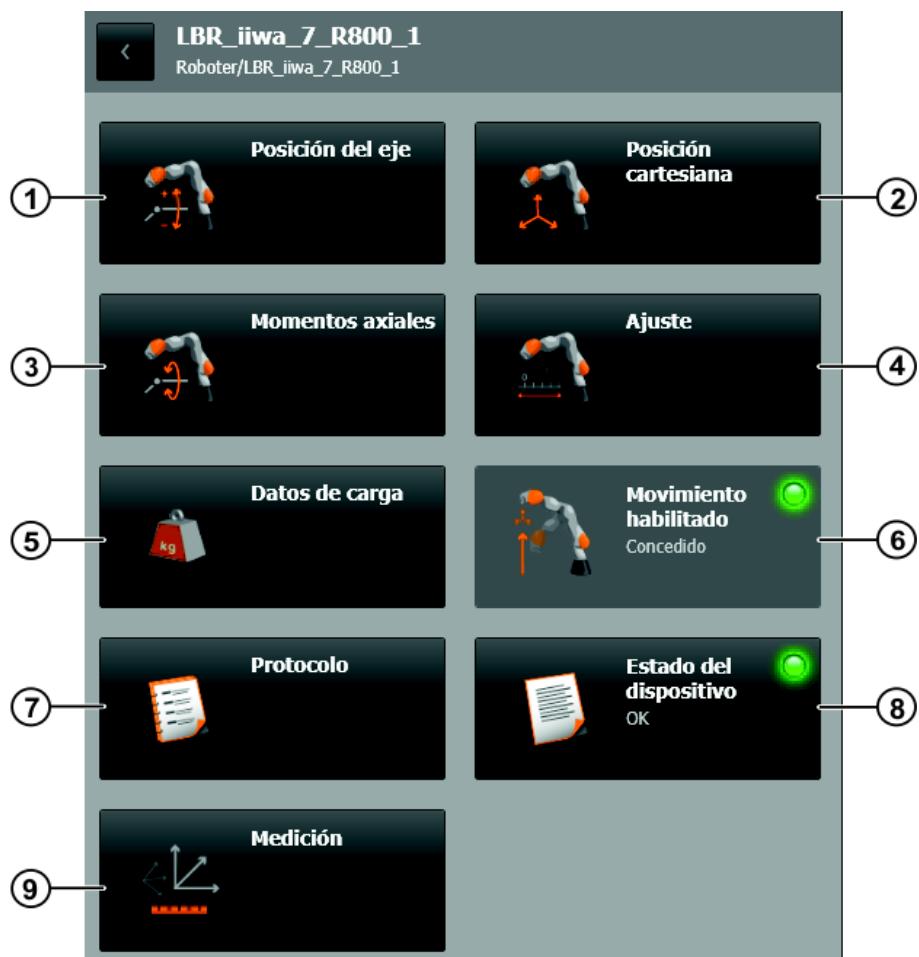


Fig. 6-7: Vista del robot

Pos.	Descripción
1	Cuadro Posición del eje Abre la vista Posición del eje . Se muestra la posición real específica del eje del robot. (>>> 6.16.2 "Visualizar la posición real específica del eje" Página 98)
2	Cuadro Posición cartesiana Abre la vista Posición cartesiana . Se muestra la posición real cartesiana del robot. (>>> 6.16.3 "Visualizar la posición real cartesiana" Página 99)
3	Cuadro Momentos axiales Abre la vista Momentos axiales . Se muestran los momentos del eje del robot. (>>> 6.16.4 "Visualizar momentos específicos del eje" Página 100)
4	Cuadro Ajuste Abre la vista Ajuste . Se muestra el estado de ajuste de los ejes del robot. Los ejes pueden ser ajustados o desajustados individualmente. (>>> 7.1 "Ajuste de posición" Página 105)
5	Cuadro Datos de carga Abre la vista Datos de carga para el cálculo automático de los datos de carga. (>>> 7.3 "Determinar datos de carga de la herramienta" Página 114)
6	Cuadro Movimiento habilitado Se muestra si el robot ha recibido el movimiento habilitado.
7	Cuadro Protocolo Abre la vista Protocolo y muestra los eventos y los cambios de estado del sistema protocolizados. La visualización puede filtrarse según diferentes criterios. Por defecto, el filtro Fuente(s) ya está activado en el robot correspondiente. (>>> 18.2 "Visualizar el protocolo" Página 413)
8	Cuadro Estado del dispositivo Se visualiza el estado del sistema de accionamiento del robot.
9	Cuadro Medición Abre la vista Medición que contiene los cuadros Medición de base y Medición de herramienta . (>>> 7.2 "Medición" Página 106)

6.5 Activar el menú principal

Procedimiento

- Pulsar la tecla de menú principal en el smartPAD. Se abre la vista **Menú principal**.

Descripción**Propiedades de la vista Menú principal:**

- En la columna izquierda se muestra el menú principal.
- Los primeros 4 botones vistos desde arriba son iguales que los niveles de la barra de navegación.
- Al tocar un botón con flecha se muestra las zonas correspondientes del nivel, p. ej. **Estación**.
- Las demás posibilidades de navegación se describen en la siguiente tabla.

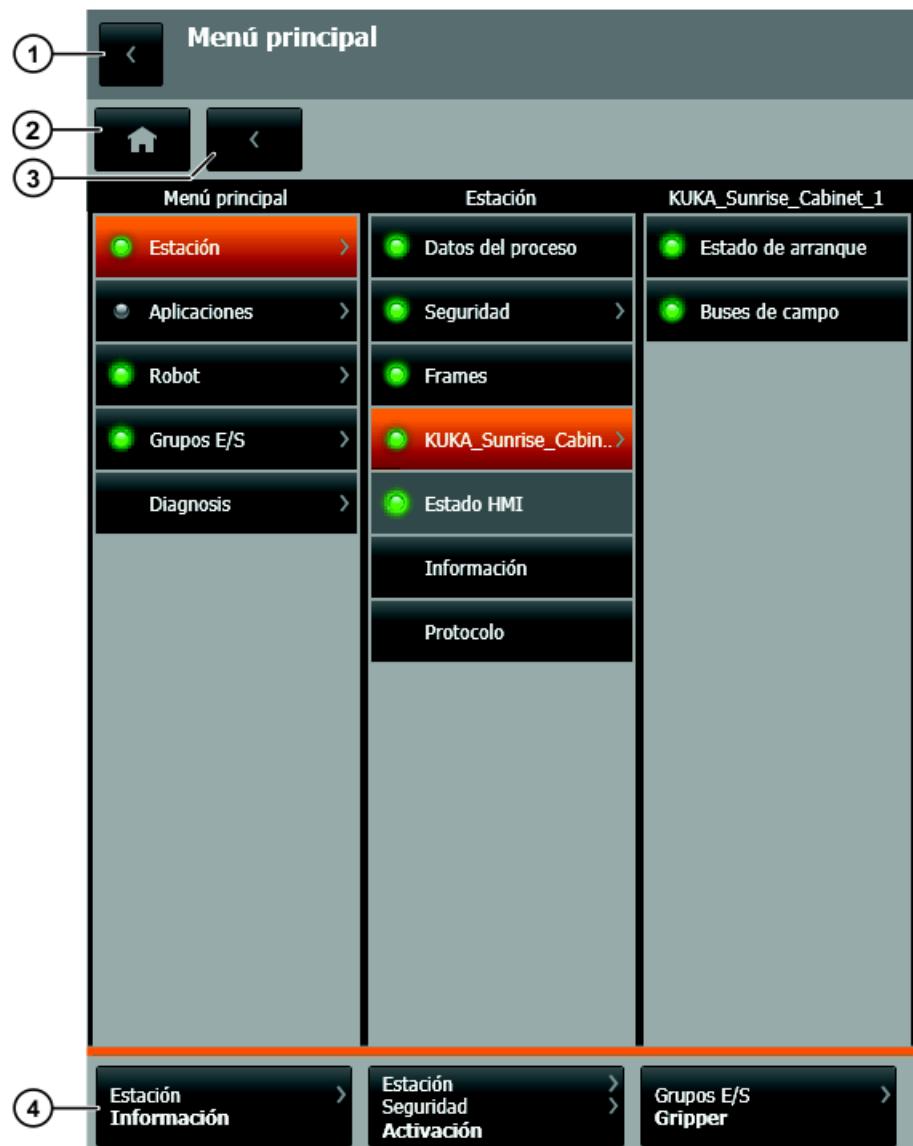


Fig. 6-8: Vista de ejemplo del menú principal

Pos.	Descripción
1	Botón Atrás Mediante este botón se retrocede a la vista que estaba visualizada antes de abrir el menú principal.
2	Botón Inicio Oculta todas las zonas abiertas.

Pos.	Descripción
3	Botón para cerrar el nivel Oculta el nivel abierto más profundo.
4	Aquí se muestran las vistas abiertas por última vez a través del menú principal (máximo 3). Tocando la indicación correspondiente se puede cambiar directamente a estas vistas, sin tener que navegar a través del menú principal.

6.6 Cambiar modo de servicio

Descripción El modo de servicio se puede ajustar con el smartPAD mediante el administrador de conexiones.



Existe la posibilidad de cambiar el modo de servicio cuando se está ejecutando una aplicación en la unidad de control del robot. En este caso, el robot industrial se detendrá con una parada de seguridad 1 y la aplicación es pausada. Cuando el nuevo modo de servicio está ajustado, puede continuar la aplicación.

Condición previa ■ La llave está insertada en el interruptor para activar el administrador de conexiones

- Procedimiento**
1. En el smartPAD, girar a la derecha el interruptor del administrador de conexiones. Se visualiza el administrador de conexiones.
 2. Seleccionar el modo de servicio.
 3. En el smartPAD, girar a la izquierda el interruptor del administrador de conexiones.

En la barra de navegación de la smartHMI se muestra el modo de servicio seleccionado.

Modo de servicio	Utilización	Velocidades
T1	Programación, programación por aprendizaje y prueba de programas	<ul style="list-style-type: none"> ■ Verificación del programa: velocidad reducida programada, máximo 250 mm/s ■ Modo de servicio manual: velocidad de desplazamiento manual, máximo 250 mm/s
T2	Prueba de programas Solo es posible con la puerta de protección cerrada	<ul style="list-style-type: none"> ■ Verificación del programa: velocidad programada ■ Modo de servicio manual: no es posible

Modo de servicio	Utilización	Velocidades
AUT	Ejecución automática de programas Para robots industriales con y sin unidad de control superior	<ul style="list-style-type: none"> ■ Servicio con programa: velocidad programada ■ Modo de servicio manual: no es posible
KRF	<ul style="list-style-type: none"> ■ Extraer el robot industrial de una zona vulnerable cartesiana o específica de eje. ■ Extraer el robot industrial de una zona vulnerable de la orientación de herramienta ■ Liberar el robot industrial de situaciones de bloqueo en caso de vulneración de límites de fuerza o de momento ■ Desplazar el robot industrial si con el control de velocidad cartesiano activo, existe una pérdida de ajuste para al menos un sensor de posición <p>KRF es un modo de servicio que está disponible si el control de seguridad detiene al robot industrial por alguna de las siguientes causas:</p> <ul style="list-style-type: none"> ■ El robot industrial vulnera un espacio controlado de forma segura. ■ La orientación de la herramienta destinada a la seguridad está fuera del sector controlado de forma segura. ■ El robot industrial vulnera un límite de fuerza o de momento controlado de forma segura. ■ Un sensor de posición no está ajustado con el control cartesiano de velocidad activo. 	<ul style="list-style-type: none"> ■ Servicio con programa: velocidad reducida programada, máximo 250 mm/s ■ Modo de servicio manual: velocidad de desplazamiento manual, máximo 250 mm/s

6.7 Sistemas de coordenadas

Los sistemas de coordenadas o Frames determinan la posición y la orientación de un objeto en el espacio.

Vista general

Los siguientes sistemas de coordenadas son relevantes para la unidad de control del robot:

- Mundo
- Pie del robot
- Base
- Brida
- Herramienta

Descripción

Sistema de coordenadas Mundo

El sistema de coordenadas Mundo es un sistema de coordenadas cartesianas de definición fija. Es el sistema de coordenadas de origen para todos los sis-

temas de coordenadas, especialmente para los sistemas de coordenadas base y el sistema de coordenadas del pie del robot.

Por defecto, el sistema de coordenadas Mundo se encuentra en el pie del robot.

Sistema de coordenadas del pie del robot

El sistema de coordenadas del pie del robot es un sistema de coordenadas cartesianas que siempre se encuentra en el pie del robot. Describe la posición del robot en relación con el sistema de coordenadas Mundo.

Por defecto, el sistema de coordenadas del pie del robot se cubre con el sistema de coordenadas Mundo. Se puede definir un desplazamiento del robot respecto al sistema de coordenadas Mundo modificando la dirección de montaje al crear el proyecto Sunrise. Por defecto se encuentra ajustada la dirección de montaje del robot montado en el suelo. ($A=0^\circ$, $B=0^\circ$, $C=0^\circ$).

(>>> 5.3 "Crear proyecto Sunrise con plantilla" Página 51)

Sistema de coordenadas base

Para definir movimientos en la zona cartesiana, es necesaria la indicación de un sistema de coordenadas de referencia (base).

Por defecto se utiliza el sistema de coordenadas Mundo como sistema de coordenadas base para un movimiento. Se pueden definir otros sistema de coordenadas base en relación con el sistema de coordenadas mundial.

(>>> 7.2.2 "Medición base: Método de 3 puntos" Página 112)

Sistema de coordenadas de la brida

El sistema de coordenadas de la brida describe la posición y la orientación actuales del punto central de la brida del robot. No está sujeto a un lugar fijo y se mueve junto con el robot.

El sistema de coordenadas de la brida se utiliza como origen para sistemas de coordenadas que describen las herramientas montadas en la brida.

Sistema de coordenadas de la herramienta

El sistema de coordenadas de la herramienta es un sistema de coordenadas cartesianas cuyo punto de trabajo se encuentra en la herramienta montada. Dicho punto se denomina como Tool Center Point (TCP) (punto central de la herramienta).

Para una herramienta se puede definir el número deseado de Frames que pueden seleccionarse como TCP. El origen del sistema de coordenadas de la herramienta es generalmente idéntico al del sistema de coordenadas de la brida.

(>>> 9.3.1 "Estructura geométrica de las herramientas" Página 144)

El sistema de coordenadas de herramientas es desplazado por el usuario al punto de trabajo de la herramienta.

(>>> 7.2.1 "Medir la herramienta" Página 106)

Posición y orientación

Para determinar la posición y la orientación de un objeto, se indican la translación y la rotación referidas a un sistema de coordenadas de referencia. Para ello se utilizan 6 coordenadas.

Translación

Coordenadas	Descripción
Trayecto X	Desplazamiento a lo largo del eje X del sistema de referencia

Coordenadas	Descripción
Trayecto Y	Desplazamiento a lo largo del eje Y del sistema de referencia
Trayecto Z	Desplazamiento a lo largo del eje Z del sistema de referencia

Rotación

Coordenadas	Descripción
Ángulo A	Giro alrededor del eje Z del sistema de referencia
Ángulo B	Giro alrededor del eje Y del sistema de referencia
Ángulo C	Giro alrededor del eje X del sistema de referencia

6.8 Desplazar el robot de forma manual**Resumen**

Existen dos formas de desplazar el robot de forma manual:

- Desplazamiento cartesiano
El TCP ajustado es desplazado en dirección positiva o negativa a lo largo de los ejes de un sistema de coordenadas o es girado en torno a estos ejes.
- Desplazamiento específico de los ejes
Cada eje puede ser desplazado individualmente en dirección positiva y negativa.

6.8.1 Ventana Opciones de desplazamiento manual**Procedimiento**

Abrir la ventana **Opciones de desplazamiento manual**:

- Tocar el botón **Opciones de desplazamiento manual**.

Cerrar la ventana **Opciones de desplazamiento manual**:

- Tocar el botón **Opciones de desplazamiento manual** o una zona fuera de la ventana.

Descripción

Todos los parámetros para el desplazamiento manual del robot se ajustan en la ventana **Opciones de procesos manuales**.



Fig. 6-9: Ventana Opciones de desplazamiento manual

Pos.	Descripción
1	<p>Botón Opciones de desplazamiento manual</p> <p>El símbolo indicado depende del tipo de desplazamiento ajustado.</p>
2	<p>Seleccionar el tipo de desplazamiento.</p> <p>Se puede realizar el desplazamiento de forma específica del eje y de forma cartesiana en diferentes sistemas de coordenadas. El tipo de desplazamiento seleccionado está marcado en verde y se muestra en el botón Opciones de desplazamiento manual.</p> <ul style="list-style-type: none"> ■ Ejes: El desplazamiento se realiza de forma específica del eje. ■ Mundo: El TCP seleccionado se desplaza de forma cartesiana en el sistema de coordenadas mundial. ■ Base: El TCP seleccionado se desplaza de forma cartesiana en el sistema de coordenadas base seleccionado. ■ Herramienta: El TCP seleccionado se desplaza de forma cartesiana en el sistema de coordenadas propio de la herramienta.
3	<p>Seleccionar la brida del robot o la herramienta montada. No es posible mientras que se ejecute la aplicación.</p> <p>Los Frames de la herramienta seleccionada se pueden seleccionar como TCP para el desplazamiento manual cartesiano. Los datos de carga ajustados de la herramienta se tienen en cuenta.</p> <p>Si se pausa una aplicación del robot, se podrá seleccionar la herramienta utilizada actualmente en la aplicación bajo el nombre Herramienta de aplicación.</p> <p>(>>> "Herramienta de aplicación" Página 79)</p>
4	<p>Seleccionar TCP.</p> <p>Como TCP se encuentran disponibles todos los Frames de la herramienta seleccionada.</p> <p>El TCP ajustado aquí manualmente se conserva. Este también es el caso si se pausa una aplicación que ha utilizado otro TCP.</p> <p>Excepciones:</p> <ul style="list-style-type: none"> ■ En las Opciones de procesos manuales se ha seleccionado la herramienta de aplicación. Si en este caso se modifica la herramienta empleada en la aplicación y a continuación se ejecuta una instrucción de movimiento, también se adaptan las herramientas de aplicación en las Opciones de procesos manuales. Como TCP ajustado en las Opciones de procesos manuales se toma automáticamente el Frame de la herramienta de aplicación con el que se ha ejecutado el último movimiento ordenado. ■ En la opción de desplazamiento manual, está seleccionado el TCP del programa de la herramienta de aplicación. En este caso, el TCP para el desplazamiento manual cambia de acuerdo con el TCP utilizado actualmente en la aplicación.
5	<p>Seleccionar base. Solo es posible si se ha seleccionado el tipo de desplazamiento Base.</p> <p>Como base se encuentran disponibles todos los Frames que han sido identificados en Sunrise.Workbench como base.</p>

Herramienta de aplicación

La herramienta de aplicación se compone de todos los Frames que se encuentran por debajo de la brida del robot durante el tiempo de ejecución. Pue-

den ser Frames de una herramienta o una pieza conectadas a la brida del robot mediante la instrucción attachTo, así como Frames creados en la aplicación y enlazadas directa o indirectamente con la brida.

La herramienta de aplicación solo se puede seleccionar en las opciones de desplazamiento manual, cuando una aplicación del robot está pausada y antes de pausar se ha enviado una instrucción de movimiento a la unidad de control del robot.

Si se selecciona la herramienta de aplicación, en las opciones de desplazamiento manual se ajustará automáticamente el Frame como TCP, con el que se ejecuta la instrucción de movimiento actual en la aplicación. Todos los demás Frames que se encuentran jerárquicamente debajo del sistema de coordenadas de la brida durante el tiempo de ejecución, también se pueden seleccionar como TCP para el desplazamiento manual.

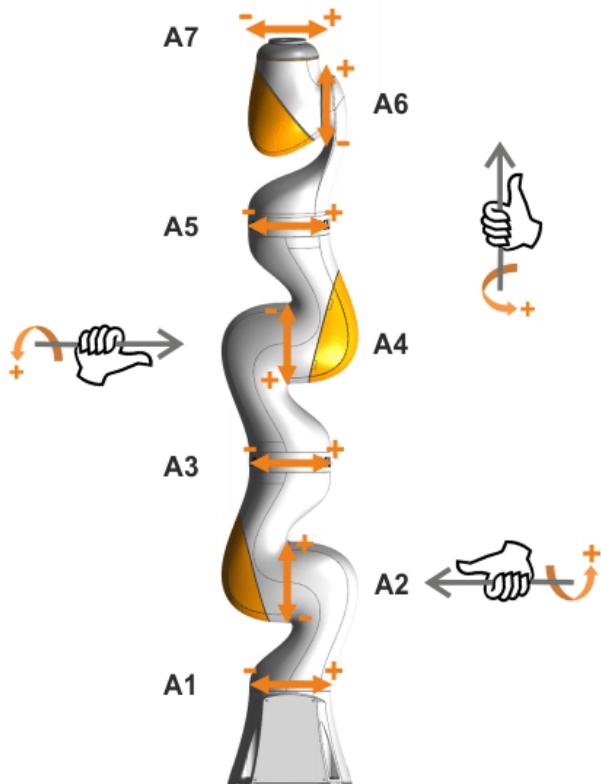
El Frame de la brida del robot está disponible con el nombre **Application-Tool(Root)** como TCP para el desplazamiento manual.

6.8.2 Ajustar el override manual (HOV)

Descripción	El override manual determina la velocidad del robot en el desplazamiento en modo manual. La velocidad que el robot alcanza realmente con el 100 % del override manual depende de diferentes factores, entre otros, del tipo de robot. Sin embargo, la velocidad del punto de trabajo ajustado no debe superar 250 mm/s.
Procedimiento	<ol style="list-style-type: none">1. Tocar el botón HOV. Se abre la ventana Velocidad de desplazamiento manual.2. Ajustar el override manual deseado. Se puede ajustar mediante las teclas positivas y negativas o mediante el regulador.<ul style="list-style-type: none">■ Teclas positivas y negativas: El override se puede ajustar paso a paso con los siguientes valores: 100%, 75%, 50%, 30%, 10%, 5%, 3%, 1%, 0%.■ Regulador: El Override puede modificarse en pasos de 1 %.3. Tocar el botón HOV o una zona fuera de la ventana para cerrar la ventana.
Procedimiento alternativo	Como alternativa, el override se puede ajustar con la tecla positiva y negativa que se encuentra a la derecha en el smartPAD. Ajuste en los pasos 100 %, 75 %, 50 %, 30 %, 10 %, 5 %, 3 %, 1 %, 0 %.

6.8.3 Con las teclas de desplazamiento, desplazar en específicos del eje

Condición previa	<ul style="list-style-type: none">■ Modo de servicio T1
Procedimiento	<ol style="list-style-type: none">1. En las opciones de desplazamiento manual seleccionar Ejes como tipo de desplazamiento. Junto a las teclas de desplazamiento se muestran los ejes A1 a A7.2. Ajustar el override manual.3. Presionar el pulsador de validación y mantenerlo presionado. Si el movimiento habilitado se ha concedido, se resaltarán las indicaciones junto a las teclas de desplazamiento.4. Pulsar la tecla de desplazamiento positiva o negativa para mover un eje en dirección positiva o negativa.

Descripción**Fig. 6-10: Desplazamiento específico del eje**

El sentido de giro positivo de los ejes del robot se puede determinar con la regla de la mano derecha. Para ello, debe imaginarse el haz de cables que va dentro del robot desde el pie hasta la brida. Los dedos de la mano derecha rodean (figuradamente) el haz de cables en el punto del eje observado. En este caso, es importante que el pulgar quede extendido. De este modo, el pulgar se encuentra sobre el haz de cables, de forma que señale en la misma dirección en la que el haz de cables transcurra hacia la brida en el interior del eje. El resto de dedos de la mano derecha señala en el sentido de giro positivo del eje del robot.

6.8.4 Con las teclas de desplazamiento, desplazar de forma cartesiana

Condición previa ■ Modo de servicio T1

Procedimiento 1. Seleccionar el sistema de coordenadas para el desplazamiento manual cartesiano. Están disponibles **Mundo**, **Base** y **Herramienta**.

Junto a las teclas de desplazamiento se muestran las siguientes denominaciones:

- **X, Y, Z:** Para los movimientos lineales a lo largo de los ejes del sistema de coordenadas seleccionado
- **A, B, C:** Para los movimientos de rotación alrededor de los ejes del sistema de coordenadas seleccionado
- **R:** para el movimiento de espacio cero

2. Seleccionar la herramienta deseada y el TCP deseado de la herramienta.
3. Si está seleccionado el sistema de coordenadas **Base** para el desplazamiento manual cartesiano, seleccionar el Frame base deseado.



Los Frames que se desee que estén disponibles como base para el desplazamiento manual deben marcarse antes en Sunrise.Workbench como base.

(>>> 9.2.2 "Marcar Frame como base" Página 140)

4. Ajustar el override manual.
5. Accionar y mantener pulsado el pulsador de validación.
Si el movimiento habilitado se ha concedido, se resaltarán las indicaciones junto a las teclas de desplazamiento.
6. Pulsar la tecla de desplazamiento positiva o negativa para mover un eje en dirección positiva o negativa, respectivamente.

6.8.4.1 Movimiento de espacio cero

Descripción

El robot de estructura liviana posee 7 ejes y, de este modo, tiene una redundancia cinemática. Esto significa que puede desplazarse hasta cualquier punto del campo de trabajo teóricamente con un número infinito de configuraciones del eje.

Debido a la redundancia cinemática, se puede ejecutar un denominado como movimiento de espacio cero para el desplazamiento. Durante el movimiento de espacio cero, los ejes giran de forma que la posición y la orientación del TCP ajustado se pueda conservar durante el movimiento.



Fig. 6-11: Movimiento de espacio cero

Propiedades

- El movimiento de espacio cero se ejecuta con el "codo" del brazo del robot.
- La posición del codo se describe mediante el ángulo del codo (R).
- En el desplazamiento cartesiano con las teclas de desplazamiento se puede modificar la posición del ángulo del codo (R).

Campos de aplicación

- Para una posición y una orientación dadas del TCP se puede adoptar la configuración óptima del eje. Esto resulta especialmente útil en un campo de trabajo limitado.
- Al alcanzar un interruptor de final de carrera de software, se puede intentar desplazar el robot fuera de los interruptores de final de carrera mediante la modificación del ángulo del codo.

6.9 Modo de servicio KRF – Desbloquear el robot de forma controlada

Descripción

KRF es un modo de servicio que está disponible cuando el robot ha sido detenido por el control de seguridad y la línea de activación en la configuración de seguridad contiene uno de siguientes controles:

- Control del campo del eje

- Control de velocidad cartesiana
- Control del campo de trabajo cartesiano
- Monitorización cartesiana de las zonas de protección
- Orientación de la herramienta
- Control del momento del eje
- Detección de colisión
- Control de fuerza TCP

En el modo de servicio KRF se puede desplazar el robot si ha sido detenido por el control de seguridad debido a una de las siguientes causas.

- El robot vulnera un espacio controlado de forma segura.
- La orientación de la herramienta destinada a la seguridad está fuera del sector controlado de forma segura.
- El robot vulnera un límite de fuerza o de momento controlado de forma segura.
- Un sensor de posición no está ajustado con el control cartesiano de velocidad activo.

La velocidad de desplazamiento del punto de trabajo ajustado en el modo de servicio KRF corresponde a la velocidad de desplazamiento manual en el modo de servicio T1, como máximo 250 mm/s.

En el modo de servicio KRF, el robot puede desplazarse independientemente de los controles activados. Al pasar por otros límites de control no se activa ninguna parada. Los controles de velocidad siguen estando activos en el servicio KRF.

Procedimiento

1. Cambiar al modo de servicio KRF.
2. Desplazar el robot manualmente hasta una posición en la que ya no esté vulnerando el control que activa la parada.

Si los parámetros supervisados vuelven a estar fuera de la zona vulnerable y continúan en la zona permitida después de 4 segundos, se cambiará automáticamente al modo de servicio T1.

6.10 Guiar el robot manualmente

⚠ ATENCIÓN

El operario tiene que llevarse el smartPAD para guiar el robot manualmente. Debe asegurarse que durante el desplazamiento manual ninguna otra persona puede manejar de forma inadvertida el smartPAD y desplazar al robot. Si no se tiene en cuenta, pueden producirse lesiones o daños materiales.

6.11 Reanudar control de seguridad

Descripción

En caso de errores de conexión o errores periféricos se detiene el control de seguridad (según el error se detiene a la primera aparición o en apariciones sucesivas). Debido a la detención del control de seguridad, se detiene el robot y se desconectan todas las salidas seguras. Cuando el error se haya solucionado, se puede reanudar el control de seguridad.

Procedimiento

1. En la vista de la estación, seleccionar **Seguridad > Estado**. Se abre la vista **Estado**.
En la vista se indica la causa del error. El botón **Continuar el control de seguridad** está inactivo.
2. Subsanar el error. Ahora está activo el botón **Continuar el control de seguridad**.

3. Pulsar **Continuar el control de seguridad**. Se continúa con el control de seguridad.

6.12 Abrir los frenos de retención

Descripción

Si el robot entra en una situación de bloqueo, por la que el par de torsión máximo del sensor de momentos de articulaciones se supera en al menos un eje, se pausará el control de seguridad. El robot ya no se puede desplazar.

En este caso, existe la posibilidad de reducir el momento que está actuando sobre los ejes afectados, mediante la apertura breve de los frenos de retención.

⚠ ATENCIÓN

Tras la superación del par de torsión máximo se rechaza la referenciación de los sensores de posición y de momentos de los ejes afectados. Para garantizar la integridad de la seguridad de los controles dependiente de la posición y del momento del eje, se deberán referenciar de nuevo los sensores de posición y de momentos de los ejes afectados después de eliminar el atrapamiento.

Par de torsión máximo

El par de torsión máximo se encuentra entre el par de torsión nominal y límite del sensor. El sensor se puede cargar hasta el par de torsión límite sin sufrir daños permanentes. Para tener en cuenta la precisión de medición aumentada más allá del par de torsión nominal, el control de seguridad ya se pausará al alcanzar el par de torsión máximo.

Mientras que el valor determinado por el sensor no supere el par de torsión máximo, está asegurado que el par de torsión límite no se ha superado.

Condición previa

- El par de torsión máximo del sensor de momentos de articulaciones se ha superado en al menos un eje.

Procedimiento

1. En la vista de la estación, seleccionar **Seguridad > Estado**. Se abre la vista **Estado**.
En la vista se indica un mensaje de error por superar el par de torsión máximo.
2. Accionar y mantener pulsado el pulsador de validación. Los frenos de retención de los ejes afectados se abren durante pocos milisegundos.



Al abrir los frenos, el robot se mueve ligeramente.

3. Soltar el pulsador de validación después de que los frenos se hayan cerrado de nuevo.
4. En caso de que en al menos un eje se siga superando el par de torsión máximo del sensor de momentos de articulaciones: Repetir el paso 1 y 2 hasta que el momento se encuentre por debajo del par de torsión máximo en todos los ejes.
5. Pulsar **Continuar el control de seguridad**. Se continúa con el control de seguridad.
6. En caso necesario, desplazar el robot mediante el desplazamiento manual hasta una posición no crítica.
7. Si se han configurado funciones de seguridad, emplear los AMF basados en el momento de eje o en la posición: realizar una referenciación de momento y/o posición.
(>>> 13.10.2 "Referenciación del momento" Página 228)
(>>> 13.10.1 "Referenciación de posición" Página 227)

6.13 Programar por aprendizaje y desplazar manualmente los Frames

6.13.1 Visualizar Frames

Procedimiento

- En la vista de la estación, seleccionar **Frames**. Se abre la vista de Frames.

Descripción

La vista contiene los Frames creados para el proyecto Sunrise. Aquí se programan por aprendizaje los Frames. Mediante la programación por aprendizaje se registran la posición y la orientación de un Frame en el espacio, así como la información de redundancia correspondiente.

- El desplazamiento hasta los Frames programados por aprendizaje se puede realizar manualmente.
 - Los Frames programados por aprendizaje se puede utilizar como puntos de destino de movimientos. Si se ejecuta una aplicación y se desplaza el Frame de destino de un movimiento, se marcará en la vista de Frames.
- (>>> 6.16.1 "Visualizar el Frame de destino del movimiento ejecutado actualmente" Página 97)

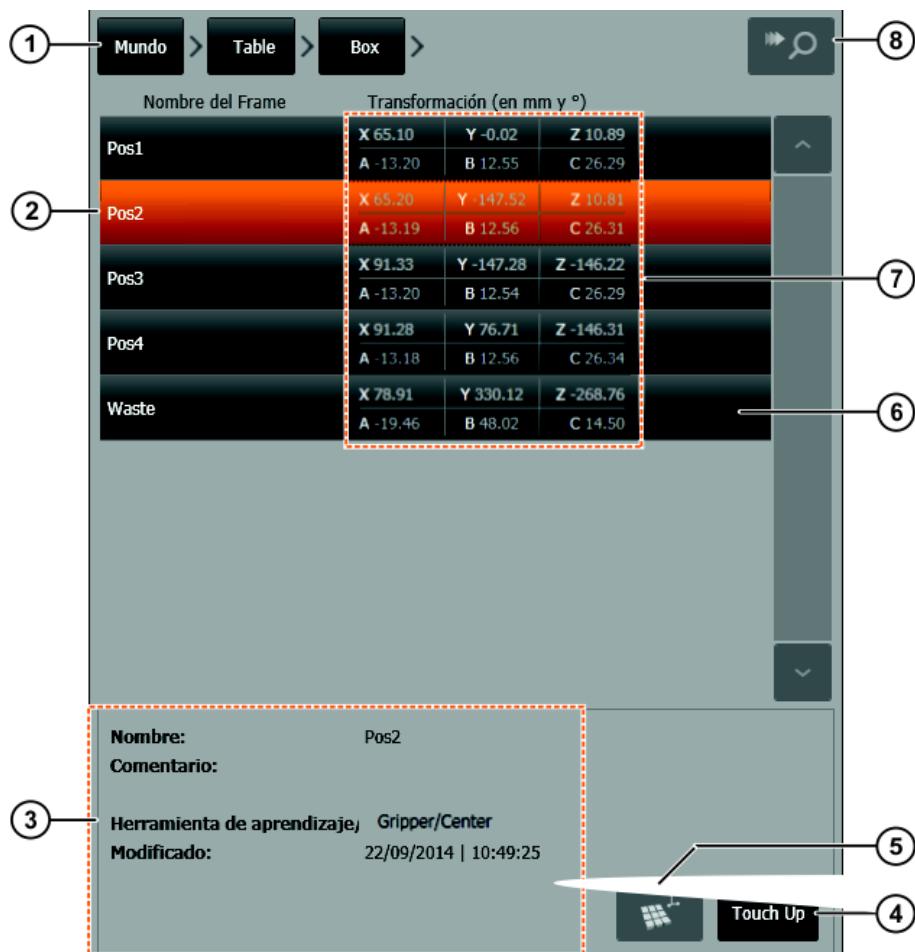


Fig. 6-12: Vista de Frames

Pos.	Descripción
1	Ruta del Frame Ruta hasta los Frames del nivel de jerarquía mostrado actualmente. Partiendo de World hasta el Frame padre directo (aquí Box).
2	Frames del nivel de jerarquía actual Un Frame se puede marcar tocándolo. El Frame marcado aquí está identificado con el símbolo de una mano. El símbolo de la mano significa que este Frame se puede utilizar como base para el desplazamiento manual y se puede medir.
3	Propiedades del Frame marcado <ul style="list-style-type: none"> ■ Nombre del Frame ■ Comentario ■ Herramienta utilizada para programar por aprendizaje el Frame ■ Fecha y horario de la última modificación
4	Botón Touch Up Un Frame marcado se puede programar por aprendizaje. Si no se encuentra marcado ningún Frame, el botón está inactivo.
5	Botón para ajustar la base para el desplazamiento manual Mediante el botón se ajusta el Frame marcado en las opciones de desplazamiento manual como base para el desplazamiento manual. (>>> 6.8.1 "Ventana Opciones de desplazamiento manual" Página 78) El botón solo está activo si en las opciones de desplazamiento manual está seleccionado el tipo de desplazamiento Base y el Frame marcado se ha identificado como base en Sunrise.Workbench.
6	Botón para mostrar Frames hijos El botón solo está disponible si un Frame posee subelementos. Mediante el botón se muestran los subelementos directos de un Frame.
7	Coordinadas del Frame referidas al Frame padre
8	Botón Buscar El botón Buscar solo está activo cuando se está ejecutando una aplicación y se realiza el desplazamiento de un Frame de destino de un movimiento. Mediante el botón se puede conmutar hasta este Frame de destino, en caso de que aún no se muestre.

6.13.2 Programar por aprendizaje los Frames

Descripción

Las coordenadas de un Frame pueden modificarse mediante la smartHMI. Para ello, con el TCP deseado se avanza hasta la nueva posición del Frame y se programa el Frame. Se aceptan la nueva posición y la orientación.



Después de programar por aprendizaje los Frames, se recomienda sincronizar el proyecto para que los datos del Frame también se actualicen en el proyecto correspondiente en Sunrise.Workbench.

Condición previa

- La herramienta con el TCP deseado está ajustada en las opciones de desplazamiento manual.
- (>>> 6.8.1 "Ventana Opciones de desplazamiento manual" Página 78)

i Se recomienda no usar la herramienta de aplicación para programar Frames por aprendizaje, ya que la herramienta de aplicación solo está disponible en pausas de aplicación en las opciones de desplazamiento manual. En su lugar, se puede emplear la herramienta actual correspondiente a la herramienta de aplicación que se ha creado en las plantillas de objetos.

- Modo de servicio T1

Procedimiento

- Desplazar la posición deseada del Frame con el TCP.
- En la vista de Frames, marcar el Frame cuya posición se debe programar por aprendizaje.
- Pulsar **Touch Up** para aceptar las coordenadas TCP actuales para el Frame marcado.
- Las coordenadas y la información de redundancia del punto programado por aprendizaje se muestran en el diálogo **Aceptar los datos Touchup**. Pulsar **Aceptar** para guardar los nuevos valores.

ADVERTENCIA Si se modifica un Frame, la modificación se producirá en todas las aplicaciones en las que se utilice el Frame. Los programas modificados siempre se deben probar primero en el modo de servicio Manual Velocidad reducida (T1).

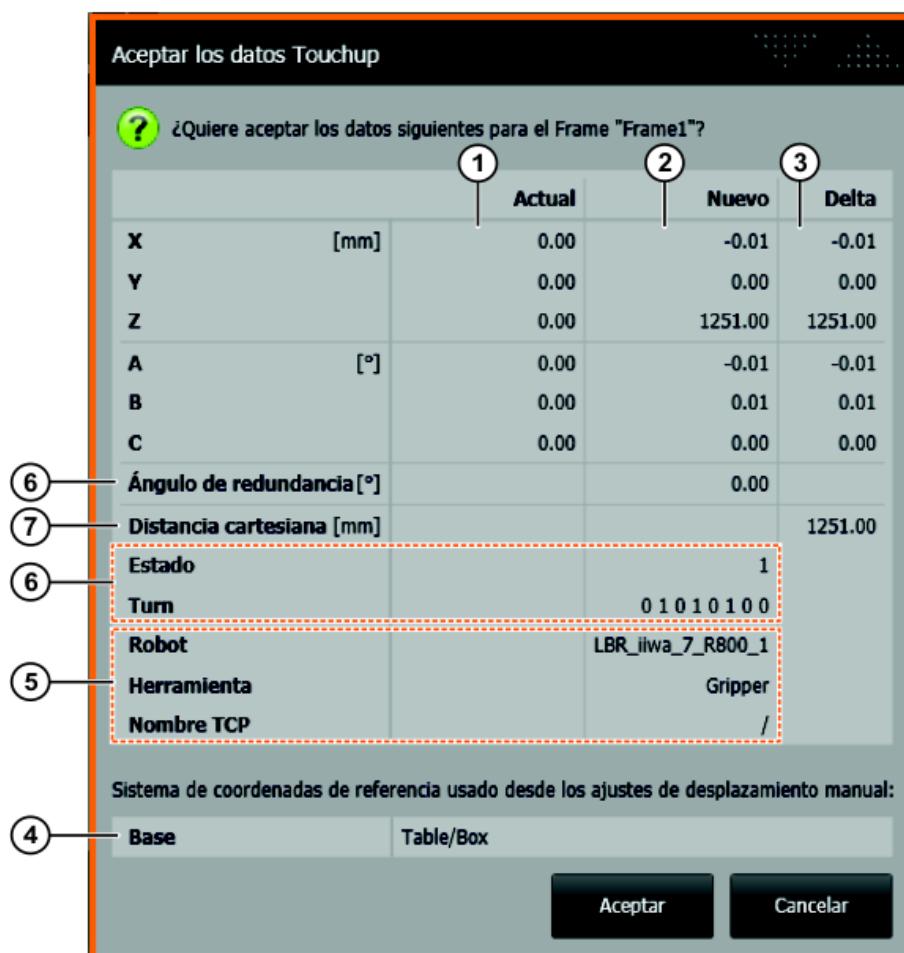


Fig. 6-13: Aceptar los datos Touch Up

Pos.	Descripción
1	Valores guardados anteriormente
2	Valores nuevos
3	Diferencias entre los valores guardados hasta ahora y los nuevos
4	Base para desplazamiento manual Todos los valores de coordenadas indicados en el diálogo del Frame se refieren a la base ajustada para el desplazamiento manual en las opciones de desplazamiento manual. Por lo general, estos valores se diferencian de los valores de coordenadas del Frame con respecto a su Frame padre. (>>> 6.8.1 "Ventana Opciones de desplazamiento manual" Página 78)
5	Información acerca del robot y la herramienta que se han utilizado durante la programación por aprendizaje.
6	Información de redundancia acerca del punto programado por aprendizaje
7	Distancia cartesiana entre la posición actual y la posición nueva del Frame

6.13.3 Ventana Tipo de desplazamiento

Procedimiento

Abrir la ventana **Tipo de desplazamiento**:

- Tocar el botón **Tipo de desplazamiento** junto a la tecla de arranque.

Cerrar la ventana **Tipo de desplazamiento**:

- Tocar el botón **Tipo de desplazamiento** o una zona fuera de la ventana.

Descripción

En la ventana **Tipo de desplazamiento** se puede configurar la funcionalidad de la tecla de arranque.



Fig. 6-14: Ventana Tipo de desplazamiento

Pos.	Descripción
1	Botón Tipo de desplazamiento El símbolo indicado depende del tipo de desplazamiento ajustado.
2	Tipo de desplazamiento Modo de aplicación En este tipo de desplazamiento, la tecla de arranque sirve para iniciar aplicaciones. Indicación: Al cambiar al modo de servicio T2 o automático se ajusta automáticamente el tipo de desplazamiento Modo de aplicación .
3	Tipos de desplazamiento Desplazamiento a PTP / Desplazamiento a LIN En estos tipos de desplazamiento, la tecla de arranque sirve para el desplazamiento manual hasta Frames con un movimiento PTP o LIN. Solo se pueden seleccionar si un Frame está marcado en la vista de Frames y está ajustado el modo de servicio T1. Indicación: En el tipo de desplazamiento Desplazamiento a PTP se tiene en cuenta el estado del Frame de destino. De este modo puede producirse el movimiento de los ejes, incluso si el punto de destino ya se ha alcanzado desde el punto de vista cartesiano. Indicación: En el tipo de desplazamiento Desplazamiento a LIN no se tiene en cuenta el estado del Frame de destino.
4	Botón Abrir la vista de marcos Mediante el botón se puede cambiar a la vista de Frames.
5	Indicación de Frames Aquí se indica el nombre del Frame que está marcado actualmente en la vista de Frames.

Símbolos

Los símbolos se muestran en función del tipo de desplazamiento ajustado en el botón **Tipo de desplazamiento**:

Símbolo	Descripción
	Tipo de desplazamiento Modo de aplicación
	Tipo de desplazamiento Desplazar al PTP
	Tipo de desplazamiento Desplazar al LIN

6.13.4 Desplazamiento manual hasta Frames**Descripción**

Los Frames programados por aprendizaje se pueden desplazar manualmente con un movimiento PTP o un movimiento LIN. Con un movimiento PTP, el Frame se desplaza por el recorrido más rápido y con un movimiento LIN en una trayectoria previsible.

Durante el desplazamiento de Frames se mostrará un mensaje de advertencia en los siguientes casos:

- La herramienta seleccionada no se corresponde con la herramienta con la que el Frame se ha programado por aprendizaje.

- El TCP seleccionado no se corresponde con el TCP con el que el Frame se ha programado por aprendizaje.
- La transformación del Frame TCP se ha modificado.

Si a pesar de ello, el Frame se puede alcanzar, se podrá realizar un desplazamiento.

Condición previa

- Se puede realizar el desplazamiento hasta el Frame con el TCP seleccionado.
- Modo de servicio T1

Procedimiento

1. Marcar el Frame deseado en la vista de Frames.
2. En la ventana **Tipo de desplazamiento** seleccionar el tipo de desplazamiento deseado.
3. Presionar el pulsador de validación y mantenerlo presionado.
4. Pulsar la tecla de arranque y mantenerla pulsada hasta que se alcance el Frame.



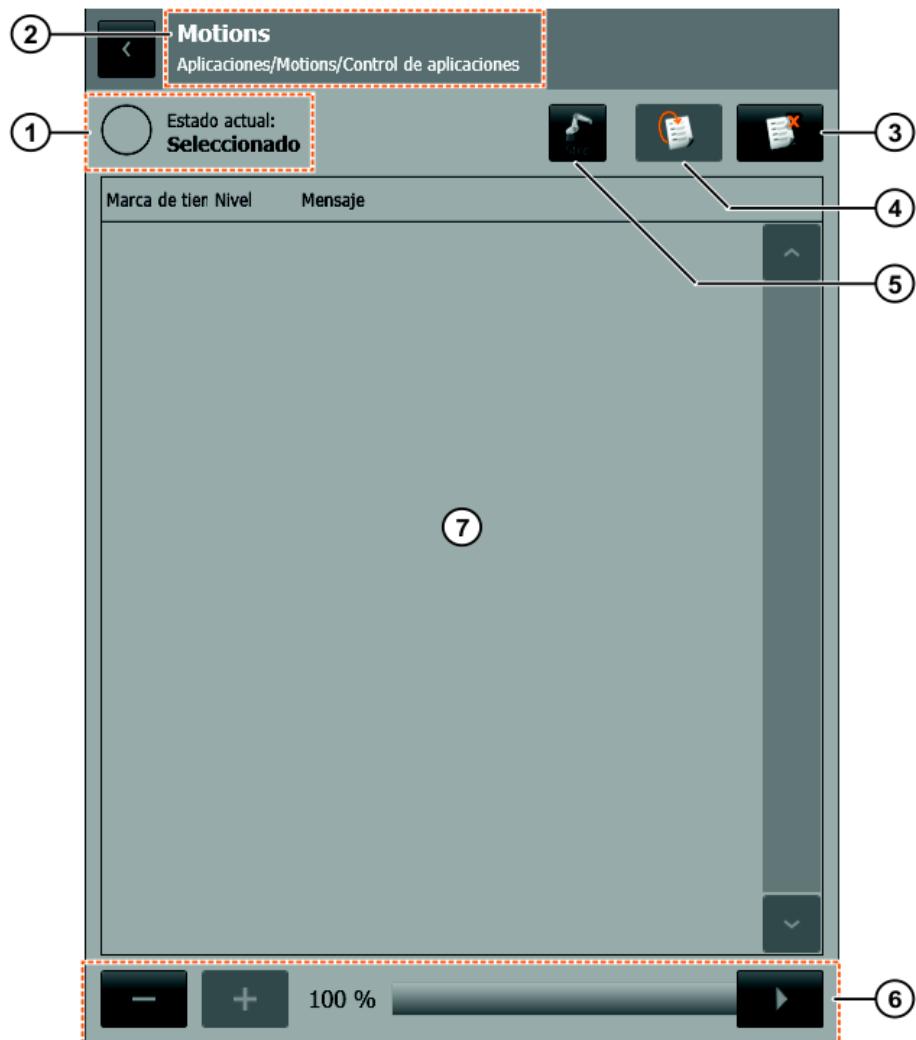
Si el punto de trabajo seleccionado ya se encuentra en la posición de destino o si no se puede acceder al Frame con los ajustes actuales, el robot no ejecuta ningún movimiento.

6.14 Ejecución del programa

6.14.1 Seleccionar la aplicación de robot

Procedimiento

- En la barra de navegación, seleccionar en **Aplicaciones** la aplicación deseada. La vista de la aplicación se abre y la aplicación pasa al estado **Seleccionada**.

Descripción**Fig. 6-15: Vista de la aplicación - Aplicación seleccionada**

Pos.	Descripción
1	Estado actual de la aplicación El estado se visualiza en forma de texto y como símbolo.
2	Visualización de la aplicación Se muestra el nombre de la aplicación seleccionada, en este caso Motions .
3	Botón Deseleccionar Deselecciona la aplicación seleccionada y cierra la vista de aplicación. Una aplicación en pausa se restablece antes de deselegirla. El botón está activo solamente cuando la aplicación está en el estado Seleccionado , Movimiento pausado o Error .
4	Botón Restablecer Restablece una aplicación en pausa. Restablecer significa que la aplicación se vuelve a ajustar al inicio del programa y pasa al estado Seleccionado . El botón está activo solamente cuando la aplicación está en pausa.

Pos.	Descripción
5	<p>Botón Step</p> <p>Pulsando el botón puede cambiarse entre el modo Step y estándar.</p> <p>(>>> 6.14.2 "Seleccionar el modo de ejecución del programa" Página 93)</p>
6	<p>Override manual</p> <p>El Override manual se puede ajustar mediante las teclas + y - o mediante el regulador.</p>
7	<p>Ventana de mensajes</p> <p>Aquí se visualizan mensajes de error y los mensajes de usuario programados en la aplicación.</p>

Indicación de estado de la aplicación

Símbolo	Estado	Descripción
	Seleccionado	La aplicación está seleccionada.
	Arrancar	La aplicación se inicializa.
	Ejecución	La aplicación se ejecuta.
	Movimiento pausado	<p>La aplicación está pausada.</p> <p>Si la aplicación detiene a través del smartPAD, p. ej. pulsando la tecla STOP, se detiene la ejecución del movimiento. Otras instrucciones, p. ej. la conmutación de salidas, se realizan en el estado Movimiento pausado hasta que se haya alcanzado una instrucción de movimiento sincronizada.</p>
	Error	Se ha producido un error durante la ejecución de la aplicación.
	Reposicionamiento	El robot retrocede a la posición previa. La aplicación está pausada, ya que el robot ha salido de la trayectoria.
	Detener	La aplicación se restablece al inicio del programa y pasa al estado Seleccionado .

Tecla de arranque

En el modo de aplicación se encuentran disponibles las siguientes funcionalidades a través de la tecla de arranque:

Símbolo	Descripción
(▶)	Iniciar la aplicación. Una aplicación seleccionada se puede iniciar o una aplicación pausada puede continuar.
(●←)	Retroceder el robot. Si el robot ha abandonado la trayectoria, se deberá retroceder para poder continuar la aplicación.

(i)	Si una aplicación está en pausa, el robot puede desplazarse manualmente. La herramienta utilizada en una aplicación que está en pausa y el TCP actual no están ajustados automáticamente como herramienta y TCP para el procedimiento manual cartesiano. (>>> 6.8.1 "Ventana Opciones de desplazamiento manual" Página 78)
-----	---

Tecla STOP

A través de la tecla STOP está disponible la siguiente funcionalidad:

Símbolo	Descripción
(II)	Pausar la aplicación. Una aplicación en marcha puede detenerse en el modo automático.

6.14.2 Seleccionar el modo de ejecución del programa

- Condición previa**
- Se ha seleccionado la aplicación.
 - Modo de servicio T1 o T2

- Procedimiento**
- Seleccionar un tipo de flujo de programa con el botón **Step**.

Descripción	Modo de ejecución del programa	Descripción
	Continuous	Modo estándar El programa se ejecuta sin detención hasta el final del mismo.
	Step	Modo Step El programa se ejecuta con una parada después de cada instrucción de movimiento. La tecla de arranque debe ser pulsada nuevamente para cada instrucción de movimiento. <ul style="list-style-type: none">■ El punto de destino de un movimiento de posicionamiento aproximado no se aproxima, sino que el desplazamiento se realiza con una parada exacta. Excepción: Los movimientos de posicionamiento aproximado, que ya hayan sido enviados de forma asíncrona a la unidad de control del robot antes de la activación del modo Step y que esperan allí a la ejecución, se detienen en el punto de posicionamiento aproximado. Para estos movimientos aún se recorre el arco de aproximación para continuar.■ Para un movimiento Spline, se recorre el bloque Spline completo como un movimiento y seguidamente se detiene.■ Para un MotionBatch, no se recorre el Batch completo, sino que después de cada movimiento individual del Batch se ejecuta una parada exacta.



El modo de ejecución del programa también se puede ajustar y consultar en el código fuente de la aplicación. ([>>> 15.18 "Modificar y consultar el modo de ejecución del programa"](#) Página 320)

6.14.3 Ajustar el Override manual

Descripción	El Override manual determina la velocidad del robot durante la ejecución del programa. El Override manual no tiene por qué ser idéntico al Override efectivo con el que se está desplazando realmente el robot. Si se ha programado un Override de aplicación que es fijado por la aplicación, el Override efectivo se obtiene de la siguiente forma: Override efectivo = Override manual * Override de aplicación (>>> 15.19 "Modificar y consultar el Override" Página 321) El Override se indica con un valor porcentual y se refiere a la velocidad del programa. En el modo de servicio T1 la velocidad máxima es de 250 mm/s, independientemente del Override ajustado.
Condición previa	<ul style="list-style-type: none">■ La aplicación está abierta.
Procedimiento	<ul style="list-style-type: none">■ Ajustar el Override manual deseado. Se puede ajustar mediante las teclas positivas y negativas o mediante el regulador.

- Teclas positivas y negativas: El override se puede ajustar paso a paso con los siguientes valores: 100%, 75%, 50%, 30%, 10%, 5%, 3%, 1%, 0%.
- Regulador: El Override puede modificarse en pasos de 1 %.

6.14.4 Arrancar el programa hacia adelante (manual)

Condición previa	<ul style="list-style-type: none"> ■ Se ha seleccionado la aplicación. ■ Modo de servicio T1 o T2
Procedimiento	<ol style="list-style-type: none"> 1. Ajustar Override manual. 2. Seleccionar el modo de ejecución del programa. 3. Accionar y mantener pulsado el pulsador de validación. 4. Pulsar y mantener pulsada la tecla de arranque. La aplicación se ejecuta. <p>Para detener un programa arrancado manualmente, soltar la tecla de arranque. Si la aplicación está en pausa, se puede restablecer mediante el botón Restablecer.</p>

6.14.5 Arrancar el programa hacia adelante (automático)

Condición previa	<ul style="list-style-type: none"> ■ Se ha seleccionado la aplicación. ■ Modo de servicio Automático ■ El proyecto no se controla de forma externa.
Procedimiento	<ul style="list-style-type: none"> ■ Pulsar la tecla de arranque. La aplicación se ejecuta. <p>Para pausar un programa iniciado en el modo automático, pulsar la tecla de PARADA. Si la aplicación está pausada, se puede restablecer mediante el botón Restablecer.</p>

6.14.6 Retroceder el robot después de abandonar la trayectoria

Descripción	<p>El robot puede desviarse de su trayectoria planificada debido a las siguientes incidencias:</p> <ul style="list-style-type: none"> ■ Activación de una parada fuera de la trayectoria ■ Desplazamiento manual si la aplicación está pausada <p>El robot puede retroceder con la ayuda de la tecla de arranque. Retroceso significa que el robot es desplazado de vuelta hasta la posición cartesiana en la que se abandonó previamente la trayectoria. Desde allí puede continuar la aplicación.</p> <p>Propiedades del movimiento con las que se retrocede a la trayectoria:</p> <ul style="list-style-type: none"> ■ Se ejecuta un movimiento PTP. El curso de la trayectoria al retroceder al retroceder a la trayectoria es diferente en el momento de abandonar la trayectoria. ■ El desplazamiento se realiza con un 20 % de la velocidad de eje máxima posible y el Override efectivo (Override efectivo = Override manual * Override de aplicación).
--------------------	---



El Override manual actualmente ajustado no es relevante a la hora del retorno a posición.

- Se realiza el desplazamiento con los datos de carga que estaban ajustados en el momento en el que se canceló la aplicación.

- Se realiza el desplazamiento con el modo de regulación que estaba ajustado en el momento en el que se canceló la aplicación.
Las fuerzas o modulaciones de fuerza adicionales obtenidas a través de un regulador de impedancia, se retiran a la hora de realizar el retorno a posición.

AVISO

Si se retorna a posición un robot regulado por impedancia, se pueden producir movimientos inesperados del robot. Como siempre se retorna a la posición nominal, en el caso de un robot regulado por impedancia, la posición real después del retorno no coincide necesariamente con la posición real en la que abandonó la trayectoria. En situaciones de contacto esto puede llevar a fuerzas inesperadamente elevadas.

Antes de realizar el retorno a posición, desplazar un robot regulado por impedancia de forma manual a una posición que se acerque lo máximo posible a la posición en la que abandonó la trayectoria. Si no se tiene en cuenta esta indicación, pueden surgir daños materiales.

AVISO

Solo se puede retroceder si no puede producirse ninguna colisión en el recorrido de vuelta a la trayectoria. Si esto no está asegurado, primero desplazar el robot hasta una posición adecuada desde la que se puede retroceder sin peligro.

Procedimiento

1. En el modo de servicio T1 o T2: Pulsar el pulsador de validación y mantenerlo pulsado.
2. Pulsar y mantener apretada la tecla de arranque. El robot retrocede en la trayectoria.

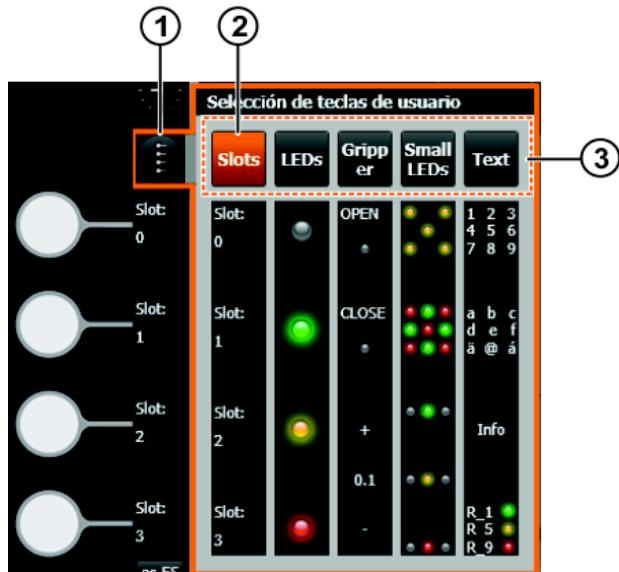
6.15 Activar las teclas de usuario

Descripción

Las teclas de usuario en el smartPAD pueden estar ocupadas con funciones. El operario tiene a su disposición todas las funciones de las teclas de usuario que estén definidas en una aplicación del robot en ejecución o una tarea en segundo plano en ejecución. Para poder utilizar las funciones deseadas, el operario deberá activar la correspondiente barra de teclas de usuario.

Procedimiento

1. Tocar el botón **Selección de teclas de usuario**.
Se abre la ventana **Selección de teclas de usuario**. Se muestran las barras de teclas de usuario disponibles actualmente.
2. Seleccionar la barra de teclas de usuario deseada mediante el botón de nombre correspondiente.
La rotulación o el gráfico de la smartHMI junto a las teclas de usuario cambian de acuerdo con la barra seleccionada. Las teclas de usuario poseen ahora las funciones correspondientes.
3. Tocar el botón **Selección de teclas de usuario** o una zona fuera de la ventana.
La ventana **Selección de teclas de usuario** se cierra.

Ejemplo**Fig. 6-16: Ventana Selección de teclas de usuario**

- 1 Botón **Selección de teclas de usuario**
- 2 Barra de teclas de usuario activa actualmente
- 3 Nombres de las barras de teclas de usuario disponibles

6.16 Funciones de indicación en pantalla**6.16.1 Visualizar el Frame de destino del movimiento ejecutado actualmente****Descripción**

Si se realiza el desplazamiento hasta un Frame del árbol de Frames en una aplicación, se identificará en la vista de Frames. Si el Frame de destino del movimiento ejecutado actualmente se encuentra en el nivel jerárquico indicado, el nombre del Frame aparecerá marcado con un símbolo de flecha (3 puntas de flecha):

Start	X 586.45	Y 0.02	Z 555.34
	A 180.00	B 47.24	C 179.99

Fig. 6-17: El símbolo de flecha marca el Frame de destino actual

Si el Frame de destino se encuentra jerárquicamente debajo de un Frame indicado, el botón para mostrar los Frames hijos aparecerá identificado con un símbolo de flecha adicional (3 puntas de flecha):

Triangle		X 427.07	Y -401.86	Z 555.35	
		A 136.74	B 47.22	C 179.99	

Fig. 6-18: El botón conmuta directamente hasta el Frame de destino actual

Con la ayuda del botón Buscar en la zona superior derecha de la vista de Frames se puede conmutar directamente al Frame de destino actual:

Frame name	Transformation (in mm and deg)		
Start	X 586.45	Y 0.02	Z 555.34
	A 180.00	B 47.24	C 179.99
Triangle	X 427.07	Y -401.86	Z 555.35
	A 136.74	B 47.22	C 179.99

Fig. 6-19: El botón Buscar conmuta directamente hasta el Frame de destino

El botón Buscar está inactivo si no se realiza el desplazamiento hasta ningún Frame.

Condición previa

- Se ha seleccionado la aplicación.
- Estado de la aplicación **Ejecución o Movimiento pausado**.
- El movimiento utiliza un Frame de destino creado en los datos de la aplicación.

Procedimiento

1. En la vista de la estación seleccionar **Frames**. Se abre la vista de Frames.
2. Conmutar al Frame de destino mediante el botón para mostrar Frames hijos o mediante el botón Buscar.

6.16.2 Visualizar la posición real específica del eje

Procedimiento

- En la vista del robot, seleccionar **Posición del eje**.

Descripción

Se muestra la posición actual de los ejes A1 hasta A7. Adicionalmente para cada eje se muestra la zona (barra blanca) en la que se puede desplazar (limitación mediante topes finales).

La posición real también puede mostrarse mientras el robot se encuentra en movimiento.

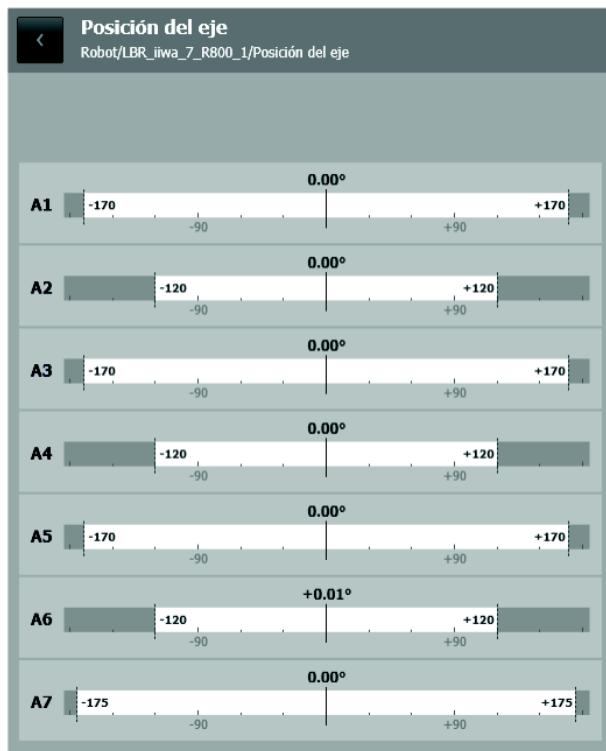


Fig. 6-20: Posición real específica del eje

6.16.3 Visualizar la posición real cartesiana

Procedimiento

1. En la vista del robot, seleccionar **Posición cartesiana**.
2. Ajustar el TCP y la base en la ventana **Opciones de desplazamiento manual**.

Descripción

Se muestra la posición real cartesiana del TCP seleccionado. Los valores hacen referencia a la base ajustada en las opciones de desplazamiento manual.

La indicación contiene los siguientes datos:

- Posición actual (X, Y, Z)
- Orientación actual (A, B, C)
- Información de redundancia actual: Status, Turn, ángulo de redundancia (E1)
- Herramienta actual, TCP actual y base actual

La posición real también puede mostrarse mientras el robot se encuentra en movimiento.



Fig. 6-21: Posición real cartesiana

6.16.4 Visualizar momentos específicos del eje

Procedimiento

- En la vista del robot, seleccionar **Momentos axiales**.

Descripción

Se visualizan los valores de momentos actuales para los ejes A1 y A7. Adicionalmente se visualiza para cada eje el rango de medición del sensor (barra blanca).

Si se supera el momento absoluto permitido en una unión articulada, se coloreará en naranja la zona gris oscura de la barra del eje correspondiente. Solo se colorea la zona vulnerable, es decir, la parte negativa o la parte positiva.



La tasa de actualización de los valores mostrados está limitada. Por ello, en determinadas circunstancias no se muestran los valores máximos existentes brevemente.

La indicación contiene los siguientes datos:

- Pares de torsión absolutos actuales
- Pares de torsión externos actuales



Los pares de torsión externos solo se muestran correctamente si se ha indicado la herramienta correcta.

- Herramienta actual

Los momentos específicos del eje también pueden indicarse mientras el robot se encuentra en movimiento.

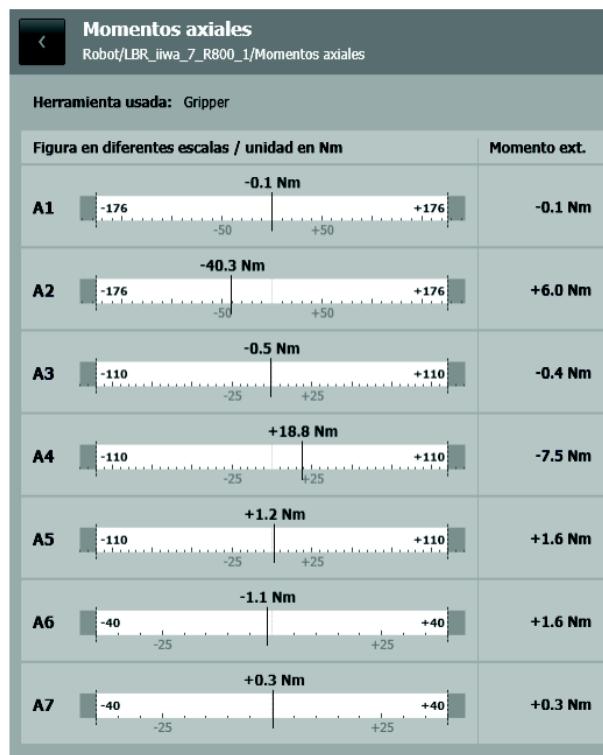


Fig. 6-22: Momentos específicos de los ejes

6.16.5 Visualizar el grupo E/S y modificar el valor de una salida

Condición previa

- El grupo E/S se ha creado para el proyecto Sunrise y las señales correspondientes se han conectado con WorkVisual.
- Para modificar una salida: Modo de servicio T1, T2 o KRF

i Las salidas se pueden modificar independientemente del estado del control de seguridad, por ejemplo, también con una PARADA DE EMERGENCIA pulsada.

Procedimiento

1. Seleccionar el grupo E/S deseado en la barra de navegación en **Grupos E/S**. Se muestran las entradas/salidas del grupo seleccionado.
2. Marcar la salida que debe modificarse.
3. En el caso de las salidas numéricas se muestra un campo de entrada: Introducir el valor deseado.
4. Presionar el pulsador de validación y mantenerlo presionado. Modificar el valor de la salida mediante el botón correspondiente.

Descripción

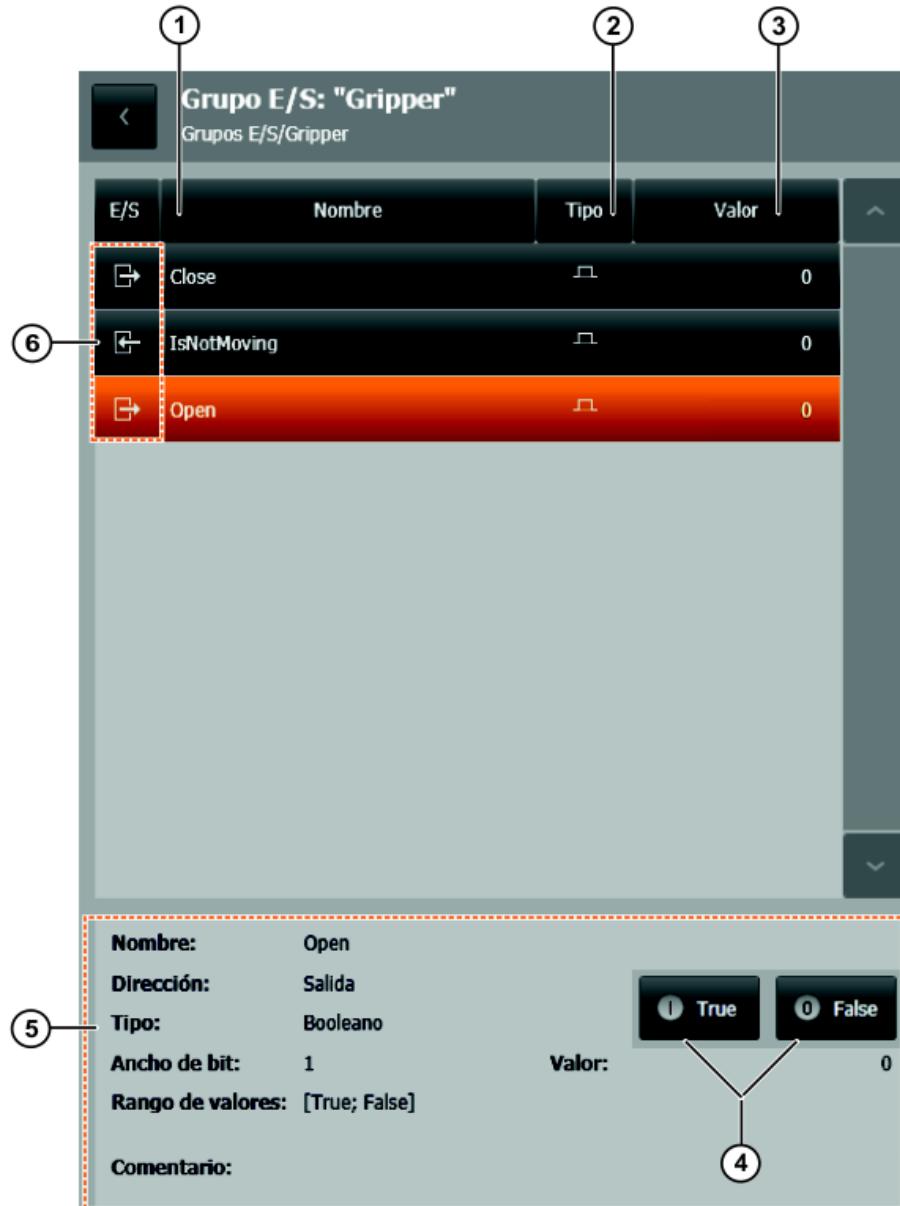


Fig. 6-23: Entradas/salidas de un grupo E/S

Pos.	Descripción
1	Nombre de la entrada/salida.
2	Tipo de la entrada/salida

Pos.	Descripción
3	Valor de la entrada/salida El valor se indica como número decimal.
4	Botones para modificar salidas Si una salida está marcada, se podrá modificar su valor. Condición previa: El interruptor de parada está pulsado. Los botones disponibles dependen del tipo de salida.
5	Propiedades de la señal Se muestran las propiedades y el valor actual de la entrada/salida marcada.
6	Dirección de la señal Los símbolos indican si en una señal se trata de una entrada o una salida.

Los siguientes botones están disponibles en función del tipo de la salida marcada:

Botón	Descripción
True	Botones para modificar salidas booleanas
False	Ajustar las salidas booleanas marcadas al valor True (1) o False (0).
Establecer	Botón para modificar salidas numéricas Ajusta la salida numérica marcada al valor introducido.

Dirección de la señal

Los siguientes símbolos señalizan la dirección de una señal:

Símbolo	Descripción
	Símbolo para una salida
	Símbolo para una entrada

Tipos de E/S

Los siguientes símbolos identifican el tipo de una entrada/salida:

Símbolo	Descripción
	Símbolo para una señal analógica
	Símbolo para una señal binaria
	Símbolo para una señal digital con signo
	Símbolo para una señal digital sin signo

6.16.6 Mostrar la dirección IP y la versión de software

Procedimiento

- En la vista de la estación, seleccionar el cuadro **Información**.

En el nodo **Estación** se visualizan la versión actualmente instalada del software y la dirección IP de la unidad de control del robot.

6.16.7 Visualizar el tipo de robot y el número de serie

Procedimiento

- En la vista de la estación seleccionar el cuadro **Información**.
En el nodo **Nombre del robot/placa de tipo** se muestra información sobre el robot utilizado actualmente.
 - **Número de serie:** Número de serie del robot conectado
 - **Robot conectado:** Tipo de robot del robot conectado
 - **Robot instalado:** Tipo de robot indicado en la configuración de la estación del proyecto Sunrise

7 Puesta en servicio y reanudación del servicio

7.1 Ajuste de posición

En el ajuste de posición, se asigna un ángulo de motor a una posición del eje del robot ajustada mecánicamente. El desplazamiento con una repetibilidad exacta de las posiciones programadas solo se puede realizar con un robot ajustado. Un robot desajustado solo se puede desplazar manualmente (de forma específica del eje) (modo de servicio T1 o KRF).

7.1.1 Ajustar ejes

Descripción El LBR iiwa posee en cada eje un sensor de ajuste basado en el efecto de Hall. La posición de ajuste del eje (posición cero) se encuentra en el centro de una secuencia de imanes definida. Se detecta automáticamente por el sensor de ajuste cuando es desplazado sobre la secuencia de imanes por un giro del eje.

Antes del propio ajuste, se ejecuta un desplazamiento de búsqueda automático para localizar una posición de preajuste definida.

Si el desplazamiento de búsqueda tiene éxito, el eje se desplaza hasta la posición de preajuste. A continuación, el eje se desplaza de forma que el sensor de ajuste sea desplazado sobre la secuencia de imanes. Para ello se guarda la posición del motor en el momento de la detección de la posición de ajuste del eje como posición cero del motor.



Solo mediante un procedimiento siempre igual se garantiza la repetibilidad y la capacidad de reproducción del ajuste. Las siguientes reglas se deben tener en cuenta para el ajuste:

- Al ajustar un eje, todos los ejes se deberán encontrar en posición vertical. Si esto no es posible, el ajuste siempre se deberá realizar en la misma posición del eje.
- Los ejes individuales se deben ajustar siempre en el mismo orden.
- Realizar el ajuste siempre sin carga. El ajuste con carga actualmente no es soportado.

Condición previa

- Modo de servicio T1 o KRF

Procedimiento

1. En la pantalla del robot seleccionar **Ajuste**. Se abre la pantalla **Ajuste**.
2. Presionar el pulsador de validación y mantenerlo presionado.
3. Pulsar el botón **Ajustando** del eje desajustado.

Primero se determina la posición de preajuste mediante un desplazamiento de búsqueda. A continuación se ejecuta el desplazamiento de ajuste. Despues del ajuste exitoso, el eje se desplaza hasta la posición de ajuste determinada (posición cero).



Si fallan el desplazamiento de búsqueda o el ajuste, se cancelará el proceso y el robot se detiene.

7.1.2 Desajustar los ejes de forma manual

Descripción La posición de ajuste guardada de un eje se puede borrar. Como consecuencia se desajusta el eje. Durante el desajuste no se ejecuta ningún movimiento.

Condición previa

- Modo de servicio T1

Procedimiento

1. En la pantalla del robot seleccionar **Ajuste**. Se abre la pantalla **Ajuste**.
2. Pulsar el botón **Desajustando** del eje ajustado. El eje se desajusta.

7.2 Medición

7.2.1 Medir la herramienta

Descripción

En la medición de la herramienta, el usuario asigna a una herramienta montada en la brida de acople del robot un sistema de coordenadas cartesianas (sistema de coordenadas de herramientas).

El sistema de coordenadas de herramientas tiene su origen en un punto definido por el usuario. Éste se denomina TCP (Tool Center Point). Por lo general, el TCP se coloca en el punto de trabajo de la herramienta. Una herramienta puede contar con varios TCP.

Ventajas de la medición de la herramienta:

- La herramienta puede desplazarse en línea recta siguiendo la dirección de trabajo.
- La herramienta puede girar alrededor del TCP sin que la posición del TCP varíe.
- Servicio con el programa: La velocidad programada se mantiene en el TCP a lo largo de toda la trayectoria.

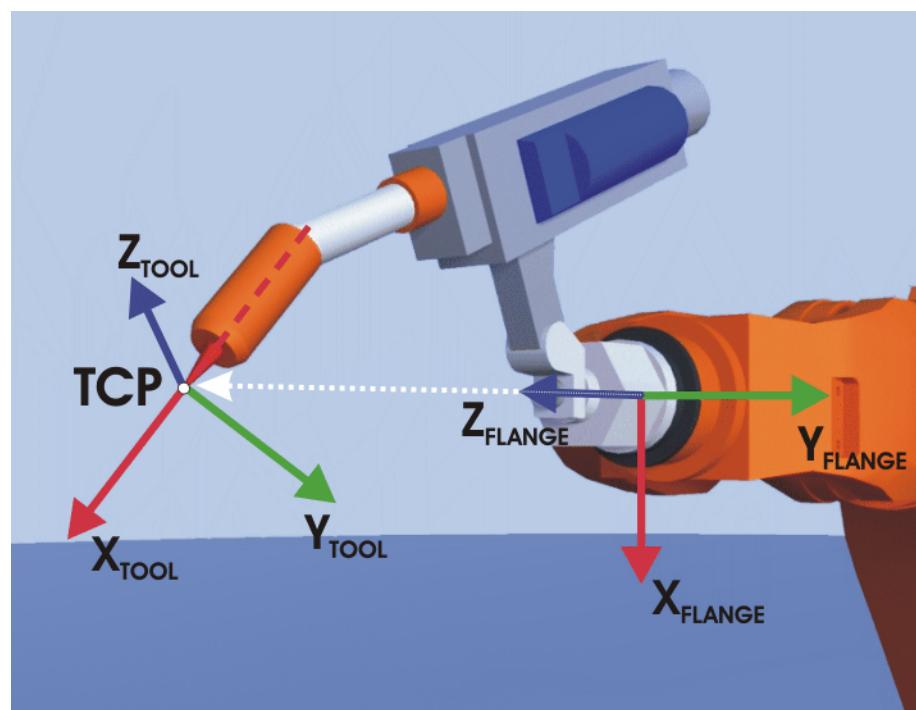


Fig. 7-1: Principio de la medición TCP

Vista general

La medición de la herramienta consta de 2 pasos:

Paso	Descripción
1	<p>Definir el origen del sistema de coordenadas de herramientas</p> <p>Se puede elegir entre los siguientes métodos:</p> <ul style="list-style-type: none"> ■ Punto XYZ-4 (>>> 7.2.1.1 "Medir el TCP: Método XYZ 4 puntos" Página 107)
2	<p>Definir la orientación del sistema de coordenadas de herramientas</p> <p>Se puede elegir entre los siguientes métodos:</p> <ul style="list-style-type: none"> ■ Punto ABC-2 (>>> 7.2.1.2 "Definir la orientación: Método ABC 2 puntos" Página 109) ■ ABC World (>>> 7.2.1.3 "Definir la orientación: Método ABC mundial" Página 111)

7.2.1.1 Medir el TCP: Método XYZ 4 puntos

Descripción	<p>Con el TCP de la herramienta que se desea medir debe desplazarse el robot a un punto de referencia desde 4 direcciones diferentes. El punto de referencia puede ser cualquiera. La unidad de control del robot calcula el TCP a partir de las distintas posiciones de la brida.</p> <p>Las 4 posiciones de la brida con las cuales el robot se desplaza al punto de referencia deben contar con una distancia mínima entre sí. Si los puntos están demasiado cerca uno del otro, los datos de posición no se pueden guardar. Se emite un mensaje de error correspondiente.</p> <p>La calidad de la medición se puede evaluar a través del error de cálculo translatorial que se determina durante la medición. Si este error supera un valor límite determinado, se recomienda volver a medir el TCP.</p> <p>Es posible modificar la distancia mínima y el error de cálculo máximo en Sunrise.Workbench. (>>> 10.1.1 "Configurar parámetros para medición" Página 159)</p>
--------------------	--

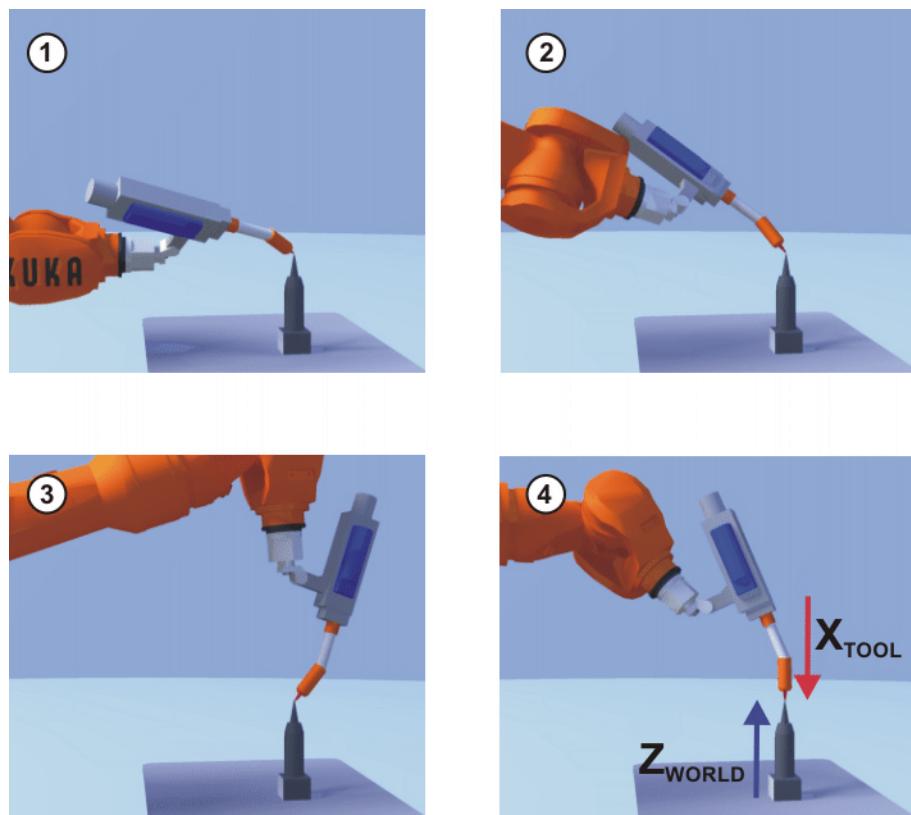


Fig. 7-2: Método XYZ 4-Puntos

- | | |
|-------------------------|---|
| Condición previa | <ul style="list-style-type: none"> ■ La herramienta que se desea medir se encuentra montada sobre la brida de acople. ■ La herramienta que se va a medir y el Frame que se emplea como TCP se han creado en las plantillas de objetos del proyecto y se han transferido a la unidad de control del robot por sincronización. ■ Modo de servicio T1 |
| Procedimiento | <ol style="list-style-type: none"> 1. En la vista del robot seleccionar Medición > Medición de herramienta. Se abre la vista Medición de herramienta. 2. Seleccionar la herramienta que se desee medir y el TCP correspondiente. 3. Seleccionar el método Medición de TCP(Punto XYZ 4). Los puntos de medición del método se indican como botones: <ul style="list-style-type: none"> ■ Punto de medición 1 ... Punto de medición 4
Para poder integrar un punto de medición, debe estar marcado (el botón aparece en naranja). 4. Con el TCP desplazarse a un punto de referencia cualquiera. Pulsar Registrar punto de medición. Los datos de posición se aceptan e indican para el punto de medición marcado. 5. Con el TCP desplazarse al punto de referencia desde otra dirección. Pulsar Registrar punto de medición. Los datos de posición se aceptan e indican para el punto de medición marcado. 6. Repetir 2 veces el paso 5. 7. Pulsar Determinar datos de herramienta. Los datos de medición y el fallo de cálculo detectado se indican en el diálogo Aceptar datos de herramienta. 8. Si el fallo de cálculo supera el valor máximo admisible, se indica una advertencia. Pulsar Cancelar y volver a medir TCP. 9. Si el fallo de cálculo está por debajo del límite configurado, pulsar Aceptar para guardar los datos de medición. |

10. Cerrar la vista de medición o establecer la orientación del sistema de coordenadas de herramientas con el método ABC de 2 puntos o el método ABC mundial.
(>>> 7.2.1.2 "Definir la orientación: Método ABC 2 puntos" Página 109)
(>>> 7.2.1.3 "Definir la orientación: Método ABC mundial" Página 111)
11. Sincronizar el proyecto para aceptar los datos de medición, incluido el archivo de cálculo, en Sunrise.Workbench.

7.2.1.2 Definir la orientación: Método ABC 2 puntos

Descripción A la unidad de control del robot se le comunican los ejes del sistema de coordenadas de herramienta desplazando el robot a un punto del eje X y un punto en el nivel XY.

Los puntos deben estar separados por una distancia mínima. Si los puntos están demasiado cerca uno del otro, los datos de posición no se pueden guardar. Se emite un mensaje de error correspondiente.

Es posible modificar la distancia mínima en Sunrise.Workbench.
(>>> 10.1.1 "Configurar parámetros para medición" Página 159)

Este método se emplea con herramientas que tienen esquinas y bordes por los que se puede guiar el usuario. Este método también se utiliza cuando las direcciones de los ejes deben establecerse con la mayor exactitud posible.

Este método no está disponible para herramientas destinadas a la seguridad.

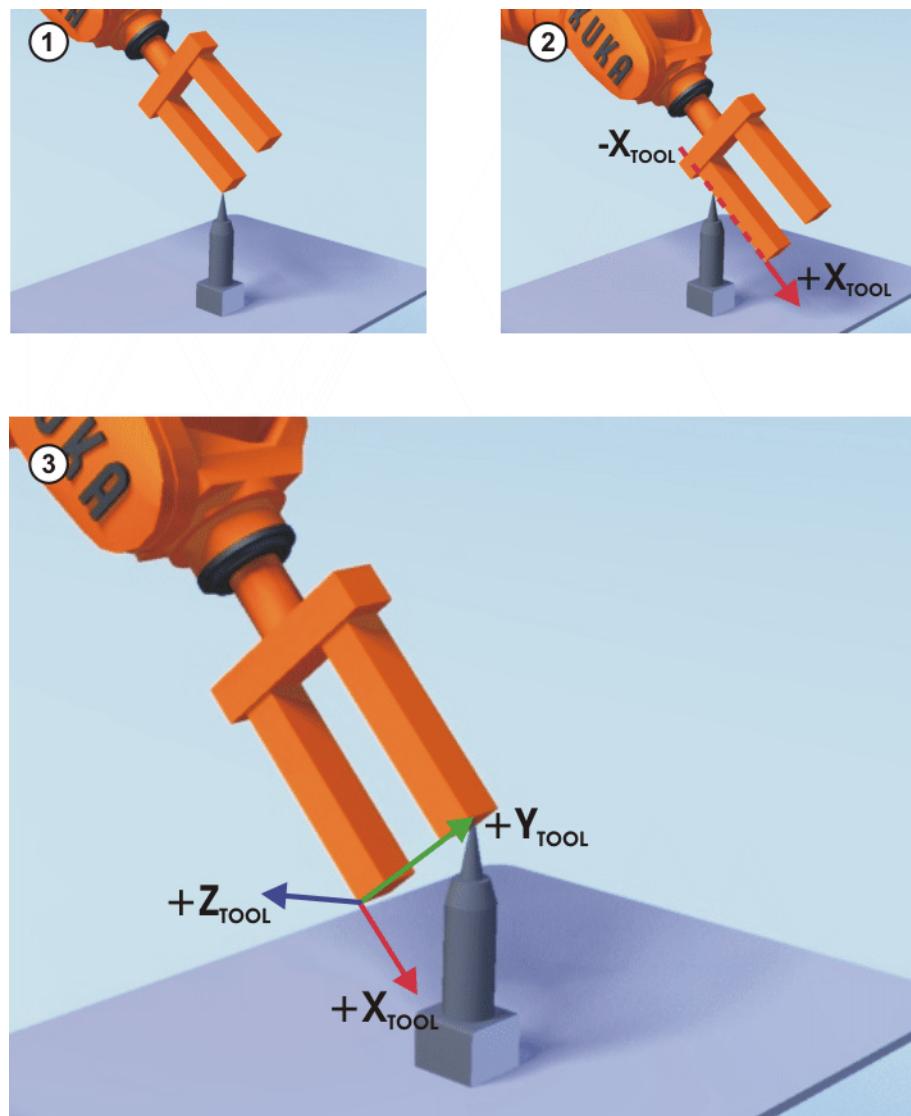


Fig. 7-3: Método ABC 2 puntos

Condición previa

- La herramienta que se desea medir se encuentra montada sobre la brida de acople.
- El TCP de la herramienta ha sido medido.
- Modo de servicio T1

Procedimiento

1. Solo si la vista de medición se ha cerrado después de la unidad de calibración para TCP:
En la vista del robot seleccionar **Medición > Medición de herramienta**. Se abre la vista **Medición de herramienta**.
2. Solo si la vista de medición se ha cerrado después de la unidad de calibración para TCP:
Seleccionar la herramienta montada y la TCP correspondiente de la herramienta.
3. Seleccionar el método **Definir la orientación(Punto ABC 2)**. Los puntos de medición del método se indican como botones:
 - **TCP**
 - **Eje X negativo**
 - **Valor Y positivo en el plano XY**
 Para poder integrar un punto de medición, debe estar marcado (el botón aparece en naranja).

4. Con el TCP desplazarse a un punto de referencia cualquiera. Pulsar **Registrar punto de medición**. Los datos de posición se aceptan e indican para el punto de medición marcado.
5. Desplazar la herramienta de tal modo que el punto de referencia sobre el eje X se encuentre sobre un punto de valor X negativo (es decir, en contra de la dirección de trabajo). Pulsar **Registrar punto de medición**. Los datos de posición se aceptan e indican para el punto de medición marcado.
6. Desplazar la herramienta de modo tal que el punto de referencia sobre el plano XY se encuentre sobre un valor Y positivo. Pulsar **Registrar punto de medición**. Los datos de posición se aceptan e indican para el punto de medición marcado.
7. Pulsar **Determinar datos de herramienta**. Los datos de medición se indican en el diálogo **Aceptar datos de herramienta**.
8. Pulsar **Aceptar** para guardar los datos de medición.
9. Sincronizar el proyecto para aceptar los datos de medición en Sunrise.Workbench.

7.2.1.3 Definir la orientación: Método ABC mundial

Descripción	<p>El usuario alinea los ejes del sistema de coordenadas de herramientas de forma paralela a los ejes del sistema de coordenadas mundial. De este modo, se informa a la unidad de control del robot de la orientación del sistema de coordenadas de herramienta.</p> <p>Este método tiene 2 variantes:</p> <ul style="list-style-type: none"> ■ 5D: El usuario informa a la unidad de control del robot de la dirección de avance de la herramienta. Por defecto, la dirección de avance es el eje X. La orientación de los demás ejes la determina el sistema y el usuario no la puede cambiar. <p>El sistema siempre establece igual la orientación de los demás ejes. Si en un futuro se debe volver a medir la herramienta, p. ej., tras una colisión, bastará con establecer de nuevo la dirección de avance. No se debe tener en cuenta el giro en la dirección de avance.</p> <ul style="list-style-type: none"> ■ 6D: El usuario informa a la unidad de control del robot de la dirección de los 3 ejes. <p>Este método se emplea con herramientas que no tienen bordes por los que se pueda guiar el usuario, p. ej. herramientas redondas como toberas de encolado o de soldadura.</p>
Condición previa	<ul style="list-style-type: none"> ■ La herramienta que se desea medir se encuentra montada sobre la brida de acople. ■ El TCP de la herramienta ha sido medido. ■ Modo de servicio T1
Procedimiento	<ol style="list-style-type: none"> 1. Solo si la vista de medición se ha cerrado después de la unidad de calibración para TCP: En la vista del robot seleccionar Medición > Medición de herramienta. Se abre la vista Medición de herramienta. 2. Solo si la vista de medición se ha cerrado después de la unidad de calibración para TCP: Seleccionar la herramienta montada y la TCP correspondiente de la herramienta. 3. Seleccionar el método Definir la orientación(ABC Mundo). 4. Seleccionar la opción ABC Mundo 5D o ABC Mundo 6D. 5. Si se selecciona ABC Mundo 5D:

Alinear $+X_{TOOL}$ de forma paralela a $-Z_{WORLD}$ ($+X_{TOOL}$ = dirección de avance)

Si se selecciona **ABC Mundo 6D**:

Alinear los ejes del sistema de coordenadas de herramientas del siguiente modo:

- $+X_{TOOL}$ de forma paralela a $-Z_{WORLD}$ ($+X_{TOOL}$ = dirección de avance)
- $+Y_{TOOL}$ de forma paralela a $+Y_{WORLD}$
- $+Z_{TOOL}$ de forma paralela a $+X_{WORLD}$

6. Pulsar **Determinar datos de herramienta**. Los datos de medición se indican en el diálogo **Aceptar datos de herramienta**.
7. Pulsar **Aceptar** para guardar los datos de medición.
8. Sincronizar el proyecto para aceptar los datos de medición en Sunrise.Workbench.

7.2.2 Medición base: Método de 3 puntos

Descripción

Durante la medición de base, el usuario asigna un sistema de coordenadas cartesianas (sistema de coordenadas de base) a un Frame que está marcado como base. El sistema de coordenadas de base tiene su origen en un punto definido por el usuario.

Ventajas de la medición BASE:

- El TCP puede moverse de forma manual a lo largo de los cantos de la superficie de trabajo o de la pieza.
- Puntos pueden ser programados por aprendizaje en relación a la base. Si es necesario desplazar la base, por ej. porque la superficie de trabajo se ha desplazado, se desplazan también los puntos y no hace falta volver a programarlos por aprendizaje.

En el método de 3 puntos el robot se desplaza al origen y a 2 puntos más de una base. Estos 3 puntos definen la base.

Los puntos deben cumplir con una distancia mínima determinada hasta el origen y un ángulo mínimo entre las rectas (origen – eje X y origen – nivel XY). Si los puntos están demasiado cerca o si el ángulo entre las rectas es demasiado pequeño, no es posible guardar los datos de posición. Se emite un mensaje de error correspondiente.

Es posible modificar la distancia y el ángulo mínimos en Sunrise.Workbench.
(>>> 10.1.1 "Configurar parámetros para medición" Página 159)

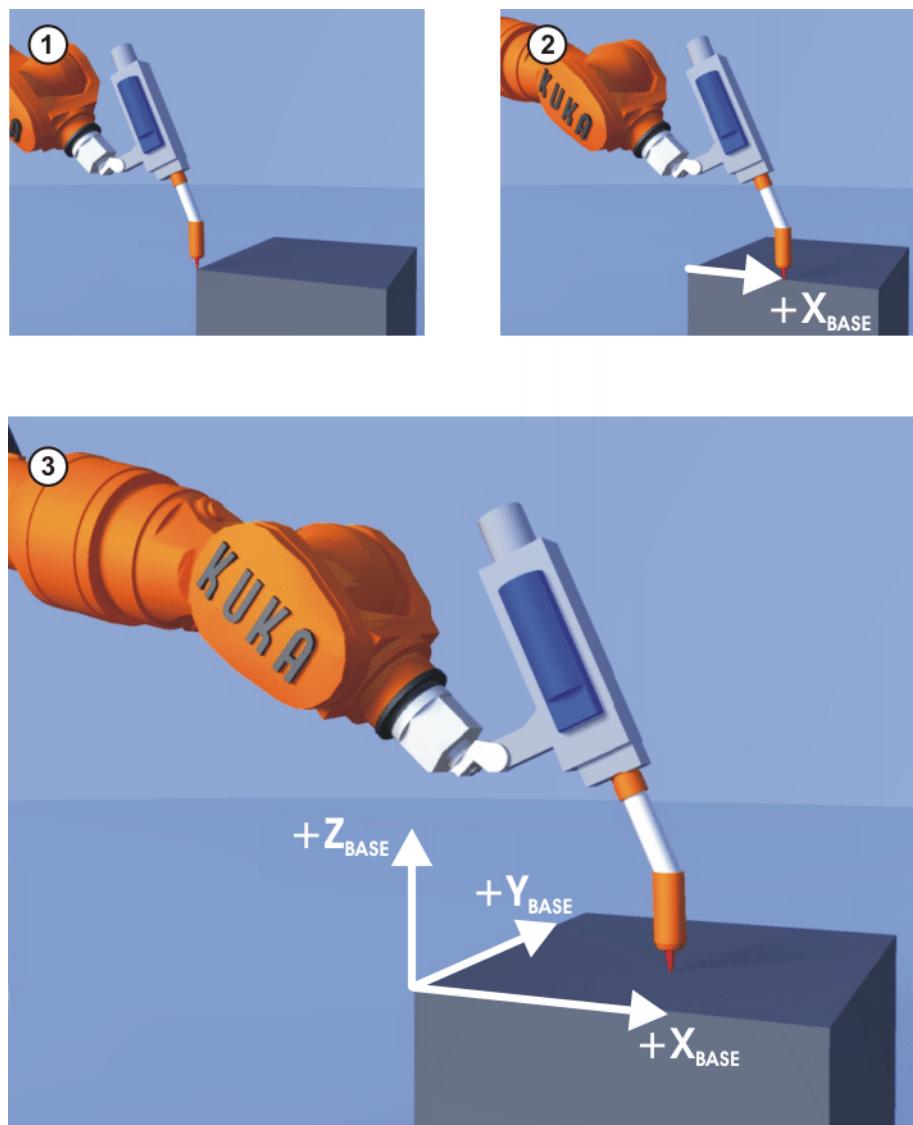


Fig. 7-4: Método de los 3 puntos

Condición previa

- Una herramienta ya medida se encuentra montada sobre la brida de acople.
- El Frame que se desea medir ha sido marcado como base en los datos de aplicación del proyecto y se ha transferido a la unidad de control del robot por sincronización.
- Modo de servicio T1

Procedimiento

1. En la vista del robot seleccionar **Medición > Medición de base**. Se abre la vista **Medición de base**.
2. Seleccionar la base que se debe medir.
3. Seleccionar las herramientas montadas y el TCP de las herramientas con el que se realizan los desplazamientos a los puntos de medición de la base.

Los puntos de medición del método de los 3 puntos se indican como botones:

- **Origen**
- **Eje X positivo**
- **Valor Y positivo en el plano XY**

Para poder integrar un punto de medición, debe estar marcado (el botón aparece en naranja).

4. Con el TCP desplazar el robot al origen de la base. Pulsar **Registrar punto de medición**. Los datos de posición se aceptan e indican para el punto de medición marcado.
5. Con el TCP desplazar el robot a un punto del eje X positivo de la base. Pulsar **Registrar punto de medición**. Los datos de posición se aceptan e indican para el punto de medición marcado.
6. Con el TCP desplazar el robot a un punto del plano XY con valor Y positivo Pulsar **Registrar punto de medición**. Los datos de posición se aceptan e indican para el punto de medición marcado.
7. Pulsar **Determinar los datos de base**. Los datos de medición se indican en el diálogo **Aceptar datos de base**.
8. Pulsar **Aceptar** para guardar los datos de medición.
9. Sincronizar el proyecto para aceptar los datos de medición en Sunrise.Workbench.

7.3 Determinar datos de carga de la herramienta

Descripción

En la determinación de datos de carga, el robot ejecuta varios desplazamientos de medición con diferentes orientaciones de los ejes de la muñeca A5, A6 y A7. Los datos de carga se determinan a partir de los datos que se registran durante los desplazamientos de medición.

De forma actual se pueden determinar la masa y la posición del centro de gravedad de la masa de la herramienta montada en la brida del robot. También es posible especificar la masa y determinar la posición del centro de gravedad de la masa basándose en la masa ya conocida.

Al principio de la determinación de datos de carga se desplaza el eje A7 hasta la posición cero y el eje A5 se ajusta de forma que el eje A6 esté alineado en sentido vertical con respecto al peso. Durante los desplazamientos de medición, el eje A6 se puede mover desplazar entre -95° y +95° y el eje A7 desde 0° hasta -90°.

El resto de ejes del robot (A1 hasta A4) no se desplazan durante la determinación de datos de carga. Durante la medición permanecen en la posición de salida.

La calidad de la determinación de datos de carga puede recibir la influencia de las siguientes condiciones límite:

■ Masa de la herramienta

La determinación de datos de carga es más fiable con mayor masa de la herramienta, ya que las incertidumbres de medición tienen un efecto mayor si la masa es menor.



La determinación de datos de carga aún no se puede utilizar actualmente de forma fiable para masas por debajo de un kilogramo.

■ Cargas adicionales

La determinación de datos de carga solo está indicada para cargas están unidas de forma fija con la brida del robot. Las cargas adicionales, es decir, las cargas que están montadas en la estructura del robot, p. ej. paquetes de tubos y cables, no son soportadas por la determinación de datos de carga.

■ Posición inicial desde la que se inicia la determinación de datos de carga

La posición inicial adecuada se debe determinar previamente y debe cumplir los siguientes criterios:

- Los ejes A1 hasta A5, si es posible, están a una amplia distancia de las posiciones singulares.

El criterio es relevante si en la determinación de datos de carga se determina la masa. Si la determinación de datos de carga solo es posible en poses en las que los ejes A1 a A5 se encuentran cerca de posiciones singulares, se puede preestablecer la masa. Si solo se determina el centro de gravedad de la masa basándose en la masa preestablecida, el criterio de la posición de eje es irrelevante.

- La idoneidad de la posición inicial para la determinación de datos de carga para el robot, con el que debe realizarse un cálculo automático, se debe comprobar antes del montaje de la carga a determinar en el robot.

Una posición del robot es adecuada como posición inicial para la determinación de datos de carga si en dicha posición solo actúan momentos externos muy reducidos (lo ideal < 0,5 Nm) sobre el robot sin carga. Esto se puede comprobar mediante la indicación de los momentos de los ejes externos.

(>>> 6.16.4 "Visualizar momentos específicos del eje" Página 100)

Si en la determinación de datos de carga se determina la masa, son relevantes todos los momentos de los ejes y, si es posible, se deben comprobar previamente para el robot sin carga. Si solo se determina el centro de gravedad de la masa basándose en una masa prefijada, solo es relevante el momento del eje externo del eje A6.

- Momentos erróneos durante los desplazamientos de medición
 - Durante los desplazamientos de medición no se pueden aplicar momentos erróneos, p. ej. por presionar o tirar del robot.
 - Las partes móviles, p. ej. los paquetes de tubos y cables, generan pares de torsión erróneos que desplazan el centro de gravedad de la masa durante el desplazamiento de medición.



La aplicación de momentos erróneos en la determinación de datos de carga da como resultado datos de carga incorrectos.

(>>> 9.3.6 "Datos de carga" Página 149)

En la determinación de datos de carga para herramientas destinadas a la seguridad debe tenerse en cuenta que los datos de carga modificados no se aceptan automáticamente en la configuración de seguridad que se encuentra en la unidad de control del robot. (>>> 9.3.7 "Herramienta destinada a la seguridad" Página 150)

A través de la sincronización del proyecto, primero se deben actualizar en Sunrise.Workbench los datos de carga determinados para una herramienta destinada a la seguridad. De este modo, se modifica la configuración de seguridad del proyecto en Sunrise.Workbench, lo cual se debe transmitir de vuelta a la unidad de control del robot mediante una nueva sincronización del proyecto.



Si se activa una configuración de seguridad modificada en la unidad de control del robot, el técnico de puesta en servicio deberá ejecutar una recepción de seguridad. (>>> 13.11 "Resumen de la recepción de seguridad" Página 230)

Para evitar una acción de verificación innecesaria, se recomienda marcar una herramienta como destinada a la seguridad si los datos de carga se han introducido o determinado correctamente y se han transmitido a Sunrise.Workbench.

Preparación

- Determinar la posición inicial desde la que se inicia la determinación de datos de carga.

Condición previa

- El útil se encuentra montado sobre la brida de acople.
- La herramienta se ha creado en las plantillas de objetos del proyecto y se ha transmitido a la unidad de control del robot por sincronización.
- El robot se encuentra en la posición inicial deseada.
- En la zona de los ejes de la muñeca existe un campo de trabajo suficientemente grande.
- Modo de servicio T1, T2 o AUT

AVISO

Si las partes de la herramienta montada alcanzan la zona detrás del nivel de la brida (en dirección Z negativa con respecto al sistema de coordenadas de la brida), existe el peligro de que la herramienta colisione con el manipulador durante los desplazamientos de medición.

Si el usuario no conoce el movimiento de los ejes durante la determinación de los datos de carga o si p. ej. no se puede descartar una colisión entre la herramienta y el manipulador durante la primera determinación de datos de carga de una herramienta, se recomienda realizar la determinación de datos de carga en el modo de servicio T1. La calidad de los valores de medición no se verá afectada como consecuencia.

Procedimiento

1. En la vista del robot, seleccionar **Datos de carga**. Se abre la vista **Datos de carga**.
2. Seleccionar la herramienta montada en la lista de selección.
3. Si está ajustado el modo de servicio T1 o T2, presionar el pulsador de validación y mantenerlo pulsado hasta que haya finalizado la determinación de datos de carga.
4. Pulsar **Determinar los datos de carga**.
5. Si la herramienta ya posee una masa, se preguntará al operario si la masa se debe determinar de nuevo.
 - Seleccionar **Mantener masa** si debe mantenerse la masa guardada actualmente.
 - Seleccionar **Determinar de nuevo la masa** si debe determinarse nuevamente la masa.



Si la masa guardada actualmente en la determinación de datos de carga se mantiene, debe prestarse atención a que el valor indicado de la masa es correcto. De lo contrario no se podrá determinar de forma precisa el centro de gravedad de la masa.

6. El robot inicia los desplazamientos de medición y se determinan los datos de carga. Se muestra una barra de progreso.



Si se retira el movimiento habilitado durante la determinación de datos de carga, p. ej. por una PARADA DE EMERGENCIA o por soltar el pulsador de validación (T1, T2), se cancelará la determinación de datos de carga y se deberá iniciar de nuevo.

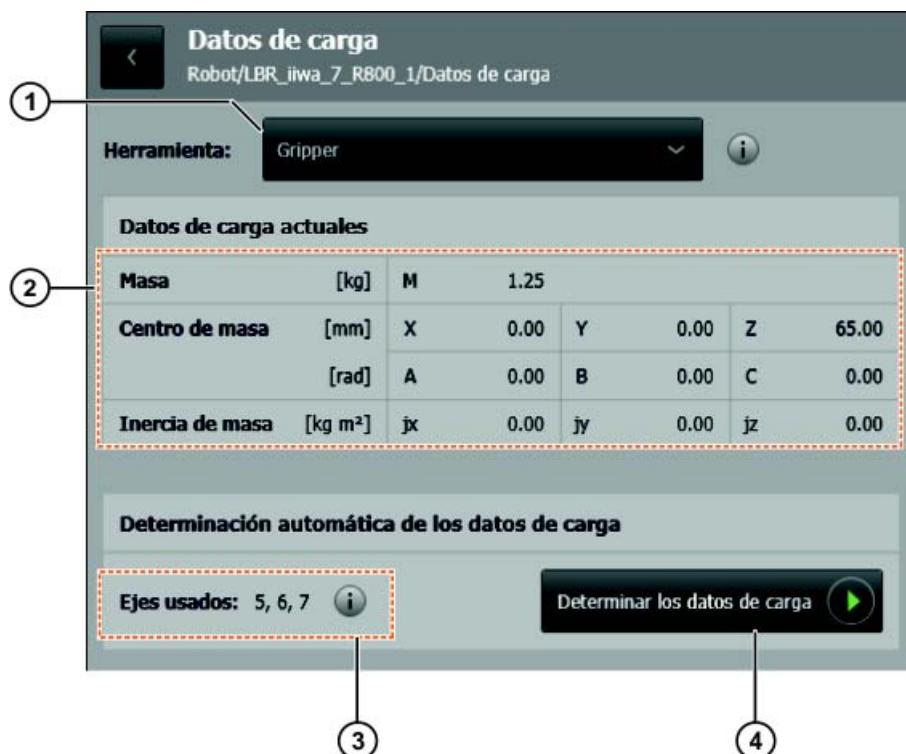
Cuando la determinación de datos de carga ha finalizado, se mostrarán los datos de carga determinados en el diálogo **Aceptar los datos de carga**.

Pulsar **Aceptar** para guardar los datos de carga determinados.

7. Sincronizar el proyecto para que los datos de carga se actualicen en Sunrise.Workbench.

Si los datos de carga se han determinado para una herramienta destinada a la seguridad, cambiará la configuración de seguridad con la sincronización del proyecto.

8. En caso necesario, sincronizar el proyecto para transmitir la configuración de seguridad modificada a la unidad de control del robot.

Resumen**Fig. 7-5: Determinar los datos de carga**

Pos.	Descripción
1	Lista de selección de herramientas Aquí se pueden seleccionar las herramientas creadas en las plantillas de objetos.
2	Indicación de datos de carga Muestra los datos de carga actuales de la herramienta seleccionada.
3	Indicación de ejes utilizados Muestra los ejes que se desplazan para la determinación de datos de carga.
4	Botón Determinar los datos de carga Inicia la determinación de datos de carga. El botón solo está activado si está seleccionada una herramienta y se ha concedido el movimiento habilitado.

8 Prueba de frenos

8.1 Resumen de la prueba de frenos

Descripción Cada uno de los ejes del robot tiene un freno de retención integrado en el ramal de accionamiento. Los frenos tienen 2 funciones:

- Detener el robot si la regulación está desconectada o en estado libre de tensión.
- Llevar el robot al estado seguro "Parada" en caso de un error.

Con la prueba de frenos se comprueba para cada freno si el momento de frenado generado por el mismo es suficiente, es decir, si supera un momento de referencia determinado. Este momento de referencia se puede especificar por el programador o se consulta en los datos del motor.

AVISO

La prueba de frenos debe de realizarse durante la puesta en servicio y la reanudación del servicio del robot industrial para cada eje.

Durante el servicio, la prueba de frenos se debe realizar diariamente. Este requisito existe independientemente del tiempo de servicio y del tipo de aplicación.

Ejecución La condición previa para la realización de la prueba de frenos es que el robot se encuentre a temperatura de servicio.

La prueba de frenos se ejecuta manualmente mediante una aplicación. KUKA Sunrise.Workbench ofrece una aplicación preparada de la prueba de frenos para el LBR iiwa.

Si se utiliza la aplicación preparada de la prueba de frenos, el robot se desplaza antes de la propia prueba de frenos y se determina el máximo par de torsión absoluto que se produce para cada eje. El momento determinado se transmite en la aplicación de la prueba de frenos como momento de frenado de referencia a la prueba de frenos.

La determinación de los máximos pares de torsión absolutos se denominará a continuación como determinación del valor del momento.



Se recomienda eliminar la determinación del valor del momento de la aplicación y comprobar el mínimo momento de frenado de los frenos.

Si se realiza la comprobación de acuerdo con un par de torsión especificado por el programador, previamente se deberá evaluar mediante un análisis de riesgos si se producen peligros cuando los frenos presentan un momento más bajo que el mínimo momento de frenado de los frenos.

Proceso En la prueba de un freno se ejecutan por defecto los siguientes pasos:

1. El eje se desplaza con velocidad constante con un ángulo del eje reducido de un máximo de 5° (en el lado de salida). Se calculan la fricción y la gravedad.
2. Cuando el eje ha alcanzado de nuevo su posición de salida y el accionamiento del eje está detenido, se cierra el freno.
3. Como momento de frenado que debe comprobarse se utiliza uno de los siguientes valores: El momento de frenado de referencia calculado, el mínimo momento de frenado de los frenos o el momento de retención de motor.

El momento de frenado que debe comprobarse se determina de forma interna en el sistema según las siguientes reglas:

- a. Si el momento de frenado de referencia es mayor que el mínimo del momento de frenado de los frenos y del momento de retención de motor.

ción de motor, se utilizará el mínimo del mínimo momento de frenado de los frenos y del momento de retención de motor como momento de frenado que debe comprobarse.

- b. Si el momento de frenado de referencia es menor que el 20 % del mínimo del mínimo momento de frenado de los frenos y del momento de retención de motor, se utilizará el 20 % del mínimo del mínimo momento de frenado de los frenos y del momento de retención de motor como momento de frenado que debe comprobarse.
- c. En todos los demás casos se utiliza el momento de frenado de referencia.

Al principio de la prueba, la prueba de frenos ajusta el momento nominal del accionamiento con el freno cerrado al 80 % del momento de frenado que debe comprobarse.



El momento de frenado de los frenos mínimo y máximo está guardado en los datos del motor. El momento de retención de motor se deriva de los datos del motor.

4. El momento de accionamiento se aumenta sucesivamente hasta que se determine una modificación de la posición o se alcance el máximo momento de frenado de los frenos (el valor se calcula a partir de los datos del motor). Al alcanzar el máximo momento de frenado de los frenos finaliza la prueba de frenos.
5. Se mide el momento generado contra el freno en caso determinación de una modificación de la posición. Este es el momento de frenado medido.
6. El momento de frenado medido se evalúa referido al momento de frenado que debe comprobarse.

La prueba de frenos se habrá realizado correctamente si el momento de frenado medido se encuentra dentro del rango siguiente:

- $\geq 105\%$ del momento de frenado que debe comprobarse ... \leq máximo momento de frenado de los frenos

Si el momento de frenado medido se encuentra por debajo del momento de frenado que debe comprobarse, habrá fallado la prueba de frenos, es decir, el freno se ha detectado como defectuoso.

El resultado de la prueba se muestra en la smartHMI.

(>>> 8.4.2 "Consultar el resultado de la prueba de frenos (indicación)"
Página 136)

7. Cuando ha finalizado la prueba de frenos y el robot está detenido, se abre el freno brevemente y se cierra de nuevo. De este modo se eliminan las posibles tensiones restantes en el freno y se evitan los movimientos indeseados del robot.



Si la prueba de frenos ha fallado para un freno, es decir, un freno ha sido detectado como defectuoso, se podrá repetir a modo de control.



Si la aplicación se pausa durante la prueba de frenos o se activa una parada de seguridad, p. ej. por una PARADA DE EMERGENCIA, se cancelará la prueba de frenos. Tras la continuación de la aplicación se repetirá la prueba de frenos para el eje.



La prueba de frenos es independiente con respecto a las cargas montadas en el robot, ya que la gravitación y la fricción se tienen en cuenta durante la ejecución de la prueba.

Resumen

Aquí se describen los pasos para ejecutar la prueba de frenos con la plantilla puesta a disposición en Sunrise.Workbench.

La aplicación de la prueba de frenos se puede adaptar y ampliar. Para ello, se deben tener en cuenta los comentarios incluidos en la plantilla.

AVISO

Si un freno está defectuoso, el eje correspondiente puede deslizarse durante la prueba de frenos y el robot puede desplomarse. La prueba de frenos se debe ejecutar en una posición en la que, dentro de lo posible, no pueden producirse daños por desplome. La posición de salida para la prueba de frenos se debe seleccionar de forma correspondiente.

AVISO

Si ha fallado la prueba de frenos para un eje, es decir, un freno se ha detectado como defectuoso, se deberá asegurar mediante la aplicación que el robot será desplazado automáticamente hasta una posición segura. Una posición segura es una posición en la que el robot está apoyado de forma que no pueda desplomarse o que no pueda ocasionar daños en caso de desplomarse.

Paso	Descripción
1	<p>Crear la aplicación de la prueba de frenos a partir de una plantilla.</p> <p>(>>> 8.2 "Crear la aplicación de la prueba de frenos a partir de la plantilla" Página 122)</p>
2	<p>Eliminar o adaptar la determinación de los máximos momentos absolutos que se generan de forma específica para aplicaciones en la aplicación de la prueba de frenos.</p> <p>Al principio de la aplicación de la prueba de frenos se realiza por defecto el desplazamiento hasta 2 posiciones de los ejes predefinidas, para determinar el máximo par de torsión absoluto generado para cada eje y transmitirlo como momento de frenado de referencia a la prueba de frenos.</p> <p>Se recomienda comprobar los frenos en cuanto al mínimo momento de frenado de los frenos, que está almacenado en los datos del motor. Para ello se debe adaptar la aplicación de la prueba de frenos preparada.</p> <p>(>>> 8.2.1 "Adaptar la aplicación de la prueba de frenos para la prueba de acuerdo con un mínimo momento de frenado de los frenos" Página 124)</p> <p>Si para la prueba de frenos son necesarios los máximos momentos absolutos que se generan durante la ejecución de una aplicación del robot específica de usuario, se podrá insertar la aplicación del robot específica de usuario en la aplicación de la prueba de frenos. Debido a que, en este caso, los frenos no se comprueban en cuanto al mínimo momento de frenado de los frenos, se debe realizar previamente un análisis de riesgos.</p> <p>(>>> 8.2.2 "Modificar la secuencia de movimientos para la determinación del valor del momento" Página 125)</p>
3	<p>Modificar la posición de salida para la prueba de frenos.</p> <p>La posición de salida es por defecto la posición vertical. En caso necesario, se puede desplazar otra posición de salida.</p> <p>(>>> 8.2.3 "Modificar la posición de salida para la prueba de frenos" Página 125)</p>

Paso	Descripción
4	<p>En caso necesario, realizar otras adaptaciones específicas de usuario en la aplicación de la prueba de frenos.</p> <p>Ejemplos:</p> <ul style="list-style-type: none"> ■ Ajustar la salida en una prueba de frenos fallido. ■ Guardar los resultados de la prueba en un archivo. <p>(>>> 8.3 "Interfaz de programación para la prueba de frenos" Página 126)</p>
5	Sincronizar el proyecto para transmitir la aplicación de la prueba de frenos a la unidad de control del robot.
6	Ejecutar la aplicación de la prueba de frenos. (>>> 8.4 "Ejecutar la prueba de frenos" Página 135)

8.2 Crear la aplicación de la prueba de frenos a partir de la plantilla

Procedimiento

1. Marcar el proyecto Sunrise en el **Explorador de paquetes**.
 2. Seleccionar la secuencia del menú **Archivo > Nuevo > Otros...**.
 3. En la carpeta **Sunrise** marcar la opción **Aplicación para la prueba de frenos del LBR iiwa** hacer clic en **Finalizar**.
- La aplicación **BrakeTestApplication.java** se crea en la carpeta de origen del proyecto y se abre en la zona de los editores de Sunrise.Workbench.

Descripción

En el método `run()` de la aplicación **BrakeTestApplication.java** (aquí limitada a las líneas de instrucciones relevantes) está implementada la ejecución de la prueba de frenos para todos los ejes del LBR iiwa.

También está implementada una evaluación de datos opcional, previa a la propia prueba de frenos. Se realiza el desplazamiento hasta 2 posiciones de los ejes predefinidas, para determinar el máximo par de torsión absoluto para cada eje.

```

1 public void run() {
2     ...
3     lbr_iiwa.move(ptpHome());
4     ...
5     TorqueEvaluator evaluator = new TorqueEvaluator(lbr_iiwa);
6     ...
7     evaluator.setTorqueMeasured(false);
8
9     evaluator.startEvaluation();
10    ...
11    lbr_iiwa.move(new PTP(new JointPosition(
12        0.5, 0.8, 0.2, 1.0, -0.5, -0.5, -1.5)).
13        setJointVelocityRel(relVelocity));
14    lbr_iiwa.move(new PTP(new JointPosition(
15        -0.5, -0.8, -0.2, -1.0, 0.5, 0.5, 1.5)).
16        setJointVelocityRel(relVelocity));
17    ...
18    TorqueStatistic maxTorqueData = evaluator.stopEvaluation();
19
20    boolean allAxesOk = true;
21
22    for (int axis : axes) {
23        try {
24            BrakeTest brakeTest = new BrakeTest(axis,
25                maxTorqueData.getMaxAbsTorqueValues()[axis]);

```

```

26         IMotionContainer motionContainer = lbr_iiwa.move(brakeTes
t);
27         BrakeTestResult brakeTestResult =
28             BrakeTest.evaluateResult(motionContainer);
29         switch(brakeTestResult.getState().getLogLevel())
30         {
31             case Info:
32                 getLogger().info(brakeTestResult.toString());
33                 break;
34             case Warning:
35                 getLogger().warn(brakeTestResult.toString());
36                 break;
37             case Error:
38                 getLogger().error(brakeTestResult.toString());
39                 allAxesOk = false;
40                 break;
41             default:
42                 break;
43         }
44     catch (CommandInvalidException ex) {
45         ...
46         allAxesOk = false;
47     }
48 }
49
50 if (allAxesOk) {
51     getLogger().info("Brake test was successful for all axes.");
52 }
53 else{
54     getLogger().error("Brake test failed for at least one
axis.");
55 }
56 }
```

Línea	Descripción
3	Desplazarse hasta la posición de salida desde la que se desplaza el robot, para determinar el máximo par de torsión absoluto para cada eje. La posición de salida es por defecto la posición vertical.
5	Preparar la evaluación de datos. Para poder evaluar de forma específica para el eje los pares de torsión que se determinan durante una secuencia de movimientos, se debe crear una instancia de la clase TorqueEvaluator.
7	Seleccionar los momentos que se utilizan para la evaluación de datos. No se utilizan los pares de torsión medidos, sino los pares de torsión que se calculan durante la secuencia de movimientos con la ayuda del modelo del robot. Las mediciones son propensas a los fallos. El cálculo de los valores de los pares de torsión asegura que no entren momentos defectuosos en la evaluación de datos, que se generen debido a efectos dinámicos (p. ej. aceleración del robot).
9	Iniciar la evaluación de datos. Mediante la instrucción startEvaluation() de la clase TorqueEvaluator se inicia la evaluación de datos.

Línea	Descripción
11 ... 16	Secuencia de movimientos para la determinación de los máximos pares de torsión absolutos Se desplazan 2 posiciones de los ejes predefinidas, respectivamente con un movimiento PTP.
18	Finalizar la evaluación de datos y consultar los datos. La instrucción stopEvaluation() de la clase TorqueEvaluator finaliza la evaluación de datos y envía como respuesta el resultado como valor del tipo TorqueStatistic. El resultado se guarda en la variable maxTorqueData.
20	Variable para la evaluación de la prueba de frenos Mediante la variable allAxesOk se guarda el resultado de la prueba de frenos para la evaluación posterior. Se ajusta a false si falla la prueba de frenos de un eje o se cancela debido a un error. De lo contrario se conserva el valor true.
22 ... 54	Ejecución de la prueba de frenos Los frenos se comprueban de forma consecutiva, comenzando por el freno del eje A1. <ol style="list-style-type: none"> Línea 24, 25: Se crea un objeto del tipo BrakeTest y se transmite al mismo tiempo el eje correspondiente y el máximo par de torsión absoluto como momento de frenado de referencia. Línea 26: La prueba de frenos se ejecuta como instrucción de movimiento. Línea 27 ... 54: El resultado de la prueba de frenos se evalúa y se visualiza en la smartHMI.

8.2.1 Adaptar la aplicación de la prueba de frenos para la prueba de acuerdo con un mínimo momento de frenado de los frenos

Descripción

En la prueba de frenos se debe comprobar si los frenos generan el mínimo momento de frenado de los frenos. Por ello se recomienda adaptar la aplicación de la prueba de frenos preparada de acuerdo con la siguiente descripción.

Si debe ejecutarse la prueba de frenos sin que se calculen los momentos de frenado de referencia y se transmitan a la prueba de frenos, se deberán eliminar en la aplicación de la prueba de frenos todas las líneas de instrucciones relevantes para el cálculo de los valores de momentos. La aplicación de la prueba de frenos se iniciará a continuación con el movimiento hacia la posición de salida para la prueba de frenos.

Adicionalmente, para la generación de la instancia BrakeTest se deberá eliminar el parámetro con el que se transmite el momento de referencia.

```

boolean allAxesOk = true;
for (int axis : axes) {
    try {
        ① BrakeTest brakeTest = new BrakeTest(axis, maxTorqueData.getMaxAbsTorqueValues()[axis]);
        IMotionContainer motionContainer = lbr_iwa.move(brakeTest);
        BrakeTestResult brakeTestResult = BrakeTest.evaluateResult(motionContainer);
    }
}
  
```

Fig. 8-1: Transferencia del momento de referencia para la prueba de frenos

- 1 Constructor de la clase BrakeTest con transmisión del momento de referencia

- Procedimiento**
1. Abrir la aplicación de la prueba de frenos en Sunrise.Workbench.
 2. Realizar las siguientes modificaciones en el método run() de la aplicación:
 - Borrar todas las líneas de instrucciones relevantes para el cálculo de los valores de momentos.
 - Borrar los siguientes parámetros en la activación del constructor de la clase BrakeTest:

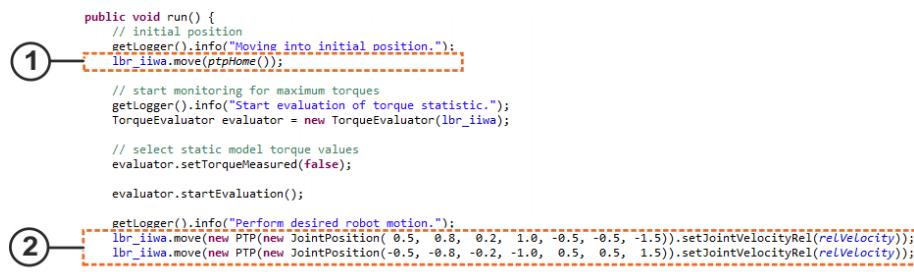
```
maxTorqueData.getMaxAbsTorqueValues() [axis]
```

En la línea se mantiene el siguiente código:

```
BrakeTest brakeTest = new brakeTest(axis);
```
 3. Guardar las modificaciones.

8.2.2 Modificar la secuencia de movimientos para la determinación del valor del momento

- Descripción**
- La aplicación de la prueba de frenos creada a partir de la plantilla, contiene una secuencia de movimientos preparada para la determinación de los máximos pares de torsión absolutos que se generan en cada eje.
- El robot es desplazado por defecto desde la posición vertical. Se puede seleccionar otra posición de salida.
- Desde la posición de salida se desplazan 2 posiciones de los ejes predefinidas, respectivamente con un movimiento PTP. Para calcular los máximos pares de torsión absolutos de una aplicación del robot específica y poder utilizarlos como momentos de frenado de referencia para la prueba de frenos, se debe insertar la aplicación del robot en la aplicación de la prueba de frenos.



```
public void run() {
    // initial position
    getLogger().info("Moving into initial position.");
    lbr_iiwa.move(ptpHome());

    // start monitoring for maximum torques
    getLogger().info("Start evaluation of torque statistic.");
    TorqueEvaluator evaluator = new TorqueEvaluator(lbr_iiwa);

    // select static model torque values
    evaluator.setTorqueMeasured(false);
    evaluator.startEvaluation();

    getLogger().info("Perform desired robot motion:");
    lbr_iiwa.move(new PTP(new JointPosition( 0.5, 0.8, 0.2, 1.0, -0.5, -0.5, -1.5)).setJointVelocityRel(relVelocity));
    lbr_iiwa.move(new PTP(new JointPosition(-0.5, -0.8, -0.2, -1.0, 0.5, 0.5, 1.5)).setJointVelocityRel(relVelocity));

    // stop monitoring for maximum torques and store results
    getLogger().info("Stop evaluation of torque statistic.");
    TorqueStatistic maxTorqueData = evaluator.stopEvaluation();
    getLogger().info("The result of evaluation:\n" + maxTorqueData.toString());
}
```

Fig. 8-2: Secuencia de movimientos para la determinación del valor del momento

- 1 Movimiento ptpHome() hacia la posición de salida
- 2 Secuencia de movimientos predefinida para la determinación del valor del momento

- Procedimiento**
1. Abrir la aplicación de la prueba de frenos en Sunrise.Workbench.
 2. En caso necesario, realizar las siguientes modificaciones en el método run() de la aplicación:
 - Sustituir el movimiento ptpHome() que lleva al robot hasta la posición de salida, por un movimiento hacia la posición de salida deseada.
 - Sustituir la secuencia de movimientos predefinida por un código de aplicación propio.
 3. Guardar las modificaciones.

8.2.3 Modificar la posición de salida para la prueba de frenos

- Descripción**
- La aplicación de la prueba de frenos creada a partir de la plantilla ejecuta la prueba de frenos por defecto en la posición final de la secuencia de movimien-

tos, para determinar el máximo par de torsión absoluto. Si esta posición no es adecuada como posición de salida para la prueba de frenos, se deberá programar un movimiento en la posición de salida deseada antes de la ejecución de la prueba de frenos.



Para determinar la gravitación y la fricción, se realiza el desplazamiento de los ejes de un LBR iiwa en dirección a la posición cero mecánica. El máximo trayecto de desplazamiento es de 5° en el lado de salida.

8.3 Interfaz de programación para la prueba de frenos

La RoboticsAPI ofrece con la clase BrakeTest una interfaz de programación para la ejecución de la prueba de frenos. La prueba de frenos se ejecuta como comando de movimiento.

Además, con la ayuda de la clase TorqueEvaluator se pueden evaluar los pares de torsión que se miden durante la secuencia de movimientos y se puede calcular el máximo par de torsión absoluto para cada eje. Este momento se puede utilizar como momento de frenado de referencia para la prueba de frenos.

8.3.1 Evaluar momentos generados y determinar el máximo valor absoluto

Descripción

Para poder evaluar de forma específica para el eje los pares de torsión que se determinan durante una secuencia de movimientos, primero se debe crear un objeto de la clase TorqueEvaluator. Al constructor de la clase TorqueEvaluator se le transmite la instancia LBR para cuyos ejes se deben calcular los máximos valores absolutos de los pares de torsión.

Con los siguientes métodos de la clase TorqueEvaluator se puede iniciar la evaluación y se puede finalizar de nuevo:

- `startEvaluation()`: Inicia la evaluación.

Después de la activación del método, se debe ordenar la secuencia de movimientos a evaluar.

- `stopEvaluation()`: Finaliza la evaluación.

El método emite un objeto del tipo `TorqueStatistic`, con el que se puede consultar el resultado de la evaluación.

Los valores de los pares de torsión que se generan, se pueden calcular de diferentes modos:

- Momentos medidos: Se utilizan los pares de torsión que son medidos por los sensores de momentos de articulaciones.
- Momentos estáticos (basados en el modelo): Se utilizan los pares de torsión que se calculan con la ayuda del modelo de robot estático.

Con el método `setTorqueMeasured(...)` de la clase `TorqueEvaluator` se puede determinar si para la evaluación se deben utilizar los momentos medidos o los momentos estadísticos (basados en el modelo).

Sintaxis

```
TorqueEvaluator evaluator = new TorqueEvaluator(lbr_iowa) ;  
evaluator.setTorqueMeasured(isTorqueMeasured) ;  
evaluator.startEvaluation() ;  
// Secuencia de movimientos  
TorqueStatistic maxTorqueData = evaluator.stopEvaluation() ;
```

Aclaración de la sintaxis

Elemento	Descripción
<i>evaluator</i>	<p>Tipo: TorqueEvaluator</p> <p>Variable a la que se asigna la instancia TorqueEvaluator creada. Mediante la variable se inicia y se finaliza la evaluación de los pares de torsión durante una secuencia de movimientos.</p>
<i>isTorque Measured</i>	<p>Tipo: boolean</p> <p>Parámetro de entrada del método setTorqueMeasured(...). Determina si para la evaluación se deben utilizar los valores de los pares de torsión medidos o calculados con la ayuda del modelo de robot estático.</p> <ul style="list-style-type: none"> ■ true: Utilización de los momentos medidos ■ false: Utilización de los momentos estáticos (basados en el modelo) <p>Indicación: Si se utilizan momentos estáticos (basados en el modelo), los efectos dinámicos que se generan, por ejemplo, por la aceleración del robot, no tendrán ningún efecto sobre los valores calculados.</p>
<i>lbr_iiwa</i>	<p>Tipo: LBR</p> <p>Instancia LBR de la aplicación. Representa al robot para que el que deben calcularse los máximos valores absolutos de los pares de torsión.</p>
<i>maxTorque Data</i>	<p>Tipo: TorqueStatistic</p> <p>Variable para el valor de retorno de stopEvaluation(). El valor de retorno contiene los máximos valores absolutos de los pares de torsión calculados y más información sobre la evaluación.</p>

8.3.2 Consultar el resultado de la evaluación de los máximos momentos absolutos

Cuando ha finalizado la evaluación de los máximos valores absolutos de los pares de torsión, se puede consultar el resultado de la evaluación.

Resumen

Se encuentran disponibles los siguientes métodos de la clase TorqueStatistic:

Método	Descripción
<code>getMaxAbs TorqueValues()</code>	<p>Tipo de retorno: double[]; unidad: Nm</p> <p>Emite un conjunto doble que contiene los máximos valores absolutos de los pares de torsión calculados (en el lado de salida) de todos los ejes.</p>
<code>getSingleMaxAbs TorqueValue(...)</code>	<p>Tipo de retorno: double; unidad: Nm</p> <p>Emite el máximo valor absoluto del par de torsión calculado (en el lado de salida) para el eje que se transmite como parámetro (tipo: JointE-num).</p>
<code>areDataValid()</code>	<p>Tipo de retorno: boolean</p> <p>Se consulta si los datos calculados son válidos (= true).</p> <p>Los datos son válidos si no se producen errores durante el procesamiento de instrucciones.</p>
<code>getStartTimestamp()</code>	<p>Tipo de retorno: java.util.Date</p> <p>Emite el momento en el que se inició la evaluación.</p>

Método	Descripción
getStopTimestamp()	Tipo de retorno: java.util.Date Emite el momento en el que finalizó la evaluación.
isTorqueMeasured()	Tipo de retorno: boolean Se consulta si durante la evaluación del máximo par de torsión absoluto se han utilizado los valores del par de torsión medidos o calculados con la ayuda del modelo de robot estático. <ul style="list-style-type: none"> ■ true: Utilización de los momentos medidos ■ false: Utilización de los momentos estáticos (basados en el modelo)

Ejemplo Los momentos máximos que se producen en una tarea de ensamblado se deben utilizar como momentos de referencia en una prueba de frenos. Para ello se evalúan los pares de torsión que se miden durante la ejecución de la tarea de ensamblado y se calcula el máximo par de torsión absoluto para cada eje.

Después del inicio de la evaluación se ordenan las instrucciones de movimiento del proceso de ensamblado. Si ha finalizado el proceso de ensamblado, finalizará la evaluación y se guarda el resultado de la evaluación para los ejes A2 y A4 en los datos de proceso. Si los datos calculados no son válidos, se ajusta una salida.

```
testEvaluator.setTorqueMeasured(true);
```

```
private LBR testLBR;
private BrakeTestIOGroup brakeTestIOS;
private Tool testGripper;
private Workpiece testWorkpiece;
...
public void run() {

    testGripper.attachTo(testLBR.getFlange());
    testWorkpiece.attachTo(testGripper.getFrame("/GripPoint"));

    // create TorqueEvaluator
    TorqueEvaluator testEvaluator = new TorqueEvaluator(testLBR);

    // select measured torque values
    testEvaluator.setTorqueMeasured(true);

    // start evaluation
    testEvaluator.startEvaluation();

    // performs assembly task
    testAssemblyTask();

    // finish evaluation and store result in variable testMaxTrqData
    TorqueStatistic testMaxTrqData = testEvaluator.stopEvaluation();

    // get maximum absolute measured torque value for joint 2
    double maxTrqA2 = testMaxTrqData
        .getSingleMaxAbsTorqueValue(JointEnum.J2);

    // save result
    getApplicationData().getProcessData("maxTrqA2").setValue(maxTrqA2);

    // get maximum absolute measured torque value for joint 4
    double maxTrqA4 = testMaxTrqData
        .getSingleMaxAbsTorqueValue(JointEnum.J4);
```

```

// save result
getApplicationData().getProcessData("maxTrqA4").setValue(maxTrqA4);

// check if evaluated data is valid
boolean areDataValid = testMaxTrqData.areDataValid();
if(areDataValid == false){
    // if data is not valid, set output signal
    brakeTestIOs.setEvaluatedTorqueInvalid(true);
}

...
}

public void exampleAssemblyTask() {

    testLBR.move(ptp(getFrame("/StartAssembly")));

    ForceCondition testForceCondition =
        ForceCondition.createNormalForceCondition
            (testWorkpiece.getDefaultMotionFrame(), CoordinateAxis.Z, 15.0);
    testWorkpiece.move(linRel(0.0, 0.0, 100.0)
        .breakWhen(testForceCondition));

    CartesianSineImpedanceControlMode testAssemblyMode =
        CartesianSineImpedanceControlMode.createLissajousPattern(
            CartPlane.XY, 5.0, 10.0, 500.0);

    testWorkpiece.move(positionHold(
        testAssemblyMode, 3.0, TimeUnit.SECONDS));

    openGripper();
    testWorkpiece.detach();

    testGripper.move(linRel(0.0, 0.0, -100.0));
}

```

8.3.3 Crear un objeto para la prueba de frenos

Descripción

Para poder ejecutar la prueba de frenos, primero se debe crear un objeto de la clase BrakeTest. Al constructor de la clase BrakeTest se le transmite el índice del eje para el que debe realizarse la prueba de frenos.

Opcionalmente, a través del parámetro *torque* se puede transmitir un momento de frenado de referencia, p. ej. el máximo par de torsión absoluto de un eje que se produce en una aplicación determinada.

En la prueba de frenos generalmente se debe comprobar si los frenos generan el mínimo momento de frenado de los frenos. Por ello se recomienda no indicar el parámetro *torque*.

Sintaxis

```
BrakeTest brakeTest = new BrakeTest(axis, <torque>);
```

Aclaración de la sintaxis

Elemento	Descripción
<i>brakeTest</i>	<p>Tipo: BrakeTest</p> <p>Variable a la que se asigna la instancia BrakeTest creada. Mediante la variable se ordena la ejecución de la prueba de frenos como instrucción de movimiento.</p>
<i>axis</i>	<p>Tipo: int</p> <p>Índice del eje cuyo freno se debe comprobar</p> <ul style="list-style-type: none"> ■ 0 ... 6: Eje A1 ... A7
<i>torque</i>	<p>Tipo: double; unidad: Nm</p> <p>Momento de frenado de referencia especificado por el usuario (en el lado de salida), p. ej. el máximo par de torsión absoluto que ha sido calculado previamente para una secuencia de movimientos específica para aplicaciones.</p> <p>Si no se indica ningún momento de frenado de referencia, en la prueba de frenos se utilizará el valor más bajo de los siguientes valores como momento de frenado a comprobar: Mínimo momento de frenado o momento de retención de motor.</p> <p>Si se indica un momento de frenado de referencia, como momento de frenado que debe comprobarse se utiliza uno de los siguientes valores: El momento de frenado de referencia especificado (<i>torque</i>, el mínimo momento de frenado de los frenos o el momento de retención de motor).</p> <p>El momento de frenado que debe comprobarse se determina de forma interna en el sistema según las siguientes reglas:</p> <ol style="list-style-type: none"> 1. Si el momento de frenado de referencia es mayor que el mínimo del mínimo momento de frenado de los frenos y del momento de retención de motor, se utilizará el mínimo del mínimo momento de frenado de los frenos y del momento de retención de motor como momento de frenado que debe comprobarse. 2. Si el momento de frenado de referencia es menor que el 20 % del mínimo del mínimo momento de frenado de los frenos y del momento de retención de motor, se utilizará el 20 % del mínimo del mínimo momento de frenado de los frenos y del momento de retención de motor como momento de frenado que debe comprobarse. 3. En todos los demás casos se utiliza el momento de frenado de referencia. <p>Indicación: El momento de frenado de los frenos mínimo y máximo está guardado en los datos del motor. El momento de retención de motor se deriva de los datos del motor.</p>

8.3.4 Iniciar la ejecución de la prueba de frenos

Descripción

La prueba de frenos se ejecuta mediante un comando de movimiento que se pone a disposición mediante la clase BrakeTest. Para ejecutar la prueba de frenos, se activa el método move(...) o moveAsync(...) mediante la instancia del robot utilizada en la aplicación y se transmite el objeto creado para la prueba de frenos.

Para evaluar el resultado de la prueba de frenos, se deberá guardar el valor de retorno del comando de movimiento en una variable del tipo IMotionContainer.

Si durante la ejecución de la prueba de frenos se detecta un error, se cancelará la prueba de frenos. Para poder reaccionar en el programa ante un caso de error, se recomienda ordenar la ejecución y la evaluación de la prueba de frenos dentro de un bloque try y tratar la CommandInvalidException que se ha generado.

Sintaxis

```
try{
    BrakeTest brakeTest = ...;
    IMotionContainer brakeTestMotionContainer =
        robot.moveToMoveAsync(brakeTest);
    ...
} catch(CommandInvalidException ex){
    ...
}
```

Aclaración de la sintaxis

Elemento	Descripción
<i>brakeTest</i>	Tipo: BrakeTest Variable a la que se asigna la instancia BrakeTest creada. La instancia determina el eje para el que debe ejecutarse la prueba de frenos y, opcionalmente, un momento de frenado de referencia establecido por el programador.
<i>brakeTest Motion Container</i>	Tipo: IMotionContainer Variable para el valor de retorno del comando de movimiento move(...) o moveAsync(...) con el que se ejecuta la prueba de frenos. Cuando ha finalizado la prueba de frenos, se podrá evaluar el resultado mediante la variable.
<i>robot</i>	Tipo: Robot Instancia del robot utilizado en la aplicación para cuyos ejes se debe ejecutar la prueba de frenos.
<i>ex</i>	Tipo: CommandInvalidException Excepción que se genera cuando la prueba de frenos se cancela debido a un error. Se recomienda tratar la excepción dentro del bloque catch, de forma que una prueba de frenos cancelado para un freno individual no provoque la cancelación de la aplicación completa de la prueba de frenos.

8.3.5 Evaluar la prueba de frenos

Descripción

Cuando ha finalizado la prueba de frenos, se puede evaluar el resultado. Para ello, el valor de retorno del comando de movimiento a través del cual se ejecuta la prueba de frenos debe asignarse a una variable del tipo IMotionContainer.

Para evaluar la prueba de frenos, la instancia IMotionContainer de la instrucción de movimiento correspondiente se transfiere al método estático evaluateResult(...). El método pertenece a la clase BrakeTest e indica un objeto del tipo BrakeTestResult. De este objeto pueden consultarse diferentes datos acerca de la prueba de frenos realizado.

Sintaxis

```
IMotionContainer brakeTestMotionContainer =
robot.moveMoveAsync(brakeTest);

BrakeTestResult result =
BrakeTest.evaluateResult(brakeTestMotionContainer);
```

Aclaración de la sintaxis

Elemento	Descripción
<i>brakeTest Motion Container</i>	Tipo: IMotionContainer Variable para el valor de retorno del comando de movimiento move(...) o moveAsync(...) con el que se ejecuta la prueba de frenos.
<i>result</i>	Tipo: BrakeTestResult Variable para el valor de retorno de evaluateResult(...). El valor de retorno contiene el resultado de la prueba de frenos y otra información sobre la prueba de frenos que puede consultarse a través de la variable.

Resumen

Para evaluar la prueba de frenos están disponibles los siguientes métodos de la clase BrakeTestResult:

Método	Descripción
getAxis()	Tipo de retorno: int Indica el índice del eje cuyo freno se ha comprobado. El índice empieza por 0 (= eje A1).
getBrakeIndex()	Tipo de retorno: int Indica el índice del freno comprobado del motor (comienza por 0). En una prueba de frenos para LBR iiwa se indica siempre el valor 0.
getFriction()	Tipo de retorno: double, unidad: Nm Indica el momento de fricción determinado (en el lado de salida) durante el desplazamiento de cálculo.
getGravity()	Tipo de retorno: double, unidad: Nm Indica el momento gravitacional determinado (en el lado de salida) durante el desplazamiento de cálculo.
getMaxBrake HoldingTorque()	Tipo de retorno: double, unidad: Nm Indica el momento calculado a partir de los datos del motor (en el lado de salida), que no debe exceder el freno. (= momento máximo de detención de los frenos)
getMeasuredBrake HoldingTorque()	Tipo de retorno: double, unidad: Nm Indica el momento de frenado medido durante la prueba de frenos (en el lado de salida). Este valor se compara con el momento de frenado que se va a comprobar.
getMinBrake HoldingTorque()	Tipo de retorno: double, unidad: Nm Indica el momento del freno calculado a partir de los datos del motor que debe alcanzarse como mínimo (en el lado de salida). (= momento mínimo de detención de los frenos)
getMotor HoldingTorque()	Tipo de retorno: double, unidad: Nm Indica el momento de retención de motor calculado a partir de los datos del motor (en el lado de salida).
getMotorIndex()	Tipo de retorno: int Indica el índice del motor comprobado del accionamiento (comienza por 0). En una prueba de frenos para LBR iiwa se indica siempre el valor 0.

Método	Descripción
getMotorMaximalTorque()	Tipo de retorno: double, unidad: Nm Indica el momento máximo de motor calculado a partir de los datos del motor (en el lado de salida).
getState()	Tipo de retorno: Enum del tipo BrakeState Indica el resultado de la prueba de frenos. <i>(>>> 8.3.5.1 "Consultar el resultado de la prueba de frenos" Página 133)</i>
getTestedTorque()	Tipo de retorno: double, unidad: Nm Indica el momento de frenado que se debe comprobar, con el que se compara el momento de frenado medido durante la prueba de frenos (en el lado de salida).
getTimestamp()	Tipo de retorno: java.util.Date Indica el momento en el que se ha iniciado la prueba de frenos.

8.3.5.1 Consultar el resultado de la prueba de frenos

Descripción El resultado de la prueba se consulta mediante el método BrakeTestResult-Methode getState(). Como resultado se ofrece un Enum del tipo BrakeState, cuyos valores describen los posibles resultados de la prueba. Además, con getLogLevel() se puede consultar el nivel Log perteneciente al resultado de la prueba.

Sintaxis

```
BrakeTestResult result = ...;
BrakeState state = result.getState();
LogLevel logLevel = state.getLogLevel();
```

Aclaración de la sintaxis	Elemento	Descripción
	<i>result</i>	Tipo: BrakeTestResult Variable para el valor de retorno del método estático evaluateResult(...), que ofrece la clase BrakeTest para la evaluación de la prueba de frenos. El valor de retorno contiene el resultado de la prueba de frenos y otra información sobre la prueba de frenos que puede consultarse a través de la variable.
	<i>state</i>	Tipo: Enum del tipo BrakeState Variable para el valor de retorno de getState(). El valor de retorno contiene el resultado de la prueba.
	<i>logLevel</i>	Tipo: Enum del tipo LogLevel Variable para el valor de retorno de getLogLevel(). El valor de retorno contiene el nivel Log del resultado de la prueba.

BrakeState El Enum del tipo BrakeState posee los siguientes valores (con indicación del nivel Log correspondiente):

Valor	Descripción
BrakeUntested	La prueba de frenos no se ha podido ejecutar debido a la existencia de averías o se ha cancelado durante la ejecución. Nivel Log: LogLevel.Error
BrakeUnknown	La prueba de frenos no se ha podido ejecutar, ya que no se ha podido crear el momento suficiente (p. ej. debido a una fricción demasiado alta). Nivel Log: LogLevel.Error
BrakeError	Fallo al efectuar la prueba de frenos. El momento de frenado medido no alcanza el momento de frenado que se debe comprobar. El freno está defectuoso. Nivel Log: LogLevel.Error
BrakeWarning	El momento de frenado medido se encuentra a menos del 5 % por encima del momento de frenado que se debe comprobar. El freno ha alcanzado el límite de desgaste y será detectado como defectuoso en un tiempo previsible. Nivel Log: LogLevel.Warning
BrakeMax Unknown	El momento de frenado que se debe comprobar se ha alcanzado, pero no se ha podido comprobar el máximo momento de frenado del freno. Nivel Log: LogLevel.Warning
BrakeExcessive	El momento de frenado medido es mayor que el máximo momento de frenado del freno. La parada con el freno puede provocar daños en la máquina. Nivel Log: LogLevel.Warning
BrakeReady	El momento de frenado medido se encuentra a más del 5 % por encima del momento de frenado que se debe comprobar. El freno está completamente en condiciones de funcionamiento. Nivel Log: LogLevel.Info

Ejemplo

Para el eje A2 se ejecuta una prueba de frenos. Si la prueba de frenos se cancela, se indicará mediante la correspondiente señal de salida. Si la prueba de frenos se ejecuta completamente, se emitirá un mensaje con el momento de frenado medido y se consultará el resultado de la prueba. En función de si el momento de frenado medido es demasiado bajo, se encuentra dentro del rango de tolerancia o en el rango ideal, también se ajustará respectivamente una salida.

```

private LBR exampleLBR_iowa;
private BrakeTestIOGroup brakeTestIOS;
...
public void run() {
    ...

    try {
        int indexA2 = 1;
        BrakeTest exampleBrakeTest = new BrakeTest(indexA2);

        IMotionContainer exampleBrakeTestMotionContainer =
            exampleLBR_iowa.move(exampleBrakeTest);

        BrakeTestResult resultA2 = BrakeTest.evaluateResult(
    }
}

```

```

        exampleBrakeTestMotionContainer);

    double measuredTorque =
        resultA2.getMeasuredBrakeHoldingTorque();

    getLogger().info("Measured torque for A2: " + measuredTorque);

    BrakeState state = resultA2.getState();
    if(state == BrakeState.BrakeError)
        brakeTestIOs.setA2_BrakeError(true);
    else if(state == BrakeState.BrakeWarning)
        brakeTestIOs.setA2_BrakeWarning(true);
    else if(state == BrakeState.BrakeReady)
        brakeTestIOs.setA2_BrakeOK(true);

} catch (CommandInvalidException ex) {
    brakeTestIOs.setBrakeTest_Aborted(true);
    ex.printStackTrace();
}
}

```

8.4 Ejecutar la prueba de frenos

AVISO

Si se detecta que un freno está defectuoso y se desconectan los accionamientos, el robot puede desplomarse.

Descripción

Si la plantilla para la aplicación de la prueba de frenos se acepta sin modificación, primero se evalúan de forma específica para el eje los pares de torsión que se miden durante una secuencia de movimientos, y se calcula el máximo momento absoluto generado para cada eje. El resultado de la evaluación se visualiza en la smartHMI. Seguidamente se comprueban los frenos de forma consecutiva, comenzando por el eje A1. El resultado de la prueba de frenos se visualiza en la smartHMI individualmente para cada eje.

Condición previa

- No deben encontrarse personas u objetos en la zona de acción del robot.
- Tipo de flujo de programa **Continuous** (modo estándar)
- El robot ha alcanzado la temperatura de servicio.

Procedimiento

- Seleccionar e iniciar la aplicación de la prueba de frenos.



Si la aplicación de la prueba de frenos se pausa durante la prueba de un freno (p. ej. pulsando la tecla de arranque en el smartPAD o mediante una solicitud de parada), se cancelará la prueba de frenos del eje.

Si se continúa la aplicación de la prueba de frenos, se repetirá la prueba de frenos para el eje afectado que ha sido cancelada anteriormente. En caso de que el eje ya no se encuentre en la posición desde la que se inició la prueba de frenos cancelada, se deberá retroceder pulsando la tecla de arranque, antes de que se pueda continuar la aplicación.

8.4.1 Resultado de la evaluación de los máximos momentos absolutos (indicación)

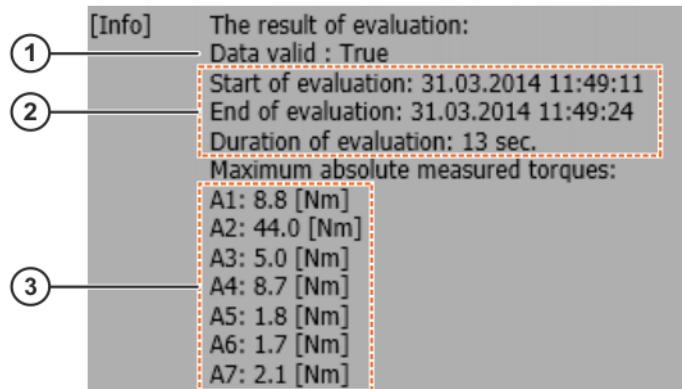


Fig. 8-3: Resultado de una evaluación de los máximos momentos absolutos

Pos.	Descripción
1	Validez Indica si los datos calculados son válidos. Los datos son válidos si no se producen errores durante el procesamiento de instrucciones.
2	Datos de tiempo Momento de inicio, momento final y duración total de la evaluación.
3	Datos determinados Para cada eje se muestra el máximo par de torsión absoluto que se ha calculado en la evaluación.

8.4.2 Consultar el resultado de la prueba de frenos (indicación)

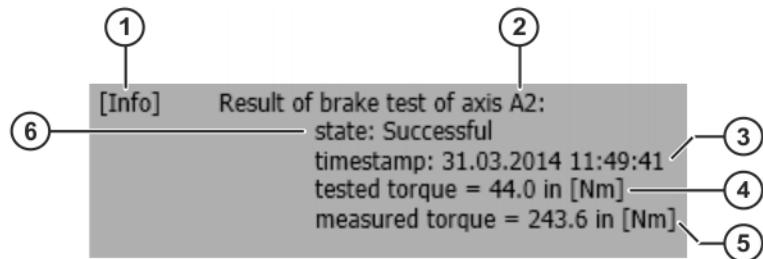


Fig. 8-4: Resultado de una prueba de frenos para el eje A2

Pos.	Descripción
1	<p>Nivel Log</p> <p>En función del resultado de la prueba de frenos se emite el mensaje con un determinado nivel Log.</p> <ul style="list-style-type: none"> ■ Info: La prueba de frenos se ha ejecutado correctamente. ■ Warning: El momento de frenado a comprobar se ha alcanzado, pero durante la ejecución de la prueba de frenos se han producido problemas (véanse las descripciones de los posibles resultados de la prueba en la pos. 6). ■ Error: La prueba de frenos no se ha podido ejecutar o ha fallado.
2	Eje comprobado
3	Marca de tiempo
4	Momento de frenado a comprobar
5	Momento de frenado medido
6	<p>Resultado de la prueba de frenos</p> <ul style="list-style-type: none"> ■ Untested: La prueba de frenos no se ha podido ejecutar debido a la existencia de averías o se ha cancelado durante la ejecución. ■ Unknown: La prueba de frenos no se ha podido ejecutar, porque a que no se ha podido el momento suficiente (p. ej. debido a una fricción demasiado alta). ■ Failed: Fallo al efectuar la prueba de frenos. El momento de frenado medido no alcanza el momento de frenado que se debe comprobar. El freno está defectuoso. ■ Warning: El momento de frenado medido se encuentra a menos del 5 % por encima del momento de frenado que se debe comprobar. El freno ha alcanzado el límite de desgaste y será detectado como defectuoso en un tiempo previsible. ■ Maximum unknown: El momento de frenado que se debe comprobar se ha alcanzado, pero no se ha podido comprobar el máximo momento de frenado del freno. ■ Excessive: El momento de frenado medido es mayor que el máximo momento de frenado del freno. La parada con el freno puede provocar daños en la máquina. ■ Successfull: El momento de frenado medido se encuentra a más del 5 % por encima del momento de frenado que se debe comprobar. El freno está completamente en condiciones de funcionamiento.

 **ADVERTENCIA**

Si no está garantizado el correcto funcionamiento de un freno, el robot puede desplomarse. Si durante la prueba de frenos se detecta al menos un eje del LBR iiwa que no puede alcanzar el momento de frenado deseado, el robot se deberá retirar inmediatamente de la circulación.

9 Gestión del proyecto

9.1 Proyecto Sunrise – Resumen

En un proyecto Sunrise están agrupados todos los datos que son necesarios para el servicio de una estación. En un proyecto Sunrise se incluyen:

- Configuración de la estación
La configuración de la estación describe las propiedades estáticas de la estación. Aquí se incluyen, p. ej. los componentes de hardware y software.
- Aplicaciones
Las aplicaciones incluyen el código fuente para la ejecución de una tarea para la estación. Se programan con KUKA Sunrise.Workbench en Java y se ejecutan en el control. Un proyecto Sunrise puede contener el número deseado de aplicaciones.
- Datos del tiempo de ejecución
Los datos del tiempo de ejecución son todos los datos que se utilizan por las aplicaciones para el tiempo de ejecución. Aquí se incluyen, p. ej. los puntos de destino para movimientos, datos de herramientas y parámetros de proceso.
- Configuración de seguridad
La configuración de seguridad incluye las funciones de seguridad configuradas.
- Configuración de E/S (opcional)
La configuración de E/S contiene las entradas/salidas de los buses de campo utilizados que están conectadas en WorkVisual. Las entradas/salidas se pueden utilizar en la aplicación.

Los proyectos Sunrise se crean y se gestionan con KUKA Sunrise.Workbench.

(>>> 5.3 "Crear proyecto Sunrise con plantilla" Página 51)

Solo puede haber siempre 1 proyecto Sunrise en la unidad de control del robot. Este proyecto se transmite desde Sunrise.Workbench a la unidad de control del robot mediante la sincronización del proyecto.

(>>> 9.4 "Sincronización de proyectos" Página 155)

9.2 Gestión de Frames

Resumen

Los Frames son transformaciones de coordenadas que describen la posición de los puntos en el espacio o de los objetos en una estación. Las transformaciones de coordenadas están dispuestas jerárquicamente en una estructura de árbol. En esta jerarquía, cada Frame posee un Frame padre superior, con el que está conectado a través de la transformación.

El elemento raíz o el origen de la transformación es el sistema de coordenadas Mundo, que se encuentra por defecto en el pie del robot. Es decir, todos los Frames hacen referencia directa o indirectamente al sistema de coordenadas Mundo.

Una transformación describe la posición relativa de 2 sistemas de coordenadas entre sí, es decir, cómo un Frame está desplazado y orientado en relación con su Frame padre.

La posición de un Frame referida a su Frame padre se define por los siguientes datos de transformación:

- X, Y, Z: Desplazamiento del origen a los largo de los ejes del Frame padre

- A, B, C: Giro del ángulo del eje del Frame padre

Ángulo de rotación de los Frames:

- Ángulo A: Giro alrededor del eje Z
- Ángulo B: Giro alrededor del eje Y
- Ángulo C: Giro alrededor del eje X

9.2.1 Crear un nuevo Frame

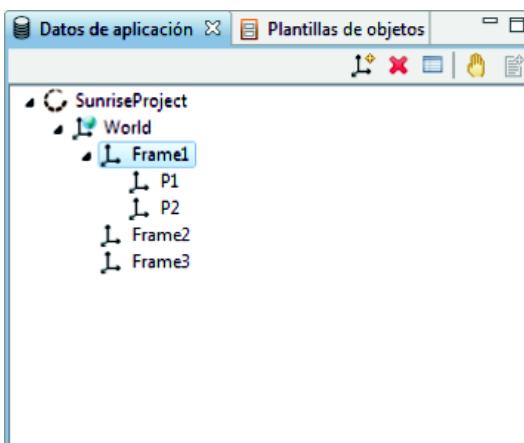
Descripción	<p>Los Frames creados en Sunrise.Workbench son específicos del proyecto y se pueden utilizar en cualquier aplicación del robot del proyecto.</p> <p>Después de la sincronización del proyecto, los Frames están disponibles en el smartHMI. En el smartHMI, los Frames se pueden programar por aprendizaje para determinar la posición de los Frames en la zona. El desplazamiento hasta los Frames programados por aprendizaje se puede realizar manualmente.</p> <p>(>>> 6.13 "Programar por aprendizaje y desplazar manualmente los Frames" Página 85)</p>
Procedimiento	<ol style="list-style-type: none"> 1. Marcar el proyecto en el Explorador de paquetes. 2. En la vista Datos de aplicación hacer clic con el botón derecho del ratón sobre el Frame padre y seleccionar Crear Frame nuevo en el menú contextual. El nuevo Frame se crea y se integra en el árbol de Frame como subelemento del Frame padre. 3. El sistema asigna automáticamente un nombre de Frame. Se recomienda modificar el nombre en la vista Propiedades. <p>Un nombre de Frame descriptivo facilita la programación y la orientación en el programa. Los nombres de Frame deben ser únicos dentro un nivel jerárquico y no se deben asignar varias veces.</p>
Ejemplo	

Fig. 9-1: Datos de aplicación – Frames

Los Frame1, 2 y 3 son subelementos de World y se encuentran en el mismo nivel jerárquico. P1 y P2 son subelementos de Frame1 y se encuentran un nivel por debajo.

9.2.2 Marcar Frame como base

Descripción	<p>Los Frames pueden marcarse como base en la vista Datos de aplicación. Solo los Frames marcados de este modo se pueden seleccionar como base para el desplazamiento manual y medir después de la sincronización del proyecto en la smartHMI.</p>
--------------------	---

(>>> 6.8.1 "Ventana Opciones de desplazamiento manual" Página 78)

(>>> 7.2.2 "Medición base: Método de 3 puntos " Página 112)



Se recomienda ponerle nombres claros e identificativos a estos Frames para facilitarle al operario en el smartPAD la selección para el desplazamiento manual.

Procedimiento

- Hacer clic con el botón derecho sobre el Frame deseado y seleccionar **Base** en el menú contextual.
- Alternativa:
- Marcar el Frame y hacer clic sobre el símbolo en forma de mano **Base**. El Frame se identifica con el símbolo en forma de mano.

Ejemplo

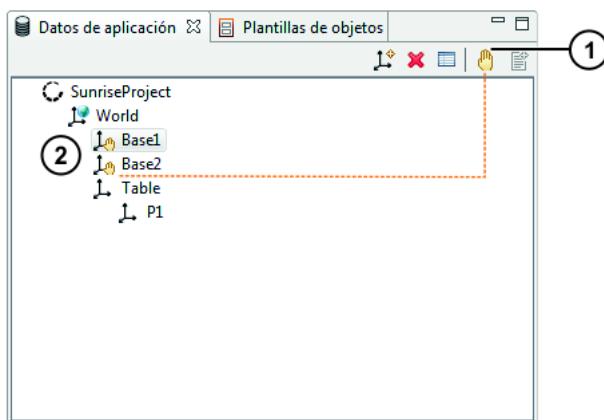


Fig. 9-2: Marcar un Frame como base

- 1 Símbolo en forma de mano **Base**
- 2 La Base1 y Base2 están marcadas como base

9.2.3 Desplazar Frames

Descripción

Un Frame se puede desplazar en la vista **Datos de aplicación** y se puede asignar a un nuevo Frame padre. Observar los siguientes puntos:

- Los Frames subordinados se incluyen automáticamente en el desplazamiento.
- Se conserva la posición absoluta de los Frames desplazados en el espacio. Se adapta la transformación relativa de los Frames con respecto al nuevo Frame padre.
- Los Frames no se pueden insertar debajo de uno de sus subelementos.
- Los nombres de los subelementos directos de un Frame deben ser únicos.



Si se desplaza un Frame, cambiará la ruta. Debido a que los Frames se utilizan a través de esta ruta en el código fuente de las aplicaciones, se deberá corregir la indicación de la ruta de forma correspondiente en las aplicaciones.

Procedimiento

1. Hacer clic en el Frame deseado y mantener pulsado el botón izquierdo del ratón.
2. Arrastrar el Frame con el ratón sobre el nuevo Frame padre.
3. Cuando el nuevo Frame padre esté marcado, soltar el botón del ratón.

9.2.4 Borrar Frames

Descripción Los Frames se pueden eliminar del árbol de Frames en la vista **Datos de aplicación**. Si un Frame posee subelementos, se encuentran disponibles las siguientes opciones para ello:

- **Modificar Frame padre:** Se borra el Frame seleccionado. Los Frames subordinados se conservan, se desplazan un nivel hacia arriba y se asignan a un nuevo Frame padre.

Se conserva la posición absoluta de los Frames desplazados en el espacio. Se adapta la transformación relativa de los Frames con respecto al nuevo Frame padre.



Si se desplaza un Frame, cambiará la ruta. Debido a que los Frames se utilizan a través de esta ruta en el código fuente de las aplicaciones, se deberá corregir la indicación de la ruta de forma correspondiente en las aplicaciones.

- **Borrar Frames padre y Frames hijo:** Borra el Frame seleccionado y todos los Frames subordinados.

Procedimiento

1. Hacer clic con el botón derecho del ratón sobre el Frame que debe borrarse y seleccionar **Borrar** en el menú contextual. Un Frame sin subelementos se borrará inmediatamente.
2. Si el Frame posee subelementos, se preguntará si éstos también se deben borrar. Seleccionar la opción deseada.
3. Sólo con la opción **Modificar Frame padre**: Si al desplazar los subelementos se produce un conflicto de nombre, se mostrará un mensaje de indicación y el proceso de borrado se cancelará.

Ayuda: Renombrar uno de los dos Frames afectados y repetir el proceso de borrado.

9.2.5 Visualizar y modificar las propiedades de un Frame



Generalmente, la posición y la orientación de un Frame se determina durante la programación por aprendizaje con el robot. Sin embargo, es posible introducir manualmente los valores de posición de un Frame o modificarlos posteriormente.

Para ello, es obligatorio observar los siguientes puntos:

- Cualquier modificación de los datos de transformación no solo desplaza el Frame actual, sino también todos los elementos subordinados y afecta a todas las aplicaciones en las que se utilizan estos Frames.
- Los valores memorizados de Estado, Turn y ángulo de redundancia se mantienen. En determinadas circunstancias no es posible avanzar hasta el Frame o sus subelementos.
- Despues de modificar los datos de transformación, deberán comprobarse todos los programas en los que se utiliza el Frame en el modo de servicio Manual velocidad reducida (T1).

Descripción

Los Frames que están marcados como base pueden medirse con el robot.

(>>> 7.2.2 "Medición base: Método de 3 puntos " Página 112)

Si los datos de una base medida se transfieren a Sunrise.Workbench a través de una sincronización, los datos de transformación del Frame se modifican según la medición. Los datos de transformación de los subelementos del Frame no se modifican debido a la medición, es decir, solo se modifica la posición del Frame con respecto al sistema de coordenadas mundial. La información de redundancia tampoco se altera.

Procedimiento

1. Marcar el Frame en la vista **Datos de aplicación**. Las propiedades del Frame se visualizan en la vista **Propiedades**.
2. Si es necesario, mostrar las propiedades ordenadas por categorías mediante el botón  (**Mostrar categorías**).
3. En el campo **Valor** de la propiedad, introducir el nuevo valor y confirmar con la tecla de introducción.



En el caso de magnitudes físicas, el valor puede introducirse con la unidad. Si esta es compatible con la unidad previamente ajustada, el valor se convertirá como corresponde (p. ej. cm en mm o ° en rad). Si no se introduce ninguna unidad, se utilizará la unidad previamente ajustada.

Resumen

Las características ordenadas por categorías incluyen la siguiente información:

Categoría	Descripción
General	Información general <ul style="list-style-type: none"> ■ Nombre: Nombre del Frame ■ Comentario: Opcional ■ Proyecto: Proyecto correspondiente ■ Última modificación: Fecha y hora de la última modificación
Transformación	Datos de transformación <ul style="list-style-type: none"> ■ X, Y, Z: Desplazamiento del Frame referido a su Frame padre ■ A, B, C: Giro del Frame referido a su Frame padre
Redundancia	Información de redundancia <ul style="list-style-type: none"> ■ E1: Valor del ángulo de redundancia ■ Estado ■ Turn
Información de aprendizaje	Información acerca del Frame programado por aprendizaje <ul style="list-style-type: none"> ■ Dispositivo: Robot utilizado ■ Herramienta: Herramienta utilizada ■ TCP: Ruta del Frame hacia el TCP utilizado ■ X, Y, Z: Desplazamiento del TCP en relación al Frame original de la herramienta ■ A, B, C: Giro del TCP en relación al Frame original de la herramienta
Medición	Información relativa a la Medición de base <ul style="list-style-type: none"> ■ Método de medición: Método utilizado ■ Última modificación: Fecha y hora de la última modificación <p>Nota: Si los datos de transformación de una base medida se editan o si el Frame se programa nuevamente por aprendizaje, la información relativa a la medición se elimina.</p>

9.2.6 Insertar un Frame en la instrucción de movimiento

Descripción	Un Frame creado en los datos de aplicación se puede insertar como punto de destino en la instrucción de movimiento.
Procedimiento	<ol style="list-style-type: none">1. Programar una instrucción de movimiento, p. ej. robot.move(ptp())... .2. En la vista Datos de aplicación hacer clic en el Frame que debe utilizarse como punto de destino y mantener pulsado el botón izquierdo del ratón.3. Arrastrar el Frame con el ratón hasta la zona de los editores y posicionarlo allí, de forma que el cursor del ratón esté entre el paréntesis del movimiento.4. Soltar la tecla del ratón. El Frame se inserta como punto de destino del movimiento.

Ejemplo

```
robot.move(ptp(getApplicationContext().getFrame("/P2/Target")));
```

El método **getApplicationContext().getFrame()** muestra que se ha insertado un Frame creado en los datos de aplicación. El punto de destino del movimiento es el Frame **Target**.

El método recibe la ruta del Frame en el árbol de Frame como parámetro de transmisión. El Frame **Target** es un subelemento de **P2**.

9.3 Gestión de objetos

Las herramientas y las piezas se crean y se gestionan en Sunrise.Workbench. Pertenecen a los datos del tiempo de ejecución de un proyecto.

Herramientas	Propiedades: <ul style="list-style-type: none">■ Las herramientas se montan en la brida del robot.■ Las herramientas se pueden utilizar en la aplicación del robot como objetos desplazables.■ Los datos de carga de la herramienta actúan sobre los movimientos del robot.■ Las herramientas pueden poseer el número deseado de puntos de trabajo (TCPs), que se definen como Frames.
piezas	Propiedades: <ul style="list-style-type: none">■ Las piezas pueden ser objetos muy diferentes que se utilizan, se editan o se mueven en el curso de una aplicación de robot.■ Las piezas pueden acoplarse a herramientas u otras piezas.■ Las piezas se pueden utilizar en la aplicación del robot como objetos móviles.■ Los datos de carga de la pieza afectan a los movimientos, p. ej. cuando una garra agarra la pieza.■ Las piezas pueden poseer cualquier número de Frames que identifican puntos relevantes, p. ej. puntos en los que una garra agarra la pieza.

9.3.1 Estructura geométrica de las herramientas

Cada herramienta posee un Frame de origen (Root). El origen de la herramienta está definido por defecto de forma que para el montaje de la herramienta en la brida del robot, coincide en cuanto a la posición y a la orientación con el punto central de la brida. El Frame de origen siempre está disponible y no se debe crear individualmente.

Una herramienta puede tener todos los puntos de trabajo (TCP) que se deseé, que se definen en relación con el Frame de origen de la pieza (Root) o en relación con uno de sus subelementos.

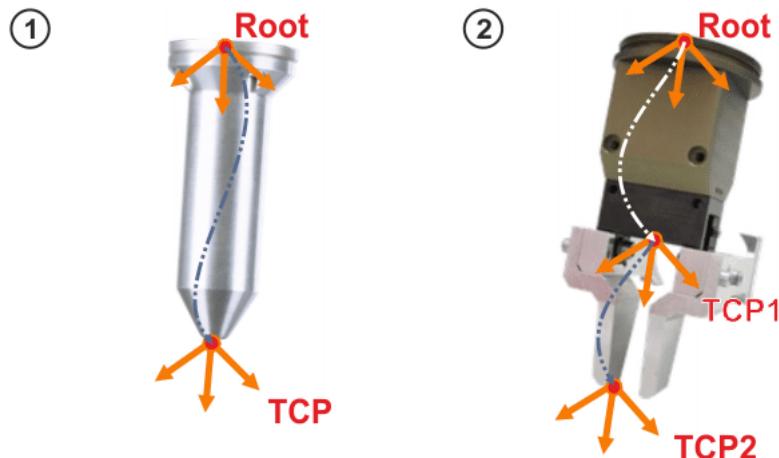


Fig. 9-3: Ejemplos de TCP para herramientas

- | | |
|--|----------------------|
| 1 Punta de guiado manual con 1
TCP | 2 Garra con 2 TCP |
|--|----------------------|

i La transformación del Frame es estática. Esto significa para las herramientas activas, p. ej. garras, que el TCP no se adapta a la posición actual de las mordazas o de los dedos.

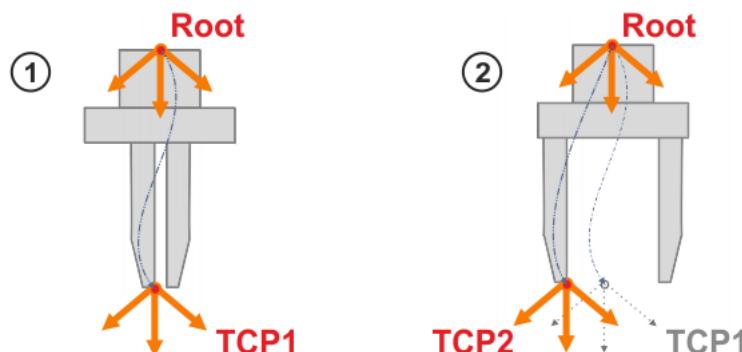


Fig. 9-4: TCP estático en una garra

- | | |
|--------------------|--------------------|
| 1 Garra cerrada | 2 Garra abierta |
|--------------------|--------------------|

9.3.2 Estructura geométrica de las piezas

Cada pieza posee un Frame de origen (Root). El Frame de origen siempre está disponible y no se debe crear individualmente.

Una pieza puede tener todos los Frames que se deseé, que se definen en relación con el Frame de origen de la pieza (Root) o en relación con uno de sus subelementos.

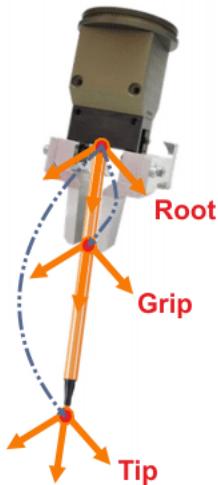


Fig. 9-5: Ejemplos de Frames de piezas

9.3.3 Crear herramienta o pieza

Descripción Las herramientas y las piezas creadas en Sunrise.Workbench son específicas del proyecto y se pueden utilizar en cualquier aplicación del robot del proyecto.

Las herramientas creadas se pueden seleccionar en las opciones de desplazamiento manual después de la sincronización del proyecto en la smartHMI.

(>>> 6.8.1 "Ventana Opciones de desplazamiento manual" Página 78)

Procedimiento

1. Marcar el proyecto en el **Explorador de paquetes**.
2. Abrir en la vista **Plantillas de objetos** la lista con las plantillas de objetos.
3. Para crear una herramienta, hacer clic con el botón derecho del ratón sobre el tipo de objeto **Herramientas** y seleccionar en el menú contextual **Crear nueva herramienta**. Se crea la plantilla de objetos para la herramienta.
4. Para crear una pieza, hacer clic con el botón derecho del ratón sobre el tipo de objeto **Piezas** y seleccionar en el menú contextual **Crear nueva pieza**. Se crea la plantilla de objetos para la pieza.
5. El sistema asigna automáticamente un nombre del objeto. Se recomienda modificar el nombre en la vista **Propiedades**.
Los nombres de los objetos deben ser únicos. Un nombre descriptivo facilita la programación y la orientación en el programa.
6. Introducir los datos de carga en la vista **Propiedades**.
(>>> 9.3.6 "Datos de carga" Página 149)

9.3.4 Crear Frame para herramienta o pieza

Descripción Cada uno de los Frames creados para una herramienta o una pieza de trabajo se puede programar en la aplicación del robot como punto de referencia para los movimientos.

Los Frames de una herramienta se pueden seleccionar como TCP para el desplazamiento manual cartesiano, después de la sincronización del proyecto en la smartHMI.

(>>> 6.8.1 "Ventana Opciones de desplazamiento manual" Página 78)

Los Frames de una herramienta (TCP) se pueden medir con el robot con referencia al sistema de coordenadas de la brida.

(>>> 7.2.1 "Medir la herramienta" Página 106)

Si los datos de una herramienta medida se transfieren a Sunrise.Workbench a través de una sincronización, los datos de transformación del Frame se modifican según la medición.



Los datos de herramienta del TCP con el que se ejecuta un movimiento cartesiano influyen en la velocidad del robot. En caso de datos de herramienta introducidos de forma incorrecta, pueden producirse velocidades cartesianas inesperadamente altas en la herramienta montada. Es posible la superación de la velocidad de 250 mm/s en el modo de servicio T1.

Procedimiento

1. Marcar el proyecto en el **Explorador de paquetes**.
2. Abrir en la vista **Plantillas de objetos** la lista con las plantillas de objetos.
3. Hacer clic con el botón derecho del ratón sobre la plantilla del objeto y seleccionar **Crear un nuevo Frame** en el menú contextual. El Frame se crea.
En el nivel superior de jerarquía, el Frame padre del Frame creado es el Frame de origen del objeto.
4. Para insertar un Frame nuevo debajo de un Frame ya existente del objeto, hacer clic con el botón derecho del ratón sobre este Frame padre y seleccionar **Crear un nuevo Frame** en el menú contextual. El Frame se crea.
5. El sistema asigna automáticamente un nombre de Frame. Se recomienda modificar el nombre en la vista **Propiedades**.
Un nombre de Frame descriptivo facilita la programación y la orientación en el programa. Los nombres de Frame deben ser únicos dentro un nivel jerárquico y no se deben asignar varias veces.
6. En la vista **Propiedades** introducir los datos de transformación del Frame referidos a su Frame padre:
 - Campos **X, Y, Z**: Desplazamiento del Frame a lo largo de los ejes del Frame padre
 - Campos **A, B, C**: Orientación del Frame referido al Frame padre

Propiedades

Las características ordenadas por categorías incluyen la siguiente información:

Categoría	Descripción
General	Información general <ul style="list-style-type: none"> ■ Nombre: Nombre del Frame ■ Comentario: Opcional ■ Orientado a la seguridad: Solo es relevante en el caso de herramientas destinadas a la seguridad <ul style="list-style-type: none"> ■ Sí: El Frame es un Frame destinado a la seguridad ■ No: El Frame no es un Frame destinado a la seguridad (>>> 9.3.7 "Herramienta destinada a la seguridad" Página 150)
Geometría	Radio de la bola en el Frame destinado a la seguridad <ul style="list-style-type: none"> ■ Radio

Categoría	Descripción
Transformación	<p>Datos de transformación</p> <ul style="list-style-type: none"> ■ X, Y, Z: Desplazamiento del Frame referido a su Frame padre ■ A, B, C: Giro del Frame referido a su Frame padre
Medición	<p>Información relativa a la medición de herramienta</p> <ul style="list-style-type: none"> ■ Método de medición: Método utilizado ■ Error de cálculo: Error de cálculo translatorio que indica la calidad de la medición (unidad: mm) ■ Última modificación: Fecha y hora de la última modificación <p>Nota: Si se editan datos de transformación de una herramienta medida, la información relativa a la medición se borra.</p>

9.3.5 Establecer Frame estándar para movimientos

Descripción	<p>Si una herramienta o una pieza dispone de un Frame con el que deben ejecutarse la mayoría de los movimientos, dicho Frame se podrá establecer como Frame estándar para movimientos.</p> <p>Si para una herramienta o una pieza se encuentra establecido un Frame estándar adecuado para movimientos, se facilita la programación de movimientos.</p> <p>(>>> 15.11.4 "Desplazar herramientas y piezas" Página 297)</p> <p>Si no se establece ningún Frame estándar, se utilizará automáticamente el Frame de origen de la herramienta o de la pieza como Frame estándar para movimientos.</p>
Procedimiento	<ol style="list-style-type: none"> 1. Marcar el proyecto en el Explorador de paquetes. 2. En la vista Plantillas de objetos seleccionar el tipo de objeto Herramientas o Piezas. 3. Seleccionar la herramienta o la pieza deseada. 4. Hacer clic con el botón derecho del ratón sobre el Frame deseado y seleccionar en el menú contextual Frame estándar para movimientos. <p>Alternativa:</p> <p>Marcar el Frame y hacer clic sobre el símbolo Frame estándar para movimientos.</p> <p>El Frame se identifica como Frame estándar para movimientos.</p>

Ejemplo

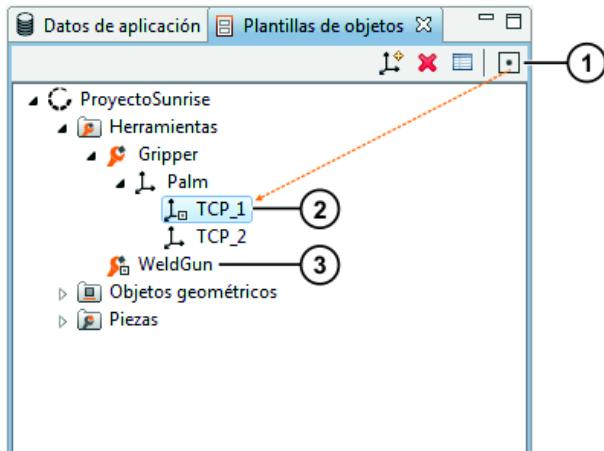


Fig. 9-6: Frame estándar para movimientos

- 1 Símbolo **Frame estándar para movimientos**
- 2 Frame estándar de la herramienta **Gripper**: TCP_1
- 3 Frame estándar de la herramienta **WeldGun**: Frame de origen

9.3.6 Datos de carga

Los datos de carga son todas las cargas montadas en la brida del robot o las cargas conectadas a la misma. Forman una masa adicional sobre el robot que debe moverse con él.

Los datos de carga de las herramientas y piezas deben indicarse al generar las plantillas de objetos correspondientes. Si hay varias herramientas y piezas conectadas al robot, a partir de los datos de carga se calcula automáticamente la carga total resultante.

Los datos de carga entran en el cálculo de las trayectorias y las aceleraciones. Que los datos de carga sean correctos es un requisito importante para el funcionamiento óptimo de la regulación y contribuyen a la optimización de los tiempos de ciclo.



ADVERTENCIA No operar el robot con datos de carga incorrectos o cargas inadecuadas. Si no se respeta esta advertencia, pueden ocurrir daños materiales o lesiones graves. Por ejemplo, porque el frenado del robot tarda demasiado debido a que los datos de carga son incorrectos.

Fuentes

Los datos de carga pueden ser consultados en las siguientes fuentes:

- Datos del fabricante
- Cálculo manual
- Programas CAD
- Los datos de carga de las herramientas pueden calcularse automáticamente.

(>>> 7.3 "Determinar datos de carga de la herramienta" Página 114)

9.3.6.1 Introducir los datos de carga

Procedimiento

1. Marcar el proyecto Sunrise en el **Explorador de paquetes**.
2. Abrir en la vista **Plantillas de objetos** la lista con las plantillas de objetos.
3. Marcar la pieza o la herramienta deseada.
4. Introducir los datos de carga en la vista **Propiedades**:

- **Masa:** Masa del objeto (herramienta o pieza de trabajo)
- Campos **MS X, MS Y, MS Z:** Posición del centro de gravedad de la masa en relación con el Frame de origen del objeto
- Campos **MS A, MS B, MS C:** Orientación de los ejes principales de inercia relativos al Frame de origen del objeto. Los ejes de inercia principales pasan por el centro de gravedad de masa.
- Campos **jX, jY, jZ:** Momentos de inercia principales

Ejemplo

Momento de inercia principal **jX**:

jX es la inercia alrededor del eje X de los ejes de inercia principales. Debido a **MS A, MS B** y **MS C**, este momento está girado hacia el Frame de origen del objeto de forma relativa y desplazado en el centro de gravedad de la masa.

Análogamente, **jY** y **jZ** son los momentos de inercia principales alrededor de los ejes Y y Z.

9.3.7 Herramienta destinada a la seguridad

Descripción

En el proyecto Sunrise se puede establecer como máximo 1 herramienta destinada a la seguridad que se modela con hasta 6 bolas configurables.

Las propiedades de la herramienta destinada a la seguridad son relevantes para las siguientes funciones de seguridad configurables:

- Control de zonas cartesianas

Las bolas se pueden controlar en relación con los límites de las zonas de control cartesianas activadas.

(>>> 13.8.9 "Zonas de control" Página 208)

- Control de la velocidad cartesiana translatoria

Se controla la velocidad de los puntos centrales de las bolas.

(>>> 13.8.8 "Controles de velocidad" Página 207)

- Detección de colisión y control de fuerza TCP

La precisión de estos controles solo está garantizada con datos de carga indicados correctamente. Deben configurarse los datos de carga de la herramienta destinada a la seguridad, especialmente la masa de la herramienta y el centro de gravedad de la masa. En las herramientas con momentos de inercia comparativamente elevados ($> 0,1 \text{ kg}^{\cdot}\text{m}^2$) también se deben indicar estos datos para garantizar la precisión de estos controles.

Una herramienta destinada a la seguridad puede poseer el número de Frames que se desee, al igual que cualquier herramienta. Para configurar las bolas de control, se deberán establecer Frames adecuados como Frames destinados a la seguridad. El origen de un Frame destinado a la seguridad es el punto central de la bola. El radio de la bola se define en las propiedades del Frame.

Uno de los Frames destinados a la seguridad se define como Frame de orientación de herramienta. La orientación de estos Frames se puede controlar de forma segura a través de la AMF *Orientación de la herramienta* en cuanto a límites admisibles.

(>>> 13.8.10 "Control de la orientación de herramienta" Página 217)

La herramienta destinada a la seguridad se transmite a la unidad de control del robot por sincronización y se activa después del reinicio de la unidad de control del robot. Es decir, está permanentemente activa, independientemente de la herramienta que se utilice en la aplicación o que esté ajustada en las opciones de desplazamiento manual.

Ejemplo

Para una garra destinada a la seguridad se encuentran configuradas 3 bolas de control.

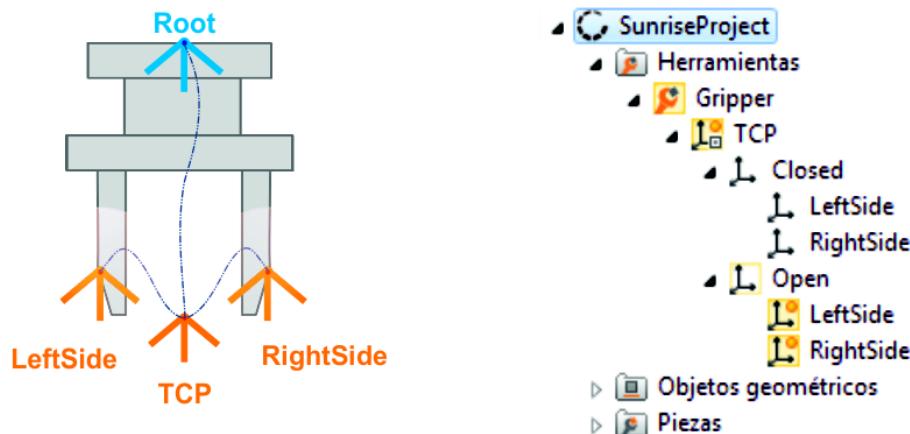


Fig. 9-7: Garra orientada a la seguridad

9.3.7.1 Definir la herramienta destinada a la seguridad

Condición previa

- La herramienta y los Frames correspondientes están creados.
- En caso de emplear los AMF *Identificación de colisión* y *Control de fuerza TCP*: Se conocen los datos de carga correctos de la herramienta, especialmente la masa de la herramienta y el centro de gravedad de la masa.

Procedimiento

1. Marcar el proyecto en el **Explorador de paquetes**.
2. En la vista **Plantillas de objetos**, marcar la herramienta que debe estar destinada a la seguridad.
3. En la vista **Propiedades**, establecer las propiedades de la herramienta destinada a la seguridad.
 - Si aún no están establecidos, introducir los datos de carga de la herramienta.
Las herramientas con datos de carga fuera de la gama de valores predefinida no pueden utilizarse como herramientas destinadas a la seguridad.
(>>> "Datos de carga" Página 152)
 - Ajustar a **Sí** la propiedad **Orientado a la seguridad**.
El fondo del icono de la herramienta de la vista **Plantillas de objetos** se pone amarillo.
Las características de herramienta se amplían con el parámetros de seguridad **Frame de orientación de la herramienta**. El Frame de origen de la herramienta (sistema de coordenadas de la brida) esta ajustado por defecto como Frame de orientación de herramienta.
4. En la vista **Plantillas de objetos**, marcar el Frame que debe estar destinado a la seguridad.
5. En la vista **Propiedades**, establecer las propiedades del Frame destinado a la seguridad.
 - Introducir el radio de la bola de control.
 - Si aún no se han establecido, introducir los datos de transformación del Frame referidos a su Frame padre.
Los Frames con datos de transformación fuera de la gama de valores predefinida no pueden utilizarse como Frames destinados a la seguridad.
(>>> "Frames" Página 152)

■ Ajustar a **Sí** la propiedad **Orientado a la seguridad**.

El fondo del ícono del Frame en la vista **Plantillas de objetos** se pone en amarillo y se marca con un símbolo de bola.

6. Repetir el paso 4 a 5 para definir otros Frames destinados a la seguridad.
7. Si se desea emplear otro Frame distinto al Frame de origen de la herramienta destinada a la seguridad como Frame de orientación de herramienta:
 - a. En la vista **Plantillas de objetos**, marcar la herramienta destinada a la seguridad.
 - b. Seleccionar el Frame deseado en la vista **Propiedades en Parámetro de seguridad > Frame de orientación de la herramienta**. Solo se pueden elegir los Frames destinados a la seguridad de la herramienta.

El ícono de Frame del Frame de orientación de herramienta en la vista **Plantillas de objetos**, se amplía con una flecha.



Procedimiento alternativo para marcar una herramienta o un Frame como destinado a la seguridad:

- En la vista **Plantillas de objetos**, hacer clic con el botón derecho del ratón en la herramienta o el Frame y seleccionar en el menú contextual **Orientado a la seguridad**.

Procedimiento alternativo para ajustar un Frame como Frame de orientación de herramienta:

- En la vista **Plantillas de objetos**, hacer clic con el botón derecho del ratón en el Frame destinado a la seguridad deseado y seleccionar en el menú contextual **Frame de orientación de la herramienta**.

Datos de carga

Datos de carga de la herramienta destinada a la seguridad:

Parámetros	Descripción
Masa	Masa de la herramienta destinada a la seguridad <ul style="list-style-type: none"> ■ ≤2000 kg
MS X, MS Y, MS Z	Posición del centro de gravedad de la masa en relación con el Frame de origen de la herramienta destinada a la seguridad <ul style="list-style-type: none"> ■ -10 000 mm ... +10 000 mm
MS A, MS B, MS C	Orientación de los ejes principales de inercia en relación con el Frame de origen de la herramienta destinada a la seguridad <ul style="list-style-type: none"> ■ cualquiera
jX, jY, jZ	Momentos de inercia de la masa de la herramienta destinada a la seguridad <ul style="list-style-type: none"> ■ 0 kg*m²... 1000 kg*m²



Para evitar un trabajo de verificación innecesario, se recomienda marcar una herramienta como destinada a la seguridad solamente si los datos de carga se han introducido o determinado correctamente y se han transmitido a Sunrise.Workbench.

Aquí hay más información sobre los datos de carga: ([>>> 9.3.6 "Datos de carga"](#) Página 149)

Frames

Propiedades de Frames orientados a la seguridad:

Parámetro	Descripción
Radio	Radio de la bola en el Frame orientado a la seguridad ■ 25 mm ... 10 000 mm
X, Y, Z	Desplazamiento del Frame orientado a la seguridad a lo largo de los ejes del Frame padre ■ -10 000 mm ... +10 000 mm
A, B, C	Orientación del Frame orientado a la seguridad referido al Frame padre ■ cualquiera

9.3.8 Piezas de trabajo destinadas a la seguridad

- Descripción** Las cargas soportadas por el robot, p. ej. un a pieza de trabajo sujetada, ejercen una fuerza adicional sobre el robot y afectan a los momentos medidos por los sensores de momentos de articulación. Por ello, los datos de carga de piezas de trabajo sujetadas deben tenerse en cuenta en el control seguro de colisiones y fuerzas. Para ello, es necesario configurar estas piezas de trabajo como piezas de trabajo destinadas a la seguridad.
- Para un proyecto Sunrise se pueden configurar un máximo de 8 piezas de trabajo destinadas a la seguridad.
- Las propiedades de la herramienta destinada a la seguridad son relevantes para las siguientes funciones de seguridad configurables:
- Detección de colisión

Los datos de carga de la pieza de trabajo destinada a la seguridad activa se tienen en cuenta a la hora de calcular el momento externo.
 (»> 13.8.13.2 "Detección de colisión" Página 221)
 - Control de fuerza TCP

Los datos de carga de la pieza de carga destinada a la seguridad más pesada se tienen en cuenta en la determinación de la fuerza cartesiana que actúa debajo de la brida.
 (»> 13.8.13.3 "Control de fuerza TCP" Página 222)
- Durante un proceso pueden surgir cambios de carga debido a que el robot coge y suelta distintas piezas de trabajo. En el control de fuerza TCP, el control de seguridad emplea automáticamente los datos de carga de la pieza de trabajo destinada a la seguridad más pesada. En la detección de colisión, el usuario debe comunicarle explícitamente al control de seguridad qué pieza de trabajo destinada a la seguridad está actualmente activa.
- (»> 15.11.5 "Ordenar cambios de carga en el control de seguridad" Página 298)
- Si una pieza de trabajo está activada, el control de seguridad tiene en cuenta sus datos de carga de forma permanente. Si se desea desactivar esta pieza de trabajo o activar otra pieza de trabajo, hay que dar un comando explícito al respecto. Si se reinicia la unidad de control del robot ya no hay ninguna pieza de trabajo destinada a la seguridad activada.
- La activación de piezas de trabajo destinadas a la seguridad no se realiza de forma destinada a la seguridad en el código fuente de aplicaciones robot y tareas en segundo plano. Por ello, en caso de error es posible que la detección de colisión y el control de fuerza TCP empleen datos de carga que difieran de la carga real de las piezas de trabajo. Estas divergencias se malinterpretan como momento externo o fuerza TCP externa. Con velocidades y aceleracio-

nes reducidas, la divergencia máxima se corresponde con el peso de la pieza de trabajo más pesada que se puede coger en la aplicación.



Al emplear las AMF *Identificación de colisión y Control de fuerza TCP* se recomienda configurar como piezas de trabajo destinadas a la seguridad a todas las piezas de trabajo que el robot coge mientras uno de los controles esté activo.



Al emplear la AMF *Control de fuerza TCP* es necesario configurar como pieza de trabajo destinada a la seguridad a la pieza de trabajo más pesada que el robot coge mientras el control está activo. Una configuración errónea de la pieza de trabajo más pesada puede llevar a la pérdida de la integridad de seguridad del control de fuerza TCP.

La manera en la que los datos de carga de una pieza de trabajo afectan al control de colisión depende de cómo se coge la pieza de trabajo. Con una pieza de trabajo destinada a la seguridad el control de seguridad da por hecho que el Frame de origen de la pieza de trabajo destinada a la seguridad es idéntico al frame estándar para movimientos de la herramienta destinada a la seguridad.

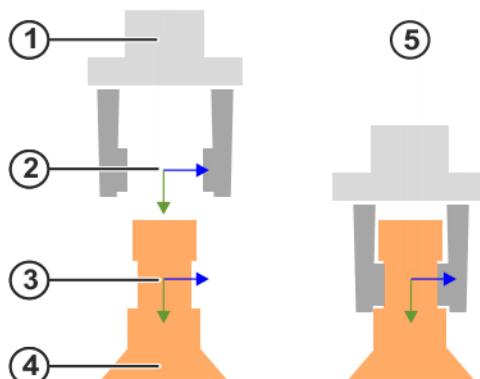


Fig. 9-8: Configuración de la pieza de trabajo destinada a la seguridad

Pos.	Descripción
1	Herramienta destinada a la seguridad
2	Frame estándar para movimientos de la herramienta destinada a la seguridad Frame de la herramienta destinada a la seguridad en el que se debe coger la pieza de trabajo destinada a la seguridad. Este Frame no tiene por qué ser un Frame destinado a la seguridad.
3	Frame origen de la pieza de trabajo destinada a la seguridad Frame de la pieza de trabajo destinada a la seguridad en el que la herramienta destinada a la seguridad debe coger la pieza de trabajo.
4	Pieza de trabajo destinada a la seguridad
5	Estado después de activar la pieza de trabajo destinada a la seguridad El Frame origen de la pieza de trabajo destinada a la seguridad es idéntico al Frame estándar para movimientos de la herramienta destinada a la seguridad.

9.3.8.1 Definir piezas de trabajo destinadas a la seguridad

Condición previa ■ La pieza de trabajo está creada.

- En caso de emplear los AMF *Identificación de colisión y Control de fuerza TCP*: Se conocen los datos de carga correctos de la pieza de trabajo, especialmente la masa de la pieza de trabajo y el centro de gravedad de la masa.

Procedimiento

1. Marcar el proyecto en el **Explorador de paquetes**.
2. En la vista **Plantillas de objetos**, marcar la pieza de trabajo que debe estar destinada a la seguridad.
3. En la vista **Propiedades**, establecer las propiedades de la pieza de trabajo destinada a la seguridad.
 - Si aún no están establecidas, introducir los datos de carga de la pieza de trabajo.
Las piezas de trabajo con datos de carga fuera de la gama de valores predefinida no pueden utilizarse como piezas de trabajo destinadas a la seguridad.
(>>> "Datos de carga" Página 155)
 - Ajustar a **Orientado a la seguridad Sí** la propiedad.
El fondo del ícono de la herramienta de la vista **Plantillas de objetos** se pone amarillo.

Después de la sincronización del proyecto se puede activar la pieza de trabajo destinada a la seguridad.



Procedimiento alternativo para marcar una pieza de trabajo como destinada a la seguridad:

- Hacer clic con el botón derecho sobre la pieza de trabajo en la vista **Plantillas de objetos** y seleccionar **Orientado a la seguridad** en el menú contextual.

Datos de carga

Datos de carga de la pieza de trabajo destinada a la seguridad:

Parámetros	Descripción
Masa	Masa de la pieza de trabajo destinada a la seguridad <ul style="list-style-type: none"> ■ 0,001 kg ... 2000 kg
MS X, MS Y, MS Z	Posición del centro de gravedad de la masa en relación con el Frame de origen de la pieza de trabajo destinada a la seguridad <ul style="list-style-type: none"> ■ -10 000 mm ... +10 000 mm
MS A, MS B, MS C	Orientación de los ejes de inercia principales en relación con el Frame de origen de la pieza de trabajo destinada a la seguridad (no es relevante para el control destinado a la seguridad) <ul style="list-style-type: none"> ■ cualquiera
jX, jY, jZ	Momentos de inercia de masa de la pieza de trabajo destinada a la seguridad (no es relevante para el control destinado a la seguridad) <ul style="list-style-type: none"> ■ cualquiera

Aquí hay más información sobre los datos de carga: (>>> 9.3.6 "Datos de carga" Página 149)

9.4 Sincronización de proyectos

Resumen

Para la sincronización de proyectos se transmiten los datos de proyecto entre Sunrise.Workbench y la unidad de control del robot. Para ello, los proyectos

se comparan entre sí. Si existen diferentes proyectos o diferencias entre versiones, el usuario podrá seleccionar la dirección en la que deben transmitirse los datos de proyecto. Se diferencian los siguientes casos:

- El proyecto solo está disponible en Sunrise.Workbench.
- El proyecto en la unidad de control del robot se sustituye por otro proyecto.
- Existen diferentes versiones del proyecto:
 - En caso de modificación de los datos de proyecto solo en Sunrise.Workbench
 - En caso de modificación de los datos de proyecto solo en la unidad de control del robot
 - En caso de modificación de los datos de proyecto en ambas partes

9.4.1 Transferir un proyecto a la unidad de control del robot

Descripción	El procedimiento descrito aquí se aplica para el caso en que aún no haya ningún proyecto en la unidad de control del robot, o haya un proyecto distinto al que debe transmitirse.
Condición previa	<ul style="list-style-type: none"> ■ Conexión de red a la unidad de control de robot ■ El proyecto que ha de transferirse contiene al menos una aplicación. ■ El software de sistema está instalado. (>>> 10.2 "Instalar el System Software" Página 160) ■ El software de sistema instalado es compatible con la configuración de la estación del proyecto que se ha de transferir.
Procedimiento	<p>El procedimiento descrito aquí se aplica para el caso en que aún no haya ningún proyecto en la unidad de control del robot, o haya un proyecto distinto al que debe transmitirse.</p> <ol style="list-style-type: none"> 1. Marcar el proyecto que debe transmitirse en el Explorador de paquetes. 2. En la barra de herramientas hacer clic en el botón Sincronizar proyecto. Se consulta si ya existen datos del proyecto en el control. Si falla la consulta, se indicará la causa del error en un mensaje. 3. Si la consulta es correcta, se abre la ventana Sincronización del proyecto.



Fig. 9-9: Transmitir un proyecto a la unidad de control

4. Hacer clic en **Ejecutar**. El proyecto se transmite a la unidad de control del robot.

El progreso se indica tanto en Sunrise.Workbench como en el smartPAD en un diálogo. Cuando la transferencia haya finalizado, el diálogo se cierra automáticamente.

5. Si la transferencia falla, se indica un diálogo correspondiente tanto en Sunrise.Workbench como en el smartPAD. En Sunrise.Workbench se indica adicionalmente la causa del error.
Confirmar el diálogo en Sunrise.Workbench y en el smartPAD con **OK**.
6. Si la transferencia se realiza satisfactoriamente, en caso de modificación de la configuración de seguridad o de la configuración de E/S, se requiere el reinicio de la unidad de control del robot.
Confirmar el requerimiento para el reinicio pulsando **OK**. La unidad de control del robot se reinicia.
7. En caso de modificación de la configuración de seguridad, activar esta modificación en la unidad de control del robot.
(>>> 13.7 "Activar la configuración de seguridad en la unidad de control del robot" Página 200)

El proyecto transmitido está ahora activo en el control. Están disponibles todos los datos de proyecto creados.

9.4.2 Actualizar el proyecto en la unidad de control del robot o en Sunrise.Workbench

Descripción	El procedimiento descrito aquí sirve para el caso de que en ambas partes exista el mismo proyecto, aunque con versiones diferentes.
Condición previa	<ul style="list-style-type: none"> ■ Conexión de red a la unidad de control de robot ■ Solo si se transfiere un proyecto a la unidad de control del robot: En la unidad de control del robot no se está ejecutando ninguna aplicación.
Procedimiento	<ol style="list-style-type: none"> 1. Marcar el proyecto que debe transmitirse en el Explorador de paquetes. 2. En la barra de herramientas hacer clic en el botón Sincronizar proyecto. Se consulta si ya existen datos del proyecto en el control. Si falla la consulta, se indicará la causa del error en un mensaje. 3. Si el proyecto en Sunrise.Workbench es idéntico al proyecto del control, no es necesaria ninguna sincronización. El proceso se cancela automáticamente. <p>Si la consulta es correcta, se abre la ventana Sincronización del proyecto.</p>



Fig. 9-10: Actualizar proyecto

Se muestra información sobre los dos proyectos. La dirección en la que se realiza la sincronización está ajustada por defecto, de forma que se transmita la versión actual del proyecto.

Si se realizaron modificaciones en los datos del proyecto en ambas partes, se detectará como conflicto y se mostrará una indicación de advertencia. La dirección en la que se realiza la sincronización se puede ajustar:

- El símbolo de confirmación está puesto en **Transmitiendo al control**: El proyecto se transmite desde Sunrise.Workbench a la unidad de control del robot.

- El símbolo de confirmación está puesto en **Transmitiendo al proyecto local**: El proyecto se transmite desde la unidad de control del robot a Sunrise.Workbench.
4. Si necesario, modificar la dirección en la que se ha de sincronizar y hacer clic sobre **Ejecutar**. Se transmite el proyecto. En caso de transmisión de la versión anterior, se mostrará una indicación de advertencia.
El progreso se indica tanto en Sunrise.Workbench como en el smartPAD en un diálogo. Cuando la transferencia haya finalizado, el diálogo se cierra automáticamente.
5. Si la transferencia falla, se indica un diálogo correspondiente tanto en Sunrise.Workbench como en el smartPAD. En Sunrise.Workbench se indica adicionalmente la causa del error.
Confirmar el diálogo en Sunrise.Workbench y en el smartPAD con **OK**.
6. Solo en caso de transmisión satisfactoria a la unidad de control del robot:
- En caso de modificación de la configuración de seguridad o de la configuración de E/S, se requiere el reinicio de la unidad de control del robot.
Confirmar el requerimiento para el reinicio pulsando **OK**. La unidad de control del robot se reinicia.
 - En caso de modificación de la configuración de seguridad, activar esta modificación en la unidad de control del robot.
(>>> 13.7 "Activar la configuración de seguridad en la unidad de control del robot" Página 200)

9.5 Cargar el proyecto desde la unidad de control del robot

Descripción	Es posible cargar desde la unidad de control del robot un proyecto que se no se encuentre en la zona de trabajo de Sunrise.Workbench.
Condición previa	<ul style="list-style-type: none">■ Conexión de red a la unidad de control de robot■ En la zona de trabajo no existe ningún proyecto con el mismo nombre que el proyecto que se va a cargar.
Procedimiento	<ol style="list-style-type: none">1. Seleccionar la secuencia de menú Archivo > Nuevo > Proyecto Sunrise. Se abre el asistente para la creación de proyectos.2. En el campo Dirección IP del control introducir la dirección IP de la unidad de control del robot desde la que debe cargarse el proyecto. <div style="border: 1px solid black; padding: 10px; margin-top: 10px;"> La dirección IP de la unidad de control del robot se puede indicar en el smartHMI. (>>> 6.16.6 "Mostrar la dirección IP y la versión de software" Página 102)</div> <ol style="list-style-type: none">3. Activar el botón de radio Cargar proyecto del control.4. Hacer clic en Siguiente. Se comprueba si existe un proyecto en el control.5. Si existe un proyecto en el control y en la zona de trabajo no existe ningún proyecto con el mismo nombre, se visualizará un resumen con información sobre el proyecto. Hacer clic en Finalizar. El proyecto se crea en la zona de trabajo y, a continuación, se visualiza en el Explorador de paquetes.

10 Configuración de la estación e instalación

10.1 Abrir la configuración de la estación

Procedimiento

- En el **Explorador de paquetes** hacer doble clic en el archivo **StationSetup.cat**.

El archivo contiene la configuración de la estación de proyecto. Ésta se puede editar e instalar mediante las siguientes pestañas:

Topología

La pestaña **Topología** muestra los componentes de hardware de la estación. La topología se puede crear de nuevo o modificarse.

Software

La pestaña **Software** muestra los componentes de software de la estación. Se pueden seleccionar los componentes a instalar y su versión. Los componentes seleccionables dependen de la topología creada.

Configuración

La pestaña **Configuración** muestra la configuración de la unidad de control del robot. La configuración se puede modificar.

Instalación

En la pestaña **Instalación** se instala el software de sistema en la unidad de control del robot.

10.1.1 Configurar parámetros para medición

Procedimiento

1. Abrir la configuración de estación y seleccionar la pestaña **Configuración**.
2. En el elemento de catálogo **SmartHMI** en **Medición** configurar los parámetros según se desee.
3. Guardar la configuración de estación. Se pregunta si las modificaciones se deben aplicar al proyecto. Hacer clic en **Guardar y aceptar**.

Resumen

Parámetros	Descripción
Distancia mínima del punto de medición (herramienta) [mm]	Distancia mínima que debe existir entre dos puntos de medición para la medición de la herramienta (método XYZ de 4 puntos y ABC de 2 puntos) <ul style="list-style-type: none"> ■ 0 ... 200 <p>Por defecto: 8</p>
Error de cálculo máximo [mm]	Error de cálculo translatorio máximo durante la medición de la herramienta hasta el que la calidad de la medición aún se considera suficiente. <ul style="list-style-type: none"> ■ 0 ... 200 <p>Por defecto: 50</p>
Distancia mínima del punto de medición (base) [mm]	Distancia mínima que debe existir entre 2 puntos de medición durante la medición de la base <ul style="list-style-type: none"> ■ 0 ... 200 <p>Por defecto: 50</p>
Ángulo mínimo en °	Ángulo mínimo entre las rectas que están definidas por los 3 puntos de medición durante la medición de la base (método de 3 puntos) y que debe respetarse en todo momento. <ul style="list-style-type: none"> ■ 0 ... 360 <p>Por defecto: 2.5</p>

10.2 Instalar el System Software

Condición previa

- La configuración de la estación ha finalizado.
- Conexión de red a la unidad de control de robot

Procedimiento

1. Seleccionar la pestaña **Instalación**.
2. En la ventana **Eventos de instalación** se muestran por defecto las advertencias y los errores que se producen durante la instalación: El símbolo de confirmación de **Mostrar solo advertencias y errores**. está puesto. Para indicar todos los eventos que surgen durante la instalación, retirar el símbolo de confirmación de **Mostrar solo advertencias y errores**.
3. Hacer clic en **Instalar**. Se prepara la instalación y se abre la ventana **Instalación**. El campo **IP configurada** está marcado en color:
 - Marcado en verde: La conexión de red con la unidad de control del robot se ha establecido y la instalación es posible.
 - Marcado en rojo: La conexión de red con la unidad de control del robot no se ha podido establecer. Las causas pueden ser que el cable de red no está conectado correctamente o que la dirección IP configurada no coincide con la dirección IP real de la unidad de control del robot.

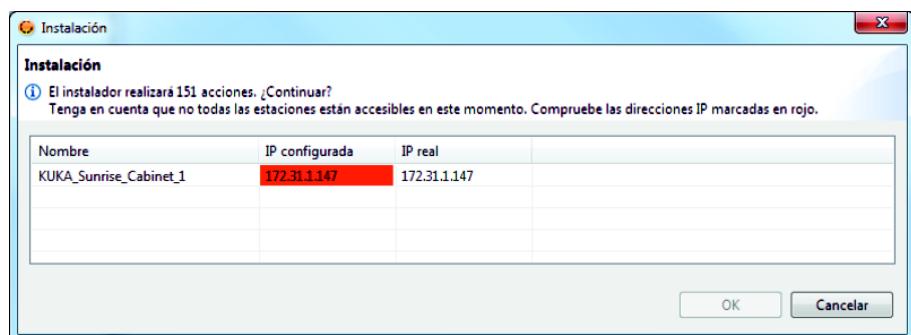


Fig. 10-1: No hay conexión con la unidad de control del robot

4. Solo si el campo **IP configurada** está marcado en rojo:
 - Si la dirección IP configurada coincide con la dirección IP real de la unidad de control del robot, no existirá ninguna conexión de red con la unidad de control del robot. Establecer la conexión de red.
 - Si la unidad de control del robot dispone de una dirección IP distinta a la configurada, se deberá introducir la dirección IP de la unidad de control del robot en el campo **IP real**. Para ello, hacer doble clic en el campo.
5. Hacer clic en **OK** para continuar la instalación.
6. Solo es relevante si se ha modificado la dirección IP en el campo **IP real**: Si se conserva la marca roja en **IP configurada** o falla la instalación, no existe ninguna conexión de red con la unidad de control del robot con esta dirección IP. Establecer la conexión de red y reiniciar el proceso de instalación (volver al paso 3).
7. Confirmar el mensaje con la solicitud de reinicio pulsando **OK**. La unidad de control del robot se reinicia y la instalación finaliza.



En la instalación se transmite la configuración de seguridad a la unidad de control del robot, pero aún no está activa. Para poder desplazar el robot se debe activar la configuración de seguridad a través del smartHMI. ([>>> 13.7 "Activar la configuración de seguridad en la unidad de control del robot"](#) Página 200)



El tipo de robot seleccionado en la configuración de la estación del proyecto Sunrise y la brida para medios ajusta se guardan durante la instalación en los datos de configuración en la unidad de control del robot. Si estos datos no coinciden con los datos correspondientes de la placa características electrónica del robot conectado a la unidad de control del robot, el robot no se puede desplazar.

Nueva instalación

En los siguientes casos es necesario instalar nuevamente el software de sistema:

- Modificación de la configuración de estación en la pestaña **Topología**
- Modificación de la configuración de estación en la pestaña **Software**

Ejemplos:

- Instalación de paquetes adicionales de software
- Actualización del software
- Modificaciones no compatibles de versiones de paquetes de software ya disponibles

Pueden surgir modificaciones no compatibles de versiones si un proyecto se abre con una versión de Sunrise.Workbench distinta a la versión con la que fue creado.

(>>> 15.6 "Información sobre la versión de RoboticsAPI" Página 280)

- Modificación de la configuración de estación en la pestaña **Configuración**

10.2.1 Convertir la configuración de seguridad a una versión nueva de software

Descripción

Si se instala una nueva versión de software de Sunrise.Workbench, se puede cargar en la zona de trabajo un proyecto Sunrise que se ha creado con la versión de software antigua y se puede seguir empleando.

Al cargar el proyecto Sunrise se modifica la configuración de estación. Al guardar la configuración de estación se transfiere la configuración de seguridad correspondiente a la nueva versión.

Condición previa

- El proyecto Sunrise está archivado o guardado en algún directorio.
- Se ha instalado la nueva versión de Sunrise.Workbench.

Procedimiento

1. Cargar el proyecto Sunrise en la zona de trabajo.
2. Abrir la configuración de estación del proyecto y hacer clic en **Guardar**.
3. Se pregunta si las modificaciones se deben aplicar al proyecto. Hacer clic en **Guardar y aceptar**.
4. La configuración de seguridad se actualiza y sus parámetros se convierten. Cuando el proceso ha finalizado se indica mediante un mensaje. Confirmar con **OK**.
5. Para poder utilizar el proyecto actualizado en la unidad de control del robot, hace falta realizar más pasos:
 - a. Instalar el software de sistema.
 - b. Sincronizar el proyecto.
 - c. Activar nuevamente la configuración de seguridad.
 - d. Efectuar la recepción de seguridad.

10.3 Instalar paquete de idioma

Descripción

La interfaz de usuario smartHMI está disponible en siguientes idiomas:

Alemán	Italiano
Inglés	Español
Francés	

Los idiomas que solo estén disponibles después del suministro de un software se pueden instalar posteriormente en caso de necesidad.

Condición previa

- El paquete de idioma está disponible en Sunrise.Workbench.
(>>> 4.4 "Instalar paquete de idioma en Sunrise.Workbench" Página 46)

Procedimiento

1. Abrir la configuración de estación y seleccionar la pestaña **Software**.
2. Seleccionar el paquete de software **SmartHMI LanguagePack** para la instalación:
 - Marcar el símbolo de confirmación en la columna **Instalar**.
3. Guardar la configuración de estación. Se pregunta si las modificaciones se deben aplicar al proyecto. Hacer clic en **Guardar y aceptar**.
4. Instalar el software de sistema nuevamente en la unidad de control del robot. Después del reinicio de la unidad de control del robot los idiomas que se acaban de instalar están disponibles en la smartHMI.

10.4 Instalar escáner de virus

Procedimiento

1. Abrir la configuración de estación y seleccionar la pestaña **Software**.
 2. Seleccionar el paquete de software **Ikarus AntiVirus** para la instalación:
 - Marcar el símbolo de confirmación en la columna **Instalar**.
 3. Guardar la configuración de estación. Se pregunta si las modificaciones se deben aplicar al proyecto. Hacer clic en **Guardar y aceptar**.
 4. Instalar el software de sistema nuevamente en la unidad de control del robot. Una vez reiniciada la unidad de control del robot, escáner de virus está activo en la unidad de control del robot.
- Los mensajes del escáner de virus se pueden indicar a través de un cuadro propio. Para ello, en la vista de estación, seleccionar **KUKA_Sunrise_Cabinet > Escáner de virus**.

11 Configuración del bus

11.1 Configuración y conexión de E/S en WorkVisual – Resumen

Paso	Descripción
1	Instalar el paquete de opciones Sunrise en WorkVisual.
2	Finalizar WorkVisual y crear en KUKA Sunrise.Workbench una nueva configuración de E/S o abrir una configuración de E/S ya existente. A continuación, WorkVisual se inicia automáticamente y se abre el proyecto de WorkVisual perteneciente a la configuración de E/S. (>>> 11.2 "Crear nueva configuración de E/S" Página 164) (>>> 11.3 "Abrir una configuración de E/S existente" Página 164)
3	Solo es necesario si se utilizan dispositivos para los que aún no se han importado archivos de descripción del dispositivo: 1. Cerrar el proyecto de WorkVisual. 2. Importar los archivos de descripción del aparato que sean necesarios. 3. Abrir de nuevo el proyecto de WorkVisual.
4	Crear la estructura del bus de campo.
5	Crear y conectar las E/S de Sunrise. (>>> 11.4 "Crear E/S Sunrise" Página 164) (>>> 11.5.3 "Conectar E/S Sunrise" Página 171)
6	Exportar la configuración de E/S en el proyecto Sunrise. (>>> 11.6 "Exportar la configuración de E/S en el proyecto Sunrise" Página 171)
7	Transmitir la configuración de E/S a la unidad de control del robot (sincronizar proyecto) y reiniciar la unidad de control del robot. (>>> 9.4 "Sincronización de proyectos" Página 155)



Puede encontrarse información acerca de la instalación y la administración de paquetes de opciones en la documentación de **WorkVisual**.



La información para la importación de archivos de descripción del aparato y en general para la estructura de los buses de campo se puede consultar en la documentación de **WorkVisual**.



La información sobre la estructura concreta de los buses de campo soportados por Sunrise se puede consultar en las correspondientes documentaciones del bus de campo.



Al conectar dispositivos EtherCAT adicionales a una brida de medios con salida EtherCAT, p. ej. brida de medios IO neumática, debe tenerse en cuenta que la gama de las señales disponibles en el bus es limitada. Si el número de participantes conectados es demasiado grande, esto puede provocar una sobrecarga del bus y una pérdida de la comunicación. En ese caso, es posible que ya no se pueda desplazar al robot.

11.2 Crear nueva configuración de E/S

- Condición previa** ■ Proyecto Sunrise sin configuración de E/S
- Procedimiento**
1. Marcar el proyecto en el **Explorador de paquetes**.
 2. Seleccionar la secuencia de menú **Archivo > Nuevo > Configuración de E/S**.
- Se inicia WorkVisual y se abre el proyecto de WorkVisual perteneciente a la configuración de E/S. En el proyecto Sunrise se añade el archivo **IO-Configuration.wvs** a través del cual se puede acceder al proyecto correspondiente de WorkVisual.

11.3 Abrir una configuración de E/S existente

- Condición previa** ■ Proyecto Sunrise con configuración de E/S
- Procedimiento**
1. Hacer doble clic en el archivo **IOConfiguration.wvs**. Se inicia WorkVisual y se abre el proyecto de WorkVisual perteneciente a la configuración de E/S.
 2. En la ventana **Estructura del proyecto** en la pestaña **Dispositivos**, hacer clic con el botón derecho del ratón en la unidad de control del robot inactiva.
 3. Seleccionar **Ajustar control activo** en el menú contextual. Se abre la ventana **Círculo EA**. Las E/S Sunrise se pueden editar.

11.4 Crear E/S Sunrise

- Condición previa** ■ La estructura del bus de campo está terminada.
■ La unidad de control del robot está activa.
- Procedimiento**
1. En la ventana **Círculo EA**, en la parte superior derecha de la pestaña **Buses de campo**, marcar un módulo de entrada o de salida del bus configurado.
(>>> 11.5.1 "Ventana Circuito EA" Página 169)
 2. En la ventana **Círculo EA** seleccionar en la parte superior izquierda la pestaña **E/S Sunrise**.
 3. Hacer clic en la ventana **Círculo EA** abajo a la izquierda sobre el botón **Generando señales en el proveedor**. Se abre la ventana **Crear señales de E/S**.
(>>> 11.4.1 "Ventana "Crear señales de E/S"" Página 165)
 4. Crear el grupo E/S y las entradas/salidas del grupo.
(>>> 11.4.2 "Crear el grupo E/S y las entradas/salidas del grupo" Página 166)
 5. Hacer clic en **OK**. Las E/S Sunrise se guardan. Se cierra la ventana **Crear señales de E/S**.

El grupo E/S creado se muestra en la ventana **Círculo EA** en la pestaña **E/S Sunrise**. Ahora las señales se pueden conectar.

(>>> 11.5.3 "Conectar E/S Sunrise" Página 171)

11.4.1 Ventana "Crear señales de E/S"

Resumen

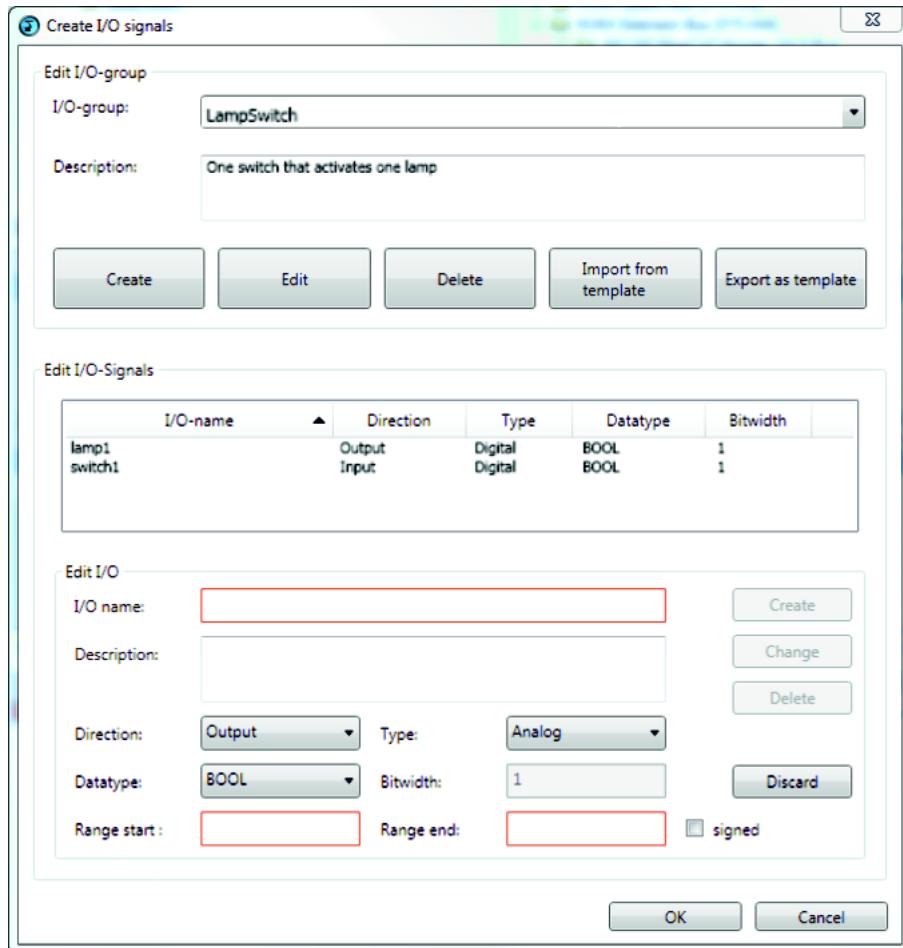


Fig. 11-1: Ventana Crear señales de E/S

La ventana para crear y editar las E/S Sunrise y los grupos E/S Sunrise se compone de las siguientes zonas:

Zona	Descripción
Editar grupo E/S	En esta zona se crean y se editan grupos E/S. Existe la posibilidad de guardar grupos E/S como plantilla y de importar plantillas ya creadas.
Editar señales de E/S	En esta zona se muestran las señales de entrada/salida de un grupo E/S.
Editar E/S	En esta zona se crean y se editan las entradas/salidas de un grupo E/S.

i Los campos de entrada están rodeados con un marco rojo si se deben introducir valores obligatoriamente o si se han introducido valores incorrectos. Se muestra un texto de ayuda si se sitúa el puntero del ratón sobre el campo.

Propiedades de la señal En la zona **Editar E/S** pueden crearse señales nuevas y se definirse sus propiedades:

Propiedad	Descripción
Nombrar de la E/S	Introducir el nombre de la entrada o la salida.
Descripción	Introducir una descripción para la entrada o la salida (opcional).

Propiedad	Descripción
Dirección	Determinar si la señal es una entrada o una salida. ■ Input, Output
Tipo	Determinar si la señal es una señal analógica o digital. ■ Analog, Digital
Tipo de datos	Seleccionar el tipo de datos de la señal. En WorkVisual se puede escoger en total entre 15 tipos de datos diferentes. Estos tipos de datos se representan con los siguientes tipos de datos para la utilización con Java: ■ integer, long, double, boolean
Ancho de bit	Introducir el número de bits de los que se compone la señal. En el tipo de datos Bool, el ancho de bit siempre es 1. Nota: El valor debe ser un número entero positivo que no supere la máxima longitud permitida del tipo de datos seleccionado.
Zona inicial	Solo es relevante para entradas/salidas analógicas.
Zona final	Introducir la zona inicial y final del valor analógico y, en caso necesario, marcar el símbolo de confirmación en la casilla Con signo .
Con signo	 Nota: Estos valores se encuentran generalmente en la hoja de datos del módulo de bus de campo. La zona inicial debe ser menor que la zona final. También se pueden indicar decimales.

11.4.2 Crear el grupo E/S y las entradas/salidas del grupo

Condición previa ■ La ventana **Crear señales de E/S** está abierta.

Procedimiento 1. En la zona **Editar grupo E/S** hacer clic en **Crear**.
Se abre la ventana **Crear grupo E/S**.

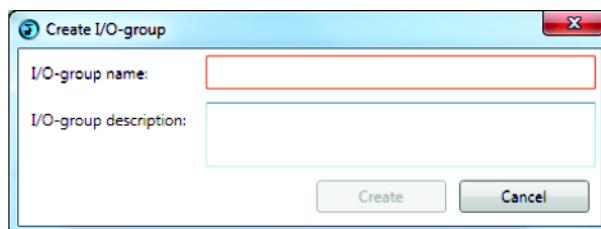


Fig. 11-2: Crear un grupo E/S

2. Indicar un nombre para el grupo E/S.
3. Introducir una descripción para el grupo E/S (opcional).



Se recomienda introducir siempre una descripción. Esta descripción se muestra posteriormente como texto auxiliar en la aplicación del robot y en el smartHMI.

4. Hacer clic en **Create**. Se crea el grupo E/S y se muestra en el menú de selección **Grupo E/S**.
5. En la zona **Editar E/S** introducir un nombre para la entrada o la salida del grupo y definir las propiedades de la señal
(>>> "Propiedades de la señal" Página 165)
6. En la zona **Editar E/S** hacer clic en **Crear**. Se crea la señal de entrada o salida y se muestra en la zona **Editar señales de E/S**.
7. Repetir el paso 5 y 6 para definir más entradas/salidas del grupo.

11.4.3 Editar grupo E/S

- Condición previa**
- La ventana **Crear señales de E/S** está abierta.
 - Las entradas/salidas del grupo no están conectadas.

- Procedimiento**
1. En el menú de selección **Grupo E/S**: seleccionar el grupo E/S deseado.
 2. Hacer clic en **Editar**. Se abre la ventana **Renombrar grupo E/S**.
 3. Modificar el nombre del grupo E/S y la descripción correspondiente (opcional). Confirmar con **Aceptar**.

11.4.4 Borrar grupo E/S

- Condición previa**
- La ventana **Crear señales de E/S** está abierta.
 - Las entradas/salidas del grupo no están conectadas.

- Procedimiento**
1. En el menú de selección **Grupo E/S**: seleccionar el grupo E/S deseado.
 2. Hacer clic en **Borrar**. Si ya se han creado señales para el grupo E/S, se mostrará una pregunta de seguridad.
 3. Responder **Sí** a la pregunta de seguridad. Se borra el grupo E/S.

11.4.5 Modificar la entrada/salida de un grupo

- Condición previa**
- La ventana **Crear señales de E/S** está abierta.
 - Las señales a modificar no están conectadas.

- Procedimiento**
1. En el menú de selección **Grupo E/S**: seleccionar el grupo E/S de la señal.
 2. En la zona **Editar señales de E/S** marcar la entrada o salida deseada.
 3. En la zona **Editar E/S**, procesar las propiedades de las señales de acuerdo con las necesidades.

(>>> "Propiedades de la señal" Página 165)



Haciendo clic una vez sobre el botón **Rechazar** se rechazarán todas las modificaciones.

4. Hacer clic en **Modificar**. Las modificaciones son aceptadas.

11.4.6 Borrar la entrada/salida de un grupo

- Condición previa**
- La ventana **Crear señales de E/S** está abierta.
 - Las señales a borrar no están conectadas.

- Procedimiento**
1. En el menú de selección **Grupo E/S**: seleccionar el grupo E/S de la señal.
 2. En la zona **Editar señales de E/S** marcar la entrada o salida deseada.
 3. Hacer clic en **Borrar**.

11.4.7 Exportar grupo E/S como plantilla

- Descripción**
- Los grupos E/S se pueden guardar como plantilla. La plantilla contiene todas las entradas y salidas que pertenecen al grupo E/S guardado. De este modo se pueden reutilizar los grupos E/S ya creados. En este caso, no se acepta la conexión de las entradas y salidas.

Después de la exportación de la plantilla, las plantillas creadas en WorkVisual estarán disponibles en Sunrise.Workbench en la carpeta **IOTemplates** del proyecto.

- Condición previa**
- La ventana **Crear señales de E/S** está abierta.
- Procedimiento**
1. En la zona **Editar grupo E/S** seleccionar el grupo E/S que debe exportarse como plantilla.
 2. Hacer clic en **Exportar plantilla**. Se abre la ventana **Exportar grupo E/S como plantilla**.
 3. Introducir un nombre para la plantilla.
-  Si ya existe una plantilla con el mismo nombre en el proyecto Sunrise, se sobrescribirá.
4. Introducir una descripción para la plantilla (opcional).
 5. Hacer clic en **Exportar**. La plantilla queda memorizada.

11.4.8 Importar grupo E/S desde la plantilla

- Condición previa**
- La ventana **Crear señales de E/S** está abierta.
 - En Sunrise.Workbench se encuentra disponible al menos un grupo E/S como plantilla en el proyecto Sunrise.
- Procedimiento**
1. En la zona **Editar grupo E/S** hacer clic en **Importar desde plantilla**. Se abre la ventana **Exportar grupo E/S desde plantilla**.
 2. En la lista de selección **Plantilla utilizada** seleccionar la plantilla que debe importarse.
 3. En el campo **Nombre del grupo E/S** introducir un nombre para el grupo E/S que se va a crear.
 4. Hacer clic en **Importar**. Se importa un grupo E/S configurado conforme a la plantilla y se puede editar.

11.5 Conectar bus

11.5.1 Ventana Circuito EA

Resumen

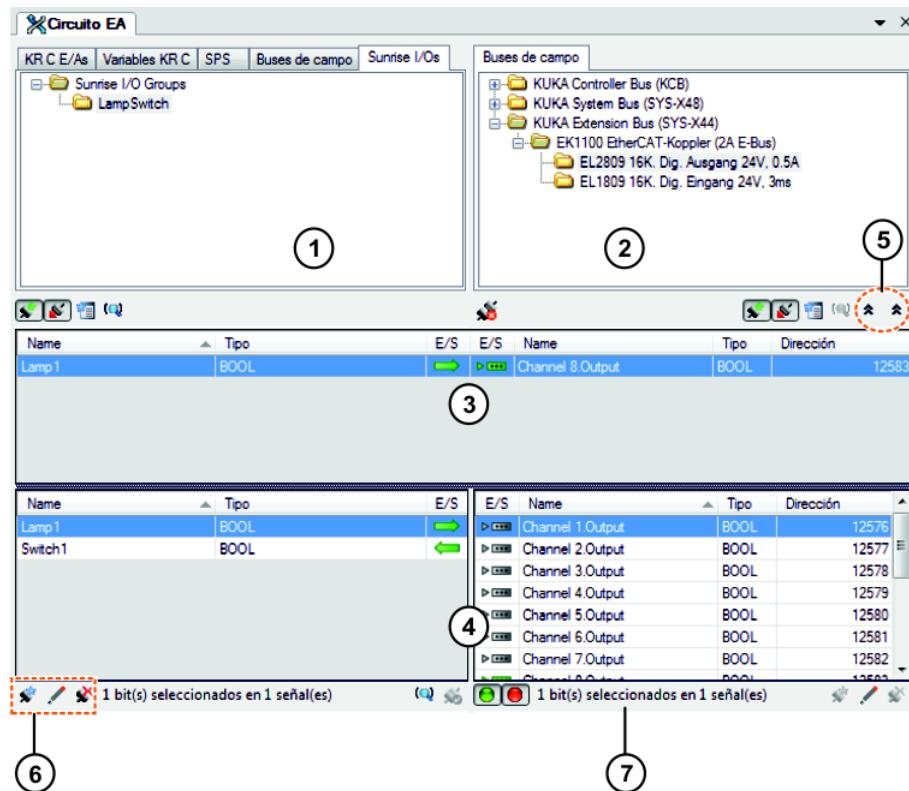


Fig. 11-3: Ventana Circuito EA

Pos.	Descripción
1	Indicación de los grupos E/S Sunrise Las señales de los grupos E/S marcados aquí se muestran en los esquemas inferiores.
2	Indicación de las entradas/salidas de los módulos de bus Las señales del módulo aquí marcado se visualizan en los esquemas inferiores.
3	Esquema de conexiones Visualización de las señales a conectar. Se trata de las señales del grupo E/S marcado en E/S Sunrise , que están conectadas con el módulo de bus marcado en Buses de campo .
4	Esquema de señales Aquí se pueden conectar las señales. (>>> 11.5.3 "Conectar E/S Sunrise" Página 171)

Pos.	Descripción
5	Con la ayuda del botón de flecha se pueden plegar y desplegar de nuevo el esquema de conexiones y de señales, separados entre sí. <ul style="list-style-type: none"> ■ Plegar esquema de conexiones (el símbolo de flecha a la izquierda señala hacia arriba) ■ Desplegar esquema de conexiones (el símbolo de flecha a la izquierda señala hacia abajo) ■ Plegar esquema de señales (el símbolo de flecha a la derecha señala hacia arriba) ■ Desplegar esquema de señales (el símbolo de flecha a la derecha señala hacia abajo)
6	Botones para crear y editar E/S Sunrise
7	Visualización sobre los bits que contienen las señales marcadas.



Para la conexión de E/S en Sunrise solo son relevantes las pestañas **E/S Sunrise** y **Buses de campo**.

11.5.2 Botones de la ventana Circuito EA

Algunos de los botones existen varias veces. Estos se refieren siempre al lado de la ventana **Círculo EA** en el que están colocados.

Editar	Botón	Nombre / descripción
		Generando señales en el proveedor Abre la ventana Crear señales de E/S . (>>> 11.4.1 "Ventana "Crear señales de E/S"" Página 165) El botón solo está activo si en la pestaña Buses de campo está marcado un módulo de entrada o salida y en el resumen de señales está marcada la señal del grupo E/S.
		Edita señales en el proveedor Abre la ventana Editar señales de E/S . El botón solo está activo si en la pestaña E/S Sunrise está marcado un grupo E/S y en el resumen de señales está marcada la señal del grupo E/S.
		Borrando señales en el proveedor Borra todas las entradas/salidas marcadas. Si están marcadas todas las entradas/salidas de un grupo, también se borrará el grupo E/S. El botón solo está activo si en la pestaña E/S Sunrise está marcado un grupo E/S y en el resumen de señales está marcada la señal del grupo E/S.

Conectar

Botón	Nombre / descripción
	Desconectar Desconecta las señales conectadas y marcadas. Es posible marcar varias conexiones y separarlas a la vez.
	Conectar Conecta entre sí las señales marcadas en el resumen de señales.

11.5.3 Conectar E/S Sunrise**Descripción**

Con este procedimiento se conectan las E/S Sunrise con las entradas/salidas del módulo de bus de campo. Solo se pueden conectar entradas con entradas y salidas con salidas del mismo tipo de datos. Es posible, por ejemplo, BOOL con BOOL o INT con INT y no es posible BOOL con INT o BYTE.

Requisitos previos

- La unidad de control del robot está activa.

Procedimiento

1. En la mitad izquierda de la ventana, marcar en la pestaña **E/S Sunrise** el grupo E/S desde el que se deben conectar las E/S.
Las señales del grupo se visualizan en la zona inferior de la ventana **Círcuito EA**.
2. En la mitad derecha de la ventana, marcar en la pestaña **Buses de campo** el módulo de entrada o de salida deseado.
Las señales del módulo de bus de campo marcado se visualizan en la zona inferior de la ventana **Círcuito EA**.
3. Arrastrar y soltar la señal del grupo sobre la entrada o la salida del módulo. (O también a la inversa, arrastrar la entrada o la salida del dispositivo sobre la señal del grupo.)
Las señales están ahora conectadas. Las señales conectadas están marcadas con flechas verdes.

Procedimiento alternativo durante la conexión:

- Marcar las señales que se deben conectar y hacer clic en el botón **Conectar**.

11.6 Exportar la configuración de E/S en el proyecto Sunrise**Descripción**

Al exportar una configuración E/S desde WorkVisual, se crea una clase Java propia para cada grupo E/S en el proyecto Sunrise correspondiente. Cada una de estas clases Java contiene los métodos necesarios para la programación, para acceder mediante lectura a las entradas/salidas de un grupo E/S y mediante escritura a las salidas de un grupo E/S.

Las clases y los métodos se almacenan en la carpeta de origen del proyecto Sunrise en el paquete Java **com.kuka.generated.ioAccess**.



El código fuente de las clases Java del paquete **com.kuka.generated.ioAccess** no se puede modificar manualmente. Para ampliar la funcionalidad de un grupo E/S, se pueden衍生 otras clases a partir de las clases creadas o se pueden seguir utilizando objetos de estas clases, p. ej. como campos de clases propias (agregar).

Aquí se describe la estructura del proyecto Sunrise después de exportar una configuración E/S:

(>>> 15.12 "Entradas/Salidas" Página 300)

Condición previa

- La unidad de control del robot está activa.
- En Sunrise.Workbench se ha activado la detección automática de modificaciones.
(>>> 5.10 "Activar la detección automática de modificaciones" Página 61)

Procedimiento

1. Seleccionar la secuencia de menú **Archivo > Importar/exportar**. Se abre el asistente para la importación/exportación de archivos.
2. Seleccionar **Exportar la configuración E/A en el proyecto con el entorno de trabajo Sunrise**.
3. Hacer clic en **Continuar>** y seguidamente en **Finalizar**. La configuración se exportará al proyecto Sunrise.
4. Se muestra si la exportación se ha finalizado correctamente. Si las E/S Sunrise no han sido conectadas, también se indicará.



No es imprescindible conectar todas las E/S Sunrise creadas.

Finalizar el asistente con **Cerrar**.

5. Cerrar WorkVisual mediante **Archivo > Finalizar**.

12 Control externo

12.1 Configurar control externo

Descripción Si los procesos de la estación se deben controlar en el modo automático por una unidad de control superior, el proyecto Sunrise que se encuentra en la unidad de control del robot debe estar configurado para el control externo.

Cada proyecto Sunrise que está configurado para el control externo utiliza una aplicación por defecto que se selecciona automáticamente al cambiar al modo de servicio automático. La aplicación por defecto del proyecto controlado de forma externa no se puede deseleccionar de nuevo en el modo automático.

La aplicación por defecto no se puede iniciar mediante la tecla de arranque en el smartPAD, sino solo mediante una entrada externa.

La unidad de control del robot y el control superior se comunican a través de señales de entrada/salida predefinidas:

- El control superior puede iniciar, pausar y continuar la aplicación por defecto mediante las señales de entrada.
- Las señales de salida se pueden utilizar para enviar información sobre el estado de la aplicación por defecto y de la estación al control superior.

Resumen

Para poder utilizar el control externo son necesarios los siguientes pasos:

Paso	Descripción
1	Configurar y conectar las entradas/salidas para la comunicación con el control superior en WorkVisual. (>>> 12.1.1 "Control externo de entradas" Página 174) (>>> 12.1.2 "Control externo de salidas" Página 174)
2	Exportar la configuración de E/S desde WorkVisual a Sunrise.Workbench.
3	Crear en Sunrise.Workbench la aplicación por defecto para el control externo del proyecto Sunrise.
4	Configurar el control externo del proyecto Sunrise en Sunrise.Workbench. (>>> 12.1.3 "Configurar el control externo en las propiedades del proyecto" Página 175)
5	Transmitir el proyecto Sunrise a la unidad de control del robot mediante sincronización.



Las entradas/salidas físicas que se utilizan para la comunicación con el control superior, no se pueden conectar varias veces.

Condición previa

Para poder iniciar una aplicación externa, deben cumplirse las siguientes condiciones previas:

- El robot está ajustado (todos los ejes).
- El proyecto Sunrise está configurado para el control externo.
- Modo de servicio AUT
- La entrada App_Start está configurada.
- Si está configurada: La entrada App_Enable entrega el nivel HIGH (TRUE).
- El movimiento habilitado está disponible.

12.1.1 Control externo de entradas

App_Start	La entrada App_Start es imprescindible para la un proyecto controlado externamente.
	Mediante un flanco ascendente de la señal de entrada (cambio de FALSE a TRUE) se inicia y se retoma la aplicación por defecto en el modo automático por el control superior.
App_Enable	La entrada App_Enable se puede configurar opcionalmente.
	Mediante un nivel LOW en esta entrada se puede pausar la aplicación por defecto en el modo automático por el control superior.
Comportamiento del sistema	La entrada App_Enable posee una prioridad mayor que la entrada App_Start. Si la entrada App_Enable está configurada, la aplicación por defecto se podrá reiniciar si en App_Enable existe un nivel High.

App_Start	App_Enable	Estado de la aplicación	Reacción
FALSE --> TRUE	FALSE	Seleccionada	Ninguno
FALSE --> TRUE	FALSE	Movimiento pausado	Ninguno
FALSE --> TRUE	TRUE	Seleccionada	Se inicia la aplicación.
FALSE --> TRUE	TRUE	Movimiento pausado	<p>Se continúa con la aplicación.</p> <p>En caso de abandonar la trayectoria: El robot retrocede a la posición previa. A continuación la aplicación es pausada.</p>
cualquiera	TRUE --> FALSE	Ejecución	La aplicación se pausa.
cualquiera	TRUE --> FALSE	Reposiciónamiento	La aplicación se pausa.

12.1.2 Control externo de salidas

La configuración de estas salidas es opcional.

AutExt_Active	Un nivel High en esta salida indica al control de orden superior de que está activo el modo de servicio automático y de que el proyecto de la unidad de control del robot puede controlarse de forma externa.
AutExt_AppRead yToStart	Un nivel High en esta salida indica al control superior que la aplicación por defecto está preparada para el arranque (estado Seleccionado o Movimiento pausado).
DefaultApp_Error	Un nivel High en esta salida indica al control superior que se ha producido un error durante la ejecución de la aplicación por defecto (estado Error).
Station_Error	Un nivel High en esta salida avisa al control superior de que la estación se encuentra en estado de error. El estado de error se presenta cuando se cumple una de las siguientes condiciones:

- Movimiento habilitado no disponible.
- Hay un error del accionamiento o del bus.
- Al menos un eje del robot no está ajustado y el modo de servicio no es T1.

12.1.3 Configurar el control externo en las propiedades del proyecto

Procedimiento

1. En el **Explorador de paquetes** hacer clic con el botón derecho del ratón sobre el proyecto Sunrise deseado y seleccionar en el menú contextual **Sunrise > Adaptar los ajustes del proyecto**.
Se abre la ventana **Propiedades del proyecto**.
2. En la estructura del directorio, seleccionar en la zona izquierda de la ventana **Sunrise > Mando externo**.
3. En la zona derecha de la ventana, realizar los ajustes para el control externo del proyecto.
 - Marcar el símbolo de confirmación de la casilla **El proyecto se controla externamente**.
 - En la zona **Aplicación por defecto** seleccionar la aplicación por defecto deseada.
 - En la zona **Configuración entradas** configurar las entradas para la comunicación externa.
 - Columna **Grupo E/S**: Seleccionar el grupo E/S que contiene la entrada deseada.
 - Columna **Entrada booleana**: Seleccionar la entrada del grupo E/S.
 - Opcional: En la zona **Configuración de salidas** configurar las salidas de aviso para el proyecto.
 - Columna **Grupo E/S**: Seleccionar el grupo E/S que contiene la salida deseada.
 - Columna **Salida booleana**: Seleccionar la salida del grupo E/S.
4. Hacer clic sobre **Aplicar** para aceptar los ajustes y cerrar la ventana con **OK**.



La aplicación por defecto está identificada con un símbolo en el **Explorador de paquetes**:

Si se modifica el nombre de la aplicación por defecto, el símbolo dejará de aparecer y deberá volver a ser marcado como aplicación por defecto.
(>>> 12.2 "Marcar la aplicación del robot como aplicación por defecto" Página 176)

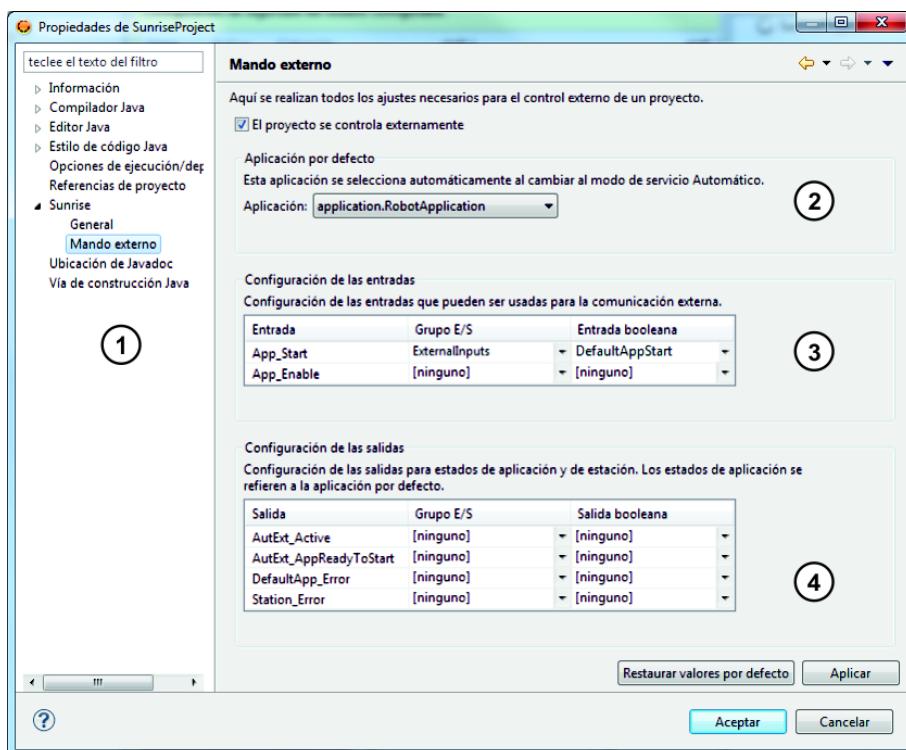
Descripción

Fig. 12-1: Configuración mínima: Control externo de un proyecto

Pos.	Descripción
1	Estructura de directorio de las propiedades del proyecto
2	Zona Aplicación por defecto Todas las aplicaciones del robot para el proyecto están disponibles para la selección.
3	Zona Configuración entradas Todos los grupos E/S de la configuración de E/S del proyecto y las entradas correspondientes están disponibles para la selección. La entrada App_Start es imprescindible para el control externo de un proyecto.
4	Zona Configuración salidas Todos los grupos E/S de la configuración de E/S del proyecto y las salidas correspondientes están disponibles para la selección.

12.2 Marcar la aplicación del robot como aplicación por defecto

Descripción

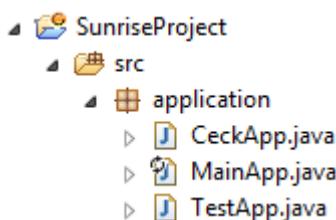
Para cada proyecto Sunrise se puede determinar una aplicación por defecto que se seleccionará automáticamente, en caso de un reinicio de la unidad de control del robot y en caso de sincronización del proyecto.

En el caso de los proyectos controlados de forma externa, es imprescindible establecer una aplicación por defecto. Ésta se selecciona automáticamente al cambiar al modo de servicio automático.

Procedimiento

- En el **Explorador de paquetes** hacer clic con el botón derecho del ratón sobre la aplicación del robot deseada y seleccionar en el menú contextual **Sunrise > Marcar como aplicación por defecto**.

La aplicación del robot se señaliza en el **Explorador de paquetes** como aplicación por defecto y se ajusta en las propiedades del proyecto automáticamente como aplicación por defecto.

Ejemplo**Fig. 12-2: Aplicación por defecto MainApp.java****12.3 Establecer salidas de aviso para el proyecto no controlado externamente****Descripción**

Las señales de salida predefinidas para la comunicación con el control superior, también se pueden utilizar en proyectos no controlados externamente, para comunicar estados de la aplicación y de la estación.

Los estados de la aplicación siempre hacen referencia a la aplicación por defecto del proyecto.

Condición previa

- La configuración de E/S del proyecto contiene las salidas configuradas y conectadas en WorkVisual.
(>>> 12.1.2 "Control externo de salidas" Página 174)

Procedimiento

1. En el **Explorador de paquetes** hacer clic con el botón derecho del ratón sobre el proyecto Sunrise deseado y seleccionar en el menú contextual **Sunrise > Adaptar los ajustes del proyecto**.
Se abre la ventana **Propiedades del proyecto**.
2. En la estructura del directorio, seleccionar en la zona izquierda de la ventana **Sunrise > General**.
3. En la zona derecha de la ventana, realizar los ajustes generales para el proyecto.
 - Si deben comunicarse los estados de la aplicación: En la zona **Aplicación por defecto** seleccionar la aplicación por defecto deseada.
 - En la zona **Configuración de salidas** configurar las salidas de aviso para el proyecto.
 - Columna **Grupo E/S**: Seleccionar el grupo E/S que contiene la salida deseada.
 - Columna **Salida booleana**: Seleccionar la salida del grupo E/S.
4. Hacer clic sobre **Aplicar** para aceptar los ajustes y cerrar la ventana con **OK**.

13 Configuración de seguridad

13.1 Resumen de la configuración de seguridad

Con la configuración de seguridad se establecen las funciones orientadas a la seguridad para integrar al robot industrial de forma segura en la instalación. Las funciones orientadas a la seguridad sirven para proteger a las personas durante el trabajo con el robot.

La configuración de seguridad forma parte de manera fija de un proyecto Sunrise y se gestiona en forma de tabla. Las funciones de seguridad individuales se combinan en KUKA Sunrise.Workbench de forma específica para la aplicación. A continuación, la configuración de seguridad se transmite con el proyecto a la unidad de control y se activa allí.



ADVERTENCIA Como consecuencia de una configuración de seguridad incorrecta, se pueden producir daños materiales y lesiones graves que pueden provocar la muerte. Si se activa una configuración de seguridad nueva o modificada, el técnico de puesta en servicio debe comprobar mediante pruebas que los parámetros de seguridad configurados se han aceptado correctamente y que las funciones de seguridad de la configuración están completamente en condiciones de funcionamiento (recepción de seguridad).



La configuración de las funciones de seguridad, la activación y la desactivación de la configuración de seguridad y la recepción de seguridad, solo se deberán llevar a cabo por un técnico de puesta en servicio instruido. El técnico de puesta en servicio tiene la responsabilidad de que la configuración de seguridad solo se active en aquellos robots para los que esté prevista.

La configuración de seguridad no se comprueba actualmente por KUKA Sunrise.Workbench en cuanto a la plausibilidad.



Si la puesta en servicio de la instalación se lleva a cabo de forma incompleta, será necesario aplicar y documentar medidas compensatorias para reducir el riesgo como p.ej. la colocación de una valla de protección o un cartel de advertencia, el enclavamiento del interruptor principal,etc. La puesta en servicio se lleva a cabo de forma incompleta cuando, por ejemplo, no se han implementado todos los controles de seguridad necesarios o no se ha comprobado que todas las funciones de seguridad funcionan correctamente.



El integrador de la instalación debe verificar que la configuración de seguridad reduce riesgos de manera suficiente en el servicio de colaboración (MRK).

Se recomienda efectuar esta verificación conforme a la información y a las indicaciones para el servicio de robots de colaboración según ISO/TS 15066.



En el marco del mecanismo ESM (Event-Driven Safety Monitoring), en la configuración de seguridad se definen estados con diferentes ajustes de seguridad, entre los que se puede comutar en la aplicación. Debido a que la comutación entre estos estados se realiza mediante señales no orientadas a la seguridad, todos los estados configurados deben ser consistentes. Es decir, cada estado debe asegurar un grado suficiente de seguridad, independientemente del momento y del lugar de su activación (es decir, independientemente del paso actual del proceso).

13.2 Concepto de seguridad

Resumen

La configuración de seguridad debe ejecutar todas las funciones de seguridad que son necesarias para el servicio del robot industrial. Una función de seguridad controla el sistema completo mediante determinados criterios. Éstos se describen mediante funciones de control individuales, denominadas AMFs (Atomic Monitoring Functions). Para la configuración de seguridad de una función de seguridad se pueden enlazar varias AMFs para los controles de seguridad complejos. La función de seguridad establece además una reacción adecuada que se activará en caso de error.

Ejemplo: En una zona determinada del campo de trabajo del robot, la velocidad en el TCP no deberá superar 500 mm/s (funciones de control para la monitorización de la zona y el control de velocidad). De lo contrario, el robot debe detenerse inmediatamente (reacción en caso de error).

PSM y ESM

El concepto de seguridad de Sunrise ofrece 2 mecanismos de control diferentes:

- Control permanente destinado a la seguridad

Las funciones de seguridad del mecanismo PSM (Permanent Safety Monitoring) están activas constantemente. La desactivación de funciones de seguridad individuales solo es posible mediante la modificación de la configuración de seguridad.

El mecanismo PSM sirve para el control continuo del sistema. Aquí se realizan ajustes básicos de seguridad que son independientes del paso de proceso ejecutado. Aquí se incluyen, por ejemplo, funciones de PARADA DE EMERGENCIA, la validación en el smartPAD, la definición de una zona de celda o funciones de seguridad dependientes de los modos de servicio.

- Control destinado a la seguridad y dependiente de las incidencias

El mecanismo ESM (Event-driven Safety Monitoring) define estados seguros entre los que se puede comutar en la aplicación. Un estado ESM seguro contiene las funciones de seguridad necesarias en el paso de proceso correspondiente.

Debido a que la conmutación se realiza mediante señales no destinadas a la seguridad, durante la determinación del estado debe tenerse asegurarse que éste debe garantizar siempre suficiente seguridad, independientemente del lugar y del momento de su activación.

El mecanismo ESM permite la adaptación en función del proceso de determinados ajustes de seguridad. Esto resulta especialmente interesante para las aplicaciones en el sector de la cooperación hombre-robot, ya que aquí a menudo son necesarios diferentes ajustes de seguridad en función de la situación. A través de un estado ESM se pueden establecer individualmente los parámetros necesarios para cada paso del proceso, por ejemplo la velocidad permitida, los valores de colisión o los límites espaciales.

AMF

La unidad más pequeña de un control de seguridad se denomina Atomic Monitoring Function (AMF).

Cada AMF envía una información elemental relevante para la seguridad, como por ejemplo, si una entrada segura está ajustada o si está seleccionado el modo de servicio automático.

Las Atomic Monitoring Functions pueden adoptar 2 estados diferentes y se comportan con el nivel LOW activado. Es decir, si se vulnera un control, el estado cambia de "1" a "0".

- Estado "0": La AMF está vulnerable.
- Estado "1": La AMF no está vulnerable.

La AMF **PARADA DE EMERGENCIA smartPAD** está vulnerada, por ejemplo si está pulsado el dispositivo de PARADA DE EMERGENCIA en la unidad de operación.

Función de seguridad

Un estado ESM seguro se define de hasta 20 funciones de seguridad. Las funciones de seguridad del mecanismo ESM utilizan exactamente una AMF. Si esta AMF está vulnerada, la función de seguridad se considerará como vulnerable y, de este modo, el estado completo.

Para las funciones de seguridad del mecanismo PSM se enlazan entre sí de forma lógica hasta 3 AMFs. De este modo, se pueden realizar controles de seguridad complejos. Si están vulneradas todas las AMFs de una función de seguridad del mecanismo PSM, toda la función de seguridad se considerará como vulnerable.

Interfaces de seguridad

Existen diferentes interfaces de seguridad para intercambiar señales destinadas a la seguridad entre el control superior y la unidad de control del robot. Mediante las entradas seguras de estas interfaces, se pueden conectar dispositivos de seguridad, por ejemplo, dispositivos de PARADA DE EMERGENCIA externos o puertas de protección, y se pueden evaluar las señales de entrada correspondientes. Las salidas seguras de estas interfaces se pueden utilizar para indicar la vulneración de funciones de seguridad.

- Interfaces de seguridad de Ethernet (solo disponible función esclavo)
 - PROFINET/PROFIsafe
 - EtherCAT/FSoE
- Interfaces de seguridad discretas
 - CIB_SR/X11



El bus PROFINET en WorkVisual se puede configurar. En la documentación del bus de campo correspondiente se puede encontrar más información sobre la estructura concreta del bus de campo.



Encontrará más información sobre la interfaz X11 en las instrucciones de servicio de la unidad de control del robot **KUKA Sunrise Cabinet**.

Reacciones

Para cada función de seguridad se establece una reacción adecuada que se debe aplicar en caso de error y debe ajustar el sistema en un estado seguro.

Se pueden configurar las reacciones siguientes:

- Se activa la parada de seguridad 0.



Se recomienda configurar únicamente una parada de seguridad 0 si se necesita una desconexión inmediata de los accionamientos y una actuación inmediata de los frenos como reacción.

- Se activa la parada de seguridad 1.
- Se activa la parada de seguridad 1 (sobre la trayectoria).

Esta es la reacción de parada recomendada. Ofrece el efecto mínimo sobre el proceso, ya que una aplicación puede continuar sin que el robot deba retroceder.



ATENCIÓN En caso de situaciones de aplastamiento, la parada de seguridad 1 y la parada de seguridad 1 (sobre la trayectoria) pueden provocar fuerzas de aplastamiento mayores debido a la parada regulada en un recorrido de frenado planeado. Por ello, se recomienda emplear la parada de seguridad 0 en caso de controles de seguridad para la detección de situaciones de aplastamiento (p. ej. mediante la AMF, *Identificación de colisión*, *Control de fuerza TCP*)

- La salida segura se ajusta a "0" (nivel LOW).



El ajuste de una salida segura solo se puede configurar como reacción para funciones de seguridad del mecanismo PSM. No se puede configurar para funciones de seguridad del mecanismo ESM.

Las reacciones se pueden utilizar para el número deseado de funciones de seguridad. Una reacción se activa en el momento en que se ha vulnerado una de las funciones de seguridad que utiliza dicha reacción. De este modo, p. ej. es posible informar a un control superior a través de una salida segura si se produce un determinado error.

El mecanismo PSM permite activar varias reacciones diferentes en caso de que esté vulnerada una combinación determinada de AMFs. De este modo, p. ej. se puede activar una parada de seguridad y adicionalmente se puede ajustar una salida segura. Para llevarlo a cabo, se deben configurar 2 funciones de seguridad con la misma combinación de AMF.

Si 2 funciones de seguridad solo se diferencian en el tipo de parada configurada si se produce una vulneración, se activará la reacción de parada más precisa. Es decir, se activa aquella reacción de parada que ejecute la desconexión de los accionamientos más rápida y destinada a la seguridad. Si varias funciones de seguridad utilizan la misma señal de salida como reacción, ésta se ajustará a "0" en cuanto se haya vulnerado una de las funciones de seguridad.

Rapidez de respuesta

Todas las salidas destinadas a la seguridad utilizan LOW como estado seguro.

Si se vulnera una función de seguridad que emplea una salida segura como reacción, esta salida se ajusta inmediatamente a LOW.

Si se anula el estado de vulneración, la salida solo se vuelve a ajustar a HIGH si se cumplen las siguientes condiciones:

- La función de seguridad no se ha vulnerado en al menos 24 ms. Siempre se reacciona de forma retardada a la anulación del estado de vulneración.
- Si se utiliza una interfaz de seguridad de Ethernet:

La salida ha estado antes al menos 500 ms en nivel LOW. Si el nivel LOW aún no estuvo ajustado tanto tiempo, no se cambia el nivel a HIGH hasta que se hayan alcanzado los 500 ms.

- Si se utiliza la interfaz de seguridad discreta:

La salida ha estado antes al menos 200 ms en nivel LOW. Si el nivel LOW aún no estuvo ajustado tanto tiempo, no se cambia el nivel a HIGH hasta que se hayan alcanzado los 200 ms.



En caso de utilización de funciones de seguridad con una salida segura como reacción, se debe tener en cuenta que el error de conexión (es decir, el error de comunicación) en entradas o salidas seguras se confirmará automáticamente por el control de seguridad cuando se haya restablecido la conexión. Como consecuencia, puede cambiar el nivel de la salida segura tras el restablecimiento de la conexión de LOW a HIGH.

Por esta razón, el técnico de puesta en servicio debe asegurar que no puede producirse ninguna reanudación automática de dispositivos periféricos..

13.3 Permanent Safety Monitoring

Las funciones de seguridad del mecanismo PSM (Permanent Safety Monitoring) están permanentemente activas y se encargan de realizar el control constante del sistema completo mediante los criterios establecidos por estas funciones.

Para una función de seguridad del mecanismo PSM se pueden enlazar entre sí hasta 3 AMFs (Atomic Monitoring Functions). La función de seguridad completa se considerará como vulnerada cuando estén vulneradas todas las AMFs utilizadas. La función de seguridad establece además una reacción. Ésta se activa cuando se ha vulnerado la función de seguridad completa.

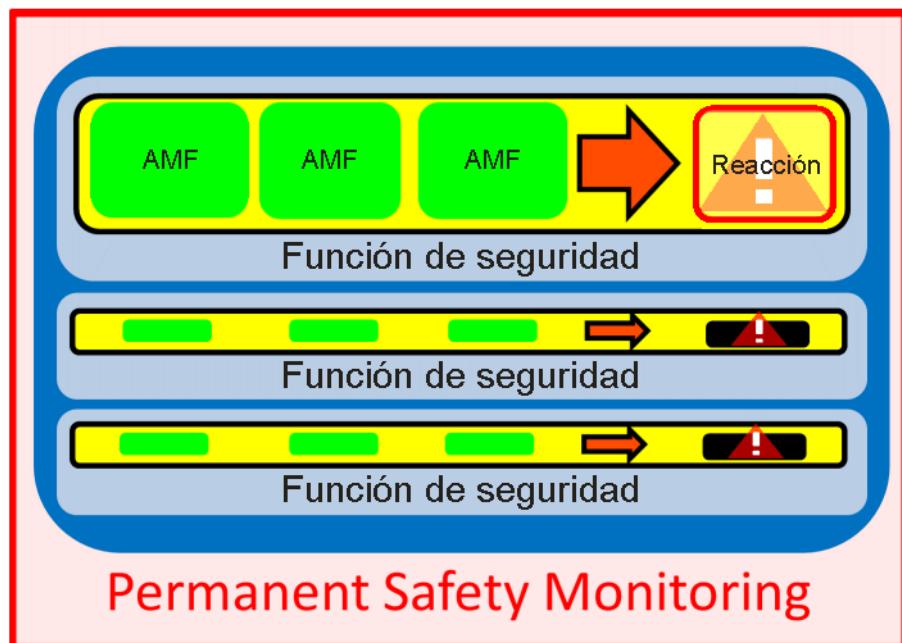


Fig. 13-1: Funciones de seguridad del mecanismo PSM

Categorías

Para el diagnóstico en caso de error, se asigna cada función de seguridad del mecanismo PSM a una categoría. En función de la categoría, los errores se visualizan en el smartPAD y se guardan en el archivo LOG. Por ello se recomienda seleccionarlas de forma práctica.

Se encuentran disponibles las siguientes categorías:

Categoría	Uso recomendado
Ninguno	Para funciones de seguridad que no se pueden asignar a ninguna categoría concreta
Salida	Para funciones de seguridad que utilizan el ajuste de una salida segura como reacción En esta categoría no se ofrece información de diagnóstico en caso de vulneración.
Validación	Para funciones de seguridad que evalúan un pulsador de validación En esta categoría no se ofrece información de diagnóstico en caso de vulneración, ya que, en caso de validación, se trata de un estado de servicio normal y no de un estado de error.
PARADA DE EMERGENCIA local	Para funciones de seguridad que evalúan una PARADA DE EMERGENCIA, que es activada por el dispositivo de PARADA DE EMERGENCIA en el smartPAD.
PARADA DE EMERGENCIA externa	Para funciones de seguridad que evalúan una PARADA DE EMERGENCIA, que es activada por el dispositivo de PARADA DE EMERGENCIA externo.
Protección del operario	Para funciones de seguridad que evalúan la señal para la protección del operario
Parada de servicio segura	Para funciones de seguridad que controlan la parada del robot

Categoría	Uso recomendado
Identificación de colisión	Para funciones de seguridad que se utilizan para la identificación de colisión o el control de fuerza
Parada de seguridad	Para funciones de seguridad que utilizan una parada de seguridad como reacción y no se pueden asignar a otra categoría. Ejemplo: Parada de seguridad externa
Control de velocidad	Para funciones de seguridad que se utilizan para el control de una velocidad específica del eje o cartesiana
Monitorización de la zona	Para funciones de seguridad que se utilizan para el control de una zona específica del eje o cartesiana

13.4 Event-driven Safety Monitoring

El mecanismo ESM (Event-driven Safety Monitoring) permite comutar entre diferentes estados ESM seguros en función de la situación.

Se pueden definir en total hasta 10 estados seguros. La conmutación entre los estados se realiza mediante la aplicación del robot o mediante una tarea en segundo plano.

(>>> 13.6.4.8 "Comutar entre estados ESM" Página 199)

Un estado ESM seguro se define con hasta 20 funciones de seguridad, que deben garantizar en cada situación un grado suficiente de seguridad. Mediante la conmutación de un estado ESM en el programa, se activa dicho estado ESM. Mientras está activo el estado ESM, se controlan todas las funciones de seguridad correspondientes de forma adicional a las funciones de seguridad activas de manera permanente.

La utilización del mecanismo ESM es opcional. El mecanismo ESM está desactivado, cuando no hay definido ningún estado ESM en la configuración de seguridad.

En caso de utilización del mecanismo ESM, siempre está activo exactamente un estado seguro. La desconexión mediante la aplicación no es posible.

Las funciones de seguridad de un estado ESM incluyen respectivamente una AMF única, a la que se asigna una reacción de parada adecuada.

En cuanto se vulnera una función de seguridad del estado ESM activo, se activa una parada. El tipo de reacción de parada se rige según la reacción de parada más precisa de todas las funciones de seguridad vulneradas en todos los estados ESM (tanto las activas como las inactivas). Es decir, se activa la reacción de parada que provoca la desconexión orientada a la seguridad más rápida de los accionamientos.

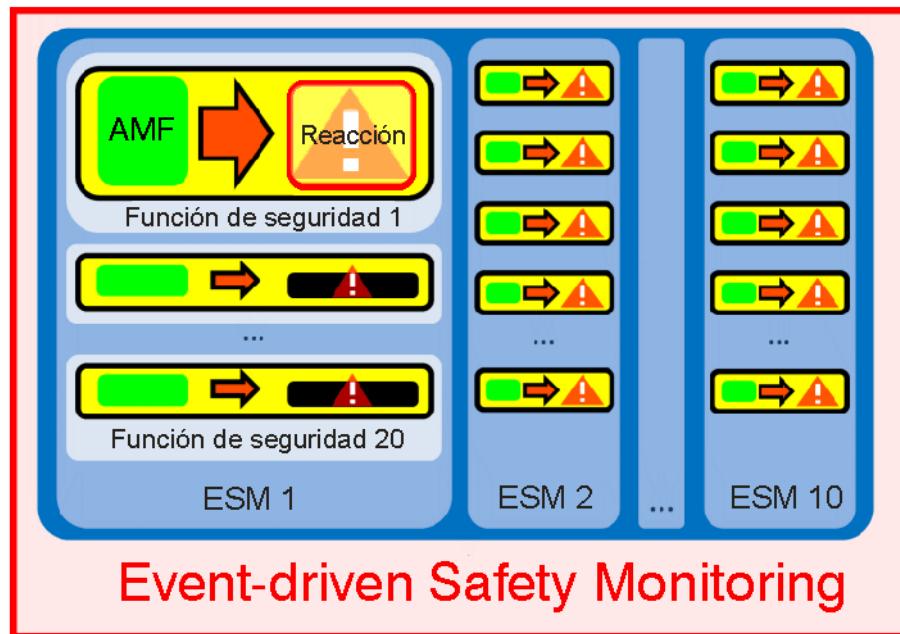


Fig. 13-2: Funciones de seguridad del mecanismo ESM

13.5 Resumen de Atomic Monitoring Functions

La unidad más pequeña de una función de seguridad se denomina Atomic Monitoring Function (AMF), por ejemplo la valoración de la validación en el smartPAD o el control de la velocidad de un eje.

Las Atomic Monitoring Functions se dividen en 3 categorías:

- AMFs estándar
- AMFs parametrizables
- AMFs extendidas

13.5.1 Standard Atomic Monitoring Functions

Descripción Las Standard Atomic Monitoring Functions ofrecen información sobre componentes del sistema o estados del sistema, como por ejemplo, los dispositivos de seguridad en el smartPAD o el modo de servicio activo. Las AMFs estándar se pueden utilizar en el número deseado de funciones de seguridad.

Resumen AMFs para la evaluación de los dispositivos de seguridad en el smartPAD:

AMF	Función
PARADA DE EMERGENCIA smartPAD	Controla el dispositivo de PARADA DE EMERGENCIA en el smartPAD
Validación smartPAD	Controla el pulsador de validación en el smartPAD
Validación pánico smartPAD	Comprueba si un pulsador de validación en el smartPAD está en posición de pánico

(>>> 13.8.1 "Evaluación de los dispositivos de seguridad en el KUKA smartPAD" Página 202)

AMFs para la evaluación del modo de servicio:

AMF	Función
Modo de servicio Test	Comprueba si está activo un modo de servicio de prueba (T1, T2 , KRF)
Modo de servicio Automático	Comprueba si está activo el modo de servicio automático (AUT)
Modo de servicio con velocidad reducida	Comprueba si está activo un modo de servicio con velocidad reducida (T1, KRF)
Modo de servicio de alta velocidad	Comprueba si está activo un modo de servicio con velocidad programada (T2, AUT)

(>>> 13.8.2 "Evaluación del modo de servicio" Página 202)

AMF para la evaluación del movimiento habilitado:

AMF	Función
Movimiento habilitado	Controla el movimiento habilitado, es decir, comprueba si está activa una parada de seguridad.

(>>> 13.8.3 "Evaluación del movimiento habilitado" Página 203)

AMFs para la evaluación del estado de referenciación:

AMF	Función
Referenciación de posición	Controla el estado de referenciación de los valores de posición de todos los ejes (>>> 13.8.4 "Evaluación de la referenciación de posición" Página 203)
Referenciación de momentos	Controla el estado de referenciación de los sensores de momentos de articulación de todos los ejes (>>> 13.8.5 "Evaluación de la referenciación del momento" Página 204)

13.5.2 Atomic Monitoring Functions parametrizables

Descripción En comparación con las AMFs estándar, las Atomic Monitoring Functions parametrizables poseen adicionalmente uno o varios parámetros. Éstos se pueden configurar dependiendo de para qué valores se debe vulnerar la AMF. Por ejemplo límites de control.

Para las AMFs parametrizables se encuentra disponible un número definido fijo de instancias. El número indica la cantidad de versiones parametrizables diferentes de la AMF que se pueden configurar y utilizar. Cada instancia puede poseer diferentes valores de parámetros. Una instancia de una AMF puede utilizarse varias veces en la tabla en la que deben configurarse las funciones de seguridad. Si en cambio son necesarios diferentes ajustes de parámetros, se deberán utilizar instancias adicionales.

Resumen AMF para la evaluación de entradas seguras:

AMF	Función
Señal de entrada	Controla una entrada segura (>>> 13.8.6 "Control de entradas seguras" Página 204)

AMF para la evaluación de la validación en equipos manuales de guiado:

AMF	Función
<i>Validación equipo manual de guiado</i>	Controla dispositivos de validación en equipos manuales de guiado (>>> 13.8.7 "Control de dispositivos de validación en equipos manuales de guiado" Página 205)

AMFs para el control de velocidad:

AMF	Función
<i>Monitorización de la velocidad del eje</i>	Controla la velocidad de un eje (>>> 13.8.8.1 "Definir el control de velocidad específico del eje" Página 207)
<i>Control de velocidad cartesiana</i>	Controla la velocidad cartesiana translatoria en todas las bolas de la herramienta y en los puntos centrales de las uniones articuladas y en la brida del robot (>>> 13.8.8.2 "Definir el control de velocidad cartesiana" Página 208)

AMFs para la monitorización de la zona:

AMF	Función
<i>Control del campo de trabajo cartesiano</i>	Comprueba si una parte de la estructura controlada se encuentra fuera de una zona de trabajo permitida (>>> 13.8.9.1 "Definir campos de trabajo cartesianos" Página 211)
<i>Monitorización cartesiana de las zonas de protección</i>	Comprueba si una parte de la estructura controlada se encuentra dentro de una zona de protección no permitida (>>> 13.8.9.2 "Definir zonas de protección cartesianas" Página 213)
<i>Control del campo del eje</i>	Controla la posición de un eje (>>> 13.8.9.3 "Definir las zonas de control específicas del eje" Página 215)

AMF para el control de la orientación de herramienta:

AMF	Función
<i>Orientación de la herramienta</i>	Comprueba si la orientación de un Frame de orientación de herramienta se encuentra fuera de una zona admisible (>>> 13.8.10 "Control de la orientación de herramienta" Página 217)

AMFs para el control seguro de fuerzas y momentos:

AMF	Función
<i>Control del campo del eje</i>	Controla el par de torsión de un eje (>>> 13.8.13.1 "Control del campo del eje" Página 221)
<i>Identificación de colisión</i>	Controla el momento externo de todos los ejes (>>> 13.8.13.2 "Detección de colisión" Página 221)
<i>Control de fuerza TCP</i>	Controla la fuerza externa que actúa en el TCP de la herramienta o en la brida del robot (>>> 13.8.13.3 "Control de fuerza TCP" Página 222)

13.5.3 Extended Atomic Monitoring Functions

Descripción

Las Extended Atomic Monitoring Functions se distinguen de las AMF estándar y de las AMF parametrizables en que los parámetros de control no se determinan hasta el servicio. Los parámetros se fijan en el momento de la activación. Por ejemplo, en la AMF **Control de parada de todos los ejes**, los ángulos de los ejes se fijan en el momento de la activación como ángulos de referencia para el control. Una Extended AMF se activa si están vulneradas todas las AMF de la función de seguridad utilizadas. Mientras al menos una de las otras AMF no esté vulnerada, la Extended AMF no estará activa y no se evaluará.



Las Extended AMF no se evalúan hasta un ciclo después de su activación. De esta forma puede producirse una prolongación del tiempo de reacción de hasta 12 ms.

Para las Extended AMF hay disponibles varias instancias. Incluso en el caso de Extended AMF no parametrizables, se recomienda utilizar cada instancia solamente una vez en la configuración de seguridad.



Las Extended AMF no están disponibles para las funciones de seguridad del mecanismo de ESM.

Resumen

AMF para el control de parada:

AMF	Función
<i>Control de parada de todos los ejes</i>	Controla la parada de todos los ejes. (>>> 13.8.11 "Control de parada (parada de servicio segura)" Página 219)

AMF para la conmutación de un retardo temporal:

AMF	Función
<i>Retardo de tiempo</i>	Retrasa la activación de la reacción de una función de seguridad un tiempo definido. (>>> 13.8.12 "Retardo de conexión para las funciones de seguridad" Página 220)

13.6 Configuración de seguridad con KUKA Sunrise.Workbench

La configuración de seguridad forma parte de manera fija de un proyecto Sunrise. Se gestiona en forma de tabla.

La configuración de seguridad define las funciones de seguridad del mecanismo PSM y los estados seguros del mecanismo ESM.

Al crear un nuevo proyecto Sunrise se genera de forma automática una configuración de seguridad estándar (>>> 5.3 "Crear proyecto Sunrise con plantilla" Página 51).

La configuración de seguridad estándar contiene funciones de seguridad predefinidas por KUKA y activas de forma permanente. El mecanismo ESM no posee estados preconfigurados y, por ello, está desactivado.

Aquí se puede encontrar más información sobre la configuración de seguridad estándar: (>>> 3 "Seguridad" Página 23)

En KUKA Sunrise.Workbench la configuración de seguridad se muestra, se procesa y a través de la instalación del System Software o de la sincroniza-

ción del proyecto se transmite a la unidad de control. A través de la smartHMI se puede activar y desactivar la configuración de seguridad.

13.6.1 Cambiar la vista de la configuración de seguridad y activar en el control

Paso	Descripción
1	Abrir la configuración de seguridad. (>>> 13.6.2 "Abrir la configuración de seguridad" Página 190)
2	Editar las funciones de seguridad en la tabla <i>Usuario PSM</i> o crear nuevas funciones de seguridad. (>>> 13.6.3 "Configurar las funciones de seguridad del mecanismo de PSM" Página 192)
3	Si es necesario, configurar los controles en función de las incidencias. Para ello, generar estados ESM seguros y configurar las funciones de seguridad correspondientes. Los estados ESM existentes pueden ser modificados adaptando funciones de seguridad ya configuradas o añadiendo nuevas. (>>> 13.6.4 "Configurar estados seguros del mecanismo ESM" Página 195)
4	Guardar la configuración de seguridad.
5	En caso de utilización del mecanismo ESM Programar la conmutación necesaria entre los estados seguros en aplicaciones de robot y tareas en segundo plano. (>>> 13.6.4.8 "Conmutar entre estados ESM" Página 199)
6	En caso de utilización de AMF basadas en la posición (>>> "AMFs basadas en la posición" Página 233) Si es necesario, generar la aplicación preparada por KUKA para la referenciación de posición y del momento del LBR iiwa o una aplicación de recorrido de referencia propia. (>>> 13.10.1 "Referenciación de posición" Página 227)
7	En caso de utilización de AMF basadas en el momento axial (>>> "AMF basadas en el momento del eje" Página 233) Generar la aplicación preparada por KUKA para la referenciación de la posición y el momento del LBR iiwa e integrar la herramienta destinada a la seguridad en la aplicación. Puede ser necesario realizar otras adaptaciones en la aplicación. (>>> 13.10.2 "Referenciación del momento" Página 228)
8	Transmitir el proyecto con la configuración de seguridad a la unidad de control del robot. <ul style="list-style-type: none"> ■ Mediante la instalación del software de sistema o a través de la sincronización del proyecto.
9	Reiniciar la unidad de control del robot para aceptar la configuración de seguridad modificada.
10	Activar la configuración de seguridad en la unidad de control del robot. (>>> 13.7 "Activar la configuración de seguridad en la unidad de control del robot" Página 200)

Paso	Descripción
11	En caso de utilización de AMF basadas en la posición (>>> "AMFs basadas en la posición" Página 233) Efectuar una referenciación de posición.
12	En caso de utilización de AMF basadas en el momento axial (>>> "AMF basadas en el momento del eje" Página 233) Realizar una referenciación del momento.
13	Probar las funciones de seguridad de la configuración de seguridad activada. (>>> 13.11 "Resumen de la recepción de seguridad" Página 230)

13.6.2 Abrir la configuración de seguridad

- | | |
|----------------------|---|
| Procedimiento | <ul style="list-style-type: none"> ■ Hacer doble clic en el archivo SafetyConfiguration.sconf en el proyecto Sunrise. |
| Descripción | <p>La configuración de seguridad contiene varias tablas.</p> <ul style="list-style-type: none"> ■ KUKA PSM
La tabla contiene las funciones de seguridad prescritas por KUKA. Éstas no se pueden modificar ni desactivar. ■ La tabla sirve para documentar el comportamiento del sistema y en combinación con la tabla Usuario PSM ofrece una descripción completa de las funciones de seguridad que están activas continuamente. ■ Usuario PSM
En esta tabla se configuran las funciones de seguridad específicas del usuario. Contiene las funciones de seguridad preconfiguradas por KUKA. Éstas pueden desactivarse, modificarse o borrarse. ■ ESM
Para cada estado ESM se crea una tabla. Ésta contiene las funciones de seguridad del estado. La configuración estándar no contiene estados ESM preconfigurados. |

13.6.2.1 Evaluación de la configuración de seguridad

Durante la evaluación de la configuración de seguridad siempre se comprueban de forma paralela las tablas **Usuario PSM** y **KUKA PSM**. Es posible que las dos tablas contengan una función de seguridad idéntica con diferentes reacciones. Si en este caso se encuentran configuradas diferentes reacciones de parada si se produce una vulneración, se activará la reacción de parada más precisa. Es decir, se activa aquella reacción de parada que ejecute la desconexión de los accionamientos más rápida y orientada a la seguridad.

Si se utiliza el mecanismo ESM, se controlarán adicionalmente todas las funciones de seguridad del estado ESM activo actualmente.

13.6.2.2 Resumen de la configuración de seguridad de la interfaz de usuario

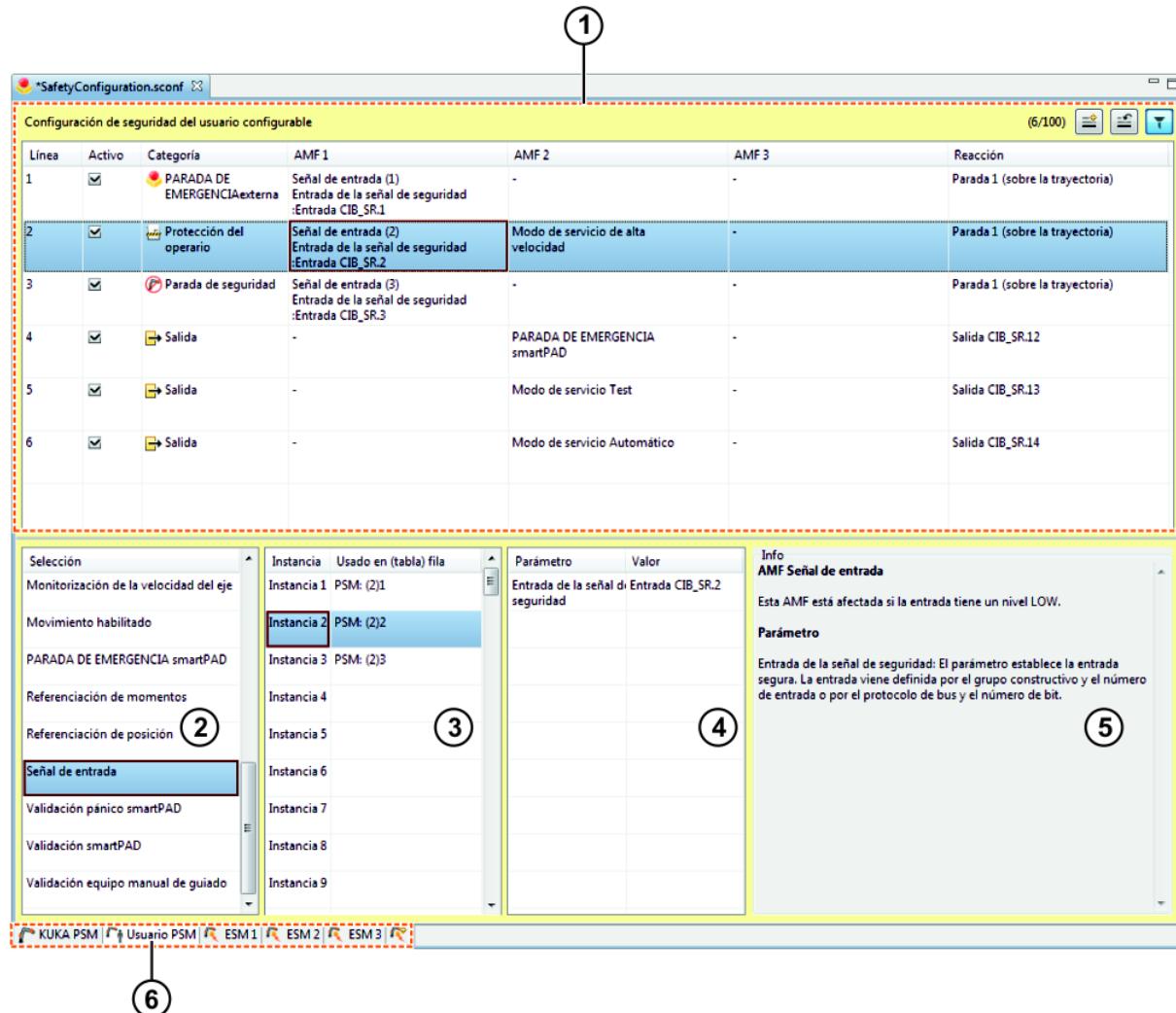


Fig. 13-3: Configuración de seguridad de la interfaz de usuario

Pos.	Descripción
1	Tabla seleccionada Contiene las funciones de seguridad configuradas de la tabla PSM o del estado ESM seleccionado.
2	Tabla de selección En función de la celda que esté marcada en la línea seleccionada de la tabla, aquí se puede seleccionar la categoría, la AMF o la reacción de una función de seguridad.
3	Tabla de instancias Aquí se muestran las instancias de la AMF seleccionada en la tabla de selección, así como las líneas de la tabla en las que se utilizan.
4	Tabla de parámetros Aquí se muestran los valores de parámetros de la instancia de la AMF que está marcada en la table de instancias. Los valores se pueden modificar.

Pos.	Descripción
5	Indicación de información Indicación de la descripción de la categoría, la AMF o la reacción seleccionadas
6	Directorio de tablas Aquí se pueden seleccionar las tablas deseadas y se pueden añadir nuevos estados ESM.

Mediante el directorio de tablas en la zona inferior del editor, se selecciona, se muestra y se edita la tabla.

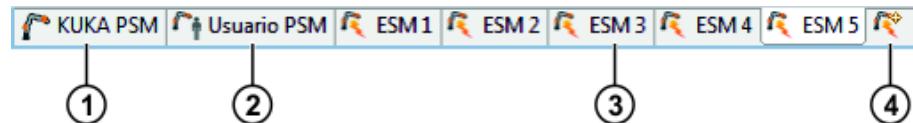


Fig. 13-4: Directorio de tablas de configuración de seguridad

Pos.	Descripción
1	Pestaña KUKA PSM Abre la tabla KUKA PSM. No es posible editar la tabla.
2	Pestaña Usuario PSM Abre la tabla PSM de usuario. La tabla se puede editar.
3	Pestaña para el estado ESM Abre el estado ESM El estado ESM se puede editar.
4	Botón Añadir nuevo estado ESM Añade un nuevo estado ESM. El nuevo estado se abre automáticamente y se puede editar.

13.6.3 Configurar las funciones de seguridad del mecanismo de PSM

El mecanismo de PSM define funciones de seguridad que están activas de forma permanente.

Las funciones de seguridad se representan en forma de tabla. Cada fila de la tabla contiene una función de seguridad.

En la tabla de PSM **Usuario PSM** se añaden nuevas funciones de seguridad y se adaptan los ajustes existentes. Pueden modificarse la categoría, las Atomic Monitoring Functions (AMF) utilizadas, la parametrización de las instancias AMF y la reacción. Las funciones de seguridad pueden activarse o desactivarse individualmente.

13.6.3.1 Abrir la tabla PSM de usuarios

Procedimiento

1. Abrir la configuración de seguridad.
2. Seleccionar la pestaña **Usuario PSM** en el directorio de tablas. La tabla PSM de usuarios se visualiza y puede editarse.

Fig. 13-5: Tabla PSM de usuarios

Pos.	Descripción
1	Columna Activo Determina si la función de seguridad está activa. Las funciones de seguridad desactivadas no serán controladas. <ul style="list-style-type: none"> ■ Símbolo de confirmación marcado: La función de seguridad está activa. ■ Símbolo de confirmación no marcado: La función de seguridad está desactiva.
2	Columna Categoría Establece la categoría de la función de seguridad. La categoría se indica en caso de error en la smartHMI como causa del error.
3	Columnas AMF1, AMF2, AMF3 Establecen las AMFs individuales de la función de seguridad. Es posible utilizar hasta 3 AMFs. La función de seguridad se habrá vulnerado si todas las AMFs utilizadas están vulneradas.
4	Columna Reacción Establece la reacción de la función de seguridad. Se activa cuando se ha vulnerado la función de seguridad.
5	Indicación del número de funciones de seguridad configuradas actualmente En total se encuentran disponibles 100 líneas para la configuración de controles de seguridad específicos del usuario.
6	Botones para la edición de la tabla
7	Línea seleccionada La línea que contiene la función de seguridad seleccionada actualmente aparece con fondo gris.

Están disponibles los siguientes botones:

Botón	Descripción
	Línea nueva Añade una línea nueva en la tabla (solo es posible si están ocultas las líneas por defecto.). La nueva línea dispone de la configuración estándar y se activa automáticamente.
	Restablecer línea Restablece la configuración de la línea marcada a la configuración estándar. La función de seguridad se desactiva.
	Mostrar líneas por defecto / Ocultar líneas por defecto Todas las líneas no configuradas están desactivadas y previamente ocupadas con una configuración estándar. <ul style="list-style-type: none"> ■ Categoría: Ninguno ■ AMF1, AMF2, AMF3: Ninguno ■ Reacción: Parada 1 Estas líneas por defecto pueden mostrarse u ocultarse. De forma estándar, las líneas por defecto están ocultas.

13.6.3.2 Crear funciones de seguridad para mecanismos PSM

- Condición previa** ■ La tabla PSM de usuarios está abierta.
- Procedimiento**
- Se muestran las líneas por defecto:**
 1. Seleccionar una línea por defecto en la tabla.
 2. Ajustar la categoría, las AMFs utilizadas y la reacción de la función de seguridad en las columnas correspondientes.
 3. Marcar el símbolo de confirmación en la columna **Activo** si la línea se debe activar.
 - Se ocultan las líneas por defecto:**
 1. Hacer clic en **Nueva línea**. Se inserta una línea preconfigurada en la tabla. La línea está ajustada automáticamente como activa (símbolo de confirmación en la columna **Activo**).
 2. Ajustar la categoría, las AMFs utilizadas y la reacción de la función de seguridad en las columnas correspondientes.

13.6.3.3 Borrar la función de seguridad del mecanismo PSM

- Condición previa** ■ La tabla PSM de usuarios está abierta.
- Procedimiento**
1. Marcar en la tabla la línea con la función de seguridad que debe borrarse.
 2. Hacer clic en **Restablecer línea**. La función de seguridad se desactiva y recibe la configuración de seguridad (categoría: Ninguna, AMF: Ninguna, reacción: Parada 1).

13.6.3.4 Editar funciones de seguridad existentes del mecanismo PSM

- Condición previa** ■ La tabla PSM de usuarios está abierta.
- Procedimiento**
- Modificar una categoría:**

1. Marcar la columna **Categoría** en las líneas deseadas. En la tabla de selección se muestran las categorías que están disponibles.
2. Marcar la categoría deseada en la tabla de selección. La categoría se acepta para la función de seguridad.

Modificar una AMF utilizada:

1. Marcar la columna **AMF1**, **AMF2** o **AMF3** en la línea deseada. En la tabla de selección se muestran las AMFs que están disponibles.
2. Marcar la AMF deseada en la tabla de selección. La AMF se acepta para la función de seguridad.
3. Para AMFs instanciadas varias veces: Seleccionar la instancia deseada en la tabla de instancias. La instancia se acepta para la función de seguridad.
4. Para AMFs parametrizables: En la tabla de parámetros, ajustar los parámetros de la AMF en la columna **Valor** y aceptar los ajustes con la tecla de entrada.

Modificar una reacción:

1. Marcar la columna **Reacción** en las líneas deseadas. En la tabla de selección se muestran las reacciones que están disponibles.
2. Marcar la reacción deseada en la tabla de selección. La reacción se acepta para la función de seguridad.
3. Si se ha seleccionado la reacción **Salida**: Seleccionar el bit de salida en la tabla de parámetros, cuya señal se debe ajustar en el nivel LOW en caso de vulneración de la función de seguridad. Aceptar el ajuste con la tecla de entrada.

Activar/desactivar una función de seguridad:

- En la línea deseada hacer clic en la columna **Activo**. El símbolo de confirmación se marca/retira.



Solo las funciones de seguridad activas estarán disponibles en la unidad de control del robot después de la transmisión y la activación de la configuración de seguridad.

13.6.4 Configurar estados seguros del mecanismo ESM

Mediante el mecanismo ESM se establecen diferentes ajustes de seguridad mediante estados seguros configurables. Se pueden crear hasta 10 estados seguros. Los estados se numeran de forma consecutiva de 1 a 10 y, de este modo, se pueden identificar de forma única.

Un estado seguro se define en una tabla con hasta 20 funciones de seguridad. Estas funciones de seguridad establecen los ajustes de seguridad que deben aplicarse para el estado.

Un estado seguro se representa en una tabla. Cada fila de la tabla contiene una función de seguridad.

La utilización del mecanismo ESM es opcional. El mecanismo ESM está activado si está configurado al menos un estado ESM. Si no hay estados ESM configurados, el mecanismo está desactivado.

Si el mecanismo ESM está activo, se aplicará siempre exactamente un estado seguro, cuyas funciones de seguridad se controlan de forma adicional a las funciones de seguridad activas de manera permanente. Entre los estados seguros configurados se puede comutar en función de la situación. La comutación se realiza mediante la aplicación del robot o mediante una tarea en segundo plano.

(>>> 13.6.4.8 "Comutar entre estados ESM" Página 199)

Un estado ESM está activo hasta que se ordena la conmutación a otro estado ESM.

Después del arranque del control estará activo automáticamente el estado ESM configurado con el número más bajo.

13.6.4.1 Añadir nuevo estado ESM

Para el mecanismo ESM se pueden crear hasta 10 estados seguros. Cuando se ha alcanzado este número, se oculta la pestaña para añadir estados nuevos.

Procedimiento

1. Abrir la configuración de seguridad.
2. Seleccionar la pestaña **Añadir nuevo estado ESM** en el directorio de tablas. Se crea un nuevo estado ESM.

El nuevo estado ESM contiene el número de estado más pequeño no utilizado. Posee una función de seguridad activa con configuración estándar. En el directorio de tablas se añade una nueva pestaña para el estado. La tabla del estado se abre automáticamente y puede editarse.

13.6.4.2 Abrir la tabla de un estado ESM

Condición previa

- El mecanismo ESM está activo.

Procedimiento

1. Abrir la configuración de seguridad.
2. Seleccionar la pestaña del estado ESM deseado en el directorio de tablas. La tabla del estado ESM se visualiza y puede editarse.

The diagram illustrates the 'Estado para la monitorización de seguridad basada en eventos (ESM)' (Event-based safety monitoring state) table. It features a header row with columns for 'Línea' (Line), 'Activo' (Active), 'AMF' (Advanced Motion Function), and 'Reacción' (Reaction). Below this is a data row for Line 1, which contains a checked checkbox in the 'Activo' column, the text 'Identificación de colisión (1) Momento externo máximo : 15 Nm' in the 'AMF' column, and 'Parada 1' in the 'Reacción' column. A red dashed box highlights the 'AMF' column for Line 1. Callouts numbered 1 through 6 point to specific elements: 1 points to the table title; 2 points to the 'AMF' column header; 3 points to the 'Reacción' column header; 4 points to the '(2/20)' count in the top right; 5 points to the toolbar icons; and 6 points to the bottom-left corner of the table.

Línea	Activo	AMF	Reacción
1	<input checked="" type="checkbox"/>	Identificación de colisión (1) Momento externo máximo : 15 Nm	Parada 1
2	<input checked="" type="checkbox"/>	Control de velocidad cartesiana (2) Velocidad máxima : 100 mm/s	Parada 1

Fig. 13-6: Tabla de un estado ESM

Pos.	Descripción
1	<p>Columna Activo</p> <p>Determina si la función de seguridad está activa. Las funciones de seguridad desactivadas no serán controladas.</p> <ul style="list-style-type: none"> ■ Símbolo de confirmación marcado: La función de seguridad está activa. ■ Símbolo de confirmación no marcado: La función de seguridad está desactiva. <p>La función de seguridad en la primera línea de la tabla siempre está activa. No se puede desactivar (señalizada con el símbolo de candado).</p>
2	<p>Columna AMF</p> <p>Establece la AMF de la función de seguridad. Para las funciones de seguridad de estados ESM solo se utiliza una AMF. Si esta AMF está vulnerada, se habrá vulnerado la función de seguridad y, de este modo, el estado completo.</p>
3	<p>Columna Reacción</p> <p>Establece la reacción de la función de seguridad. Se activa cuando se ha vulnerado la función de seguridad.</p>
4	<p>Indicación del número de funciones de seguridad configuradas actualmente</p> <p>En total se encuentran disponibles 20 líneas para la configuración de controles de seguridad de un estado ESM.</p>
5	Botones para la edición de la tabla
6	Línea seleccionada La línea que contiene la función de seguridad seleccionada actualmente aparece con fondo gris.

Están disponibles los siguientes botones:

Botón	Descripción
	<p>Borrar estado</p> <p>Borra el estado completo. El proceso de borrar se debe confirmar mediante un diálogo.</p>
	<p>Línea nueva</p> <p>Añade una línea nueva en la tabla (solo es posible si están ocultas las líneas por defecto.). La nueva línea dispone de la configuración estándar y se activa automáticamente.</p>

Botón	Descripción
	Restablecer línea Restablece la configuración de la línea marcada a la configuración estándar. La función de seguridad se desactiva (excepción: La primera línea de la tabla siempre está activa).
	Mostrar líneas por defecto / Ocultar líneas por defecto Todas las líneas no configuradas están desactivadas y previamente ocupadas con una configuración estándar. <ul style="list-style-type: none"> ■ AMF: Ninguno ■ Reacción: Parada 1 Estas líneas por defecto pueden mostrarse u ocultarse. De forma estándar, las líneas por defecto están ocultas.

13.6.4.3 Borrar estado ESM

Procedimiento

1. Abrir la configuración de seguridad.
2. Seleccionar la pestaña del estado ESM que se desea borrar en el directorio de tablas.
3. Hacer clic en **Borrar estado**.
4. Responder **Sí** a la pregunta de seguridad. El estado se borra.

El estado ESM se elimina automáticamente una vez guardada y cerrada la configuración de seguridad, siempre y cuando posea los siguientes ajustes:

- Todas las filas poseen la configuración estándar (AMF: ninguna, reacción: Parada 1).
- La primera fila está activada, el resto de filas está desactivado.

13.6.4.4 Crear una función de seguridad para el estado ESM

Condición previa

- La tabla del estado ESM deseado está abierta.

Procedimiento

Se muestran las líneas por defecto:

1. Seleccionar una línea por defecto en la tabla.
2. Ajustar la AMF utilizada y la reacción de la función de seguridad en las columnas correspondientes.
3. Marcar el símbolo de confirmación en la columna **Activo** si la línea se debe activar.

Se ocultan las líneas por defecto:

1. Hacer clic en **Nueva línea**. Se inserta una línea preconfigurada en la tabla. La línea está ajustada automáticamente como activa (símbolo de confirmación en la columna **Activo**).
2. Ajustar la AMF utilizada y la reacción de la función de seguridad en las columnas correspondientes.

13.6.4.5 Borrar la función de seguridad de un estado ESM

Condición previa

- La tabla del estado ESM deseado está abierta.

Procedimiento

1. Marcar en la tabla la línea con la función de seguridad que debe borrarse.

2. Hacer clic en **Restablecer línea**. La función de seguridad se desactiva y recibe la configuración de seguridad (AMF: ninguna, reacción: Parada 1).

13.6.4.6 Editar una función de seguridad existente de un estado ESM

Condición previa ■ La tabla del estado ESM deseado está abierta.

Procedimiento **Modificar una AMF utilizada:**

1. Marcar la columna **AMF** en las líneas deseadas. En la tabla de selección se muestran las AMFs que están disponibles.
2. Marcar la AMF deseada en la tabla de selección. La AMF se acepta para la función de seguridad.
3. Para AMFs instanciadas varias veces: Seleccionar la instancia deseada en la tabla de instancias. La instancia se acepta para la función de seguridad.
4. Para AMFs parametrizables: En la tabla de parámetros, ajustar los parámetros de la AMF en la columna **Valor** y aceptar los ajustes con la tecla de entrada.

Modificar una reacción:

1. Marcar la columna **Reacción** en las líneas deseadas. En la tabla de selección se muestran las reacciones que están disponibles.
2. Marcar la reacción deseada en la tabla de selección. La reacción se acepta para la función de seguridad.

Activar/desactivar una función de seguridad:

- En la línea deseada hacer clic en la columna **Activo**. El símbolo de confirmación se marca/retira.

13.6.4.7 Desactivar el mecanismo ESM

La utilización del mecanismo ESM es opcional. Se puede desactivar.

Procedimiento ■ Borrar todos los estados ESM.

13.6.4.8 Comutar entre estados ESM

Con el método `setESMState(...)` se puede activar un estado ESM y se puede comutar entre los diferentes estados ESM. El método pertenece a la clase LBR y se puede utilizar en aplicaciones del robot o en tareas en segundo plano.

Sintaxis

`lbr.setESMState(state);`

Aclaración de la sintaxis

Elemento	Descripción
<code>lbr</code>	Tipo: LBR Nombre del robot para el que se activa el estado ESM
<code>state</code>	Tipo: String Número del estado ESM que se activa ■ 1 ... 10 Si se indica un estado ESM no configurado, el robot se detiene con una parada de seguridad 1.

Ejemplo

El LBR iiwa se debe guiar manualmente en una aplicación. Para ello se realiza el desplazamiento hasta una posición inicial adecuada. Para el desplaza-

miento hasta la posición inicial, debe activarse el estado ESM 3. El estado ESM 3 garantiza una identificación de colisión sensible y controla la velocidad cartesiana.

Después de alcanzar el punto de inicio, debe comenzar el guiado manual. Para el guiado manual se debe activar el estado ESM 8. El estado ESM 8 requiere la validación en el equipo manual de guiado, pero permite una velocidad cartesiana elevada como estado ESM 3.

```
private LBR lbr;  
...  
  
public void run() {  
    ...  
    lbr.setESMState("3");  
    lbr.move(lin(getFrame("Start")).setCartVelocity(300));  
  
    lbr.setESMState("8");  
    lbr.move(handGuiding());  
    ...  
}
```

13.7 Activar la configuración de seguridad en la unidad de control del robot

Descripción

Una configuración de seguridad es efectiva si se activa después de la transmisión a la unidad de control del robot a través de la smartHMI. Si no hay ninguna configuración de seguridad activa, el robot no puede seguir desplazándose.

Con la activación se asigna a la configuración de seguridad una ID única (= suma de control de la configuración de seguridad). Ésta se muestra en el nivel **Seguridad** en la zona **Activación**. Con esta ID, el técnico de puesta en servicio puede identificar de forma única la configuración de seguridad activada en la unidad de control del robot.

Para la instalación del System Software es necesario el reinicio de la unidad de control del robot. Si con la instalación se transmite una configuración de seguridad modificada, se deberá activar después del reinicio.

Si se transmiten modificaciones de una configuración de seguridad activa a través de la sincronización del proyecto en la unidad de control del robot, el comportamiento del sistema depende de si la sincronización se ha finalizado con el reinicio de la unidad de control del robot:

- Sincronización con reinicio: La configuración de seguridad anterior ya no está activa y la nueva aún no está activa. El robot ya no se puede desplazar.
- Sincronización sin reinicio: La configuración de seguridad anterior permanece activa hasta que se reinicie la unidad de control del robot. Hasta este momento, el robot se puede desplazar.



Para el caso de que no se deba activar la configuración de seguridad transmitida, se podrá restablecer la configuración de seguridad anterior.

(>>> 13.7.2 "Restaurar la configuración de seguridad" Página 201)

Para la activación es necesaria la introducción de una contraseña. La contraseña por defecto es "argus".

! La contraseña para la activación de la configuración de seguridad ha de modificarse antes de la puesta en servicio. Dicha contraseña solo debe confiarse a técnicos de puesta en servicio con seguridad formados y que estén autorizados para activar la configuración de seguridad.
 (>>> 13.7.3 "Modificar la contraseña para la activación de la configuración de seguridad" Página 201)

- Procedimiento**
1. En la vista de la estación seleccionar **Seguridad > Activación**.
 2. Introducir la contraseña y pulsar **Activar**.
 3. Solo es necesario si no se ha ejecutado ningún reinicio después de la instalación del System Software o de la sincronización del proyecto: Reiniciar la unidad de control del robot.

13.7.1 Desactivar la configuración de seguridad

- Descripción** Una configuración de seguridad activada se puede desactivar de nuevo. El técnico de puesta en servicio debe comprobar que la desactivación se ha realizado correctamente. Si la configuración de seguridad está desactivada, el robot no puede seguir desplazándose. Para ello se visualiza un mensaje en la vista de la estación (cuadro **Seguridad**).

- Procedimiento**
1. En la vista de la estación seleccionar **Seguridad > Activación**.
 2. Introducir la contraseña y pulsar **Desactivar**.

13.7.2 Restaurar la configuración de seguridad

- Descripción** Esta función solo está disponible si se ha transmitido una nueva configuración de seguridad a la unidad de control del robot, pero no se ha activado. Con este procedimiento se puede restaurar la última configuración de seguridad activada.

- Procedimiento**
1. En la vista de la estación seleccionar **Seguridad > Activación**.
 2. Introducir la contraseña y pulsar **Restablecer**.
 3. Solo es necesario si no se ha ejecutado ningún reinicio después de la sincronización del proyecto: Reiniciar la unidad de control del robot.

13.7.3 Modificar la contraseña para la activación de la configuración de seguridad

- Procedimiento**
1. En la vista de la estación seleccionar **Seguridad > Nueva contraseña**.
 2. Introducir la contraseña anterior en el campo **Contraseña**.
 3. En los campos situados debajo, introducir la nueva contraseña 2 veces. Por motivos de seguridad los datos introducidos están codificados. Se diferencia entre mayúsculas y minúsculas.
 4. Pulsar **Cambiar**. A partir de ahora está vigente el código de acceso nuevo.

i Si se ha olvidado la contraseña, se deberá sustituir la tarjeta de memoria en la unidad de control del robot por el servicio técnico de KUKA. A continuación, se deberá modificar de nuevo la contraseña para la activación de la configuración de seguridad, basándose en la contraseña por defecto.

13.8 Utilización y parametrización de las Atomic Monitoring Functions

A continuación se describen la utilización, el modo de funcionamiento y parametrización de las AMFs individuales. La descripción se complementa con ejemplos de configuración.

13.8.1 Evaluación de los dispositivos de seguridad en el KUKA smartPAD

El KUKA smartPAD dispone de un dispositivo de PARADA DE EMERGENCIA y de un dispositivo de validación. Las correspondientes funciones orientadas a la seguridad están preconfiguradas en la tabla KUKA PSM y no se pueden modificar.

Se pueden configurar otras funciones de seguridad que evalúan los dispositivos de seguridad en el smartPAD. Aquí se encuentran disponibles las siguientes AMFs:

AMF	Descripción
PARADA DE EMERGENCIA smartPAD	Esta AMF está vulnerada, si está pulsado el dispositivo de PARADA DE EMERGENCIA en el smartPAD. AMF estándar
Validación smartPAD	Esta AMF está vulnerada, si no está presionado ningún pulsador de validación en el smartPAD o si está presionado a fondo un pulsador de validación (posición de pánico). AMF estándar
Validación pánico smartPAD	Esta AMF está vulnerada, si está presionado a fondo un pulsador de validación (posición de pánico). AMF estándar

13.8.2 Evaluación del modo de servicio

El modo de servicio ajustado ejerce efectos considerables sobre el comportamiento del robot industrial y determina las medidas de seguridad que son necesarias.

Para configurar una función de seguridad que evalúe el modo de servicio ajustado, se encuentran disponibles las siguientes AMFs:

AMF	Descripción
Modo de servicio Test	Esta AMF está vulnerada, si se encuentra activo un modo de servicio de prueba (T1, T2, KRF). AMF estándar
Modo de servicio Automático	Esta AMF está vulnerada, si está activo un modo de servicio automático (AUT). AMF estándar
Modo de servicio con velocidad reducida	Esta AMF está vulnerada, si se encuentra activo un modo de servicio en el que la velocidad esté reducida a máx. 250 mm/s (T1, KRF). AMF estándar
Modo de servicio de alta velocidad	Esta AMF está vulnerada, si está activo un modo de servicio en el que el robot se desplaza con una velocidad programada (T2, AUT). AMF estándar

13.8.3 Evaluación del movimiento habilitado

El robot no se puede desplazar sin el movimiento habilitado. El movimiento habilitado se puede haber retirado por diferentes motivos, por ejemplo si en el modo de prueba no se ha concedido la validación o si se ha pulsado la PARADA DE EMERGENCIA en el smartPAD.

La AMF para el movimiento habilitado funciona como una señal colectiva para todas las condiciones de parada configuradas. Se puede utilizar especialmente para la desconexión de los dispositivos periféricos. Por ello, para las funciones de seguridad que contienen la evaluación del movimiento habilitado, deberá estar configurada una salida segura como reacción. Si como reacción se ajusta una parada de seguridad, el robot no se podrá desplazar.

AMF	Descripción
Movimiento habilitado	<p>Esta AMF está vulnerada, si no se ha concedido el movimiento habilitado debido a una solicitud de parada.</p> <p>Indicación: Esta AMF solo se puede utilizar de forma práctica con una salida como reacción.</p> <p>AMF estándar</p>

Ejemplo Desconectar la herramienta (categoría: **Salida**)

Una herramienta (p. ej. un láser) que esté conectada a una salida, se deberá desconectar en caso de retirada del movimiento habilitado. La desconexión solo debe tener lugar si se ha vulnerado la protección del operario.

AMF1	AMF2	AMF3	Reacción
Señal de entrada (protección del usuario)	-	Movimiento habilitado	Salida (herramienta)

13.8.4 Evaluación de la referenciación de posición

En la referenciación de posición se comprueba si la posición cero guardada del motor de un eje (= posición de ajuste guardada) coincide con la posición cero mecánica real de dicho eje.

Mientras que no se ejecute una referenciación de posición, estará limitada la integridad de seguridad de las funciones de seguridad en las que se basa. Aquí se incluyen posiciones del robot cartesianas y específicas del eje controladas de forma segura, velocidades cartesianas, fuerzas y colisiones.

Con la AMF **Referenciación de posición** se puede comprobar si los valores de posición de todos los ejes están referenciados.

AMF	Descripción
Referenciación de posición	<p>Esta AMF está vulnerada, si la posición de al menos un eje de la cinemática no ha sido referenciada, o si ha fallado la referenciación de posición de al menos un eje.</p> <p>AMF estándar</p>

Ejemplo Controlar el estado de la referenciación de posición (categoría: **Parada de seguridad**)

Para evitar un peligro en una referenciación de posición no realizado o que haya fallado, el robot no se podrá desplazar con ejes no referenciados y solo se podrá desplazar con una velocidad reducida de 250 mm/s como máximo.

Para garantizarlo, se controla el estado de referenciación de todos los ejes en los modos de servicio con velocidad elevada (T2 y AUT) y se activa una parada de seguridad 1 (sobre la trayectoria), en cuanto no esté correctamente referenciada la posición de al menos un eje.

AMF1	AMF2	AMF3	Reacción
Modo de servicio de alta velocidad	-	Referenciación de posición	Parada 1 (sobre la trayectoria)

13.8.5 Evaluación de la referenciación del momento

Durante la referenciación de los sensores de momentos de articulaciones se comprueba si el par de torsión externo esperado, que se puede calcular para un eje mediante el modelo de robot y los datos de carga indicados, coincide con el valor calculado basado en el valor de medición del sensor de momentos de articulaciones. Si la diferencia entre este valor y un valor de tolerancia determinado supera la diferencia, habrá fallado la referenciación de los sensores de momentos.

Mientras que no se ejecute correctamente la referenciación del momento, estará limitada la integridad de seguridad de las funciones de seguridad en las que se basa. Aquí se incluyen el control de momentos de los ejes y de la fuerza TCP, así como la detección de colisión.

Con la AMF *Referenciación de momentos* se puede comprobar si están referenciados los sensores de momentos de articulaciones de todos los ejes.

AMF	Descripción
<i>Referenciación de momentos</i>	La AMF está vulnerada si no está referenciado el sensor de momentos de articulaciones de al menos un eje, o si ha fallado la referenciación de al menos un sensor de momentos de articulaciones. AMF estándar

Ejemplo

Controlar el estado de la referenciación del momento (categoría: *Parada de seguridad*)

Para una estación está configurada una detección de colisión segura. Si se detecta un momento de más de 20 Nm en al menos un eje del robot, se activará una parada de seguridad 0. Debido a que la integridad de seguridad de esta función solo está garantizada si los sensores de momentos de articulaciones están referenciados correctamente, se debe controlar al mismo tiempo el estado de referenciación de los sensores. En el momento en que se detecte al menos un sensor de momentos de articulaciones no referenciado o referenciado de forma incorrecta, se deberá activar una parada de seguridad 1 (sobre la trayectoria) en los modos de servicio con una velocidad elevada (T2 y AUT).

AMF1	AMF2	AMF3	Reacción
Detección de colisión	-	-	Parada 0
Modo de servicio con velocidad elevada	-	Referenciación de momentos	Parada 1 (sobre la trayectoria)

13.8.6 Control de entradas seguras

Descripción

Como entradas seguras se pueden utilizar las entradas de la interfaz de seguridad discreta o las entradas seguras de la interfaz de seguridad de Ethernet si están configuradas en WorkVisual.

(>>> "Interfaces de seguridad" Página 181)

En las entradas seguras se pueden conectar dispositivos de seguridad, como por ejemplo, dispositivos de PARADA DE EMERGENCIA externos o puertas de protección. Para evaluar la señal de entrada correspondiente, se utiliza la AMF *Señal de entrada*.



Si se emplea un robot con una brida de medios Touch, se pueden emplear en la AMF las entradas seguras en las que hay conectadas pulsadores de confirmación y pulsadores de pánico de la brida de medios.

AMF	Descripción
<i>Señal de entrada</i>	Esta AMF está vulnerada, si la entrada segura utilizada tiene el nivel LOW (estado "0"). AMF parametrizable con 64 instancias disponibles

Parámetros de la AMF	Parámetros	Descripción
	<i>Entrada de la señal de seguridad</i>	Entrada segura que debe controlarse

Ejemplo 1	Protección del operario (categoría: <i>Protección del operario</i>) En una entrada segura se conecta una puerta de protección. Si la puerta de protección se abre en el modo de servicio automático o T2, se debe activar una parada de seguridad 1 (sobre la trayectoria).
------------------	---

AMF1	AMF2	AMF3	Reacción
<i>Señal de entrada</i>	<i>Modo de servicio de alta velocidad</i>	-	<i>Parada 1 (sobre la trayectoria)</i>

Ejemplo 2	PARADA DE EMERGENCIA externa (categoría: <i>PARADA DE EMERGENCIA externa</i>) En una entrada segura se conecta un dispositivo de PARADA DE EMERGENCIA externo. Si se pulsa el dispositivo de PARADA DE EMERGENCIA externo, se debe activar una parada de seguridad 1 (sobre la trayectoria).
------------------	--

AMF1	AMF2	AMF3	Reacción
<i>Señal de entrada</i>	-	-	<i>Parada 1 (sobre la trayectoria)</i>

13.8.7 Control de dispositivos de validación en equipos manuales de guiado

Descripción	Un caso de aplicación en el marco de la cooperación hombre-robot es el guiado del robot manualmente para, p. ej. programar por aprendizaje los puntos de una trayectoria. Para ello es necesario un equipo manual de guiado con dispositivos de validación.
--------------------	---



Cuando un robot se guía manualmente, debe estar instalado un dispositivo de PARADA DE EMERGENCIA que siempre esté al alcance del operario.

La AMF *Validación equipo manual de guiado* sirve para la evaluación de dispositivos de validación de 3 niveles. Se pueden configurar hasta 3 pulsadores de validación y hasta 3 interruptores de pánico. También se pueden evaluar dispositivos de validación de 3 niveles con solo una salida, que procesan la señal de pánico de forma interna.

La AMF cumple los siguientes requisitos y medidas normativas para evitar usos incorrectos previsibles:

- La validación no se concede si el pulsador de validación, después de ser pulsado a fondo, se suelta de nuevo hasta la posición central.
- La validación se retira con una solicitud de parada. Para volver a conceder la validación posteriormente, se deberá soltar el pulsador de validación y presionarse de nuevo.
- La validación se concede 100 ms después de presionar el pulsador de validación.

En caso de utilización de varios pulsadores de validación se aplica lo siguiente:

- Si los 3 pulsadores de validación se mantienen al mismo tiempo en la posición central, se activará una parada de seguridad 1.
- Se pueden mantener pulsados al mismo tiempo 2 pulsadores de validación hasta 15 segundos en la posición intermedia. Ello permite cambiar de un pulsador de validación a otro. Si los pulsadores de validación se mantienen pulsados a la vez en la posición intermedia durante más de 15 segundos, esto activa una parada de seguridad 1.

AMF	Descripción
<i>Validación equipo manual de guiado</i>	<p>Esta AMF está vulnerada, si todas las entradas seguras en las que está conectado un pulsador de validación, tienen el nivel LOW (estado "0"), o si al menos una de las entradas seguras en las que está conectado un interruptor de pánico, tienen el nivel LOW (estado "0").</p> <p>AMF parametrizable con una instancia disponible</p>



Si se emplea un robot con una brida de medios Touch, se pueden emplear en la AMF las entradas seguras en las que hay conectadas pulsadores de confirmación y pulsadores de pánico de la brida de medios.

Parámetros de la AMF

Parámetros	Descripción
<i>Usado pulsador de validación 1</i>	Indica si el pulsador de validación está conectado en una entrada segura.
<i>Usado pulsador de validación 2</i>	<ul style="list-style-type: none"> ■ true: Se encuentra conectada una entrada. ■ false: No se encuentra conectada ninguna entrada.
<i>Usado pulsador de validación 3</i>	
<i>Señal de entrada del pulsador de validación 1</i>	Entrada segura en la que está conectado el pulsador de validación
<i>Señal de entrada del pulsador de validación 2</i>	Como entradas seguras se pueden utilizar las entradas de la interfaz de seguridad discreta o las entradas seguras de la interfaz de seguridad de Ethernet, si están configuradas en WorkVisual.
<i>Señal de entrada del pulsador de validación 3</i>	(>>> "Interfaces de seguridad" Página 181)
<i>Usado pulsador de pánico 1</i>	Indica si el interruptor de pánico está conectado en una entrada segura.
<i>Usado pulsador de pánico 2</i>	
<i>Usado pulsador de pánico 3</i>	<ul style="list-style-type: none"> ■ true: Se encuentra conectada una entrada. ■ false: No se encuentra conectada ninguna entrada.

Parámetros	Descripción
Señal de entrada del pulsador de pánico 1	Entrada segura en la que está conectado el pulsador de pánico
Señal de entrada del pulsador de pánico 2	Como entradas seguras se pueden utilizar las entradas de la interfaz de seguridad discreta o las entradas seguras de la interfaz de seguridad de Ethernet, si están configuradas en WorkVisual.
Señal de entrada del pulsador de pánico 3	(>>> "Interfaces de seguridad" Página 181)

Ejemplo Guiado manual con validación (categoría: *Validación*)

Un robot equipado con un equipo manual de guiado debe guiarse manualmente en una zona determinada para programar por aprendizaje los puntos de una trayectoria. La zona para el guiado manual se define mediante una zona de protección cartesiana. En esta zona de protección no se deberá superar una velocidad cartesiana de 250 mm/s, a menos que se haya concedido una validación mediante el equipo manual de guiado. De lo contrario, se activará una parada de seguridad 1 (sobre la trayectoria).

AMF1	AMF2	AMF3	Reacción
Validación equipo manual de guiado	Monitorización cartesiana de las zonas de protección	Control de velocidad cartesiana	Parada 1 (sobre la trayectoria)

13.8.8 Controles de velocidad

Un robot en movimiento siempre supone un peligro para las personas que se encuentren en sus proximidades. Para la protección de las personas puede ser necesario que no se supere una velocidad definida para, por ejemplo, permitir una desviación a tiempo. Para ello se debe controlar la velocidad de forma permanente.

Se pueden controlar velocidades específicas de los ejes y la velocidad cartesiana del robot.

13.8.8.1 Definir el control de velocidad específico del eje

Para definir el control de velocidad específico del eje, se utiliza la AMF **Monitorización de la velocidad del eje**.

AMF	Descripción
Monitorización de la velocidad del eje	Esta AMF está vulnerada, si el valor de la velocidad del eje controlado supera el límite de velocidad definido. AMF parametrizable con 16 instancias disponibles

Parámetros de la AMF	Parámetro	Descripción
	Eje controlado	Eje que debe controlarse <ul style="list-style-type: none"> ■ Eje1 ... Eje16 Indicación: Para el KUKA LBR iiwa se utilizan Eje1 ... Eje7 .
	Velocidad máxima [°/s]	Máxima velocidad permitida con la que el eje controlado se puede desplazar en sentido de giro positivo y negativo <ul style="list-style-type: none"> ■ 1 ... 500 °/s

13.8.8.2 Definir el control de velocidad cartesiana

Para definir un control de velocidad cartesiano, se utiliza la AMF *Control de velocidad cartesiana*. Se controla la velocidad cartesiana translatoria en todos los puntos centrales de los ejes y en la brida del robot.

Si está activa una herramienta destinada a la seguridad en la unidad de control del robot, se controla adicionalmente la velocidad en los puntos centrales de las bolas con la que está configurada la herramienta destinada a la seguridad.

(>>> 9.3.7 "Herramienta destinada a la seguridad" Página 150)

 No se controla la vulneración de un límite de velocidad por parte de la estructura completa del robot y de la herramienta, sino solo las bolas de control. Especialmente para las herramientas de descarga, las bolas de control de las herramientas orientadas a la seguridad se deben planificar y configurar de forma que esté garantizada la integridad de la seguridad del control de velocidad.

AMF	Descripción
Control de velocidad cartesiana	<p>Esta AMF está vulnerada, si la velocidad cartesiana supera el límite definido en al menos uno de los puntos controlados si uno de los ejes del robot están desajustados o si ha fallado la referenciación de posición de un eje ajustado.</p> <p>AMF parametrizable con 8 instancias disponibles</p> <p>Indicación: En caso de que esté vulnerada la AMF por la pérdida del ajuste, el robot solo se podrá desplazarse y ajustarse de nuevo mediante un cambio al modo de servicio KRF.</p>

Parámetros de la AMF	Parámetro	Descripción
	Velocidad máxima [mm/s]	<p>Máxima velocidad cartesiana permitida que no se debe superar en ninguno de los puntos controlados</p> <ul style="list-style-type: none"> ■ 1 ... 10 000 mm/s

Ejemplo

Categoría: Control de velocidad

Si se vulnera un campo de trabajo cartesiano, la velocidad cartesiana del robot puede ser como máximo de 300 mm/s. Si esta velocidad se supera, se deberá activar una parada de seguridad 1.

AMF1	AMF2	AMF3	Reacción
Control del campo de trabajo cartesiano	-	Control de velocidad cartesiana	Parada 1

13.8.9 Zonas de control

El entorno del robot se puede dividir en zonas en las que debe permanecer para la ejecución de la aplicación y en zonas en las que no debe encontrarse o solo debe encontrarse bajo determinadas condiciones. En este caso se debe comprobar constantemente si hay partes de la estructura del robot dentro o fuera de una de estas zonas de control.

Una zona de control se puede definir como cuadrado cartesiano o mediante campos del eje individuales.

Una zona de control cartesiana se puede configurar como campo de trabajo en el que debe permanecer el robot o como zona de protección en la que no puede entrar.

Mediante la combinación con otras Atomic Monitoring Functions se pueden establecer otras condiciones que deberán cumplirse en caso de vulneración de una zona de control. Así, por ejemplo, una zona de control puede ser activable a través de una entrada segura o solo puede ser válida en el modo automático.



Si el robot ha vulnerado una zona controlada de forma segura y es detenido por el control de seguridad, puede ser desplazado fuera de la zona vulnerada en el modo de servicio KRF.

(>>> 6.9 "Modo de servicio KRF – Desbloquear el robot de forma controlada" Página 82)

Bolas en el robot

Alrededor de puntos escogidos del robot se encuentran modeladas bolas que envuelven el robot y se desplazan junto con el mismo. Estas bolas están especificadas de forma fija y se controlan por defecto en relación con los límites en las zonas de control cartesianas.

Los centros y los radios de las bolas controladas están establecidos en los datos de la máquina del robot. Para cada eje del robot, para el pie del robot y la brida del robot se encuentra definida una bola. El centro de la bola se encuentra respectivamente en el punto central del eje, del pie del robot o de la brida del robot.

Las dimensiones de las bolas controladas varían según el tipo de robot y la brida de medios empleada:

- r = radio de bola
- z, y = punto central de la bola con respecto al sistema de coordenadas del pie del robot

Variante 1: LBR iiwa 7 R800 con brida de medios Touch

	Pie	A1	A2	A3	A4	A5	A6	A7	Brida
r [mm]	135	90	125	90	125	90	80	85	65
z [mm]	50	90	340	538	740	935	1140	1130	1240
y [mm]							-30		

Variante 2: LBR iiwa 7 R800 con brida de medios (todas las variantes salvo brida de medios Touch)

	Pie	A1	A2	A3	A4	A5	A6	A7	Brida
r [mm]	135	90	125	90	125	90	80	85	65
z [mm]	50	90	340	538	740	935	1140	1130	1220
y [mm]							-30		

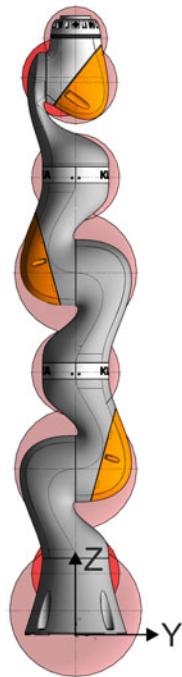


Fig. 13-7: Bolas del LBR iiwa 7 R800 (variante 2)

Variante 3: LBR iiwa 14 R820 con brida de medios Touch

	Pie	A1	A2	A3	A4	A5	A6	A7	Brida
r [mm]	150	100	140	90	131	90	80	85	65
z [mm]	50	160	360	580	780	980	1180	1170	1280
y [mm]							-30		

Variante 4: LBR iiwa 14 R820 con brida de medios (todas las variantes salvo brida de medios Touch)

	Pie	A1	A2	A3	A4	A5	A6	A7	Brida
r [mm]	150	100	140	90	131	90	80	85	65
z [mm]	50	160	360	580	780	980	1180	1170	1260
y [mm]							-30		

Bolas en la herramienta Si está activa una herramienta destinada a la seguridad en la unidad de control del robot, además de las bolas en el robot, también se controlan por defecto las bolas con la que está configurada la herramienta destinada a la seguridad.

(>>> 9.3.7 "Herramienta destinada a la seguridad" Página 150)



No se controla la vulneración de la zona por parte de la estructura completa del robot y de la herramienta, sino solo las bolas de control. Especialmente para las herramientas de descarga, las bolas de control de las herramientas destinadas a la seguridad se deben planificar y configurar de forma que esté garantizada la integridad de seguridad del control del campo de trabajo y de la monitorización de la zona de protección.

Selección de las bolas de control

En cada caso de aplicación no es necesario o no resulta práctico incluir todas las bolas de los robots o las herramientas en la monitorización de la zona cartesiana.

Ejemplo: Si la entrada de la herramienta en una zona de protección debe servir para activar otros controles, solo se deben controlar las bolas de las herramientas.

La estructura a controlar se puede seleccionar durante la configuración de las zonas de control cartesianas:

- Robot y herramienta (por defecto)
- Solo herramienta
- Solo robot

Distancia de parada

Cuando un robot se detiene debido al control, necesita recorrer una distancia de parada hasta detenerse definitivamente.

La distancia de parada depende esencialmente de los siguientes factores:

- Tipo de robot
- Velocidad del robot
- Posición de los ejes del robot
- Carga

 **ADVERTENCIA**

La distancia de parada del robot depende esencialmente de la dinámica del tipo de robot. Dependiendo del tipo de robot, en caso de error, el robot acelerará con diferente intensidad dentro del tiempo de reacción de las funciones de control. Esto influye en la distancia de parada real.

En lo que se refiere a la evaluación de la seguridad, el integrador de sistemas debe tener en cuenta este aspecto durante la parametrización de las funciones de control.

13.8.9.1 Definir campos de trabajo cartesianos

Descripción

Un campo de trabajo cartesiano se define como cuadrado, cuya posición y orientación en el espacio se establece en relación con el sistema de coordenadas Mundo.

Las bolas de control se controlan con respecto a los límites de los campos de trabajo cartesianos activados y se deben mover dentro de estos campos de trabajo.

Para definir un campo de trabajo cartesiano, se utiliza la AMF **Control del campo de trabajo cartesiano**. Esta AMF está vulnerada, si una de las bolas controladas ya no se encuentra totalmente dentro del campo de trabajo definido.

Adicionalmente, una AMF está vulnerada en los siguientes casos:

- Un eje no está ajustado.
- Ha fallado la referenciación de un eje ajustado.

Desde esta AMF parametrizable se encuentran disponibles 8 instancias, es decir, se pueden configurar como máximo 8 campos de trabajo cartesianos.

Parámetros de la AMF

Parámetros	Descripción
Estructura monitoreada	<p>Estructura que debe controlarse</p> <ul style="list-style-type: none"> ■ <i>Robot y herramienta</i>: Se controlan las bolas en el robot y las bolas con las que está configurada la herramienta destinada a la seguridad. (por defecto) ■ <i>Robot</i>: Se controlan las bolas en el robot. ■ <i>Herramienta</i>: Se controlan las bolas con las que está configurada la herramienta destinada a la seguridad. <p>Nota: Si no está configurada ninguna herramienta destinada a la seguridad y la herramienta está seleccionada como estructura a controlar, se controla la bola en la brida del robot. (>>> "Bolas en el robot" Página 209)</p>

Una esquina del cuadrado se define en relación con el sistema de coordenadas Mundo. Es el origen del campo de trabajo y se define mediante los siguientes parámetros:

Parámetro	Descripción
X, Y, Z [mm]	<p>Desplazamiento del origen del campo de trabajo a lo largo del eje X, Y y Z del sistema de coordenadas Mundo.</p> <ul style="list-style-type: none"> ■ -100 000 mm ... +100 000 mm
A, B, C [°]	<p>Orientación del origen del campo de trabajo en torno a los ejes del sistema de coordenadas Mundo, indicada mediante el ángulo de rotación A, B, C</p> <ul style="list-style-type: none"> ■ 0° ... 359°

Partiendo desde el origen establecido de este modo, se determinar el tamaño del campo de trabajo a lo largo de los ejes de coordenadas:

Parámetro	Descripción
Longitud [mm]	<p>Longitud del campo de trabajo (= trayecto a lo largo del eje X positivo del origen)</p> <ul style="list-style-type: none"> ■ 0 mm ... 100 000 mm
Ancho [mm]	<p>Anchura del campo de trabajo (= trayecto a lo largo del eje Y positivo del origen)</p> <ul style="list-style-type: none"> ■ 0 mm ... 100 000 mm
Altura [mm]	<p>Altura del campo de trabajo (= trayecto a lo largo del eje Z positivo del origen)</p> <ul style="list-style-type: none"> ■ 0 mm ... 100 000 mm



Tras la vulneración de un campo de trabajo cartesiano, la vulneración se suprime cuando todas las bolas controladas se encuentran de nuevo dentro de los límites del campo de trabajo y presentan una distancia mínima de 10 mm con respecto a los mismos.

Ejemplo

La figura muestra un ejemplo de un campo de trabajo cartesiano. El origen está desplazado en relación con el sistema de coordenadas Mundo en dirección Y y en dirección X negativa.

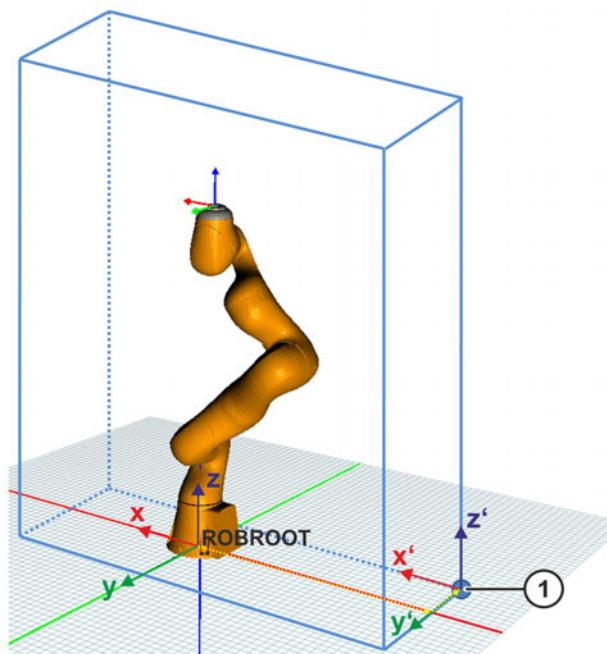


Fig. 13-8: Ejemplo de un campo de trabajo cartesiano

1 Origen del campo de trabajo

13.8.9.2 Definir zonas de protección cartesianas

Descripción Una zona de protección cartesiana se define como cuadrado, cuya posición y orientación en el espacio se establece en relación con el sistema de coordenadas Mundo.

Las bolas de control se controlan con respecto a los límites de las zonas de protección cartesianas activadas y se deben mover fuera de las zonas de protección.

Para definir una zona de protección cartesiana, se utiliza la AMF *Monitorización cartesiana de las zonas de protección*. Esta AMF está vulnerada, si una de las bolas controladas ya no se encuentra totalmente fuera de la zona de protección definida.

Adicionalmente, una AMF está vulnerada en los siguientes casos:

- Un eje no está ajustado.
- Ha fallado la referenciación de un eje ajustado.

Desde esta AMF parametrizable se encuentran disponibles 8 instancias, es decir, se pueden configurar como máximo 8 zonas de protección cartesianas.

Si se ha configurado una zona de protección muy estrecha, puede suceder que el robot se desplace a la zona de protección y la vuelva a abandonar sin que se detecte la vulneración del espacio. Causa posible: Debido a la velocidad de herramienta muy elevada, la zona de protección solo se vulnera durante un período de tiempo muy corto.

Imagine que se han configurado los siguientes valores mínimos:

- Radio de las bolas de herramientas: 25 mm
- Espesor de la zona de protección: 0 mm

En este caso se necesitarían velocidades de herramienta superiores a 4 m/s para que la zona de protección se atraviese sin detección.

Se recomiendan las siguientes medidas para evitar que se atraviesen zonas de protección sin detección:

- Configurar un control de velocidad de 4 m/s.
- O BIEN: Seleccionar valores suficientemente grandes para la longitud, el ancho y la altura de la zona de protección a la hora de configurarlo.
- O BIEN: Seleccionar valores suficientemente grandes para el radio a la hora de configurar las bolas de herramienta.

Parámetros de la AMF

Parámetros	Descripción
<i>Estructura monitoreada</i>	<p>Estructura que debe controlarse</p> <ul style="list-style-type: none"> ■ <i>Robot y herramienta</i>: Se controlan las bolas en el robot y las bolas con las que está configurada la herramienta destinada a la seguridad. (por defecto) ■ <i>Robot</i>: Se controlan las bolas en el robot. ■ <i>Herramienta</i>: Se controlan las bolas con las que está configurada la herramienta destinada a la seguridad. <p>Nota: Si no está configurada ninguna herramienta destinada a la seguridad y la herramienta está seleccionada como estructura a controlar, se controla la bola en la brida del robot. (» "Bolas en el robot" Página 209)</p>

Una esquina del cuadrado se define en relación con el sistema de coordenadas Mundo. Es el origen de la zona de protección y se define mediante los siguientes parámetros:

Parámetros	Descripción
X, Y, Z [mm]	<p>Desplazamiento del origen de la zona de protección a lo largo del eje X, Y y Z del sistema de coordenadas Mundo.</p> <ul style="list-style-type: none"> ■ -100 000 mm ... +100 000 mm
A, B, C [°]	<p>Orientación del origen de la zona de protección en torno a los ejes del sistema de coordenadas Mundo, indicada mediante el ángulo de rotación A, B, C</p> <ul style="list-style-type: none"> ■ 0° ... 359°

Partiendo desde el origen establecido de este modo, se determinar el tamaño de la zona de protección a lo largo de los ejes de coordenadas:

Parámetros	Descripción
<i>Longitud [mm]</i>	<p>Longitud de la zona de protección (= trayecto a lo largo del eje X positivo del origen)</p> <ul style="list-style-type: none"> ■ 0 mm ... 100 000 mm
<i>Anchura [mm]</i>	<p>Anchura de la zona de protección (= trayecto a lo largo del eje Y positivo del origen)</p> <ul style="list-style-type: none"> ■ 0 mm ... 100 000 mm
<i>Altura [mm]</i>	<p>Altura de la zona de protección (= trayecto a lo largo del eje Z positivo del origen)</p> <ul style="list-style-type: none"> ■ 0 mm ... 100 000 mm

i La vulneración se suprime, cuando las bolas controladas se encuentran de nuevo fuera de los límites de la zona de protección después de la vulneración de una zona de protección cartesiana y presentan una distancia mínima de 10 mm con respecto a los mismos.

Ejemplo

La figura muestra un ejemplo de una zona de protección cartesiana. El origen está desplazado en relación con el sistema de coordenadas Mundo en dirección Y negativa y en dirección X positiva.

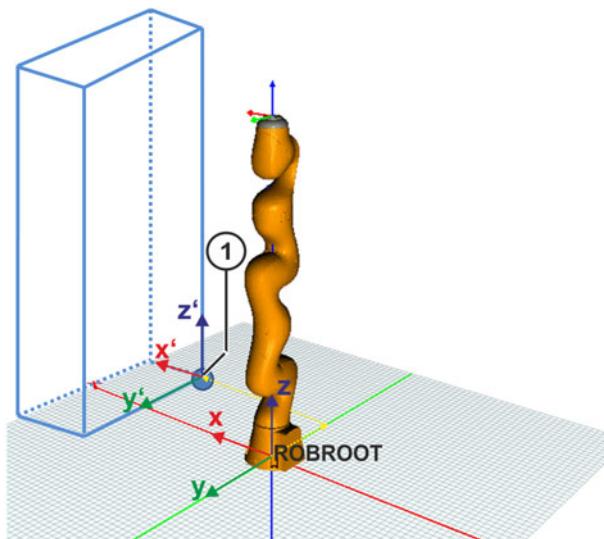


Fig. 13-9: Ejemplo de una zona de protección cartesiana

1 Origen de la zona de protección

13.8.9.3 Definir las zonas de control específicas del eje

Descripción

Los límites de los ejes se pueden ajustar individualmente para cada eje y controlarse de forma segura. El ángulo del eje debe encontrarse dentro del campo del eje definido.

Para definir una zona de control específica del eje, se utiliza la AMF **Control del campo del eje**. Esta AMF está vulnerada, si un eje se encuentra fuera del campo del eje definido.

Adicionalmente, una AMF está vulnerada en los siguientes casos:

- Un eje no está ajustado.
- Ha fallado la referenciación de un eje ajustado.

Desde esta AMF parametrizable se encuentran disponibles 16 instancias, es decir, se pueden configurar como máximo 16 zonas de control específicas de los ejes.

Parámetros de la AMF

Parámetro	Descripción
Ejes controlados	Eje que debe controlarse <ul style="list-style-type: none"> ■ Eje1 ... Eje16 Indicación: Para el KUKA LBR iiwa se utilizan Eje1 ... Eje7 .

Parámetro	Descripción
Límite inferior [°]	Límite inferior del campo del eje permitido en el que puede moverse el eje controlado ■ -180° ... +180°
Límite superior [°]	Límite superior del campo del eje permitido en el que puede moverse el eje controlado ■ -180° ... +180°

i El campo del eje permitido transcurre en el sentido de giro posición del eje desde el límite inferior al superior.
Si el eje se debe encontrar con ±180° en el rango angular permitido, el límite inferior debe ser mayor que el límite superior.

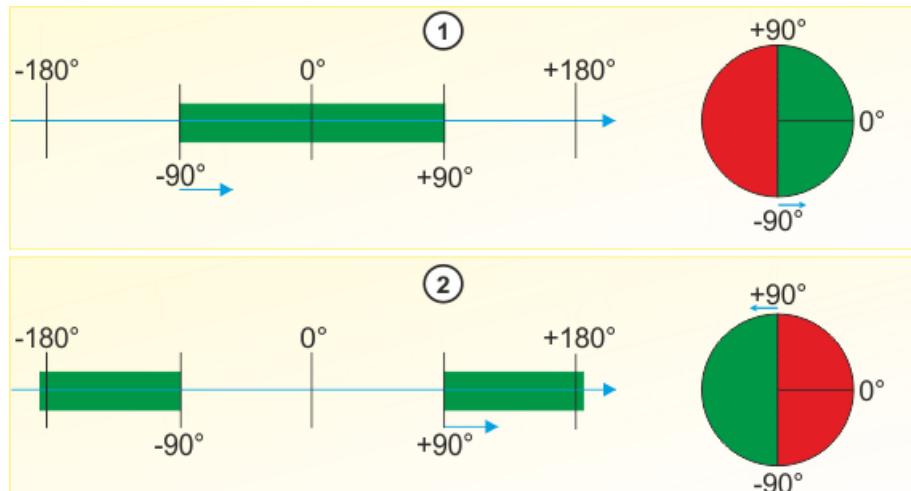


Fig. 13-10: Ejemplos de campos de trabajo específicos del eje

- 1 Límite inferior: -90°; Límite superior: +90°
- 2 Límite inferior: +90°; Límite superior: -90°

i Para la protección de personas solo es relevante la posición del eje. Por ello, también se realiza la conversión de las posiciones al rango del eje de -180° ... +180° para los ejes que pueden girar más de 360°.

Ejemplo

Los ejes A1, A2 y A4 deben controlarse de forma que el robot solo pueda desplazarse en una zona limitada. El control se activa a través de una entrada segura. La zona permitida de cada eje se determina mediante un límite superior y un límite inferior se representa en color verde en el gráfico correspondiente en la tabla PSM.

En cuanto se vulnera una de las zonas controladas de los ejes, se debe activar una parada de seguridad 1 (sobre la trayectoria). Para lograrlo, se debe utilizar una línea propia de la tabla para cada eje.

Configuración de seguridad del usuario configurable						(9/100)
Línea	Activo	Categoría	AMF 1	AMF 2	AMF 3	Reacción
8	<input checked="" type="checkbox"/>	Monitorización de la zona	Señal de entrada (4) Entrada de la señal de seguridad : EntradaCIB_SR4	Control del campo del eje (1) Eje controlado : Eje1		Parada 1 (sobre la trayectoria)
9	<input checked="" type="checkbox"/>	Monitorización de la zona	Señal de entrada (4) Entrada de la señal de seguridad : EntradaCIB_SR4	Control del campo del eje (2) Eje controlado : Eje1		Parada 1 (sobre la trayectoria)
10	<input checked="" type="checkbox"/>	Monitorización de la zona	Señal de entrada (4) Entrada de la señal de seguridad : EntradaCIB_SR4	Control del campo del eje (3) Eje controlado : Eje1		Parada 1 (sobre la trayectoria)

Fig. 13-11: Tabla PSM – Control simultáneo de 3 ejes

13.8.10 Control de la orientación de herramienta

Descripción

Mediante la AMF *Orientación de la herramienta* se puede supervisar la orientación de la herramienta destinada a la seguridad. En este proceso se comprueba si un eje determinado del Frame de orientación de herramienta está dentro de un rango de orientación admisible.

Esta función se puede emplear por ejemplo, para evitar en aplicaciones MRK que partes peligrosas de la herramienta montada, como bordes afilados, estén orientadas hacia las personas.

Se controla la orientación del eje Z del Frame de orientación de herramienta de la herramienta destinada a la seguridad. Si no se ha configurado ninguna herramienta destinada a la seguridad, se controla el eje Z del sistema de coordenadas de brida.

(>>> 9.3.7.1 "Definir la herramienta destinada a la seguridad" Página 151)

La zona admisible para el ángulo de orientación se define a través de un vector de referencia con una orientación fija con respecto al sistema de coordenadas mundial y un ángulo admisible de divergencia de este vector de referencia.

El vector de referencia se define a través del giro del vector unitario del eje Z del sistema de coordenadas mundial sobre los 3 ángulos Euler A, B y C con respecto al sistema de coordenadas mundial. Se fija un cono de control alrededor del vector de referencia. La abertura del cono se define a través de un ángulo de divergencia ajustable. El ángulo de divergencia establece el ángulo entre la orientación de la herramienta y el vector de referencia. Los valores de los ángulos del vector de referencia y el ángulo de divergencia se determinan durante la parametrización de la AMF.

El cono de control define la zona admisible para la orientación de la herramienta.

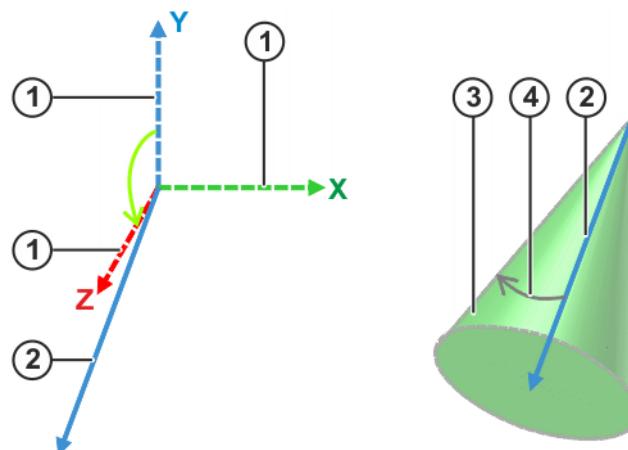


Fig. 13-12: Cono de control para la orientación de herramienta

Pos.	Descripción
1	Ejes del sistema de coordenadas mundial
2	Vector de referencia El vector de referencia determina una orientación fija con respecto al sistema de coordenadas mundial.
3	Cono de control Fija la zona admisible para la orientación de herramienta.
4	Ángulo de divergencia El ángulo de divergencia determina la abertura del cono de control.

La AMF *Orientación de la herramienta* se considera vulnerada si el ángulo entre el vector de referencia y el eje Z del Frame de orientación de la herramienta es mayor que el ángulo de divergencia ajustado.

Adicionalmente, una AMF está vulnerada en los siguientes casos:

- Un eje no está ajustado.
- Ha fallado la referenciación de posición de un eje ajustado.

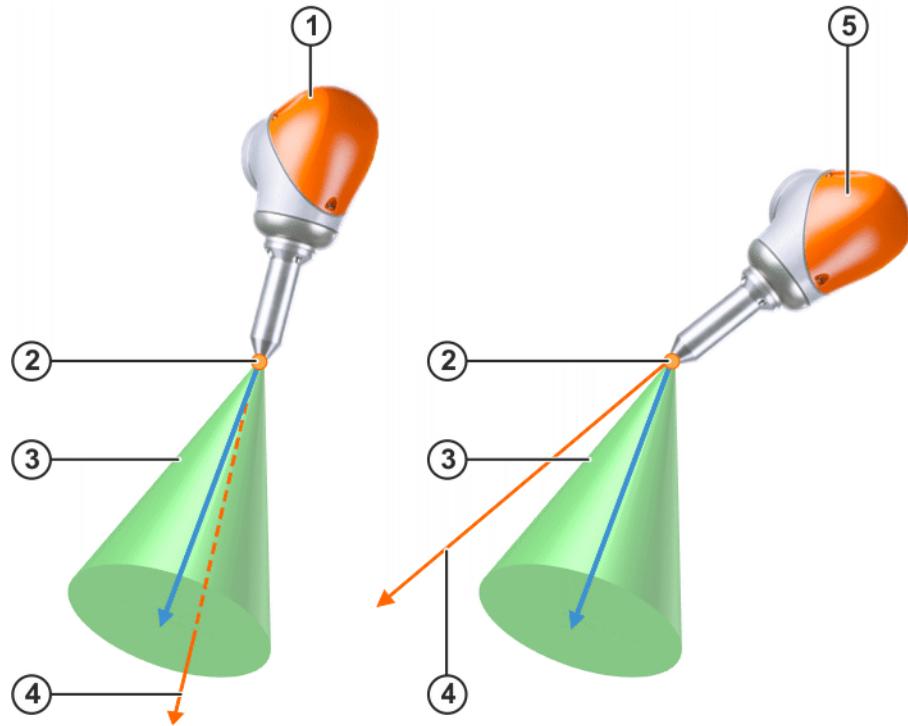


Fig. 13-13: Orientación de herramienta (no infringida e infringida)

Pos.	Descripción
1	El robot no vulnera a la AMF <i>Orientación de la herramienta</i> . El eje Z del Frame de orientación de herramienta está dentro de la zona definida por el cono de control.
2	Origen del Frame de orientación de herramienta
3	Cono de control
4	Eje Z del Frame de orientación de herramienta
5	El robot vulnera a la AMF <i>Orientación de la herramienta</i> . El eje Z del Frame de orientación de herramienta está fuera de la zona definida por el cono de control.

Parámetros de la AMF

Parámetros	Descripción
A [°]	Rotación del vector de referencia sobre el eje Z del sistema de coordenadas mundial ■ 0° ... 359°
B [°]	Rotación del vector de referencia sobre el eje Y del sistema de coordenadas mundial ■ 0° ... 359°

Parámetros	Descripción
C [°]	Rotación del vector de referencia sobre el eje X del sistema de coordenadas mundial ■ 0° ... 359°
Zona de trabajo [°]	Zona de trabajo de la orientación de herramienta Fija el ángulo de divergencia máximo admisible entre el vector de referencia y el eje Z del Frame de orientación de herramienta. ■ 1° ... 179°

13.8.11 Control de parada (parada de servicio segura)

Descripción Si el robot no debe moverse bajo determinadas condiciones, pero debe permanecer regulado, se deberá controlar de forma segura la parada de todos los ejes. Para ello utilizar la AMF **Control de parada de todos los ejes**.

En el caso de esta AMF se trata de una Extended AMF, es decir, el control comienza cuando estén vulneradas todas las demás AMFs de la función de seguridad.



Las Extended AMF no están disponibles para las funciones de seguridad del mecanismo de ESM.

La parada está definida como conservación de las posiciones de los ejes. Al principio del control de parada se guardan las posiciones de los ejes y se comparan con los valores actuales de las articulaciones, mientras que el control esté activo.

Debido a que el control de parada controla en un rango de tolerancia reducido, el control también puede ser vulnerado si el movimiento del robot es ocasionado por la actuación de una fuerza exterior, por ejemplo, un choque del robot.

AMF	Descripción
Control de parada de todos los ejes	La AMF está vulnerada, si el valor de la articulación de un eje se encuentra fuera de un rango de tolerancia de +/- 0,1°, de acuerdo con el valor guardado con la activación del control de parada, o si uno de los ejes mueve más de 1 °/s conforme a la referencia. Extended AMF no parametrizable con 8 instancias disponibles

Ejemplo Categoría: **Parada de servicio segura**

Si el robot se encuentra fuera de su zona de trabajo, debe asegurarse que no se moverá en el momento en que haya personas dentro de su alcance. La zona de trabajo está configurada mediante un campo de trabajo cartesiano. En una entrada segura se encuentra conectado un sensor, mediante el que se detectan las personas en peligro. Si se han vulnerado tanto el campo de trabajo como la señal de entrada, se activará el control de parada.

AMF1	AMF2	AMF3	Reacción
Señal de entrada (sensor)	Control del campo de trabajo cartesiano	Control de parada de todos los ejes	Parada 1

13.8.12 Retardo de conexión para las funciones de seguridad

Descripción Mediante la AMF *Retardo de tiempo* se puede retardar el accionamiento de la reacción de una función de seguridad durante un tiempo ajustado.

En el caso de esta AMF se trata de una Extended AMF, es decir, el tiempo de retardo comienza a contar cuando se hayan vulnerado todas las demás AMF de la función de seguridad.



Las Extended AMF no están disponibles para las funciones de seguridad del mecanismo de ESM.

AMF	Descripción
<i>Retardo de tiempo</i>	La AMF está vulnerada, si el tiempo ajustado ha transcurrido. Extended AMF parametrizable con 16 instancias disponibles.



Si se emplea la misma instancia de la AMF para varias funciones de seguridad, el tiempo de retardo empieza a contar desde la primera activación.

Parámetros de la AMF	Parámetros	Descripción
	<i>Tiempo de retardo</i>	<p>Período de tiempo que se retarda la reacción de una función de seguridad.</p> <ul style="list-style-type: none"> ■ 12 ms ... 24 h <p>El tiempo se puede introducir en milisegundos (ms), segundos (s), minutos (min) y horas (h). Cada retardo es un múltiplo de 12 ms, es decir, se redondea hasta el siguiente múltiplo.</p>

Ejemplo

Categoría: *Parada de seguridad*

Se desea desplazar un robot con ejes no referenciados en el modo automático durante un tiempo limitado. Una vez transcurrido este tiempo, p.ej. 2 horas, se desea activar una parada de seguridad 1 (sobre la trayectoria).

AMF1	AMF2	AMF3	Reacción
<i>Modo de servicio Automático</i>	<i>Referenciación de posición</i>	<i>Retardo de tiempo</i>	<i>Parada 1 (sobre la trayectoria)</i>

13.8.13 Control de fuerzas y momentos

El LBR iiwa está equipado con sensores de posición y de momentos de articulaciones en todos los ejes. De este modo, se pueden medir las fuerzas y los momentos que actúan desde el exterior y se puede reaccionar de forma correspondiente.



Para la utilización de las AMF *Identificación de colisión*, *Control de fuerza TCP* y *Control del campo del eje* se debe tener en cuenta que, debido a las distancias de parada del robot, las fuerzas de interacción pueden seguir aumentando si se vulnera a la AMF y se acciona una parada de seguridad. Por esta razón, se recomienda efectuar el modo de colaboración (MRK) solo con velocidades reducidas, es decir, combinar estas AMF con la AMF *Control de velocidad cartesiana* o *Monitorización de la velocidad del eje*.

! Para la utilización de las AMF *Identificación de colisión y Control de fuerza TCP* se debe tener en cuenta que las fuerzas externas que actúan sobre la estructura del robot con una distancia reducida con respecto a los ejes del robot, en determinadas circunstancias, solo pueden generar momentos reducidos de ejes externos en los ejes del robot. Esto puede representar un riesgo para la seguridad durante el servicio de colaboración, sobre todo en situaciones de aplastamiento. Las situaciones de aplastamiento pueden surgir debido a situaciones de atrapamiento dentro de la estructura del robot o situaciones de atrapamiento del robot con su entorno.

Por ello, se recomienda evitar posibles situaciones de aplastamiento críticas mediante una configuración adecuada de la célula de robot y/o un uso adicional de las AMF *Control del campo de trabajo cartesiano*, *Monitorización cartesiana de las zonas de protección*, *Control del campo del eje* o *Orientación de la herramienta*.

! Durante la utilización de las AMF *Identificación de colisión y Control de fuerza TCP* el operario debe asegurarse de que las piezas de trabajo recogidas no se suelten y se caigan de forma involuntaria mientras uno de los controles esté activo.

13.8.13.1 Control del campo del eje

Con el control del campo del eje se pueden limitar y controlar los momentos de ejes individuales.



Si debido a una situación de aprisionamiento se supera de forma permanente el momento del eje permitido, el robot podrá desbloquearse mediante un cambio al modo de servicio KRF.

AMF	Descripción
Control del campo del eje	Esta AMF está vulnerada, si el par de torsión del eje controlado supera o no alcanza la limitación del momento definida. AMF parametrizable con 16 instancias disponibles

Parámetros de la AMF

Parámetro	Descripción
Ejes controlados	Eje que debe controlarse <ul style="list-style-type: none"> ■ Eje1 ... Eje16 Indicación: Para el KUKA LBR iiwa se utilizan Eje1 ... Eje7 .
Momento mínimo [Nm]	Mínimo par de torsión permitido del eje indicado <ul style="list-style-type: none"> ■ -500 ... 500 Nm
Momento máximo [Nm]	Máximo par de torsión permitido del eje indicado <ul style="list-style-type: none"> ■ -500 ... 500 Nm

13.8.13.2 Detección de colisión

Descripción	En la detección de colisión se controlan los momentos del eje externos con respecto a un valor límite que se puede definir.
	El momento del eje externo se define como la parte del par de torsión de un eje que se produce a causa de las fuerzas y los momentos generados durante la interacción del robot con el entorno. No se mide directamente, sino que se calcula con la ayuda del modelo del robot dinámico. La precisión de los valo-

res calculados depende de la dinámica del movimiento del robot y de la dinámica de las fuerzas de interacción del robot con el entorno.

Los siguientes puntos se deben tener en cuenta para la utilización de la detección de colisión:

- Se presupone que se ha realizado la referenciación de la posición y del momento.
- Se tienen en cuenta los datos de carga de la herramienta destinada a la seguridad (si está configurada).
- Si la herramienta destinada a la seguridad está configurada, también deberá estar montada en la brida del robot.
- Los datos de carga de piezas de trabajo destinadas a la seguridad (si están configuradas) solo se tendrán en cuenta si se comunica la pieza de trabajo destinada a la seguridad actualmente activa al control de seguridad.

(>>> 15.11.5 "Ordenar cambios de carga en el control de seguridad" Página 298)



En la AMF *Identificación de colisión* no se tienen en cuenta automáticamente posibles errores al activar la pieza de trabajo destinada a la seguridad.

Por ello, en la configuración de la detección de colisiones deben ajustarse valores más bien reducidos para el momento externo máximo admisible para que las divergencias en los datos de carga se interpreten como colisiones y provoquen así una vulneración de la AMF.

AMF	Descripción
<i>Identificación de colisión</i>	<p>Esta AMF está vulnerable, si el momento externo de al menos un eje excede el valor límite configurado.</p> <p>AMF parametrizable con 8 instancias disponibles</p>

Parámetros de la AMF	Parámetros	Descripción
	<i>Momento externo máximo [N m]</i>	<p>Momento externo máximo admisible</p> <ul style="list-style-type: none"> ■ 0-30 Nm

13.8.13.3 Control de fuerza TCP

Descripción En el control de fuerza TCP se controla la fuerza externa que actúa en el TCP de la herramienta destinada a la seguridad, de acuerdo con un valor límite definible. Si no se utiliza ninguna herramienta, se controla la fuerza externa que actúa en el punto central de la brida del robot.

La fuerza externa en el TCP no se mide directamente, sino que se determina con la ayuda del modelo del robot dinámico. La precisión de la fuerza externa determinada depende, entre otras cosas, de la dinámica del movimiento del robot y de la dinámica de la fuerza que actúa realmente.

Los siguientes puntos se deben tener en cuenta para la utilización del control de fuerza TCP:

- Se presupone que se ha realizado la referenciación de la posición y del momento.
- Se tienen en cuenta los datos de carga de la herramienta destinada a la seguridad (si está configurada).
- Si la herramienta destinada a la seguridad está configurada, también deberá estar montada en la brida del robot.

- Se tienen en cuenta los datos de carga de la herramienta destinada a la seguridad más pesada (si está configurada).



En la AMF *Control de fuerza TCP* no se tienen en cuenta automáticamente posibles errores al activar una pieza de trabajo destinada a la seguridad.

Por ello, durante la configuración del control de fuerza TCP es necesario que para la fuerza externa máxima admisible en el TCP se ajuste un valor que sea mayor que el peso de la pieza de trabajo más pesada que se va a recoger.

AMF	Descripción
<i>Control de fuerza TCP</i>	<p>La AMF está vulnerada si la fuerza externa que actúa en el TCP de la herramienta destinada a la seguridad (o del punto central de la brida del robot) supera el valor límite configurado.</p> <p>AMF parametrizable con 8 instancias disponibles</p>

Parámetros de la AMF

Parámetros	Descripción
<i>Fuerza TCP máxima [N]</i>	Máxima fuerza externa permitida en el TCP ■ 50 ... 1000 N

Precisión de la detección de fuerza

La precisión de la detección de fuerza del TCP depende de la posición del robot. El control de seguridad detecta poses no permitidas y ajusta la AMF *Control de fuerza TCP* como vulnerada con un mensaje de diagnóstico correspondiente.

No están permitidas las posiciones en las que pueden producirse fuerzas TCP que presentan una distancia reducida con respecto a todos los ejes del robot. Esto se aplica para posiciones singulares y posiciones cerca de singularidades. Deben evitarse especialmente los ángulos de eje próximos a 0° en los ejes A2, A4 y A6:

- Eje A2: -35° hasta 35°
- Eje A4: -55° hasta 55°
- Eje A6: -20° hasta 20°
- Las fuerzas externas que actúan sobre la estructura del robot reducen la precisión de la detección de fuerza del TCP. El control de seguridad puede detectar en muchos casos automáticamente la acción de las fuerzas externas sobre la estructura del robot. En este caso, se ha vulnerado la AMF *Control de fuerza TCP*.



No se puede garantizar que control de seguridad detecte siempre automáticamente que están actuando fuerzas externas sobre la estructura del robot. El usuario debe asegurarse de que las fuerzas externas actúan exclusivamente sobre el TCP, para garantizar la integridad de seguridad de la AMF *Control de fuerza TCP*.

13.9 Ejemplo de una configuración de seguridad



Este ejemplo sirve exclusivamente para la demostración de la configuración de seguridad con KUKA Sunrise.Workbench.



La base para la configuración de seguridad de una instalación es siempre un análisis de riesgos. La configuración de seguridad mostrada a continuación sirve como ejemplo y no pretende ser exhaustivo.

13.9.1 Función

El LBR iiwa se utiliza en una aplicación en cooperación con una persona. La herramienta montada en el robot está ajustada como herramienta orientada a la seguridad.

El operario prepara una pieza en un alojamiento de la pieza en intervalos regulares. Una tarea del robot consiste en comprobar la presencia de la pieza. Para ello, se desplaza partiendo de una posición inicial con un movimiento de transferencia a través de la zona accesible para la persona (zona de colaboración). El objetivo de este movimiento de transferencia es una posición previa que se encuentra a 20 cm sobre la pieza preparada. A continuación, se desplaza en dirección a la pieza.

Este movimiento de descenso está parametrizado con una condición de cancelación de fuerza (valor umbral del proceso: 20 N). El robot utiliza la posición actual tras alcanzar el valor umbral del proceso para determinar si la pieza está presente o no. La pieza está presente si puede desplazarse hasta el alojamiento de la pieza. Después de la comprobación, el robot sale de la zona de colaboración a través de la posición previa.

13.9.2 Requisito

En el marco de una evaluación de riesgos, se solicitan las siguientes funciones de seguridad para el proceso descrito:

1. El robot se debe detener a través de una PARADA DE EMERGENCIA dentro del alcance del operario.
2. El robot no puede salir de un campo de trabajo establecido. La zona de colaboración es una parte del campo de trabajo.
3. Durante el movimiento de transferencia entre la posición inicial y la posición previa, pueden producirse colisiones indeseadas con el operario. Sin embargo, la zona está configurada de manera que las personas no puedan quedar atrapadas. Por ello se ha establecido la máxima velocidad permitida del robot en esta zona a 500 mm/s.
4. Las colisiones se deben detectar de forma segura durante el movimiento de transferencia y deben provocar la parada del robot si debido a las colisiones se supera un momento de 15 Nm en al menos un eje.
5. Durante los movimientos entre la posición previa y el alojamiento de la pieza, pueden producirse atrapamientos de la mano y el brazo del operario. Para que el operario pueda reaccionar adecuadamente ante un movimiento del robot y las distancias de frenado sean suficientemente reducidas, la velocidad del robot no deberá superar 100 mm/s.
6. Además, el robot se deberá detener durante los movimientos entre la posición previa y el alojamiento de la pieza si se generan fuerzas de fijación superiores a 50 N. El valor de fuerza a partir del que se cancela el movimiento de descenso en el proceso, se encuentra con 20 N suficientemente por debajo de este límite.

13.9.3 Propuesta de solución para la tarea

Para cumplir los requisitos, deben configurarse los controles de seguridad permanentes y comutables:

- Control permanente del dispositivo externo de PARADA DE EMERGENCIA y del campo de trabajo
- Estado ESM para el movimiento de transferencia entre la posición de inicio y la posición previa

- Estado ESM para el movimiento entre la posición previa y el alojamiento de la pieza

Para garantizar un desarrollo del proceso estable y fluido, la aplicación debe disponerse de forma que se respeten los valores límite establecidos para las funciones de seguridad (p. ej. para la velocidad y el campo de trabajo).

La aplicación de robot que realiza el proceso descrito no está especificada aquí.

Control de seguridad permanente

Durante todo el servicio debe estar activa la función de PARADA DE EMERGENCIA y el robot no debe abandonar el campo de trabajo. Las funciones de seguridad correspondientes se configuran en la tabla PSM de usuarios.

Configuración de seguridad del usuario configurable						(9/100)				
Línea	Activo	Categoría	AMF 1	AMF 2	AMF 3	Reacción				
1	<input checked="" type="checkbox"/>	PARADA DE EMERGENCIA externa	Señal de entrada (I) Entrada de la señal de seguridad : EntradaCIB_SR.I	-	-	Parada 1 (sobre la trayectoria)				
2	<input checked="" type="checkbox"/>	Monitorización de la zona	Monitorización cartesiana de las zonas de protección (I)	-	-	Parada 1 (sobre la trayectoria)				

Fig. 13-14: Control de seguridad permanente

Línea	Descripción
1	<p>PARADA DE EMERGENCIA externa</p> <p>Cumple el requisito 1</p> <p>Una PARADA DE EMERGENCIA externa está conectada a una entrada segura. Si el operario acciona la PARADA DE EMERGENCIA, se ejecuta una parada de seguridad 1 (sobre la trayectoria).</p>
2	<p>Control del campo de trabajo cartesiano 2</p> <p>Cumple el requisito 2</p> <p>El campo de trabajo se representa mediante un campo de trabajo cartesiano controlado de forma segura. Si el robot abandona la zona configurada, se ejecuta una parada de seguridad 1 (sobre la trayectoria).</p>

Estado ESM para el movimiento de transferencia

Se define un estado ESM para el movimiento de transferencia entre la posición de inicio y la posición previa a través de la zona de colaboración. Este se activa en la aplicación antes de comenzar el movimiento de transferencia.

Durante el movimiento de transferencia deben estar activas la monitorización de la velocidad y la detección de colisión para reducir el riesgo de colisión entre las personas y el robot a una medida adecuada.

Además, para descartar en todo momento cualquier atrapamiento, de forma adicional se define una zona de protección que detiene al robot cuando entre el robot o la herramienta y el alojamiento de la pieza hay una distancia inferior a 15 cm.

State of the event-driven safety monitoring (ESM)				(3/20)					
Row	Active	AMF	Reaction						
1	<input checked="" type="checkbox"/>	Cartesian velocity (3) Maximum velocity : 100 mm/s	Stop 1 (on-path)						
2	<input checked="" type="checkbox"/>	Collision detection (I) Maximum external torque : 15 Nm	Stop 1 (on-path)						
3	<input checked="" type="checkbox"/>	Cartesian protected space monitoring (I)	Stop 1 (on-path)						

Fig. 13-15: Estado ESM para el movimiento de transferencia

Línea	Descripción
1	<p>Control de velocidad cartesiana</p> <p>Cumple el requisito 3</p> <p>Si se excede una velocidad cartesiana de 500 mm/s, se ejecuta una parada de seguridad 1 (sobre la trayectoria).</p>
2	<p>Detección de colisión</p> <p>Cumple el requisito 4</p> <p>En una colisión que provoque un momento externo de más de 15 N m en al menos un eje del robot, se activará una parada de seguridad 1 (sobre la trayectoria).</p>
3	<p>Monitorización de la zona de protección</p> <p>Lleva a cabo la seguridad del estado independientemente del tiempo y el lugar de la activación</p> <p>La zona de protección controlada de forma segura representa la zona situada por encima del alojamiento de la pieza. Cuando el robot o la herramienta controlada de forma segura entran en esta zona, se ejecuta una parada de seguridad 1 (sobre la trayectoria).</p>

Estado ESM para el alojamiento de la pieza

Para los movimientos entre la posición previa y el alojamiento de la pieza se define un estado ESM propio. Este se activa en la aplicación antes de comenzar el movimiento de descenso.

Durante el movimiento deben estar activos la monitorización de la velocidad y el control de fuerza para reducir a una medida adecuada el riesgo de que queden atrapados la mano o el antebrazo del operario.

El estado debe garantizar una seguridad suficiente, independientemente del momento y el lugar de la activación. Debido a la reducida velocidad admisible y al control de fuerza activo, esto se cumple sin otras medidas.

Estado para la monitorización de seguridad basada en eventos (ESM)

(2/20)

Línea	Activo	AMF	Reacción
1	<input checked="" type="checkbox"/>	<p>Control de velocidad cartesiana (2)</p> <p>Velocidad máxima : 100 mm/s</p>	Parada 1 (sobre la trayectoria)
2	<input checked="" type="checkbox"/>	<p>Control de fuerza TCP (1)</p> <p>Fuerza TCP máxima : 50 N</p>	Parada 0

Fig. 13-16: Estado ESM para el alojamiento de la pieza

Línea	Descripción
1	<p>Control de velocidad cartesiana</p> <p>Cumple el requisito 5</p> <p>Si se excede una velocidad cartesiana de 100 mm/s, se ejecuta una parada de seguridad 1 (sobre la trayectoria).</p>
2	<p>Control de fuerza</p> <p>Cumple el requisito 6</p> <p>En caso de un contacto a través del cual se provoque una fuerza de más de 50 N en el TCP del robot, se ejecuta una parada de seguridad 0.</p>

13.10 Referenciación de posición y del momento

13.10.1 Referenciación de posición

Descripción En la referenciación de posición se comprueba si la posición cero guardada del motor de un eje (= posición de ajuste guardada) coincide con la posición cero mecánica real de dicho eje.

La referenciación se ejecuta de forma continua por el sistema cuando un eje se desplaza con menos de 30 °/s. La referenciación tiene lugar cuando el sensor de ajuste detecta la posición cero mecánica del eje en una zona estrecha alrededor de la posición cero guardada del motor. La referenciación falla cuando el sensor de ajuste no detecta la posición cero mecánica del eje en la zona de la posición cero guardada del motor, o si se detecta en una posición inesperada.

Mientras que no se ejecute una referenciación de posición, estará limitada la integridad de seguridad de las funciones de seguridad en las que se basa. Aquí se incluyen posiciones del robot cartesianas y específicas del eje controladas de forma segura, velocidades cartesianas, fuerzas y colisiones.

Si ha fallado la referenciación de posición de al menos un eje, estarán vulneradas todas las AMFs basadas en posiciones seguras de los ejes.
(>>> "AMFs basadas en la posición" Página 233)

Requisito La posición de un eje no está referenciada después de las siguientes incidencias:

- La unidad de control del robot se reinicia.
- El eje se reajusta.
- La referenciación del momento del eje falla.
- El par de torsión máximo del sensor de momentos de articulaciones del eje se supera.



Después de estas incidencias no estarán vulneradas las funciones de seguridad basadas en posiciones seguras. El robot se puede desplazar, pero no se ofrece la integridad de la seguridad de las funciones de seguridad.

Las funciones de seguridad basadas en la posición estarán vulneradas después de estas incidencias si falla la referenciación de posición de un eje. Antes de la ejecución de aplicaciones críticas para la seguridad, se debe haber ejecutado correctamente la referenciación.

El estado de la referenciación de posición se puede utilizar en la configuración de seguridad como AMF. (>>> 13.8.4 "Evaluación de la referenciación de posición" Página 203)

Condición previa La posición de un eje está referenciada cuando se pasa por la posición cero guardada del motor y al mismo tiempo se detecta la posición cero del eje por el sensor de ajuste en un margen de 0° +/- 0,5°. Condición previa para ello:

- La velocidad con la que se pasa por la posición cero es < 30 °/s.
- Se pasa como mínimo por una zona específica del eje antes y después de la posición cero. La dirección de desplazamiento no es relevante.

La zona de desplazamiento específica del eje es específica del robot:

Variante del robot	A1	A2	A3	A4	A5	A6	A7
LBR iiwa 7 R800	±10.5°	±10.5°	±10.5°	±10.5°	±10.5°	±14°	±14°
LBR iiwa 14 R820	±9.5°	±9.5°	±10.5°	±10.5°	±10.5°	±14°	±14°

Ejecución	<p>La referenciación de posición de todos los ejes se ejecuta de forma continua da por el sistema si se cumplen las condiciones mencionadas anteriormente. La referenciación de posición se puede realizar con precisión de las siguientes maneras:</p> <ul style="list-style-type: none">■ Automáticamente durante el flujo de programa si un eje pasa por la posición cero con menos de 30 °/s.■ Por desplazamiento manual debe desplazarse cada eje individualmente por la posición cero.■ Ejecutar la aplicación preparada por KUKA para la referenciación. Los ejes se desplazan desde la posición vertical sobre la posición cero. Sunrise.Workbench ofrece una aplicación preparada para la referenciación de posición y del momento del LBR iiwa. Con esta aplicación se ejecutan al mismo tiempo la referenciación de posición y del momento. (>>> 13.10.3 "Crear la aplicación para la referenciación de posición y del momento" Página 229)
------------------	---



En caso de que no sea posible realizar la referenciación desde la posición vertical, es necesario crear y ejecutar una aplicación propia para la referenciación de posición.

13.10.2 Referenciación del momento

Descripción	<p>El LBR iiwa dispone en cada eje de un sensor de momentos de articulaciones mediante el que se calcula de forma segura el par de torsión que está actuando actualmente sobre el eje. Basándose en estos datos, se calculan y se controlan, por ejemplo, los momentos que actúan desde el exterior o fuerzas cartesianas.</p> <p>Durante la referenciación de los sensores de momentos de articulaciones se comprueba si el par de torsión externo esperado, que se puede calcular para un eje mediante el modelo de robot y los datos de carga indicados, coincide con el valor calculado basado en el valor de medición del sensor de momentos de articulaciones. Si la diferencia entre este valor y un valor de tolerancia determinado supera la diferencia, habrá fallado la referenciación de los sensores de momentos.</p> <p>Mientras que no se ejecute correctamente la referenciación del momento, estará limitada la integridad de seguridad de las funciones de seguridad en las que se basa. Aquí se incluyen el control de momentos de los ejes y de la fuerza TCP, así como la detección de colisión.</p> <p>Si ha fallado la referenciación del momento de al menos un eje, estarán vulneradas todas las AMFs basadas en valores de momentos seguros. (>>> "AMF basadas en el momento del eje" Página 233)</p>
Requisito	<p>El sensor de momentos de articulaciones de un eje no está referenciada después de las siguientes incidencias:</p> <ul style="list-style-type: none">■ La unidad de control del robot se reinicia.■ La referenciación de posición del eje falla.■ El par de torsión máximo del sensor de momentos de articulaciones del eje se supera.



Después de estas incidencias no estarán vulneradas las funciones de seguridad basadas en momentos seguros. El robot se puede desplazar, pero no se ofrece la integridad de seguridad de las funciones de seguridad.

Las funciones de seguridad basadas en momentos estarán vulneradas después de estas incidencias si falla la referenciación del momento de un eje. Antes de la ejecución de aplicaciones críticas para la seguridad, se debe haber ejecutado correctamente la referenciación.

El estado de la referenciación del momento se puede utilizar en la configuración de seguridad como AMF. ([>>> 13.8.5 "Evaluación de la referenciación del momento" Página 204](#))

Ejecución

Sunrise.Workbench ofrece una aplicación preparada para la referenciación de posición y del momento del LBR iiwa. Con esta aplicación se ejecutan al mismo tiempo la referenciación de posición y del momento.

([>>> 13.10.3 "Crear la aplicación para la referenciación de posición y del momento" Página 229](#))

Para la referenciación del momento deben estar disponibles en total 10 valores medidos de momentos de articulaciones por eje. Para ello se encuentran definidas 5 posiciones de medición en la aplicación que se desplazan respectivamente con giro del eje positivo y negativo. Si no es posible el desplazamiento hasta las posiciones, se deberán adaptar en la aplicación.

La referenciación del momento se debe ejecutar con la herramienta destinada a la seguridad montada en la brida. Esta herramienta debe integrarse en la aplicación.



Antes de ejecutar la referenciación de momentos, el usuario debe asegurarse de que los datos de carga de la herramienta fijada al robot coincidan con los datos de carga indicados para la herramienta destinada a la seguridad y de que los datos de carga de la pieza sujetada de trabajo (si la hay) coincidan con los datos de carga de la pieza de trabajo destinada a la seguridad activada. De lo contrario, no está asegurada la integridad de seguridad de la referenciación de los sensores de momentos de articulaciones.

Sobre todo no pueden estar fijadas cargas adicionales en el robot, p. ej. cargas fijadas en la estructura del robot o piezas de trabajo recogidas que no han sido tenidas en cuenta por el control de seguridad.



Durante la referenciación del momento no deben actuar fuerzas externas sobre el robot ni sobre la herramienta montada en la brida o la pieza de trabajo sujetada (si la hay). Esto se deberá asegurar por parte del usuario durante la ejecución de la referenciación. De lo contrario, no está asegurada la integridad de seguridad de la referenciación de los sensores de momentos de articulaciones.

El control de seguridad evalúa el momento externo cada 10 valores de medición y calcula el valor medio del momento externo de cada eje. Si el valor medio está por debajo de una tolerancia determinada, la referenciación será correcta. En caso contrario, habrá fallado la referenciación.

13.10.3 Crear la aplicación para la referenciación de posición y del momento

Descripción

Los siguientes puntos se deben tener en cuenta si la aplicación para la referenciación del momento se debe modificar debido a posiciones de medición hasta las que no es posible desplazarse:

- Se miden los valores medidos de momentos de articulaciones mientras el robot está parado.
Entre la llegada a la posición de medición y la medición se debe mantener un tiempo de espera mínimo de 2,5 segundos, durante el que no se desplaza el robot. Si los tiempos de espera son muy cortos, se puede reducir la precisión de referenciación debido a las oscilaciones de la estructura del robot.
- La medición se inicia con el método sendSafetyCommand().
- Entre 2 mediciones consecutivas puede haber como máximo 15 s.

Procedimiento

1. Marcar el proyecto Sunrise en el **Explorador de paquetes**.
2. Seleccionar la secuencia del menú **Archivo > Nuevo > Otros....**
3. En la carpeta **Sunrise** marcar la opción **Aplicación para la referenciación de posición y GMS del LBR iiwa** hacer clic en **Finalizar**.
La aplicación **PositionAndGMSReferencing.java** se crea en la carpeta de origen del proyecto y se abre en la zona de los editores de Sunrise.Workbench.
4. Realizar las siguientes adaptaciones en la aplicación:
 - Integrar la herramienta orientada a la seguridad en la aplicación.
 - Si no es posible desplazarse hasta las posiciones de medición debido a la configuración de la instalación, éstas deberán adaptarse.
5. Sincronizar el proyecto para transmitir la aplicación a la unidad de control del robot.

13.11 Resumen de la recepción de seguridad

Solo se permite el servicio de la instalación después de haber efectuado correctamente la recepción de seguridad. Para una recepción de seguridad exitosa, deben procesarse completamente los puntos de las listas de verificación y confirmarse por escrito por el técnico de puesta en servicio.



Las listas de verificación trabajadas y confirmadas por escrito deben guardarse para comprobación.

La recepción de seguridad debe efectuarse en los siguientes casos:

- Despues de la primera puesta en servicio o de la reanudación del servicio de un robot industrial
- Despues de una modificación en el robot industrial
- Despues de la modificación de la configuración de seguridad
- Despues de la actualización del software, p. ej., del software del sistema
La recepción de seguridad después de una actualización del software solo es necesaria si como consecuencia de la actualización se modifica la ID de la configuración de seguridad (= suma de control).

El integrador de sistemas determina las funciones de seguridad necesarias basándose en el análisis de riesgos. El funcionamiento correcto de las funciones de seguridad debe comprobarse después de la activación de la configuración de seguridad en la unidad de control del robot.



Si para una prueba es necesario permanecer en la zona de peligro, se deberá realizar la prueba en el modo de servicio T1.

Mediante las listas de verificación siguientes se debe comprobar si los parámetros de seguridad configurados se han aceptado correctamente. Las listas de verificación se deben procesar en el orden indicado aquí:

1. Lista de verificación para una comprobación básica de la configuración de seguridad
(>>> 13.11.1 "Lista de verificación de funciones de seguridad general" Página 231)
2. Lista de verificación para comprobar la herramienta destinada a la seguridad
(>>> 13.11.2 "Listas de verificación de la herramienta orientada a la seguridad" Página 233)
3. Lista de verificación para la comprobación de piezas de trabajo destinadas a la seguridad
(>>> 13.11.3 "Liste de verificación de las piezas de trabajo destinadas a la seguridad" Página 235)
4. Lista de verificación para comprobar las líneas utilizadas de la tabla PSM de usuarios
(>>> 13.11.4 "Lista de verificación de las líneas de la tabla de usuarios PSM" Página 236)
5. Listas de verificación para comprobar los estados ESM utilizados y no utilizados
(>>> 13.11.5 "Listas de verificación de estados ESM" Página 236)
6. Lista de verificación para comprobar las AMFs utilizadas
(>>> 13.11.6 "AMFs utilizadas en listas de verificación" Página 238)

Se puede elaborar un informe relativo a la configuración de seguridad actual.

(>>> 13.11.7 "Generar informe relativo a la configuración de seguridad" Página 244)

13.11.1 Lista de verificación de funciones de seguridad general

- | | |
|------------------------------|---|
| Lista de verificación | <ul style="list-style-type: none"> ■ Número de serie del robot: _____ ■ ID de la configuración de seguridad: _____ ■ Nombre del técnico de puesta en servicio: _____ |
|------------------------------|---|

N.º	Tarea	Sí	No relevante
1	Protección del usuario: ¿Todos los dispositivos de protección del operario están configurados, conectados correctamente y se ha garantizado su correcto funcionamiento?		
2	Protección del usuario: Se activa una parada si está activo el modo de servicio AUT o T2 estando abierta la protección del operario.		
3	Protección del usuario: La función manual de restablecimiento está disponible y en condiciones de funcionamiento.		
4	Validación equipo manual de guiado: ¿El dispositivo de validación del equipo manual de guiado está configurado, conectado correctamente y se ha garantizado su correcto funcionamiento?		
5	PARADA DE EMERGENCIA local: ¿Todos los dispositivos de PARADA DE EMERGENCIA locales están configurados, conectados correctamente y se ha garantizado su correcto funcionamiento?		
6	PARADA DE EMERGENCIA externa: ¿Todos los dispositivos de PARADA DE EMERGENCIA externos están configurados, conectados correctamente y se ha garantizado su correcto funcionamiento?		

N.º	Tarea	Sí	No relevante
7	PARADA DE EMERGENCIA local y externa: ¿La PARADA DE EMERGENCIA local y externa está configurada respectivamente como AMF única en una línea de la tabla PSM?		
8	Parada de seguridad: ¿Todos los dispositivos de parada de seguridad están configurados, conectados correctamente y se ha garantizado su correcto funcionamiento?		
9	Parada de servicio segura: ¿Todos los dispositivos para la parada de servicio segura están configurados, conectados correctamente y se ha garantizado su correcto funcionamiento?		
10	Para la utilización de AMFs basadas en la posición: ¿Se ha tenido en cuenta la integridad limitada de la seguridad de las AMFs basadas en la posición en caso de falta de la referenciación de posición? (>>> "AMFs basadas en la posición" Página 233) Nota: El inicio del estado seguro en caso de que falte la referenciación de posición se puede configurar con la utilización de la AMF <i>Referenciación de posición</i> .		
11	Para la utilización de AMFs basadas en la posición: ¿Se ha ejecutado correctamente la referenciación de posición?		
12	Control de velocidad: ¿Se han configurado y comprobado todos los controles de velocidad necesarios?		
13	Monitorización de la zona: ¿Se han configurado y comprobado todos las monitorizaciones de las zonas necesarias?		
14	Monitorizaciones cartesianas de la zona: ¿Se ha tenido en cuenta que no se comprueba toda la estructura del robot, la herramienta y la pieza de trabajo en cuanto a vulneración de la zona, sino solo las bolas de control del robot y la herramienta?		
15	Detección de colisión: ¿Se han configurado las funcionalidades MRK necesarias?		
16	Detección de colisión: ¿Se ha ajustado la configuración de forma que si la detección de colisión está activa también esté siempre activo el control de velocidad?		
17	Detección de colisión: ¿Se ha ajustado la configuración de forma que si el control de fuerza TCP está activo también esté siempre activo un control de velocidad?		
18	Detección de colisión: ¿Se ha configurado una parada de seguridad 0 para todos los controles de seguridad para la detección de situaciones de aplastamiento?		
19	Para la utilización de AMFs basadas en el momento del eje: ¿Se ha tenido en cuenta la integridad limitada de la seguridad de las AMFs basadas en el momento del eje en caso de falta de la referenciación de posición y/o de la referenciación de momentos? (>>> "AMF basadas en el momento del eje" Página 233) Nota: El inicio del estado seguro en caso de que falte la referenciación de posición y/o la referenciación de momentos se puede configurar con la utilización de la AMF <i>Referenciación de posición</i> y la AMF <i>Referenciación de momentos</i> .		

N.º	Tarea	Sí	No relevante
20	¿Se ha tenido en cuenta para la configuración de todas las líneas de la tabla PSM y los estados ESM que el estado seguro de las AMFs es el estado "vulnerable" (estado "0")? Nota: En caso de error, una AMF pasa a un estado seguro.		
21	Configuración PSM: ¿Durante la configuración de señales de salida como reacción de seguridad se ha tenido en cuenta que una salida tenga en estado seguro el nivel LOW (estado "0")?		
22	Configuración ESM: ¿Son todos los estados ESM consistentes por sí mismos, es decir, cada estado ESM reduce individualmente todos los peligros de un modo suficiente?		
23	¿Se han ejecutado correctamente la referenciación del momento y la referenciación de posición?		

Lugar, fecha	
Firma	

El firmante certifica con su firma la ejecución correcta y completa de la recepción de seguridad.

AMFs basadas en la posición La integridad de la seguridad de las AMFs basadas en la posición solo existe de forma ilimitada si la referenciación de posición se ha realizado correctamente.

AMF	Referenciación de posición	Referenciación del momento
Control de parada de todos los ejes	✓	✗
Control del campo del eje	✓	✗
Control de velocidad cartesiana	✓	✗
Control del campo de trabajo cartesiano	✓	✗
Monitorización cartesiana de la zona de protección	✓	✗

AMF basadas en el momento del eje La integridad de la seguridad de las AMF basadas en el momento del eje solo existe de forma ilimitada si la referenciación de posición y/o del momento se ha realizado correctamente.

AMF	Referenciación de posición	Referenciación del momento
Control del campo del eje	✗	✓
Identificación de colisión	✓	✓
Control de fuerza TCP	✓	✓

13.11.2 Listas de verificación de la herramienta orientada a la seguridad

13.11.2.1 Datos geométricos de la herramienta destinada a la seguridad

Descripción Si se utiliza una de las siguientes AMFs en la configuración de seguridad, debe comprobarse si los datos geométricos de la herramienta se han indicado correctamente:

- Control de velocidad cartesiana

- *Control del campo de trabajo cartesiano*

Solo en zonas en las que las bolas de control en la herramienta se han configurado como estructura a controlar.

- *Monitorización cartesiana de las zonas de protección*

Solo en zonas en las que las bolas de control en la herramienta se han configurado como estructura a controlar.

Los datos geométricos de la herramienta se pueden comprobar, vulnerando conscientemente una de las zonas de control configuradas con cada bola de la herramienta y comprobando la reacción.

Si no se utilizan monitorizaciones de la zona, solo es relevante la posición de los puntos centrales de las bolas. El límite de velocidad cartesiana configurado se puede comprobar, superándolo conscientemente para cada bola de la herramienta y comprobando la reacción.

Condición previa

- La referenciación de posición se ha ejecutado correctamente.

Lista de verificación

N.º	Tarea	Sí	No relevante
1	Bola de la herramienta (nombre de Frame): _____ ¿El radio y la posición de la bola de la herramienta se han indicado correctamente y se han comprobado?		
2	Bola de la herramienta (nombre de Frame): _____ ¿El radio y la posición de la bola de la herramienta se han indicado correctamente y se han comprobado?		
3	Bola de la herramienta (nombre de Frame): _____ ¿El radio y la posición de la bola de la herramienta se han indicado correctamente y se han comprobado?		
4	Bola de la herramienta (nombre de Frame): _____ ¿El radio y la posición de la bola de la herramienta se han indicado correctamente y se han comprobado?		
5	Bola de la herramienta (nombre de Frame): _____ ¿El radio y la posición de la bola de la herramienta se han indicado correctamente y se han comprobado?		
6	Bola de la herramienta (nombre de Frame): _____ ¿El radio y la posición de la bola de la herramienta se han indicado correctamente y se han comprobado?		

13.11.2.2 Datos de carga de la herramienta destinada a la seguridad**Descripción**

Si se utiliza una de las siguientes AMFs en la configuración de seguridad, debe comprobarse si los datos de carga de la herramienta destinada a la seguridad se han indicado correctamente.

- *Identificación de colisión*
- *Control de fuerza TCP*

Se recomienda comprobar los datos de carga mediante una referenciación del momento en varias posiciones adecuadas. Las posiciones adecuadas son, por ejemplo, las que tienen ángulos de ejes similares en posición extendida horizontal con las siguientes propiedades:

- Los ejes A2, A4 y A6 soportan cargas.
- Las posiciones se diferencian 90° en el valor del eje de A7.

Con los datos de carga correctos, la referenciación del momento debe ser correcta.

Condición previa

- La referenciación de la posición y del momento se ha efectuado correctamente.

Lista de verificación

N.º	Tarea	Sí	No relevante
1	¿Se han introducido correctamente y se han comprobado los datos de carga de la herramienta?		

13.11.3 Lista de verificación de las piezas de trabajo destinadas a la seguridad

Descripción

Si se utiliza una de las siguientes AMF en la configuración de seguridad, debe comprobarse si los datos de carga de las piezas de trabajo destinadas a la seguridad se han indicado correctamente.

- *Identificación de colisión*
- *Control de fuerza TCP*

Se recomienda comprobar los datos de carga mediante una referenciación del momento en varias posiciones adecuadas. Las posiciones adecuadas son, por ejemplo, las que tienen ángulos de ejes similares en posición extendida horizontal con las siguientes propiedades:

- Los ejes A2, A4 y A6 soportan cargas.
- Las posiciones se diferencian 90° en el valor del eje de A7.

Con los datos de carga correctos, la referenciación del momento debe ser correcta.

Condición previa

- La referenciación de la posición y del momento se ha efectuado correctamente.
- La pieza de trabajo destinada a la seguridad correcta está activa.

Lista de verificación

N.º	Tarea	Sí	No relevante
1	Nombre de la pieza de trabajo: _____ ¿Se han introducido correctamente y se han comprobado los datos de carga de la pieza de trabajo?		
2	Nombre de la pieza de trabajo: _____ ¿Se han introducido correctamente y se han comprobado los datos de carga de la pieza de trabajo?		
3	Nombre de la pieza de trabajo: _____ ¿Se han introducido correctamente y se han comprobado los datos de carga de la pieza de trabajo?		
4	Nombre de la pieza de trabajo: _____ ¿Se han introducido correctamente y se han comprobado los datos de carga de la pieza de trabajo?		
5	Nombre de la pieza de trabajo: _____ ¿Se han introducido correctamente y se han comprobado los datos de carga de la pieza de trabajo?		

N.º	Tarea	Sí	No relevante
6	Nombre de la pieza de trabajo: _____ ¿Se han introducido correctamente y se han comprobado los datos de carga de la pieza de trabajo?		
7	Nombre de la pieza de trabajo: _____ ¿Se han introducido correctamente y se han comprobado los datos de carga de la pieza de trabajo?		
8	Nombre de la pieza de trabajo: _____ ¿Se han introducido correctamente y se han comprobado los datos de carga de la pieza de trabajo?		

13.11.4 Lista de verificación de las líneas de la tabla de usuarios PSM

Descripción

Para cada línea de la tabla *Usuario PSM* se debe comprobar si se activa la reacción esperada. Si la reacción es la desconexión de una salida, se deberá asegurar adicionalmente mediante la prueba que la salida se ha conectado correctamente.

Una línea de la table PSM se puede comprobar, vulnerando respectivamente 2 AMFs. Seguidamente se puede comprobar individualmente y de forma precisa la AMF restante. Si se utilizan menos de 3 AMFs en una línea, se considerarán las columnas no ocupadas como AMFs vulneradas.

(>>> 13.11.6 "AMFs utilizadas en listas de verificación" Página 238)



Los puntos de la lista de verificación se deben procesar para cada línea de la tabla PSM y deben documentarse por separado.

Lista de verificación

■ N.º de línea: _____

cación

N.º	Tarea	Sí	No relevante
1	La AMF 1 se ha comprobado correctamente. Requisito: AMF 2 y AMF 3 están vulneradas. AMF 1: _____		
2	La AMF 2 se ha comprobado correctamente. Requisito: AMF 1 y AMF 3 están vulneradas. AMF 2: _____		
3	La AMF 3 se ha comprobado correctamente. Requisito: AMF 1 y AMF 2 están vulneradas. AMF 3: _____		

13.11.5 Listas de verificación de estados ESM

13.11.5.1 Estados ESM utilizados

Descripción

Para cada línea de un estado ESM se debe comprobar si se activa la reacción esperada, cuando está vulnerada la AMF configurada.

(>>> 13.11.6 "AMFs utilizadas en listas de verificación" Página 238)



Los puntos de la lista de verificación se deben procesar para cada estado ESM utilizado y deben documentarse por separado.

Lista de verificación ■ Estado ESM: _____

N.º	Tarea	Sí	No relevante
1	AMF línea 1 se ha comprobado correctamente. AMF línea 1: _____		
2	AMF línea 2 se ha comprobado correctamente. AMF línea 2: _____		
3	AMF línea 3 se ha comprobado correctamente. AMF línea 3: _____		
4	AMF línea 4: se ha comprobado correctamente. AMF línea 4: _____		
5	AMF línea 5 se ha comprobado correctamente. AMF línea 5: _____		
6	AMF línea 6 se ha comprobado correctamente. AMF línea 6: _____		
7	AMF línea 7 se ha comprobado correctamente. AMF línea 7: _____		
8	AMF línea 8 se ha comprobado correctamente. AMF línea 8: _____		
9	AMF línea 9 se ha comprobado correctamente. AMF línea 9: _____		
10	AMF línea 10 se ha comprobado correctamente. AMF línea 10: _____		
11	AMF línea 11 se ha comprobado correctamente. AMF línea 11: _____		
12	AMF línea 12 se ha comprobado correctamente. AMF línea 12: _____		
13	AMF línea 13 se ha comprobado correctamente. AMF línea 13: _____		
14	AMF línea 14 se ha comprobado correctamente. AMF línea 14: _____		
15	AMF línea 15 se ha comprobado correctamente. AMF línea 15: _____		
16	AMF línea 16 se ha comprobado correctamente. AMF línea 16: _____		
17	AMF línea 17 se ha comprobado correctamente. AMF línea 17: _____		
18	AMF línea 18 se ha comprobado correctamente. AMF línea 18: _____		
19	AMF línea 19 se ha comprobado correctamente. AMF línea 19: _____		
20	AMF línea 20 se ha comprobado correctamente. AMF línea 20: _____		

13.11.5.2 Estados ESM no utilizados

Descripción Para todos los estados ESM utilizados se debe comprobar si se activa una parada de seguridad al seleccionar el estado ESM.

Listado de verificación

N.º	Tarea	Sí	No relevante
1	La selección del estado ESM no utilizado 1 se ha comprobado correctamente.		
2	La selección del estado ESM no utilizado 2 se ha comprobado correctamente.		
3	La selección del estado ESM no utilizado 3 se ha comprobado correctamente.		
4	La selección del estado ESM no utilizado 4 se ha comprobado correctamente.		
5	La selección del estado ESM no utilizado 5 se ha comprobado correctamente.		
6	La selección del estado ESM no utilizado 6 se ha comprobado correctamente.		
7	La selección del estado ESM no utilizado 7 se ha comprobado correctamente.		
8	La selección del estado ESM no utilizado 8 se ha comprobado correctamente.		
9	La selección del estado ESM no utilizado 9 se ha comprobado correctamente.		
10	La selección del estado ESM no utilizado 10 se ha comprobado correctamente.		

13.11.6 AMFs utilizadas en listas de verificación

Una AMF que se utiliza en varias líneas de la tabla PSM, se debe comprobar en cada línea individualmente. ([>>> 13.11.4 "Lista de verificación de las líneas de la tabla de usuarios PSM" Página 236](#))

13.11.6.1 AMF PARADA DE EMERGENCIA smartPAD

Listado de verificación

N.º	Tarea	Sí	No relevante
1	La reacción configurada se activa al presionar la PARADA DE EMERGENCIA en la smartPAD.		

13.11.6.2 AMF Validación smartPAD

Listado de verificación

N.º	Tarea	Sí	No relevante
1	La reacción configurada se activa al soltar el pulsador de validación.		

13.11.6.3 AMF Validación pánico smartPAD**Lista de verificación**

N.º	Tarea	Sí	No relevante
1	La reacción configurada se activa al presionar a fondo el pulsador de validación.		

13.11.6.4 AMF Validación equipo manual de guiado**Lista de verificación**

- Entrada usada para el pulsador de validación 1: _____
- Entrada usada para el pulsador de validación 2: _____
- Entrada usada para el pulsador de validación 3: _____
- Entrada usada para el pulsador de pánico 1: _____
- Entrada usada para el pulsador de pánico 2: _____
- Entrada usada para el pulsador de pánico 2: _____

N.º	Tarea	Sí	No relevante
1	La reacción configurada se activa al soltar cada pulsador de validación.		
2	La reacción configurada se activa al presionar a fondo cada pulsador de validación.		

13.11.6.5 AMF Modo de servicio Test**Lista de verificación**

N.º	Tarea	Sí	No relevante
1	La reacción configurada se activa en T1.		
2	La reacción configurada se activa en T2.		
3	La reacción configurada se activa en KRF.		

13.11.6.6 AMF Modo de servicio Automático**Lista de verificación**

N.º	Tarea	Sí	No relevante
1	La reacción configurada se activa en AUT.		

13.11.6.7 AMF Modo de servicio con velocidad reducida**Lista de verificación**

N.º	Tarea	Sí	No relevante
1	La reacción configurada se activa en T1.		
2	La reacción configurada se activa en KRF.		

13.11.6.8AMF Modo de servicio de alta velocidad

Lista de verificación

N.º	Tarea	Sí	No relevante
1	La reacción configurada se activa en T2.		
2	La reacción configurada se activa en AUT.		

13.11.6.9AMF Señal de entrada

Lista de verificación

- Entrada utilizada: _____
- Instancia de la entrada utilizada: _____

N.º	Tarea	Sí	No relevante
1	La reacción configurada se activa cuando la entrada tiene un nivel LOW (estado "0").		

13.11.6.10AMF Control de parada de todos los ejes

Lista de verificación

N.º	Tarea	Sí	No relevante
1	La reacción configurada se activa cuando se mueve un eje.		

13.11.6.11AMF Movimiento habilitado

Lista de verificación

N.º	Tarea	Sí	No relevante
1	La reacción configurada se activa cuando, por ejemplo, se pulsa la PARADA DE EMERGENCIA en el smartPAD.		

13.11.6.12AMF Control del campo del eje

Descripción La AMF se puede comprobar, indicando los momentos de los ejes medidos actualmente en el smartPAD y, seguidamente, el eje controlado se carga manualmente o mediante la fuerza de la gravedad.

Lista de verificación

- Eje controlado: _____
- Instancia del eje controlado: _____
- Máximo momento del eje permitido: _____
- Mínimo momento del eje permitido: _____

N.º	Tarea	Sí	No relevante
1	La reacción configurada se activa cuando se supera el máximo momento del eje permitido.		
2	La reacción configurada se activa cuando no se alcanza el mínimo momento del eje permitido.		

13.11.6.13AMF Monitorización de la velocidad del eje

Descripción Para comprobar la AMF, se debe desplazar el eje controlado con una velocidad de aprox. el 10 % por encima del límite de velocidad configurado.

Lista de verificación

- Eje controlado: _____
- Instancia del eje controlado: _____
- Máxima velocidad del eje permitida: _____

N.º	Tarea	Sí	No relevante
1	La reacción configurada se activará cuando la velocidad del eje controlado supere la máxima velocidad permitida.		

13.11.6.14AMF Referenciación de posición



Esta AMF está vulnerable después del reinicio de la unidad de control del robot.

Lista de verificación

N.º	Tarea	Sí	No relevante
1	La reacción configurada se activa cuando uno de los ejes está referenciado.		

13.11.6.15AMF Referenciación de momentos



Esta AMF está vulnerable después del reinicio de la unidad de control del robot.

Lista de verificación

N.º	Tarea	Sí	No relevante
1	La reacción configurada se activa cuando uno de los sensores de momento de giro de un eje está referenciado.		

13.11.6.16AMF Control del campo del eje

Lista de verificación

- Eje controlado: _____
- Instancia del eje controlado: _____
- Límite inferior del campo del eje permitido: _____
- Límite superior del campo del eje permitido: _____

N.º	Tarea	Sí	No relevante
1	La reacción configurada se activa cuando no se alcanza el límite inferior del campo del eje permitido.		
2	La reacción configurada se activa cuando se supera el límite superior del campo del eje permitido.		

13.11.6.17AMF Control de velocidad cartesiana

Descripción

Para comprobar la AMF, se debe desplazar un punto controlado con una velocidad cartesiana de aprox. el 10 % por encima del límite de velocidad configurado.

Lista de verificación

- Instancia de la velocidad controlada: _____
- Máxima velocidad cartesiana permitida: _____

N.º	Tarea	Sí	No relevante
1	La reacción configurada se activará cuando la velocidad cartesiana de un punto controlado supere la máxima velocidad permitida.		

13.11.6.18AMF Control del campo de trabajo cartesiano / Monitorización cartesiana de las zonas de protección

Descripción En el primer paso debe comprobarse si la orientación de la zona de control está configurada correctamente. Para ello, se deben vulnerar 2 superficies espaciales que limiten entre sí, en al menos 3 puntos diferentes respectivamente.

En el segundo paso debe comprobarse si la extensión de la zona de control está configurada correctamente. Para ello, se deben vulnerar las otras superficies espaciales respectivamente en 1 punto. En total se debe realizar el desplazamiento hasta un mínimo de 10 puntos.

En el tercer paso debe comprobarse si la estructura que debe controlarse está configurada correctamente. Para ello, la monitorización de la zona se debe vulnerar con las bolas de control en el robot y en la herramienta si deben controlarse las dos estructuras, o solo con las bolas de control en el robot o en la herramienta.

- Lista de verificación**
- Tipo de zona de control: _____
 - Instancia de la zona de control: _____
 - Estructura monitorizada: _____
 - Desplazamiento del origen de la zona de control:
 - X: _____ mm
 - Y: _____ mm
 - Z: _____ mm
 - Orientación del origen de la zona de control:
 - S: _____ °
 - B: _____ °
 - C: _____ °
 - Longitud de la zona de control: _____ mm
 - Anchura de la zona de control: _____ mm

N.º	Tarea	Sí	No relevante
1	La configuración correcta de la zona de control se ha comprobado cómo se ha descrito anteriormente y la reacción configurada se activará cuando se haya vulnerado la zona de control.		
2	La reacción configurada se activará cuando se haya vulnerado la monitorización de la zona por las bolas de control en el robot.		
3	La reacción configurada se activará cuando se haya vulnerado la monitorización de la zona por las bolas de control en la herramienta.		

13.11.6.19AMF Identificación de colisión

Descripción La AMF se puede comprobar, mostrando los momentos del eje medidos actualmente en el smartPAD y, seguidamente, cargando ejes individuales.

- Lista de verificación**
- Instancia de la detección de colisión: _____
 - Máximo momento externo admisible: _____

N.º	Tarea	Sí	No relevante
1	La reacción configurada se activa cuando el momento externo de un eje supera el máximo momento externo admisible.		

13.11.6.20AMF Control de fuerza TCP

Descripción Para comprobar la AMF, es necesario un medio de medición adecuado, p. ej. una balanza de muelle.

- Lista de verificación**
- Instancia del control de fuerza TCP: _____
 - Máxima fuerza TCP externa permitida: _____

N.º	Tarea	Sí	No relevante
1	La reacción configurada se activa cuando la fuerza externa que actúa sobre el TCP, supera la máxima fuerza permitida.		

13.11.6.21AMF Retardo de tiempo

- Lista de verificación**
- Instancia del retardo de tiempo: _____
 - Tiempo de retardo: _____

N.º	Tarea	Sí	No relevante
1	La reacción configurada se activa después del tiempo configurado.		

13.11.6.22AMF Orientación de la herramienta

Descripción Para comprobar la AMF es necesario vulnerar el cono de control configurado a través del vector de referencia y el ángulo de divergencia en 3 rectas que estén posicionadas aprox. 120° desajustadas entre sí. Esta comprobación debe realizarse para dos ángulos notablemente distintos del eje A7 (diferencia recomendada: 20° como mínimo). De esta forma se asegura que el ángulo de orientación permitido, la orientación del vector de referencia y la orientación de la herramienta estén configurados correctamente.

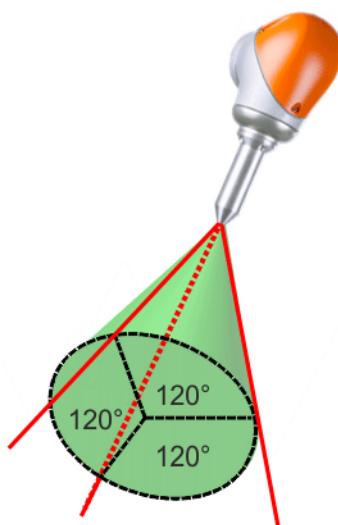


Fig. 13-17: Posición de las rectas en el cono de control

A través de 3 rectas posicionadas 120° desajustadas entre sí que se encuentren en el borde del cono de control, se fijan los ángulos de orientación del eje Z del Frame de orientación de herramientas que deben de ajustarse para re-

visar la AMF *Orientación de la herramienta*. La AMF debe estar vulnerada al superar los 3 ángulos de orientación de herramienta.

Procedimiento

El procedimiento explica a modo de ejemplo cómo se puede comprobar la configuración correcta del cono de control.

1. Alinear el eje Z del Frame de orientación de herramienta según el vector de referencia con respecto al sistema de coordenadas mundial.
2. Superar el ángulo de divergencia admisible inclinando el Frame de orientación de herramienta en B o C.
Debe ejecutarse la reacción configurada.
3. Alinear el eje Z del Frame de orientación de herramienta según el vector de referencia con respecto al sistema de coordenadas mundial.
Si se ha configurado una reacción de parada, es necesario comutar al modo de servicio KRF para desplazar el robot.
4. Girar el Frame de orientación de herramienta 120° en A.
5. Superar el ángulo de divergencia admisible inclinando el Frame de orientación de herramienta en B o C.
Debe ejecutarse la reacción configurada.
6. Alinear el eje Z del Frame de orientación de herramienta según el vector de referencia con respecto al sistema de coordenadas mundial.
Si se ha configurado una reacción de parada, es necesario comutar al modo de servicio KRF para desplazar el robot.
7. Girar el Frame de orientación de herramienta 120° en A.
8. Superar el ángulo de divergencia admisible inclinando el Frame de orientación de herramienta en B o C.
Debe ejecutarse la reacción configurada.

Lista de verificación

- Instancia de la orientación de herramienta: _____
- Orientación del vector de referencia con respecto al sistema de coordenadas mundial:
 - S: _____ °
 - B: _____ °
 - C: _____ °
- Zona de trabajo admisible (ángulo de divergencia): _____ °

N.º	Tarea	Sí	No relevante
1	Se ha comprobado la configuración correcta del cono de supervisión y la reacción configurada se ejecuta si se supera el ángulo admisible para las 3 rectas.		

13.11.7 Generar informe relativo a la configuración de seguridad

Descripción

Se puede elaborar un informe relativo a la configuración de seguridad actual e indicarse en el editor. El informe es editable y se puede imprimirse para propósitos de documentación.

El informe relativo a la configuración de seguridad facilita de acuerdo con esta las siguientes listas de verificación:

- Lista de verificación para comprobar las líneas utilizadas de la tabla PSM de usuarios
- Listas de verificación para comprobar los estados ESM utilizados y no utilizados
- Lista de verificación para comprobar las AMFs utilizadas



Las listas de verificación facilitadas por el informe relativo a la configuración de seguridad no son suficiente para una recepción completa de seguridad. Las siguientes listas de verificación adicionales deben comprobarse para una recepción completa de seguridad:

- Lista de verificación para una comprobación básica de la configuración de seguridad
- Lista de verificación para comprobar la herramienta destinada a la seguridad
- Lista de verificación para la comprobación de piezas de trabajo destinadas a la seguridad

El informe de configuración de seguridad contiene la siguiente información para poder asignar claramente la configuración de seguridad:

- Nombre del proyecto Sunrise al que pertenece la configuración de seguridad
- Versión de seguridad utilizada
- ID de seguridad (suma de control de la configuración de seguridad)
La ID de seguridad debe coincidir con la ID de la configuración de seguridad que está activada en la unidad de control del robot y que está activada.
- Fecha y horario de la última modificación de la configuración de seguridad

Procedimiento

- En el **Explorador de paquetes** hacer clic con el botón derecho del ratón sobre el proyecto deseado y seleccionar en el menú contextual **Sunrise > Crear informe de configuración de seguridad**.

El informe relativo a la configuración de seguridad se genera y se abre en la zona de los editores.

14 Principios de la programación de movimiento

En este capítulo se describen los fundamentos teóricos de la programación de movimientos.

La programación de movimientos en KUKA Sunrise.Workbench se describe en el siguiente capítulo: ([>>> 15 "Programación"](#) Página 269)

14.1 Resumen Tipos de movimiento



El punto de arranque de un movimiento siempre es el punto de destino del movimiento anterior.

Se pueden programar los siguientes tipos de movimiento como movimiento individual:

- Movimiento Point-to-Point (PTP)
([>>> 14.2 "Tipo de movimiento PTP"](#) Página 247)
- Movimiento lineal (LIN)
([>>> 14.3 "Tipo de movimiento LIN"](#) Página 248)
- Movimiento circular (CIRC)
([>>> 14.4 "Tipo de movimiento CIRC"](#) Página 248)
- Movimiento de guiado manual con equipo de guiado manual
([>>> 14.7 "Tipo de movimiento guiado manual"](#) Página 255)

Los siguientes tipos de movimiento se pueden programar como segmentos de un bloque Spline CP:

- Movimiento lineal (LIN)
- Movimiento circular (CIRC)
- Movimiento polinomial (SPL)

Los siguientes tipos de movimiento se pueden programar como segmentos de un bloque Spline JP:

- Movimiento Point-to-Point (PTP)
([>>> 14.6 "Tipo de movimiento Spline"](#) Página 249)

Los siguientes movimientos se resumen bajo el término "Movimientos CP" ("Continuous Path"):

- LIN, CIRC, SPL, bloques Spline CP

Los siguientes movimientos se resumen bajo el término "Movimientos JP" ("Joint Path"):

- Bloque Spline PTP, JP

14.2 Tipo de movimiento PTP

El robot desplaza el TCP al punto de destino a lo largo de la trayectoria más rápida. La trayectoria más rápida generalmente no es la trayectoria más corta en el espacio y por ello no es una recta. Dado que los ejes del robot se mueven al mismo tiempo de forma rotatoria, las trayectorias curvas se pueden ejecutar de forma más rápida que las trayectorias rectas.

El PTP es un movimiento de aproximación rápido. El recorrido exacto del movimiento no es previsible, pero permanece igual mientras que no se modifiquen las condiciones marco.

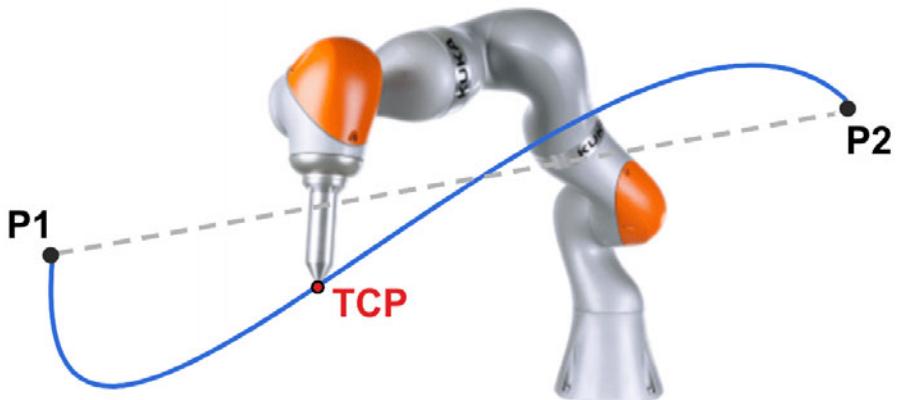


Fig. 14-1: Movimiento PTP

14.3 Tipo de movimiento LIN

El robot guía el TCP a la velocidad definida hasta el punto de destino a lo largo de una recta en la zona.

Para el movimiento LIN no se tiene en cuenta la configuración del robot de la posición de destino.

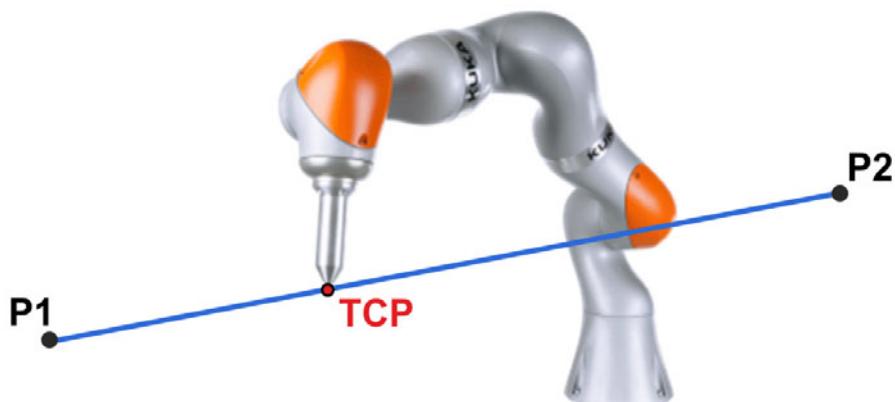


Fig. 14-2: Movimiento LIN

14.4 Tipo de movimiento CIRC

El robot conduce el TCP con una velocidad definida al punto de destino a lo largo de la trayectoria circular. La trayectoria circular queda definida por el punto de arranque, un punto intermedio y el punto de destino.

Para un movimiento CIRC no se tiene en cuenta la configuración del robot de la posición de destino.

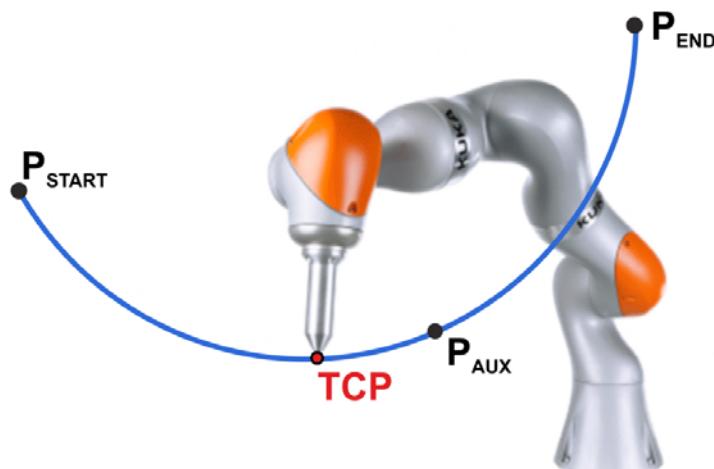


Fig. 14-3: Movimiento CIRC

14.5 Tipo de movimiento SPL

Con la ayuda del tipo de movimiento SPL se generan trayectorias curvas. Los movimientos SPL se agrupan siempre en bloques Spline. Se crean trayectorias que transcurren lisas a través de los puntos de destino del movimiento SPL.

Para el movimiento SPL no se tiene en cuenta la configuración del robot de la posición de destino.



Mediante la interconexión de 2 o más segmentos SPL se crean líneas curvas. Si se ejecuta un único segmento SPL, se comportará como una instrucción LIN.

14.6 Tipo de movimiento Spline

Spline es un tipo de movimiento especialmente apropiado para trayectorias curvas complejas. Con un movimiento Spline, el robot puede recorrer estas trayectorias complejas en un movimiento continuado. En principio, dichas trayectorias también se pueden crear con movimientos LIN y CIRC aproximados; no obstante, el Spline presenta una serie de ventajas.

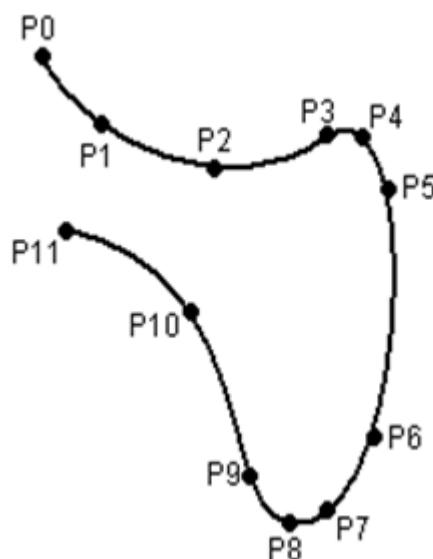
Los Splines se programan en bloques Spline. Con un bloque Spline, se agrupan varios movimientos en un mismo movimiento global. La unidad de control del robot configura y ejecuta el bloque Spline como 1 conjunto de movimientos.

Los movimientos que se pueden incluir en un bloque Spline se denominan segmentos Spline.

- Un bloque Spline CP puede incluir segmentos SPL, LIN y CIRC.
- Un bloque Spline JP puede incluir segmentos PTP.

Para un movimiento Spline cartesiano no se tiene en cuenta la configuración del robot de la posición de destino.

La configuración de la posición de destino de un segmento Spline depende de la configuración del robot en el inicio del segmento Spline.

Curso de trayectoria del bloque Spline**Fig. 14-4: Trayectoria curva con bloque spline**

- La trayectoria se define con puntos que se encuentran en la propia trayectoria. Estos puntos son los puntos de destino de cada uno de los segmentos Spline.
 - Todos los puntos se recorren sin parada exacta.
Excepción: la velocidad se reduce a 0.
(>>> 14.6.1 "Perfil de velocidad para movimientos Spline" Página 250)
 - Si todos los puntos se encuentran en un plano, la trayectoria se encuentra en este plano.
 - Si todos los puntos se encuentran en una recta, la trayectoria es una recta.
- Solo en muy pocos casos se produce una reducción de la velocidad.
(>>> 14.6.1 "Perfil de velocidad para movimientos Spline" Página 250)
- El curso de la trayectoria siempre es el mismo, independientemente del Override, de la velocidad o de la aceleración.
- Los círculos y los radios estrechos se recorren con gran precisión.

14.6.1 Perfil de velocidad para movimientos Spline

La unidad de control del robot tiene en cuenta los límites físicos del robot durante la planificación. El robot se mueve tan rápido como le es posible dentro de la velocidad programada, es decir, tanto como le permiten sus límites físicos.

La trayectoria siempre es la misma, independientemente del override, de la velocidad o de la aceleración.

Solo los efectos dinámicos que se producen, por ejemplo, por cargas de herramienta elevadas o por el ángulo de montaje del robot, pueden provocar desviaciones mínimas de la trayectoria si existen velocidades diferentes.

Reducción de la velocidad

En los siguientes casos de un Spline, no se alcanza la velocidad programada:

- Esquinas salientes, p. ej. debido a un cambio de dirección brusco
- Grandes cambios de orientación
- Cerca de singularidades

En los segmentos Spline se puede evitar una reducción de la velocidad provocada por grandes cambios de orientación, programando el control de la orientación `SplineOrientationType.Ignore`.

(>>> 14.9 "Control de la orientación LIN, CIRC, SPL" Página 258)

Reducción de la velocidad a 0

En los siguientes casos de un Spline se ejecuta una parada exacta:

- Segmentos Spline consecutivos con los mismos puntos de destino
- Segmentos LIN y/o CIRC consecutivos. Causa: Transcurso discontinuo de la dirección de la velocidad.

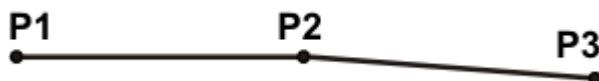


Fig. 14-5: Parada exacta en P2

En las transiciones LIN-CIRC, la velocidad también es 0 si la recta pasa tangencialmente al círculo. La causa de ello es que en el punto de transición entre la recta (curvatura igual a 0) y el círculo (curvatura no igual a 0), el recorrido de la curvatura es irregular.

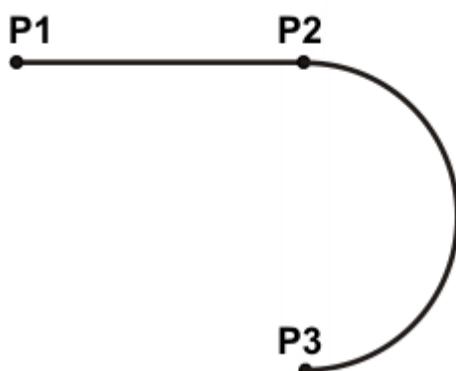


Fig. 14-6: Parada exacta en P2

Excepciones:

- Si se suceden una serie de segmentos LIN que crean una recta y que cambian la orientación uniformemente, la velocidad no se reduce.



Fig. 14-7: P2 es pasado sin parada exacta.

- En una transición CIRC-CIRC la velocidad no se reduce cuando ambos círculos tienen el mismo punto central y el mismo radio, y cuando las orientaciones cambian uniformemente. Debido a que la precisión necesaria a través de la programación por aprendizaje de los puntos de destino y auxiliares es difícil de alcanzar, se recomienda calcular los puntos del círculo.

14.6.2 Cambios en bloques Spline

Descripción

■ Cambio de la posición del punto:

Si se desplaza un punto dentro del bloque Spline, el cambio máx. de la trayectoria se efectúa en los dos segmentos antes de este punto y los dos segmentos consecutivos.

Normalmente, los desplazamientos de punto menores producen modificaciones de trayectoria menores. No obstante, en el caso de segmentos muy largos y segmentos muy cortos consecutivos, incluso las modificaciones menores podrían tener efectos considerables.

■ Cambio del tipo de segmento:

En caso de cambiar un segmento SPL en un segmento LIN y vice versa, la trayectoria cambia en el segmento precedente y en el segmento consecutivo.

Ejemplo 1

Trayectoria original:

```
Spline mySpline = new Spline(  
    splgetApplicationData().getFrame("/P1")),  
    splgetApplicationData().getFrame("/P2")),  
    splgetApplicationData().getFrame("/P3")),  
    splgetApplicationData().getFrame("/P4")),  
    circ(getApplicationData().getFrame("/P5")),  
        getApplicationData().getFrame("/P6")),  
    splgetApplicationData().getFrame("/P7")),  
    lin(getApplicationData().getFrame("/P8"))  
);  
...  
robot.move(ptp(getApplicationData().getFrame("/P0")));  
robot.move(mySpline);
```

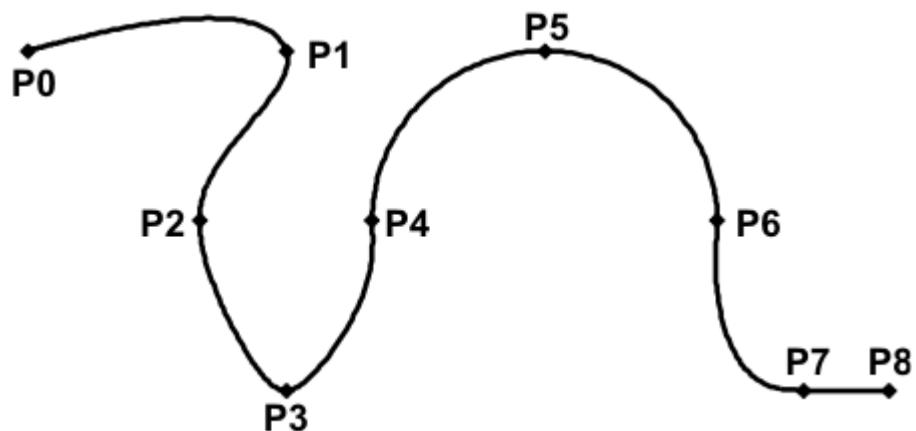


Fig. 14-8: Trayectoria original

Respecto a la trayectoria original, se desplaza un punto:

P3 se desplaza. Así, se modifica la trayectoria en los segmentos P1 - P2, P2 - P3 y P3 - P4. El segmento P4 - P5 no se modifica en este caso porque pertenece a un segmento CIRC, definiéndose así una trayectoria circular.

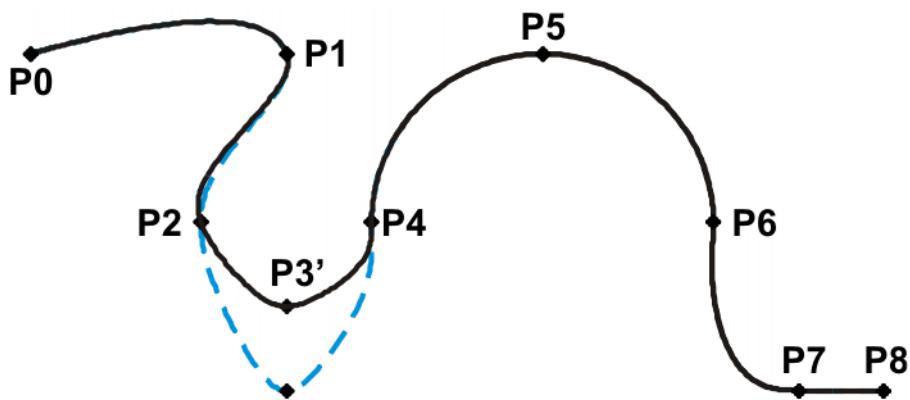


Fig. 14-9: El punto se desplazó

Por lo que respecta a la trayectoria original, se modifica el tipo de un segmento:

En el caso de la trayectoria original el tipo de segmento cambia de P2 - P3 de SPL a LIN. La trayectoria cambia en los segmentos P1 - P2, P2 - P3 y P3 - P4.

```
Spline mySpline = new Spline(
    spl(getApplicationContext().getFrame("/P1")),
    spl(getApplicationContext().getFrame("/P2")),
    lin(getApplicationContext().getFrame("/P3")),
    spl(getApplicationContext().getFrame("/P4")),
    circ(getApplicationContext().getFrame("/P5")),
    getApplicationContext().getFrame("/P6")),
    spl(getApplicationContext().getFrame("/P7")),
    lin(getApplicationContext().getFrame("/P8"))
);
...
robot.move(ptp(getApplicationContext().getFrame("/P0")));
robot.move(mySpline);
```

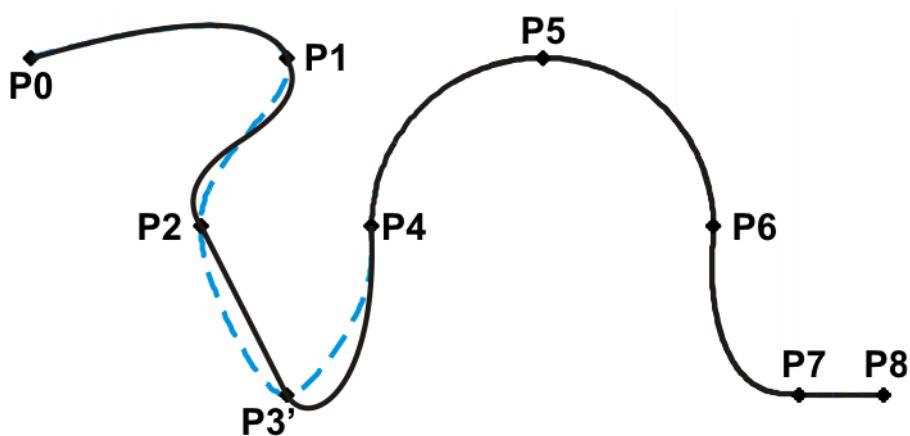


Fig. 14-10: Tipo de segmento modificado

Ejemplo 2

Trayectoria original:

```
Spline mySpline = new Spline(
    spl(getApplicationContext().getFrame("/P2")),
    spl(getApplicationContext().getFrame("/P3")),
```

```

spl(getApplicationData().getFrame("/P4")),
spl(getApplicationData().getFrame("/P5")),
);
...
robot.move(mySpline);

```

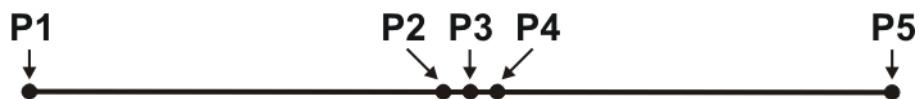


Fig. 14-11: Trayectoria original

Las siguientes coordenadas de Frame se han programado por aprendizaje:

Frame	X	Y	Z
P2	100.0	0.0	0.0
P3	102.0	0.0	0.0
P4	104.0	0.0	0.0
P5	204.0	0.0	0.0

Respecto a la trayectoria original, se desplaza un punto:

P3 se desplaza ligeramente en dirección Y. Por ello la trayectoria cambia en todos los segmentos representados.

Frame	X	Y	Z
P3	102.0	1.0	0.0

Dado que P2 - P3 y P3 - P4 son segmentos muy cortos y P1 - P2 y P4 - P5 segmentos largos, el desplazamiento pequeño provoca un cambio importante de la trayectoria.

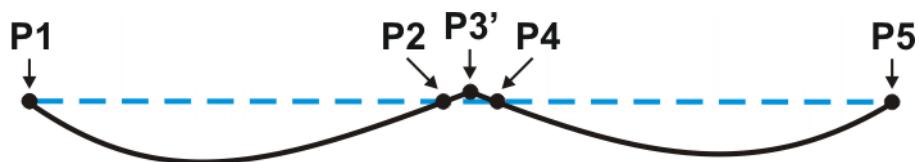


Fig. 14-12: El punto se ha desplazado.

Solución:

- Distribuir las distancias de puntos de forma más constante.
- Programar las rectas (excepto las rectas muy cortas) como segmentos LIN.

14.6.3 Transición LIN-SPL-LIN

Generalmente, en una secuencia de segmentos LIN-SPL-LIN es recomendable que el segmento SPL discurre dentro del ángulo menor entre las dos rectas. En función del punto inicial y de destino del segmento SPL, la trayectoria puede encontrarse también al exterior.

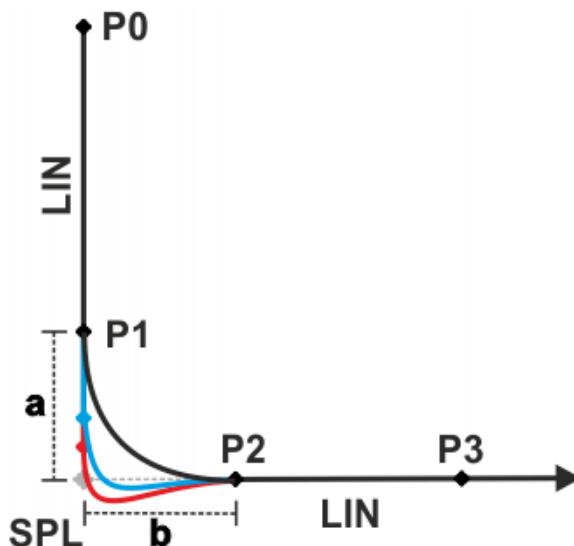


Fig. 14-13: LIN-SPL-LIN

La trayectoria se encuentra dentro del ángulo interior si se cumplen las condiciones siguientes:

- Los dos segmentos LIN se cruzan en sus prolongaciones.
 - $2/3 \leq a/b \leq 3/2$
- a = Distancia del punto de inicio del segmento SPL al punto de intersección de los segmentos LIN
- b = Distancia del punto de intersección de los segmentos LIN al punto de destino del segmento SPL

14.7 Tipo de movimiento guiado manual

Descripción El robot se puede guiar manualmente con la ayuda de un equipo manual de guiado. El equipo de guiado manual es un dispositivo que se requiere para el guiado manual del robot y que está equipado con un dispositivo de validación.

El modo de guiado manual se puede conectar con la instrucción de movimiento handGuiding() en la aplicación. El guiado manual comienza en la posición real que se haya alcanzado por última vez antes de la conexión.

(>>> 15.9 "Programar el guiado manual" Página 288)

En el modo de guiado manual, el robot reacciona de forma flexible a las fuerzas externas y puede ser guiado manualmente a cualquier punto en la zona cartesiana. Al conmutar al modo de guiado manual se fijan automáticamente los parámetros de impedancia. No es posible modificar los parámetros de impedancia para el guiado manual.

Condición previa

- Equipo de guiado manual con pulsador de validación
- Está configurado un estado ESM con la AMF **Validación equipo de guiado manual** (para el movimiento de guiado manual).

(>>> 13.8.7 "Control de dispositivos de validación en equipos manuales de guiado" Página 205)

Cuando se presiona el pulsador de validación en el equipo de guiado manual y se mantiene presionado en la posición intermedia, el robot se puede guiar manualmente. Si se aprieta hasta el fondo o se suelta el pulsador de validación, se retira la validación para el guiado manual y el robot permanece en su posición actual.

- Está configurado un estado ESM sin AMF **Validación equipo de guiado manual** (para todos los movimientos excepto el movimiento de guiado manual).
- Modo de servicio Automático
La configuración debe ser conforme a la cooperación hombre-robot (MRK).



En el guiado manual, los parámetros seleccionados incorrectamente (p. ej. datos de carga defectuosos, herramienta incorrecta), la información incorrecta (p. ej. sobre sensores de momentos defectuosos) o fuerzas adicionales conectadas se pueden interpretar como fuerzas externas y pueden provocar movimientos inesperados del robot.



Si se concede la validación para el guiado manual antes de que se conecte el modo de guiado manual en la aplicación, dicho modo de guiado manual estará activo inmediatamente después de la conexión. Es decir, la ejecución del movimiento no se pausa con la conexión, de forma que el paso entre el modo de la aplicación y el modo de guiado manual sea fluido.

14.8 Posicionamiento aproximado

Posicionamiento aproximado significa: El punto de destino de un movimiento programado no se desplaza de forma precisa, para que sea posible un movimiento continuo del robot. Para la programación del movimiento se encuentran disponibles diferentes parámetros para influir en el posicionamiento aproximado.

El punto en el que se abandona la trayectoria original y comienza el arco de aproximación se denomina punto de posicionamiento aproximado.



Para aproximar movimientos sin parada exacta, se deberán ejecutar de forma asíncrona o agruparse en un MotionBatch.
(>>> 15.7.1 "Estructura de una instrucción de movimiento (move/moveAsync)" Página 280)
(>>> 15.7.6 "MotionBatch" Página 284)



En el posicionamiento aproximado de movimientos ejecutados de forma sincronizada se ejecuta una parada exacta al principio del arco de aproximación. En la ejecución sincronizada, esto también se aplica para el último movimiento dentro de un MotionBatch.

Movimiento PTP

El TCP abandona la trayectoria sobre la cual se posicionaría exactamente en el punto de destino y, en su lugar, recorre una trayectoria con la que puede pasar por el punto de destino sin parada exacta. De este modo, la trayectoria pasa junto al punto, pero ya no a través del mismo.

En la programación se establece la distancia relativa hasta el punto de destino en el que el TCP puede abandonar como muy pronto su trayectoria original en la zona del eje. Una distancia relativa del 100 % corresponde a la trayectoria completa desde el punto de inicio hasta el punto de destino del movimiento.

El contorno de aproximación por el que se desplaza el TCP no es forzosamente la trayectoria más corta en la zona cartesiana. Por ello, el punto aproximado también puede estar dentro del arco de aproximación.

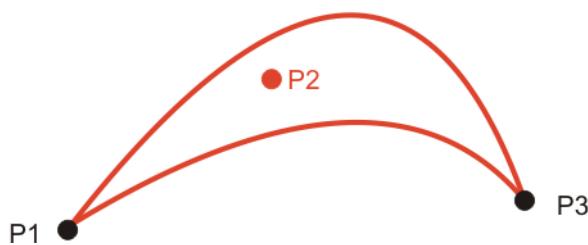


Fig. 14-14: Movimiento PTP, P2 tuvo un posicionamiento aproximado

Movimiento LIN

El TCP abandona la trayectoria sobre la cual se posicionaría exactamente en el punto de destino y recorre una trayectoria más corta. En la programación del movimiento se define la distancia mínima al punto de destino en que el TCP puede abandonar su trayectoria original.

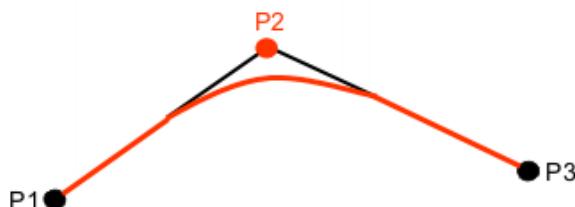


Fig. 14-15: Movimiento LIN, P2 tuvo un posicionamiento aproximado

Movimiento CIRC

El TCP abandona la trayectoria sobre la cual se posicionaría exactamente en el punto de destino y recorre una trayectoria más corta. En la programación del movimiento se define la distancia mínima al punto de destino en que el TCP puede abandonar su trayectoria original.

Puede ocurrir que el punto auxiliar entre en la zona de aproximación y no se pase de forma exacta por el mismo. Esto depende de la posición del punto auxiliar y de los parámetros de aproximación programados.

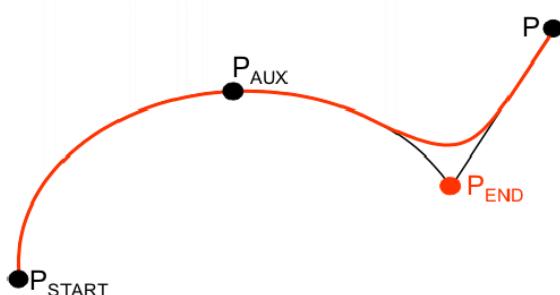


Fig. 14-16: Movimiento CIRC, P_END tuvo posicionamiento aproximado

Se puede realizar una aproximación conjunta de los bloques Spline y los movimientos individuales. No importa si se trata de bloques Spline CP o JP ni el tipo de movimiento individual.

El arco de aproximación se corresponde siempre con el segundo movimiento, en lo que respecta al tipo de movimiento. Durante la aproximación PTP-LIN, por ejemplo, el arco de aproximación es del tipo CP.

Si se realiza una aproximación a un bloque Spline, se aproximará el último segmento completo. Si el bloque Spline se compone solo de un segmento, se

aproximará como máximo la mitad del segmento (se aplica también para PTP, LIN y CIRC).

El posicionamiento aproximado no es posible debido al tiempo:

Si, debido a un retardo temporal al ordenar el movimiento, no es posible efectuar un posicionamiento aproximado, el robot espera al comienzo del arco de aproximación. El robot continúa en el momento en que se pueda planear el paso siguiente. A continuación el robot recorre el arco de aproximación. De hecho, la aproximación es posible, solo se retarda.

Sin posicionamiento aproximado en el modo Step:

En el modo Step también se realizará el desplazamiento preciso hasta el punto de destino con movimientos aproximados

En el posicionamiento aproximado de un bloque Spline a un bloque Spline, como consecuencia del comportamiento exacto, la trayectoria en el último segmento del primer bloque y en el primer segmento del segundo bloque la trayectoria es distinta que en el modo estándar.

En todos los demás segmentos en los dos bloques Spline, la trayectoria es igual en ambos tipos de flujo de programa.

Los movimientos de posicionamiento aproximado, que ya hayan sido enviados de forma asíncrona a la unidad de control del robot antes de la activación del modo Step y que esperan allí a la ejecución, se detienen en el punto de posicionamiento aproximado. Para estos movimientos aún se recorre el arco de aproximación para continuar.

14.9 Control de la orientación LIN, CIRC, SPL

Descripción

El TCP puede tener en el punto inicial una orientación distinta que en el punto de destino. En la programación de movimientos se puede establecer cómo se debe proceder con las diferentes orientaciones.

El control de la orientación se ajusta como parámetro de movimiento mediante el método `setOrientationType(...)`. El control de la orientación es un valor del tipo `Enum SplineOrientationType`.

Control de la orientación	Descripción
Constant	<p>La orientación del TCP se mantiene de forma constante durante el movimiento.</p> <p>Se mantiene la orientación del punto inicial. La orientación del punto de destino no se tiene en cuenta.</p>
Ignore	<p>La orientación del TCP se modifica durante el movimiento.</p> <p>Esta opción solo está disponible para segmentos Spline individuales, pero no para el bloque Spline completo o para movimientos individuales. El control calcula el control de la orientación basándose en la orientación de los puntos de apoyo del entorno, siempre que su orientación no sea ignorada.</p> <p>Ignore se utiliza cuando para un segmento Spline no es necesaria una orientación determinada.</p> <p>(>>> "Ignore" Página 260)</p> <p>Indicación: Con Ignore, la orientación del punto de destino no se tiene en cuenta. Si es necesario respetar la orientación programada por aprendizaje en el punto de destino para, por ejemplo evitar colisiones, no se podrá utilizar Ignore.</p>

Control de la orientación	Descripción
OriJoint	<p>La orientación del TCP se modifica continuamente durante el movimiento. Esto se logra mediante el desplazamiento lineal del ángulo del eje de la muñeca (desplazamiento específico del eje).</p> <p>Indicación: Utilizar OriJoint cuando el robot con VariableOrientation presente una singularidad de ejes de la muñeca. La orientación del TCP se modifica de forma continua durante el movimiento, pero no de forma completamente uniforme. Por ello, OriJoint no es adecuado cuando tiene que mantenerse exactamente un curso determinado de la orientación, como p. ej. en el caso de la soldadura láser.</p>
VariableOrientation	<p>La orientación del TCP es llevada durante el movimiento continuamente desde la orientación del punto de inicio hasta la orientación del punto de destino.</p> <p>Si el control de la orientación no está ajustado, se aplicará por defecto este control de la orientación.</p>

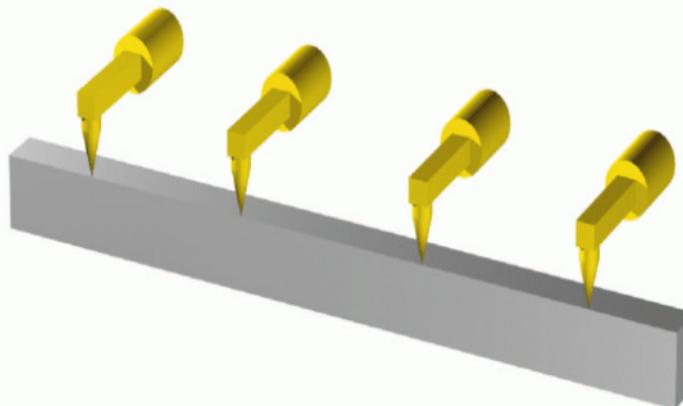


Fig. 14-17: Orientación constante (Constant)

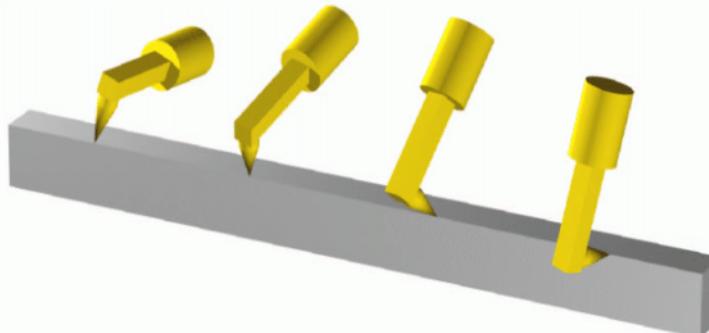


Fig. 14-18: Orientación variable (VariableOrientation o OriJoint)

Movimiento CIRC

En los movimientos CIRC se puede establecer si el control de la orientación se debe referir al espacio o a la trayectoria.

(>> 14.9.1 "CIRC – Sistema de referencia del control de la orientación" Página 260)

Para movimientos CIRC la unidad de control del robot tiene en cuenta la orientación del punto de destino. Se puede determinar si la orientación del punto auxiliar se debe tener en cuenta y en qué medida. Además se puede determinar el comportamiento de la orientación en el punto de destino.

Ignore

El tipo de orientación SplineOrientationType.Ignore se utiliza cuando en un punto no es necesaria una orientación determinada. La unidad de control del robot ignora la orientación programada por aprendizaje o programada del punto. En lugar de ello, calcula la orientación óptima para este punto en base a las orientaciones de los puntos periféricos. Esto reduce el tiempo de ciclo.

Ejemplo:

```
robot.move(P0);
Spline path6 = new Spline(
    spl(P1),
    spl(P2),
    spl(P3).setOrientationType(SplineOrientationType.Ignore),
    spl(P4).setOrientationType(SplineOrientationType.Ignore),
    spl(P5),
    spl(P6)
),
...
robot.move(path6);
```

La orientación programada por aprendizaje o programada de P3 y P4 es ignorada.

Para los siguientes segmentos Spline no está permitido SplineOrientationType.Ignore:

- El primer y el último segmento en un bloque Spline
- Segmentos CIRC con OrientationReferenceSystem.Path
- Segmentos seguidos por un segmento CIRC con OrientationReferenceSystem.Path
- Segmentos seguidos por un segmento con SplineOrientationType.Constant
- Segmentos consecutivos en un bloque Spline que poseen el mismo punto de destino

14.9.1 CIRC – Sistema de referencia del control de la orientación

Descripción

En los movimientos CIRC se puede establecer si el control de la orientación se debe referir al espacio o a la trayectoria.

El sistema de referencia del control de la orientación se ajusta como parámetro de movimiento mediante el método setOrientationReferenceSystem(...). El control de la orientación es un valor del tipo Enum OrientationReferenceSystem.

El sistema de referencia del control de la orientación solo se puede indicar delante del tipo de orientación.

Sistema de referencia	Descripción
Base	Control de la orientación referido a la base durante el movimiento circular
Path	Control de la orientación relacionada con la trayectoria durante el movimiento circular

Ejemplo

Movimiento circular referido a la trayectoria con orientación constante:

```
robot.move(circ(P6, P7)
.setOrientationReferenceSystem(OrientationReferenceSystem.Path)
.setOrientationType(SplineOrientationType.Constant));
```

Restricción

Para los siguientes segmentos Spline no está permitido OrientationReferenceSystem.Path:

- Segmentos CIRC con SplineOrientationType.Ignore
- Segmentos CIRC a los que precede un segmento con SplineOrientationType.Ignore

14.9.2 CIRC – Combinación de sistema de bus y tipo de control de la orientación

Si el sistema de bus del control de la orientación se combina con SplineOrientationType.OriJoint, el sistema de bus no influirá en el control de la orientación.

Movimiento circular referido a la trayectoria con orientación constante:

- OrientationReferenceSystem.Path
- SplineOrientationType.Constant

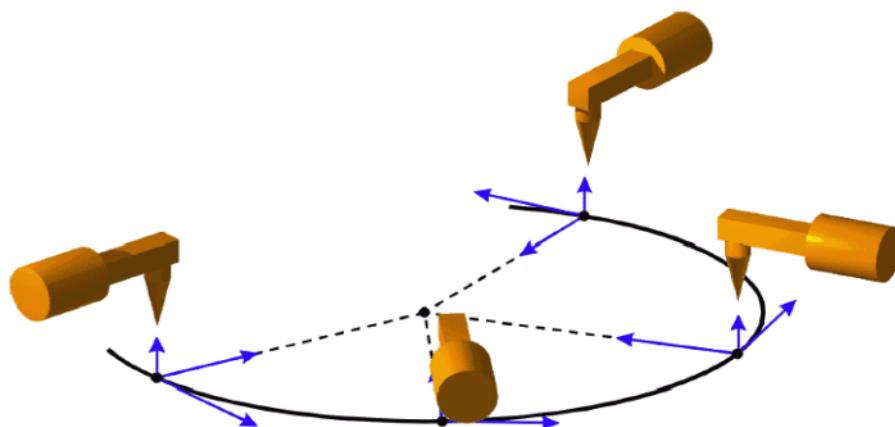


Fig. 14-19: Orientación constante, referida a la trayectoria

Movimiento circular referido a la trayectoria con orientación variable:

- OrientationReferenceSystem.Path
- SplineOrientationType.VariableOrientation

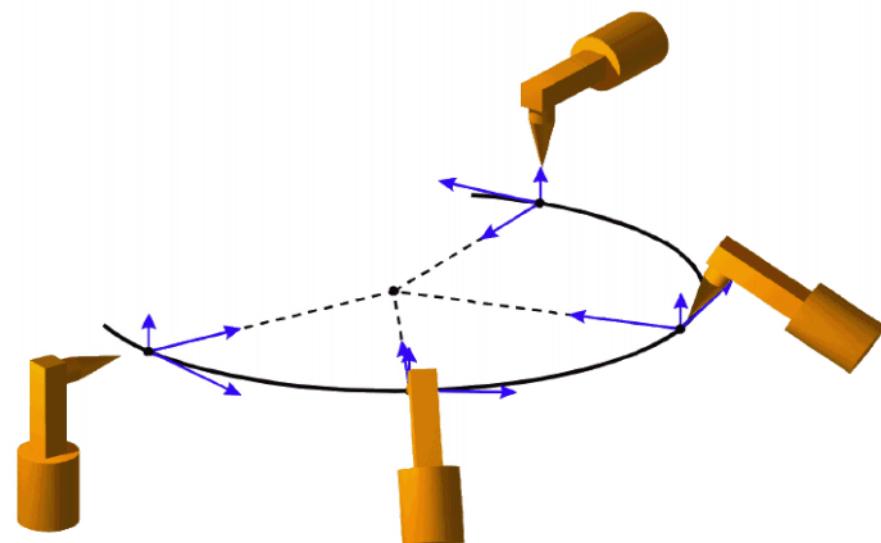
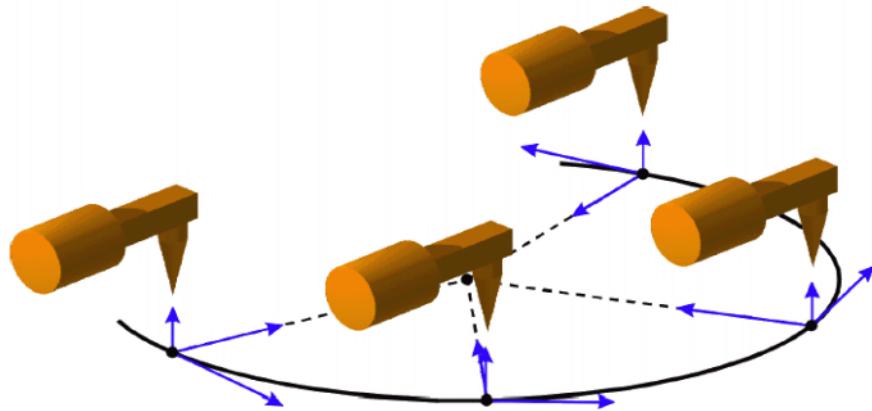


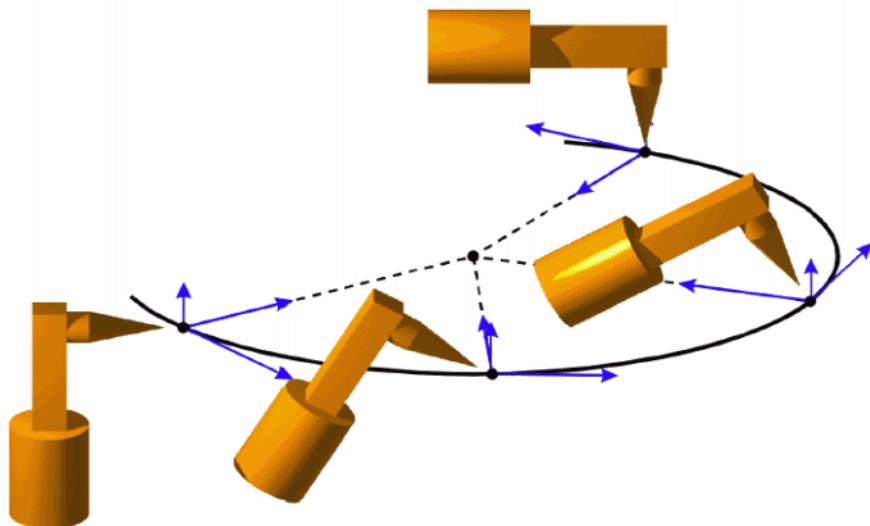
Fig. 14-20: Orientación variable, referida a la trayectoria

Movimiento circular referido a la base con orientación constante:

- OrientationReferenceSystem.Base
- SplineOrientationType.Constant

**Fig. 14-21: Orientación constante, referida a la base****Movimiento circular referido a la base con orientación variable:**

- OrientationReferenceSystem.Base
- SplineOrientationType.VariableOrientation

**Fig. 14-22: Orientación variable, referida a la base**

14.10 Información de redundancia

Para una posición dada del eje de un robot, el punto resultante está definido de forma única en la zona cartesiana en la que se encuentra el TCP. Sin embargo, si es al contrario, la posición del eje del robot no puede determinarse de forma única partiendo de la posición cartesiana X, Y, Z y la orientación A, B, C del TCP. Un punto cartesiano puede alcanzarse con varias configuraciones de eje. Para poder determinar una configuración única es necesario indicar el parámetro de estado.

En los robots con 6 ejes ya existe ambigüedad en las posiciones del eje para un punto cartesiano determinado. Gracias a su 7.^o eje adicional, KUKA LBR

iiwa puede alcanzar una posición y una orientación determinada con cualquier número de posiciones del eje, en teoría. Para determinar la posición del eje de forma unívoca, en el caso de un LBR iiwa es necesario indicar el ángulo de redundancia además del estado.

El parámetro Turn se requiere para ejes que pueden exceder el ángulo $\pm 180^\circ$. En los movimientos PTP sirve para definir de forma unívoca el sentido de giro de los ejes. En los movimientos CP, el parámetro Turn no tiene ninguna influencia.

Estado, Turn y el ángulo de redundancia se guardan durante la programación por aprendizaje de un Frame. Se gestionan como campos del tipo de datos AbstractFrame.

Programación

El estado de un Frame se tiene en cuenta solamente durante los movimientos PTP a este Frame. En los movimientos CP se utiliza el estado que la configuración del eje proporciona al principio del movimiento.

Para evitar un movimiento imprevisible al principio de una aplicación y para definir una configuración del eje unívoca, se recomienda programar el primer movimiento en una aplicación con una de las siguientes instrucciones. La configuración del eje no debe estar cerca de una posición del eje singular.

- Movimiento PTP a una configuración del eje predefinida con la indicación de todos los valores del eje:

```
ptp(double a1, double a2, double a3, double a4, double  
a5, double a6, double a7)
```

- Movimiento PTP a una configuración del eje predefinida:

```
ptp(JointPosition joints)
```

- Movimiento PTP a un Frame programado por aprendizaje (tipo: AbstractFrame):

```
ptp(getApplicationContext().getFrame(String frameName));
```

14.10.1 Ángulo de redundancia

Gracias a sus 7 ejes, el KUKA LBR iiwa está capacitado para alcanzar un punto en el espacio, con un número teórico ilimitado de diferentes configuraciones del eje. Mediante el ángulo de redundancia se define una posición unívoca.

En un LBR iiwa, el ángulo de redundancia contiene el valor del 3° eje.

Para todos los movimientos se aplica:

- El ángulo de redundancia del Frame de destino se tiene en cuenta si para la programación por aprendizaje del Frame se ha utilizado el robot que también ejecuta la instrucción de movimiento. Sobre todo debe coincidir el nombre del robot definido en la configuración de la estación con el nombre del dispositivo indicado en las propiedades del Frame.
- Si los robots no coinciden o se utilizan Frames calculados, se conservará el ángulo de redundancia indicado por la configuración del eje al comienzo del movimiento.

14.10.2 Estado

La indicación de Status (estado) evita ambigüedades en la posición de eje. El estado se describe mediante un número binario con 3 bits.

Bit 0

El bit 0 indica la posición del punto de la raíz de la muñeca (punto de intersección de los ejes A5, A6, A7) referido al eje X del sistema de coordenadas del eje A1. La orientación del sistema de coordenadas A1 es idéntico al sistema de coordenadas del pie del robot, cuando el eje A1 está en 0° . Se desplaza junto con el eje A1.

Posición	Valor
Área por encima de la cabeza El robot se encuentra en el área por encima de la cabeza si es negativo el valor x de la posición del punto de la raíz de la muñeca referido al sistema de coordenadas A1.	Bit 0 = 1
Área de base El robot se encuentra en el área de base si es positivo el valor x de la posición del punto de la raíz de la muñeca referido al sistema de coordenadas A1.	Bit 0 = 0

Bit 1

En un LBR iiwa, el Bit 1 indica la posición del eje A4.

Posición	Valor
A4 < 0°	Bit 1 = 1
A4 ≥ 0°	Bit 1 = 0

Bit 2

En un LBR iiwa, el Bit 2 indica la posición del eje A6.

Posición	Valor
A6 ≤ 0°	Bit 2 = 1
A6 > 0°	Bit 2 = 0

Para los movimientos PTP se aplica:

- El estado del Frame de destino se tiene en cuenta si para la programación por aprendizaje del Frame se ha utilizado el robot que también ejecuta la instrucción de movimiento. Sobre todo debe coincidir el nombre del robot definido en la configuración de la estación con el nombre del dispositivo indicado en las propiedades del Frame.
- Si los robots no coinciden o se utilizan Frames calculados, se conservará el estado indicado por la configuración del eje al comienzo del movimiento.

Para los movimientos CP se aplica:

- El estado del Frame de destino no se tiene en cuenta. Se conserva el estado indicado por la configuración del eje al comienzo del movimiento.
- **Excepción:** Es posible realizar la modificación del estado si el Frame de destino es desplazado con el control de la orientación `SplineOrientationType.OriJoint`. En este caso tampoco se tiene en cuenta el estado del Frame de destino. El estado del final del movimiento se determina mediante la planificación de la trayectoria que selecciona el recorrido más corto hasta la posición de destino.

14.10.3 Turn

La indicación de Turn (giro) también permite poder alcanzar ángulos del eje superiores a +180° o inferiores a -180°, sin una estrategia de desplazamiento especial (p. ej., puntos intermedios). El Turn se indica mediante un número binario con 7 bits.

En los ejes rotatorios, los bits individuales determinan el signo que precede al valor del eje del siguiente modo:

Bit = 0: Ángulo ≥ 0°

Bit = 1: ángulo < 0°

Valor	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	A7 $\geq 0^\circ$	A6 $\geq 0^\circ$	A5 $\geq 0^\circ$	A4 $\geq 0^\circ$	A3 $\geq 0^\circ$	A2 $\geq 0^\circ$	A1 $\geq 0^\circ$
1	A7 $< 0^\circ$	A6 $< 0^\circ$	A5 $< 0^\circ$	A4 $< 0^\circ$	A3 $< 0^\circ$	A2 $< 0^\circ$	A1 $< 0^\circ$

En un LBR iiwa, el Turn no se tiene en cuenta, ya que ninguno de sus ejes puede girar más de $\pm 180^\circ$.

14.11 Singularidades

Es posible que se limiten los movimientos cartesianos del robot debido a la posición de eje. El motivo es que, debido a la combinación de posiciones de eje de todo el robot, no se pueden transmitir movimientos de los accionamientos a la brida (o a un objeto en la brida, p. ej. una herramienta) al menos en una dirección cartesiana. En este caso o si las modificaciones cartesianas mínimas requieren cambios muy grandes en el ángulo de los ejes, se habla de posiciones singulares.



Se recomienda desplazar el robot de forma especialmente lenta en las inmediaciones de singularidades.

14.11.1 Singularidades cinemáticas

La flexibilidad a través de la redundancia de un robot de 7 ejes hace que, en comparación con el robot de 6 ejes, deban estar activas al mismo tiempo 2 o más condiciones cinemáticas (p. ej. posición extendida, 2 ejes de giro se encuentran uno sobre otro), para alcanzar una posición singular. Existen 4 posiciones diferentes del robot en las que ya no es posible el movimiento de la brida en una dirección cartesiana. En este caso, solo es importante la posición de uno o 2 ejes respectivamente. Los otros ejes pueden adoptar una posición cualquiera.

Singularidad A4

Esta singularidad cinemática se cumple con $A4 = 0^\circ$. Se denomina posición extendida.

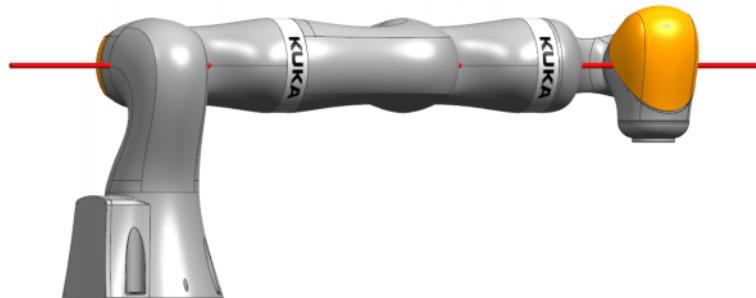


Fig. 14-23: Posición extendida $A4 = 0^\circ$

El movimiento está bloqueado en la dirección del pie del robot o de forma paralela al eje A3 o A5. Una condición cinemática adicional para esta singularidad es que se alcancen los límites del campo de trabajo. Se cumple automáticamente con $A4 = 0^\circ$.

Si el brazo del robot está extendido, se perderá un grado de libertad de movimiento para el punto de la raíz de la muñeca (ya no se puede desplazar de nuevo a lo largo del eje del brazo del robot). La posición de los ejes A3 y A5 no se puede deshacer de nuevo.

Singularidad A4/ A6

Esta singularidad cinemática se cumple con $A4 = 90^\circ$ y $A6 = 0^\circ$.

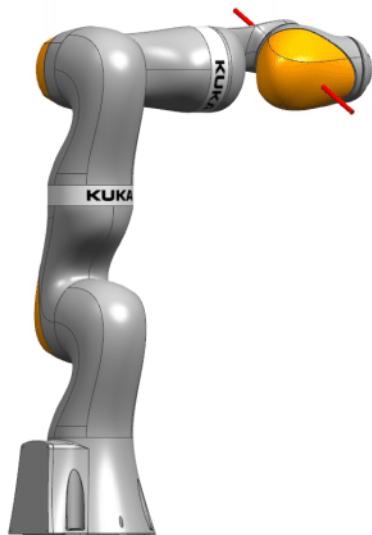


Fig. 14-24: $A_4 = 90^\circ$ y $A_6 = 0^\circ$

El movimiento de forma paralela al eje A6 o A2 está bloqueado.

- Singularidad A2/
A3** Esta singularidad cinemática se cumple con $A_2 = 0^\circ$ y $A_3 = \pm 90^\circ$ ($\pi/2$).

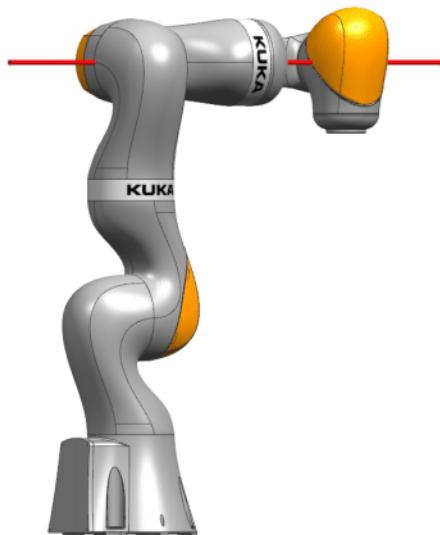


Fig. 14-25: $A_2 = 0^\circ$ y $A_3 = \pm 90^\circ$ ($\pi/2$)

El movimiento está bloqueado en la dirección del robot o de forma paralela al eje A2 o A5.

- Singularidad A5/
A6** Esta singularidad cinemática se cumple con $A_5 = \pm 90^\circ$ ($\pi/2$) y $A_6 = 0^\circ$.



Fig. 14-26: $A_5 = \pm 90^\circ (\pi/2)$ y $A_6 = 0^\circ$

El movimiento de forma paralela al eje A6 está bloqueado.

14.11.2 Singularidades condicionadas por el sistema

La construcción redundante del LBR con el 7º eje hace posible que el brazo del robot se mueva sin que se mueva la brida. En este denominado como movimiento de espacio cero, se desplazan todos los ejes, excepto A4, el "ángulo del codo". De forma adicional a la redundancia normal, con determinadas condiciones previas es posible que solo se desplacen cadenas parciales del robot y no todos los ejes.

Las posiciones del robot que se incluyen en esta categoría, tienen todas en común que con pequeñas modificaciones cartesianas, se activan grandes modificaciones del ángulo del eje. Son muy similares a las singularidades que son conocidas por los robots de 6 ejes, ya que en el LBR también se realiza la separación de la parte de posición y de orientación del punto de la raíz de la muñeca.

Singularidades de los ejes de la muñeca

Con la singularidad de los ejes de la muñeca, se denomina la posición del eje $A_6 = 0^\circ$. De este modo, la posición de los ejes A5 y A7 ya no se puede deshacer y existen posibilidades infinitas para ajustar estos dos ejes, para generar la misma posición en la brida.

Singularidad A1

Si el punto de la raíz de la muñeca se encuentra directamente sobre A1, de acuerdo con la definición anterior no se podrá indicar ningún valor de referencia para el círculo de redundancia, ya que para $A_3 = 0^\circ$ está permitido cualquier valor A1.

Cada posición del eje de A1 se puede compensar mediante una combinación de A5, A6 y A7, de forma que la posición de la brida permanezca inalterada.

Singularidad A2

Si el "hombro" está estirado, ya no se podrá deshacer la posición de los ejes A1 y A3 de acuerdo con patrón anterior.

Singularidad A2/A4

Si A1 y A7 coinciden, ya no se podrá deshacer la posición de los ejes A1 y A7 de acuerdo con patrón anterior.



Las singularidades condicionadas por el sistema se pueden evitar en la mayoría de los casos mediante el ajuste adecuado del codo.

15 Programación

15.1 Editor de Java

15.1.1 Abrir la aplicación del robot en el editor de Java

Descripción En el editor de Java se pueden tener abiertos varios archivos simultáneamente. Si es necesario se pueden mostrar uno al lado de otro o de forma superpuesta. Esto permite, p. ej., comparar cómodamente sus contenidos.

Condición previa ■ La aplicación del robot está creada.

(>>> 5.4 "Crear una nueva aplicación del robot" Página 54)

Procedimiento 1. En el **Explorador de paquetes**, hacer doble clic en un archivo Java. O bien: Marcar el archivo y seleccionar la secuencia de menú **Navegar > Abrir**. O bien: Hacer clic con el botón derecho del ratón sobre el archivo y en el menú contextual seleccionar **Abrir** o **Abrir con > Editor de Java**.
2. Para cerrar el archivo: hacer clic en la "X" situada en la parte superior derecha de la pestaña correspondiente.

15.1.2 Estructura de una aplicación del robot

```

1 package application;
2
3 import com.kuka.roboticsAPI.applicationModel.RoboticsAPIApplication;
4
5 public class RobotApplication extends RoboticsAPIApplication {
6     private Controller kuka_Sunrise_Cabinet_1;
7     private LBR lbr_iwa_7_R800_1;
8
9     public void initialize() {
10         kuka_Sunrise_Cabinet_1 = getController("KUKA_Sunrise_Cabinet_1");
11         lbr_iwa_7_R800_1 = (LBR) getRobot(kuka_Sunrise_Cabinet_1,
12                                         "LBR_iwa_7_R800_1");
13     }
14
15     public void run() {
16         lbr_iwa_7_R800_1.move(ptpHome());
17     }
18
19     /**
20      * Auto-generated method stub. Do not modify the contents of this method.
21      */
22
23     public static void main(String[] args) {
24         RobotApplication app = new RobotApplication();
25         app.runApplication();
26     }
27 }
```

Fig. 15-1: Estructura de una aplicación del robot

Pos.	Descripción
1	Esta línea contiene el nombre del paquete en el que se encuentra la aplicación.
2	La zona import contiene las clases importadas que son necesarias para la programación de la aplicación. Indicación: Haciendo clic en el símbolo "+" se abrirá la zona y se pueden mostrar las clases importadas.
3	Encabezamiento de la aplicación (contiene el nombre de la clase de la aplicación) (>>> "Encabezamiento" Página 270)

Pos.	Descripción
4	Zona de declaración Aquí se declaran los campos de datos de las clases que son necesarias en la aplicación. Al crear la aplicación, se crea automáticamente una instancia de las siguientes clases: <ul style="list-style-type: none"> ■ Controlador: Unidad de control del robot utilizada ■ LBR: Robot utilizado
5	Método initialize() En este método se asignan valores iniciales a los campos de datos creados en la zona de declaración.
6	Método run() En este método se programa el robot. Al crear la aplicación se inserta automáticamente una instrucción de movimiento que lleva al robot a la posición HOME. (>>> 15.16 "Posición HOME" Página 313)
7	Método main() Este método es necesario para que se pueda ejecutar la aplicación. No se puede modificar.

Encabezamiento

En el caso de una aplicación del robot se trata de la forma especial de una clase Java:

```
public class RobotApplication extends RoboticsAPIApplication
```

Elemento	Descripción
public	La palabra clave public identifica a una clase que está visible de forma pública. Las clases públicas se pueden utilizar en todo el paquete.
class	La palabra clave class identifica a una clase Java. El nombre de la clase se forma a partir del nombre de la aplicación.
extends	La aplicación está subordinada a la clase RoboticsAPIApplication .

15.1.3 Funciones de procesamiento**15.1.3.1 Renombrar variables**

- Descripción** Un nombre de variable se puede modificar con una sola acción en todos los lugares en los que aparezca.
- Procedimiento**
1. Marcar la variable deseada en cualquier lugar.
 2. Hacer clic con el botón derecho del ratón y seleccionar **Refactoring > Renombrar....**
 3. La variable aparece rodeada con un marco azul y se puede editar. Cambiar el nombre y confirmar mediante la tecla de entrada.

15.1.3.2 Autocompletar

- Descripción** En el editor Java se encuentra disponible una funcionalidad de Autocompletar.

A la hora de introducir el código es posible visualizar una lista de completado con entradas que se ajustan a los caracteres ya introducidos. Estas entradas se priorizan de nuevo según la frecuencia de su uso, es decir, la selección se adapta constantemente al comportamiento del usuario.

Si se desea se puede aceptar una entrada de la lista de completado en el código de programa. Esto ahorra, p. ej., el hecho de tener que escribir continuamente las compleja sintaxis de los métodos. Solo tienen que introducirse manualmente los elementos variables de la sintaxis.

Procedimiento

1. Comenzar introduciendo el código.



Al introducir el operador del punto en un campo de datos o Enum, automáticamente se visualiza la lista de completado. La lista contiene las siguientes entradas:

- Métodos disponibles de la clase correspondiente (solo en el caso de campos de datos)
- Constantes disponibles de la clase correspondiente

2. Pulsar Ctrl + barra espaciadora. Se indica la lista de completado con las entradas disponibles.



Si la lista solo consta de una entrada adecuada, esta se aceptará automáticamente en el código de programa después de pulsar Ctrl + barra espaciadora.

3. Marcar la entrada adecuada en la lista y pulsar Intro. La entrada se acepta en el código de programa.

Si una entrada está marcada, automáticamente se visualiza la información de Javadoc acerca de esta entrada.

(>>> 15.1.4 "Visualizar información Javadoc" Página 273)

4. Si se desea, completar la sintaxis.

Navegar y filtrar

Hay distintas posibilidades para navegar por la lista de completado para filtrar las entradas disponibles:

- Navegar con las teclas de flecha del teclado de una entrada a la siguiente (hacia arriba y hacia abajo)
- Desplazarse
- Añadir más caracteres al código introducido. Se filtra la lista y ya solo se indican las entradas que se adecúan a los caracteres.
- Pulsar Ctrl + barra espaciadora. Solo se indican las propuestas de plantilla disponibles.

15.1.3.3 Plantillas – Entrada rápida para instrucciones Java

Descripción

En el editor de Java se encuentran disponibles plantillas para la realización de una entrada rápida, destinadas a las instrucciones Java habituales, p. ej. un bucle for.

Procedimiento

1. Comenzar introduciendo el código.
2. Pulsar Ctrl + barra espaciadora. Se mostrará una lista con las propuestas de plantillas que se adaptan a los caracteres ya introducidos.
3. Aceptar la instrucción con la tecla de entrada. O hacer doble clic en otra instrucción.
4. Completar la sintaxis.

Procedimiento alternativo

Seleccionar las plantillas en la vista **Plantillas**:

1. Seleccionar la secuencia del menú **Ventana > Visualizar vista > Otros....**
Se abre la ventana **Visualizar vista**.
2. En la carpeta **General** marcar la entrada **Plantillas**. Confirmar con **OK**. Se abre la vista de plantillas.
3. Situar el cursor en el editor de Java en la última línea en la que debe introducirse el código predefinido.
4. En **Instrucciones Java** hacer doble clic sobre la plantilla deseada en la vista de plantillas. El código se introduce en el editor.
5. Completar la sintaxis.

15.1.3.4 Crear plantillas específicas de usuario

Descripción	El usuario puede crear sus propias plantillas, p. ej. plantillas para conjuntos de movimientos con determinados parámetros de movimiento que se utilizan con frecuencia para la programación.
Procedimiento	<ol style="list-style-type: none">1. Marcar en la vista de plantillas el contexto en el que debe insertarse el plantilla.2. Hacer clic con el botón derecho sobre el contexto y seleccionar Nuevo... en el menú contextual. O bien: Hacer clic en el símbolo Crear una plantilla nueva. Se abre la ventana Plantillas nuevas.3. En el campo Nombre introducir un nombre para la plantilla.4. En el campo Descripción introducir una descripción (opcional).5. En el campo Ejemplo introducir el código deseado.6. Confirmar las propiedades de la plantilla con OK. La plantilla se crea y se inserta en la vista de plantillas.

15.1.3.5 Extraer métodos

Descripción	Las partes del código del programa pueden sacarse de la aplicación del programa y ponerse a disposición como método propio. Esto es especialmente recomendable en el caso de las tareas recurrentes, ya que de esta forma se aumenta la claridad dentro de la aplicación de robot.
Procedimiento	<ol style="list-style-type: none">1. Marcar el código del programa deseado.2. Hacer clic con el botón derecho del ratón en la zona de los editores.3. En el menú contextual seleccionar Refactoring > Extraer método.... Se abre la ventana Extraer método.4. Introducir un nombre de método único en Nombre del método y seleccionar el Valor de modificación para el acceso. Confirmar con OK. El código del programa marcado se elimina y se genera el método nuevo. El método nuevo se activa en el lugar en el que se ha eliminado el código previamente.
Valor de modificación para el acceso	Esta opción determina las clases que pueden activar los métodos extendidos.

Opción	Descripción
Privado (private)	El método solo se puede activar individualmente por la propia clase correspondiente.
Estándar (default)	Las siguientes clases pueden activar el método: <ul style="list-style-type: none"> ■ La clase correspondiente ■ Las clases internas de la clase correspondiente. ■ Todas las clases del paquete en la que se encuentra la clase correspondiente
Protegido (protected)	Las siguientes clases pueden activar el método: <ul style="list-style-type: none"> ■ La clase correspondiente ■ Las subclases de la clase correspondiente (transmisión directa) ■ Las clases internas de la clase correspondiente. ■ Todas las clases del paquete en la que se encuentra la clase correspondiente
Público (public)	Todas las clases pueden activar el método. Independientemente de la relación de la clase correspondiente e independientemente de la pertenencia del paquete.

15.1.4 Visualizar información Javadoc

Descripción Javadoc es una documentación generada a partir de comentarios especiales de Java. En Javadoc se describen la funcionalidad y el empleo de clases, métodos y bibliotecas.

La información Javadoc puede indicarse durante la programación. La información solo está disponible en lengua inglesa.

Las distintas posibilidades de indicación se describen en el ejemplo de la clase LBR.

Procedimiento

Indicar información Javadoc en la función de autocompletado:

- Al utilizar la función de autocompletado (Ctrl + barra espaciadora), marcar una entrada en la lista de Completar. La información Javadoc correspondiente se indica en una ventana adicional en la zona del editor.

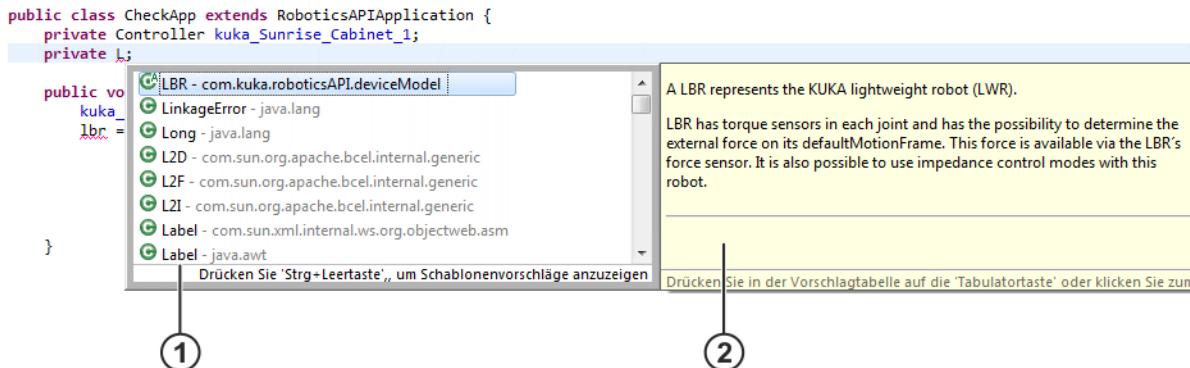


Fig. 15-2: Indicar información Javadoc en la función de autocompletado

- | | |
|-------------------------|--------------------------|
| 1 Lista de Completar | 2 Información Javadoc |
|-------------------------|--------------------------|
- Para fijar la ventana en la zona del editor, pulsar la tecla de tabulación o hacer clic dentro de la ventana.

La fijación de la ventana permite la navegación por la descripción Javadoc, p. ej. desplazando la pantalla hacia arriba y abajo.

Indicar información Javadoc mediante el cursor:

- En el código de programa, mover el cursor sobre el nombre del elemento deseado. La información Javadoc correspondiente se indica automáticamente en una ventana adicional en la zona del editor.

Los siguientes elementos reaccionan al cursor:

- Métodos
- Clases (tipos de datos, campos de datos no autodefinidos)
- Interfaces
- Enums

```
public class CheckApp extends RoboticsAPIApplication {
    private Controller kuka_Sunrise_Cabinet_1;
    private LBR lbr;
}
public v
kuka
lbr
```

com.kuka.roboticsAPI.deviceModel.LBR

A LBR represents the KUKA lightweight robot (LWR).

LBR has torque sensors in each joint and has the possibility to determine the external force on its defaultMotionFrame. This force is available via the LBR's force sensor. It is also possible to use impedance control modes with this robot.

Fig. 15-3: Indicar información Javadoc a través del cursor

- Desde aquí hay disponibles más opciones de indicación:
 - Para poder navegar por la descripción Javadoc, p. ej. desplazando la pantalla hacia arriba y abajo, mover el cursor a la ventana.
La ventana no está fijada. Si el cursor se mueve fuera de la ventana, la ventana se cierra.
 - Para fijar la ventana en la zona del editor, pulsar la tecla F2 o hacer clic dentro de la ventana.
En la ventana fija también se puede navegar por la descripción Javadoc.
 - Para indicar la información Javadoc también en la vista **Javadoc**, marcar el elemento seleccionado con un clic del botón izquierdo del ratón.
Si la ventana no está fijada en la zona del editor, se cerrará.

Navegar

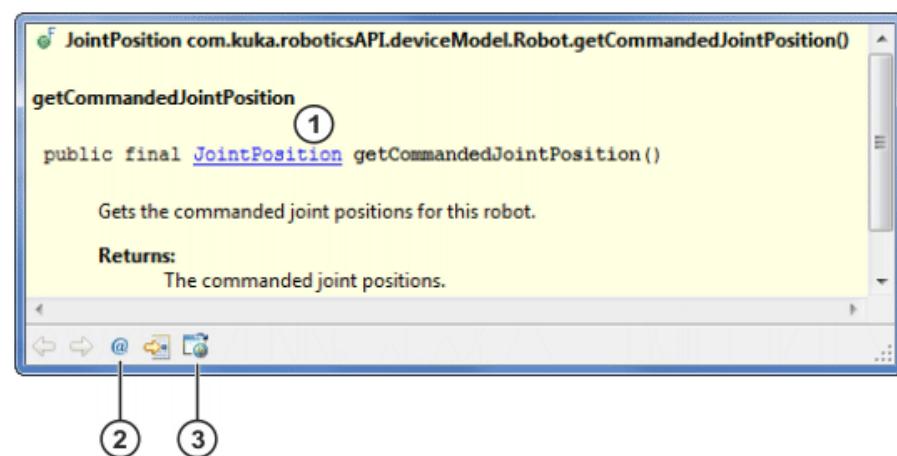


Fig. 15-4: Navegar por la descripción Javadoc

Pos.	Descripción
1	<p>Clase enlazada</p> <p>Haciendo clic con el botón izquierdo del ratón en la clase enlazada, se puede mostrar la información Javadoc completa de esta clase en el navegador de Javadoc.</p> <p>Nota: Si se selecciona el enlace correspondiente en la vista Javadoc, se mostrará automáticamente la información Javadoc completa en la vista.</p>
2	<p>Botón Visualizar en vista Javadoc</p> <p>La ventana en la zona de los editores se cierra y la información Javadoc se muestra en la vista Javadoc.</p>
3	<p>Botón Open Attached Javadoc Browser</p> <p>La ventana en la zona de los editores se cierra y la información Javadoc completa de la clase correspondiente se muestra en el navegador de Javadoc.</p>



Existe otra posibilidad de consultar la información Javadoc completa relativa a un elemento determinado en el navegador Javadoc: Marcar el elemento deseado en el código de programa y pulsar SHIFT + F2.

15.1.4.1 Estructura del navegador Javadoc

La estructura del navegador Javadoc se describe de forma abreviada en el ejemplo de la clase LBR.

Overview Package Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: NESTED | FIELD | CONSTR | METHOD

DETAIL: FIELD | CONSTR | METHOD

com.kuka.roboticsAPI.deviceModel

Class LBR

java.lang.Object

- └ com.kuka.common.TaggableObject
- └ com.kuka.roboticsAPI.geometricModel.SceneGraphNode
- └ com.kuka.roboticsAPI.geometricModel.SpatialObject
- └ com.kuka.roboticsAPI.geometricModel.PhysicalObject
- └ com.kuka.roboticsAPI.deviceModel.Device
- └ com.kuka.roboticsAPI.deviceModel.Robot
- └ com.kuka.roboticsAPI.deviceModel.LBR

All Implemented Interfaces:

com.kuka.common.ITagable, IOperationModeProvider

Direct Known Subclasses:

SunriseLBR

public abstract class LBR
extends Robot

A LBR represents the KUKA lightweight robot (LWR).

LBR has torque sensors in each joint and has the possibility to determine the external force on its defaultMotionFrame. This force is available via the LBR's force sensor. It is also possible to use impedance control modes with this robot.

Field Summary

protected	<code>_gmsSensorLimits</code> double[] torque sensor limits.
-----------	---

Fields inherited from class com.kuka.roboticsAPI.deviceModel.Device

hardwareVersion

Constructor Summary

`LBR(Controller controller, java.lang.String name)`
Creates a new LBR instance with the given controller.

Method Summary

boolean	<code>checkTorqueSensor(JointEnum joint)</code> Checks if the torque sensor of the given axis works properly.
---------	--

Field Detail

`_gmsSensorLimits`
protected double[] _gmsSensorLimits
torque sensor limits.

Constructor Detail

LBR

`public LBR(Controller controller,
 java.lang.String name)`

Creates a new LBR instance with the given controller.

Parameters:
controller - The controller to which this device belongs.
name - The name of the device.

Method Detail

initializeJointCount

`protected int initializeJointCount()`

Fig. 15-5: Estructura del navegador Javadoc

Pos.	Descripción
1	Navegación
2	<p>Jerarquía de clases (>>> Fig. 15-6)</p> <p>Aquí se muestran las relaciones de transmisión de la clase.</p>
3	<p>Descripción de la clase</p> <p>Aquí se describe la tarea de la clase y su funcionalidad. Es costumbre hacer hincapié en las particularidades del uso de la clase. También puede haber ejemplos breves relativos al uso de la clase.</p> <p>Al final de la descripción habitualmente se indica a partir de qué versión de biblioteca está disponible la clase. Además puede haber un listado con referencias a otras clases o métodos que también sean de interés.</p>
4	<p>Vistas generales</p> <ul style="list-style-type: none"> ■ Field Summary Vista general de los campos de datos correspondientes a la clase Aquí se listan los campos de datos que han sido transferidos de una clase padre. ■ Constructor Summary Vista general de los constructores correspondientes a la clase ■ Method Summary Vista general de los métodos correspondientes a la clase Aquí se listan los métodos que han sido transferidos de una clase padre. <p>Si se ha indicado durante la creación de Javadoc, las vistas generales contienen descripciones breves relativas a los campos de datos, los constructores y los métodos de la clase. Los campos de datos y métodos que han sido transferidos por unidades padre solo aparecen listados.</p> <p>Hay descripciones más detalladas relativas a los campos de datos, los constructores y los métodos en la zona Details. Mediante un clic sobre el nombre en cuestión se puede acceder directamente a la descripción detallada.</p>
5	<p>Detalles</p> <ul style="list-style-type: none"> ■ Field Detail Descripción detallada de los campos de datos correspondientes a la clase ■ Constructor Detail Descripción detallada de los constructores correspondientes a la clase ■ Method Detail Descripción detallada de los métodos correspondientes a la clase <p>La descripción detallada puede contener, por ejemplo, en el caso de los métodos, un listado y descripción de los parámetros transferidos, así como el valor de retorno. Si existen, aquí también se nombran las excepciones (Exceptions) que pueden surgir al ejecutar un método o un constructor.</p>

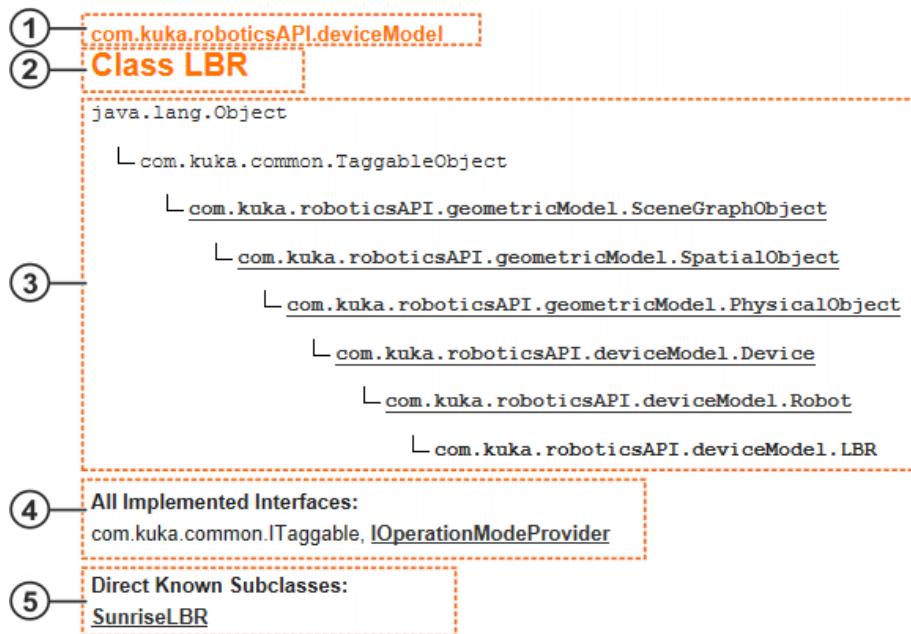


Fig. 15-6: Jerarquía de clases

Pos.	Descripción
1	Nombre del paquete al que pertenece la clase
2	Nombre de la clase
3	Jerarquía de clases (origen de la clase)
4	Enumeración de las interfaces que implementa la clase
5	Enumeración de las subclases que se derivan de la clase

15.2 Caracteres y tipos de letra

En las descripciones de la sintaxis se emplean los siguientes caracteres y tipos de letra:

Elemento de la sintaxis	Visualización
Código Java	<ul style="list-style-type: none"> ■ Tipo de letra Courier ■ Mayúsculas/minúsculas Ejemplos: private; new; linRel; Tool
Elementos que deben sustituirse por datos específicos del programa	<ul style="list-style-type: none"> ■ Cursiva ■ Mayúsculas/minúsculas Ejemplos: Punto de destino; Nombre; mode
Elementos opcionales	<ul style="list-style-type: none"> ■ Entre signos de mayor y menor Ejemplo: <.setVelocity (valor) >
Elementos que se excluyen mutuamente	<ul style="list-style-type: none"> ■ Separados por el carácter " " Ejemplo: ++ --

15.3 Tipos de datos

Resumen

En Java existen 2 variedades de tipos de datos:

- Tipos de datos primitivos
- Tipos de datos complejos

Los tipos de datos complejos se definen en Java mediante clases.

Resumen de tipos de datos importantes:

Tipo de datos	Descripción
int	Número entero ■ $-2^{31}-1 \dots +2^{31}-1$ Ejemplos: -1; 32; 8000
double	Número con coma flotante con doble precisión ■ $-1.7E+308 \dots +1.7E+308$ Ejemplos: 1.25; -98.76; 123.456
boolean	Estado lógico ■ true ■ false
char	Character (1 carácter) ■ Caracteres ASCII Ejemplos: 'A'; '1'; 'q'
String	Cadena de caracteres ■ Caracteres ASCII Ejemplos: "KUKA"; "tool"



En el editor de Java, los nombres de los tipos de datos primitivos se muestran en color de escritura violeta.

15.4 Variables

Descripción Antes de utilizar una variable en el programa debe declararse, es decir, deben determinarse el tipo de dato y el denominador. Una variable puede declararse p. ej. en el método run() de una aplicación.

Sintaxis *Tipo de datos Nombre;*

Aclaración de la sintaxis

Elemento	Descripción
<i>Tipo de datos</i>	Tipo de datos de las variables
<i>Nombre</i>	Nombre de las variables

Ejemplos

```
int counter;
double value;
boolean isObjectPlaced;
```

15.5 Comunicación de red a través de UDP y TCP/IP

En la unidad de control del robot se han autorizado determinados puertos para la comunicación con dispositivos externos a través de UDP o TCP/IP.

Se pueden emplear los siguientes números de puertos (Socket de cliente o de servidor) en una aplicación robot:

- 30 000 a 30 010

15.6 Información sobre la versión de RoboticsAPI

La RoboticsAPI es una interfaz de programación para todas las instrucciones específicas del robot. Posee un número de versión de 4 cifras. Dicho número informa sobre las modificaciones de la interfaz para una comparación de versiones.

15.6.1 Visualizar la versión de RoboticsAPI

El número de versión de RoboticsAPI se puede consultar mediante el archivo StationSetup.cat (configuración de la estación).

Procedimiento

1. Abrir el proyecto en el **Explorador de paquetes**.
2. Abrir la configuración de la estación.
3. Seleccionar la pestaña **Software**.
4. Poner el símbolo de confirmación en **Mostrar bibliotecas individuales**.

Descripción

- El número de versión de RoboticsAPI se muestra en la línea **BasicRobotics-JavaLib**:
- Columna **Versión actualmente instalada**: Versión que se instaló por último en el control
 - Columna **Selección de versión**: Versión que se utiliza actualmente en el proyecto Sunrise

15.6.2 Estructura del número de versión de RoboticsAPI

El número de versión de RoboticsAPI está definido con el formato XX.YY.ZZ.número de creación.

- XX: Versión instalada
- YY: Modificaciones incompatibles entre las versiones
En las aplicaciones que hayan sido creadas utilizando la versión anterior de API, pueden producirse errores al cambiar a una versión más actualizada.
- ZZ: Modificaciones compatibles entre las versiones
En las aplicaciones que hayan sido creadas utilizando la versión anterior de API, también se pueden utilizar con la versión actual de API.
- Número de creación: Se asigna automáticamente.

15.7 Programación de movimiento: PTP, LIN, CIRC

15.7.1 Estructura de una instrucción de movimiento (move/moveAsync)

Descripción

En Sunrise pueden aplicarse las instrucciones de movimiento a todos los objetos móviles de una estación. Un objeto móvil puede ser, p. ej., el robot, pero también una herramienta que esté conectada a la brida del robot o una pieza que esté siendo sujetada por una herramienta (p. ej. una garra).

Las instrucciones de movimientos se pueden ejecutar de forma sincronizada y asíncrona. Para ello se encuentran disponibles los métodos move(...) y moveAsync(...):

- move(...) para la ejecución sincronizada

Sincronizado significa que las instrucciones de movimientos se envían a la unidad de control en tiempo real y se procesan por pasos. La ejecución posterior del programa está cancelada hasta que se haya ejecutado el movimiento. Seguidamente se envía la siguiente instrucción.

- moveAsync(...) para la ejecución asíncrona

Asíncrono significa que directamente después de enviar la instrucción de movimiento se ejecuta la siguiente línea del programa. La ejecución asíncrona de movimientos es necesaria, p. ej. para aproximar movimientos.

El modo en que se programan los distintos tipos de movimientos se representa a modo de ejemplo en el objeto Robot.

La programación de movimientos para herramientas y piezas se describe aquí: ([>>> 15.11.4 "Desplazar herramientas y piezas" Página 297](#))



Para la programación se pueden indicar valores con una precisión mayor de la que puede alcanzar el robot. Por ejemplo, se pueden realizar indicaciones de posición en un rango de nanómetros, pero que no se pueden alcanzar con esta precisión.

Sintaxis

Ejecutar el movimiento de forma sincronizada:

Objeto.move (movimiento) ;

Ejecutar el movimiento de forma asíncrona:

Objeto.moveAsync (movimiento) ;

Aclaración de la sintaxis

Elemento	Descripción
<i>Objeto</i>	Objeto de la estación que es desplazado Aquí se indica el nombre de variable del objeto que ha sido declarado e inicializado en la aplicación.
<i>Movimiento</i>	Movimiento que se ejecuta El movimiento a ejecutar está definido por los siguientes elementos: <ul style="list-style-type: none"> ■ Tipo o bloque de movimiento: ptp, lin, circ, spl o spline, splineJP, batch ■ Posición de destino ■ Otros parámetro de movimiento opcionales

15.7.2 PTP

Descripción

Ejecuta un movimiento punto a punto al punto de destino. Las coordenadas del punto de destino son absolutas.

Para programar el punto de destino, existen las siguientes posibilidades:

- Insertar un Frame desde los datos de aplicación en la instrucción de movimiento.
- Crear el Frame en el programa y utilizar la instrucción de movimiento.



La información de redundancia sobre el punto de destino (Status, Turn, ángulo de redundancia) debe indicarse correctamente. De lo contrario, el punto de destino no se puede alcanzar de forma correcta.

- Indicar el ángulo del eje de los ejes A1 ... A7. Se deben indicar siempre todos los valores de los ejes.

Sintaxis

Movimiento PTP con indicación de Frame:

`ptp (getApplicationData () .getFrame ("punto de destino") <. Parámetro de movimiento>)`

Movimiento PTP con indicación del ángulo del eje:

`ptp (A1, A2, ... A7) <.parámetro de movimiento>)`

Aclaración de la sintaxis

Elemento	Descripción
<i>Punto de destino</i>	Ruta del Frame en el árbol de Frame o nombre de variable del Frame (si está creado en el programa)
<i>A1 ... A7</i>	Ángulo del eje de los ejes A1 ... A7 (tipo: double; unidad: rad)
<i>Parámetros de movimiento</i>	Otros parámetros de movimiento, p. ej. velocidad o aceleración

Ejemplos

Movimiento PTP para el Frame "StartPos"

```
robot.move(ptp(getApplicationContext().getFrame("/StartPos")));
```

Movimiento PTP hacia la posición vertical:

```
robot.move(ptp(0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0));
```

Movimiento PTP para el Frame "StartPos" con indicación de la velocidad relativa:

```
robot.move(ptp(getApplicationContext().getFrame("/StartPos"))
.setJointVelocityRel(0.25));
```

15.7.3 LIN

Descripción

Ejecuta un movimiento lineal hasta el punto de destino. Las coordenadas del punto de destino son absolutas desde el punto de vista cartesiano.

Para programar el punto de destino, existen las siguientes posibilidades:

- Insertar un Frame desde los datos de aplicación en la instrucción de movimiento.
- Crear el Frame en el programa y utilizar la instrucción de movimiento.

Sintaxis

`lin(getApplicationContext().getFrame("punto de destino") <.Parámetro de movimiento>)`

Aclaración de la sintaxis

Elemento	Descripción
<i>Punto de destino</i>	Ruta del Frame en el árbol de Frame o nombre de variable del Frame (si está creado en el programa) La información de redundancia al punto de destino (Status y Turn) se ignora en el caso de movimientos LIN (al igual que con los movimientos CIRC). Tan solo se tiene en cuenta el ángulo de redundancia.
<i>Parámetros de movimiento</i>	Otros parámetros de movimiento, p. ej. velocidad o aceleración

Ejemplos

Movimiento LIN al Frame "/Table/P1":

```
robot.move(lin(getApplicationContext().getFrame("/Table/P1")));
```

Movimiento LIN con indicación de la velocidad cartesiana:

```
robot.move(lin(getApplicationContext().getFrame("/Table/P1"))
.setCartVelocity(150.0));
```

15.7.4 CIRC

Descripción	Ejecuta un movimiento circular. Para que el control pueda calcular el movimiento circular, debe indicarse un punto auxiliar y un punto de destino. Las coordenadas del punto auxiliar y del punto de destino son cartesianas y absolutas. Para programar el punto auxiliar y de destino, existen las siguientes posibilidades:								
Sintaxis	<pre>circ(getApplicationData().getFrame("Punto auxiliar"), getApplicationData().getFrame("Punto de destino") <.Parámetro de movimiento>)</pre>								
Aclaración de la sintaxis	<table border="1"> <thead> <tr> <th>Elemento</th><th>Descripción</th></tr> </thead> <tbody> <tr> <td><i>Punto auxiliar</i></td><td>Ruta del Frame en el árbol de Frame o nombre de variable del Frame (si está creado en el programa) La información de redundancia sobre el punto auxiliar (Status, Turn, ángulo de redundancia) es ignorada.</td></tr> <tr> <td><i>Punto de destino</i></td><td>Ruta del Frame en el árbol de Frame o nombre de variable del Frame (si está creado en el programa) La información de redundancia sobre el punto de destino (Status y Turn) se ignora en el caso de movimientos CIRC (al igual que con los movimientos LIN). Solo se tiene en cuenta el ángulo de redundancia.</td></tr> <tr> <td><i>Parámetros de movimiento</i></td><td>Otros parámetros de movimiento, p. ej. velocidad o aceleración</td></tr> </tbody> </table>	Elemento	Descripción	<i>Punto auxiliar</i>	Ruta del Frame en el árbol de Frame o nombre de variable del Frame (si está creado en el programa) La información de redundancia sobre el punto auxiliar (Status, Turn, ángulo de redundancia) es ignorada.	<i>Punto de destino</i>	Ruta del Frame en el árbol de Frame o nombre de variable del Frame (si está creado en el programa) La información de redundancia sobre el punto de destino (Status y Turn) se ignora en el caso de movimientos CIRC (al igual que con los movimientos LIN). Solo se tiene en cuenta el ángulo de redundancia.	<i>Parámetros de movimiento</i>	Otros parámetros de movimiento, p. ej. velocidad o aceleración
Elemento	Descripción								
<i>Punto auxiliar</i>	Ruta del Frame en el árbol de Frame o nombre de variable del Frame (si está creado en el programa) La información de redundancia sobre el punto auxiliar (Status, Turn, ángulo de redundancia) es ignorada.								
<i>Punto de destino</i>	Ruta del Frame en el árbol de Frame o nombre de variable del Frame (si está creado en el programa) La información de redundancia sobre el punto de destino (Status y Turn) se ignora en el caso de movimientos CIRC (al igual que con los movimientos LIN). Solo se tiene en cuenta el ángulo de redundancia.								
<i>Parámetros de movimiento</i>	Otros parámetros de movimiento, p. ej. velocidad o aceleración								
Ejemplos	<p>Movimiento CIRC para el Frame de destino "/Table/P4" mediante el Frame auxiliar "/Table/P3":</p> <pre>robot.move(circ(getApplicationContext().getFrame("/Table/P3"), getApplicationContext().getFrame("/Table/P4")));</pre> <p>Movimiento CIRC con indicación de la aceleración absoluta:</p> <pre>robot.move(circ(getApplicationContext().getFrame("/Table/P3"), getApplicationContext().getFrame("/Table/P4")).setCartAcceleration(25));</pre>								

15.7.5 LIN REL

Descripción	Ejecuta un movimiento lineal hasta el punto de destino. Las coordenadas del punto de destino están ajustadas en relación con la posición de destino del movimiento previo, a menos que dicho movimiento previo se cancela por una condición de cancelación. En este caso, las coordenadas del punto de destino están ajustadas en relación con la posición de cancelación. En un movimiento relativo, el punto de destino se desplaza por defecto en el sistema de coordenadas del Frame desplazado. Opcionalmente se puede indicar otro sistema de coordenadas de referencia, en el que se ejecuta el movimiento relativo. En este caso, las coordenadas del punto de destino hacen referencia a este sistema de coordenadas de referencia. Éste puede ser, por ejemplo, un Frame creado en los datos de la aplicación o una base medida. Para programar el punto de destino, existen las siguientes posibilidades:
--------------------	--

- Indicar individualmente los valores de offset cartesianos.
- Utilizar una transformación del Frame del tipo Transformación. La transformación del Frame tiene la ventaja de que la rotación también se puede indicar en grados.

Sintaxis

Movimiento LinRel con valores de offset:

```
linRel(x, y, z<, a, b, c>
<, Sistema de referencia>)
```

Movimiento LinRel con transformación del Frame:

```
linRel(Transformation.ofDeg|ofRad(x, y, z, a, b, c)
<, Sistema de referencia>)
```

Aclaración de la sintaxis

Elemento	Descripción
x, y, z	Desplazamiento en dirección X, Y, Z (tipo: double; unidad: mm)
a, b, c	Rotación alrededor del eje Z, Y, X (tipo: double) La unidad depende del método utilizado: <ul style="list-style-type: none"> ■ Valores de offset y Transformation.ofRad: Rueda ■ Transformation.ofDeg: Grados
Sistema de referencia	Tipo: AbstractFrame Sistema de coordenadas de referencia en el que se ejecuta el movimiento

Ejemplos

El Frame desplazado es el TCP de una garra. En el sistema de coordenadas de la herramienta, este TCP se desplaza de la posición actual 100 mm en dirección X y 200 mm en dirección Z negativa. La orientación del TCP no se modifica.

```
gripper.getFrame("/TCP2").move(linRel(100, 0, -200));
```

El robot se desplaza respectivamente 10 mm en el sistema de coordenadas del Frame P1, partiendo desde la posición actual. Adicionalmente, el robot gira 30° alrededor del eje Z y Y del sistema de coordenadas del Frame P1

```
robot.move(linRel(Transformation.ofDeg(10, 10, 10, 30, 30, 0),
getApplicationData().getFrame("/P1")));
```

15.7.6 MotionBatch**Descripción**

En un MotionBatch se pueden agrupar varios movimientos individuales y, de este modo, transmitirse al mismo tiempo a la unidad de control del robot. De este modo se pueden aproximar movimientos dentro del MotionBatch.

Los parámetros de movimiento, p. ej. velocidad, aceleración, control de la orientación, etc., se pueden programar para el Batch completo o por movimientos.



Para el Batch solo se pueden indicar parámetros de movimiento específicos del eje (setJoint...Rel(...)). Los parámetros de movimiento cartesianos (setCart...(...)) se deben indicar el conjunto individual.

Las dos formas pueden aparecer mezcladas, p. ej. para asignar otro valor de parámetro a un movimiento individual como Batch.



El parámetro de conjunto individual sobrescribe el parámetro Batch. Esto también se aplica si para el Batch se indica un valor de parámetro más bajo que para el conjunto individual.

Sintaxis

```
Objeto.move (batch  

Movimiento,  

Movimiento,  

...  

Movimiento,  

Movimiento  

) < .Parámetros de movimiento> ;
```

Aclaración de la sintaxis

Elemento	Descripción
<i>Objeto</i>	Objeto de la estación que es desplazado
<i>Movimiento</i>	Movimiento, con o sin parámetros de movimiento <ul style="list-style-type: none"> ■ ptp, lin, circ o spline
<i>Parámetros de movimiento</i>	Los parámetros de movimiento que se programan al final del Batch, son válidos para el Batch completo. Solo se pueden programar parámetros de movimiento específicos del eje.

15.8 Programación de movimiento: Spline

15.8.1 Sugerencias de programación para movimientos Spline

- Un bloque Spline solo debe contener un proceso (por ej. 1 cordón de pegamento). Si se incluyeran varios procesos en un bloque Spline, el programa resultaría muy complejo y dificultaría modificaciones.
- Si a través de la pieza se especifican rectas y segmentos circulares, se deben utilizar segmentos LIN y CIRC. (Excepción: para rectas muy cortas utilizar segmentos SPL). De lo contrario, utilizar segmentos SPL; especialmente para distancias de punto cortas.
- Procedimiento para la definición de la trayectoria:
 - a. En primer lugar, programar por aprendizaje o calcular pocos puntos característicos. Ejemplo: Puntos en los cuales cambia la curvatura.
 - b. Testar la trayectoria. En los lugares en los cuales la precisión ya no es suficiente, insertar otros puntos SPL.
- Evitar segmentos LIN y/o CIRC consecutivos porque, caso contrario, la velocidad se reducirá a menudo a 0. Para evitarlo:
 - Programar entre los segmentos LIN y CIRC, segmentos SPL. La longitud de los segmentos SPL debe ser, como mínimo, > 0,5 mm. En función del curso concreto de la trayectoria, puede que se necesiten segmentos SPL mucho mayores.
 - Sustituir un segmento LIN por varios segmentos SPL que se encuentren sobre una recta. De este modo, la trayectoria también se convierte en una recta.
- Evitar puntos consecutivos con coordenadas cartesianas idénticas porque la velocidad se puede reducir a 0.
- Si el robot se desplaza a puntos que se encuentran sobre una superficie de trabajo, al desplazar al primer punto se puede producir una colisión con la superficie de trabajo.

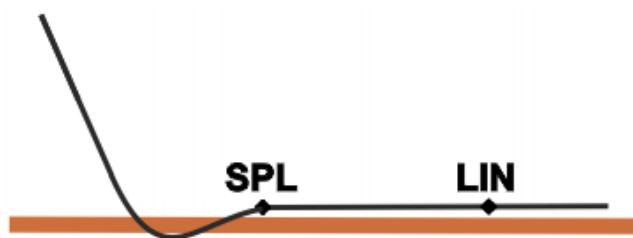


Fig. 15-7: Colisión con superficie de trabajo

Las colisiones se pueden evitar mediante la inserción de un segmento antes de la superficie de trabajo. Tener en cuenta las recomendaciones para la transición LIN-SPL-LIN.

(>>> 14.6.3 "Transición LIN-SPL-LIN" Página 254)

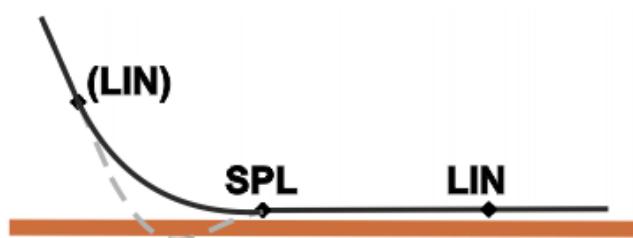


Fig. 15-8: Evitar colisión con superficie de trabajo

- Dentro de lo posible, no utilizar los segmentos SPL si se realiza el desplazamiento cerca del límite del campo de trabajo. Puede ocurrir que con SPL se supere el límite del campo de trabajo, a pesar de que el robot puede alcanzar el Frame de destino en otro tipo de movimiento o en el desplazamiento manual.

15.8.2 Crear un bloque Spline CP

Descripción

Con un bloque Spline CP se pueden agrupar varios segmentos SPL, LIN y/o CIRC en un mismo movimiento total. El número máximo de segmentos Spline en un bloque Spline está limitado actualmente a 20. Para generar trayectorias con más de 20 segmentos, se recomienda programar varios bloques Spline y aproximar estos bloques Spline.

Un bloque Spline no debe contener otras instrucciones, p. ej. asignaciones de variables o instrucciones lógicas.

Los parámetros de movimiento, p. ej. velocidad, aceleración, control de la orientación, etc., se pueden programar para el bloque Spline completo o por segmentos. Las dos formas pueden aparecer mezcladas, p. ej. para asignar otro valor de parámetro a un segmento individual como bloque.



El parámetro de conjunto individual sobrescribe el parámetro de bloque. Esto también se aplica si para el bloque se indica un valor de parámetro más bajo que para el conjunto individual.

Sintaxis

```
Spline Nombre = new Spline(
    Segment,
    Segment,
    ...
    Segment,
    Segment
) <. Parámetros de movimiento>;
```

Aclaración de la sintaxis

Elemento	Descripción
Nombre	Nombre del bloque Spline
Segmento	Movimiento, con o sin parámetros de movimiento ■ spl, lin o circ
Parámetros de movimiento	Los parámetros de movimiento que se programan al final del bloque Spline, son válidos para el bloque Spline completo.

Ejemplo

```
Spline mySpline = new Spline(
    splgetApplicationData().getFrame("/P1"),
    circ(getApplicationData().getFrame("/P2"),
        getApplicationData().getFrame("/P3")),
    spl(getApplicationData().getFrame("/P4")).setCartVelocity(150),
    lin(getApplicationData().getFrame("/P5"))
).setCartVelocity(250);
```

15.8.3 Crear un bloque Spline JP

Descripción

Con un bloque Spline JP se pueden agrupar varios segmentos PTP en un mismo movimiento total. El número máximo de segmentos Spline en un bloque Spline está limitado actualmente a 20. Para generar trayectorias con más de 20 segmentos, se recomienda programar varios bloques Spline y aproximar estos bloques Spline.

Un bloque Spline no debe contener otras instrucciones, p. ej. asignaciones de variables o instrucciones lógicas.

Los parámetros de movimiento, p. ej. velocidad, aceleración, etc., se pueden programar para el bloque Spline completo o por segmentos. Las dos formas pueden aparecer mezcladas, p. ej. para asignar otro valor de parámetro a un segmento individual como bloque.



El parámetro de conjunto individual sobrescribe el parámetro de bloque. Esto también se aplica si para el bloque se indica un valor de parámetro más bajo que para el conjunto individual.

Sintaxis

```
SplineJP Nombre = new SplineJP(
    Segment,
    Segment,
    ...
    Segment,
    Segment
) <. Parámetros de movimiento>;
```

Aclaración de la sintaxis

Elemento	Descripción
Nombre	Nombre del bloque Spline
Segmento	Movimiento PTP, con o sin parámetros de movimiento
Parámetros de movimiento	Los parámetros de movimiento que se programan al final del bloque Spline, son válidos para el bloque Spline completo.

Ejemplo

```
SplineJP mySpline = new SplineJP(
    ptp(getApplicationData().getFrame("/P1")),
```

```
ptpgetApplicationData().getFrame("/P2"))
).setJointVelocityRel(0.75);
```

15.8.4 Utilizar Spline en una instrucción de movimiento

Descripción El movimiento Spline programado en un bloque Spline se utiliza como tipo de movimiento en una instrucción de movimiento.

Sintaxis *Objeto.move (Nombre Spline) ;*

Aclaración de la sintaxis

Elemento	Descripción
<i>Objeto</i>	Objeto de la estación que es desplazado
<i>Nombre Spline</i>	Nombre del bloque Spline

Ejemplo

```
robot.move(mySpline);
```

15.9 Programar el guiado manual

Descripción El robot se puede guiar manualmente con la ayuda de un equipo manual de guiado. El modo de guiado manual se puede conectar con la instrucción de movimiento handGuiding() en la aplicación. El guiado manual comienza en la posición real que se haya alcanzado por última vez antes de la conexión.

Si se utiliza el modo de guiado manual en la aplicación, deben estar configurados al menos 2 estados ESM:

- Estado ESM para el movimiento de guiado manual

El estado ESM contiene la AMF *Validación equipo manual de guiado* que controla el pulsador de validación en el equipo manual de guiado.

(>>> 13.8.7 "Control de dispositivos de validación en equipos manuales de guiado" Página 205)

Se recomienda configurar para la AMF *Validación equipo manual de guiado* una parada de seguridad 1 (sobre la trayectoria) como reacción de parada. Después de una parada sobre la trayectoria, la aplicación puede continuar directamente pulsando la tecla de arranque.

Si para la AMF *Validación equipo manual de guiado* se configura una parada de reacción que no sea sobre la trayectoria, el robot deberá retornar a la posición previa después de un guiado manual, antes de que continúe la aplicación.



En un análisis de riesgos se deberá evaluar si está permitido configurar una parada de reacción sobre la trayectoria para el estado ESM que controla al pulsador de validación en el equipo manual de guiado.

- Estado ESM para todos los movimientos excepto para el movimiento de guiado manual

El estado ESM no contiene ninguna AMF *Validación equipo manual de guiado*. La validación en el equipo manual de guiado no se evalúa.

En la aplicación generalmente son necesarios movimientos antes y después del guiado manual. Se recomienda controlar estos movimientos respectivamente con un estado ESM, que no evalúe la validación en el equipo manual de guiado y que cambie al estado ESM para el movimiento de guiado manual directamente antes de la conmutación al modo de guiado manual. Si esto se lleva a cabo en la aplicación, el comportamiento será el siguiente:

- Si se concede la validación para el guiado manual antes de que se conecte el modo de guiado manual en la aplicación, dicho modo de guiado manual estará activo inmediatamente después de la conexión. Es decir, la aplicación no se pausa con la conexión, de forma que el paso entre el modo de la aplicación y el modo de guiado manual sea fluido.
- Si la validación para el guiado manual se concede cuando el modo de guiado manual ya está conectado en la aplicación, se deberá pulsar la tecla de arranque para poder desplazar el robot manualmente. La pausa de la aplicación permite al operario mover la mano hacia el equipo manual de guiado.
- El modo de guiado manual finaliza cuando se retira la validación para el guiado manual, p. ej. al soltar el pulsador de validación. La aplicación se pausa y solo puede continuar tras pulsar la tecla de arranque. La pausa de la aplicación permite al operario retirar la mano del equipo manual de guiado.

 **ATENCIÓN**

Si durante la comutación al modo de guiado manual, la aplicación está en un estado ESM que no contiene ninguna AMF **Validación equipo manual de guiado**, el robot se podrá desplazar igualmente de forma manual en caso necesario. El pulsador de validación en el equipo manual de guiado se presiona y está disponible una AMF **Validación equipo manual de guiado** en otro estado ESM cualquiera o de la tabla PSM.

La constelación se debe evitar obligatoriamente, ya que en un caso como este la aplicación no se pausa tras finalizar el guiado manual, cuando se suelta el pulsador de validación en el equipo manual de guiado. En su lugar se continúa la aplicación sin más acciones de manejo. Si después del guiado manual continúan otros movimientos, se ejecutarán directamente de forma que el operario aún tiene la mano en el equipo manual de guiado y se encuentra en la zona de movimiento del robot.



La comutación entre los estados ESM se realiza mediante señales no destinadas a la seguridad. Por ello, durante la determinación de un estado ESM debe tenerse en cuenta que debe garantizar siempre suficiente seguridad, independientemente del lugar y del momento de su activación. ([>>> 13.2 "Concepto de seguridad" Página 180](#))

Preparación

La instrucción de movimiento handGuiding() pertenece a la clase MMCMotions. La clase se debe insertar manualmente en la zona de importación de la aplicación del robot. Se debe programar la siguiente línea:

```
import static com.kuka.roboticsAPI.motionModel.MMCMotions.*;
```

Sintaxis

```
Objeto.move (handGuiding());
```

Aclaración de la sintaxis

Elemento	Descripción
Objeto	Objeto de la estación que es desplazado

Ejemplo

```
1 robot.setESMState("1");
2 robot.move(ptpgetApplicationData().getFrame("/P1"));
3 robot.setESMState("2");
4 robot.move(handGuiding());
5 robot.setESMState("1");
6 robot.move(ptpgetApplicationData().getFrame("/P2"));
```

Línea	Descripción
1	El estado ESM 1 se activa para el robot. El estado ESM 1 controla en este ejemplo la protección del operario.
2	El desplazamiento hasta el Frame "/P1" se realiza con un movimiento PTP.
3	El estado ESM 2 se activa para el robot. El estado ESM 2 controla al pulsador de validación en el equipo manual de guiado. Si aún no se ha concedido ninguna validación mediante el interruptor, se activará la reacción de parada configurada y la aplicación se pausará.
4	Se conecta el modo de guiado manual. Cuando se presiona el pulsador de validación en el equipo manual de guiado y se mantiene presionado en la posición central, el robot se puede guiar manualmente. Si se retira la validación para el guiado manual, p. ej. al soltar el pulsador de validación, habrá finalizado el modo de guiado manual. La reacción de parada configurada para el estado ESM 2 se activa y la ejecución del movimiento se pausa.
5	El estado ESM 1 se activa para el robot. El estado ESM 1 controla en este ejemplo la protección del operario. La ejecución del movimiento aún sigue pausada. Para continuar la aplicación se deberá pulsar la tecla de arranque.
6	El desplazamiento hasta el Frame "/P2" se realiza con un movimiento PTP.

15.10 Parámetros de movimiento

Los parámetros de movimiento necesarios pueden añadirse a la instrucción de movimiento en cualquier orden. Para ello se utilizan el operador del punto y los métodos set.

Resumen

Método	Descripción
setCartVelocity(...)	<p>Velocidad cartesiana absoluta (tipo: double; unidad: mm/s)</p> <p>■ > 0.0</p> <p>El valor indica la máxima velocidad cartesiana con la que puede desplazarse el robot para el movimiento. Debido a otras limitaciones de planificación de la trayectoria, puede ocurrir que dicha velocidad máxima no se alcance y la velocidad real sea más baja.</p> <p>Si no se indica ninguna velocidad, el movimiento se ejecutará con la velocidad más rápida posible.</p> <p>Indicación: Este parámetro no se puede ajustar para los movimientos PTP.</p>
setJointVelocityRel(...)	<p>Velocidad relativa específica del eje (tipo: double; unidad: %)</p> <p>■ 0.0 ... 1.0</p> <p>Se refiere al valor máximo de la velocidad del eje en los datos de máquina.</p> <p>(>>> 15.10.1 "Programar parámetros de movimiento específicos del eje" Página 292)</p>

Método	Descripción
setCartAcceleration(...)	<p>Aceleración cartesiana absoluta (tipo: double; unidad: mm/s²)</p> <ul style="list-style-type: none"> ■ > 0.0 <p>Si no se indica ninguna aceleración, el movimiento se ejecutará con la aceleración más rápida posible.</p> <p>Indicación: Este parámetro no se puede ajustar para los movimientos PTP.</p>
setJointAccelerationRel(...)	<p>Aceleración relativa específica del eje (tipo: double; unidad: %)</p> <ul style="list-style-type: none"> ■ 0.0 ... 1.0 <p>Se refiere al valor máximo de la aceleración del eje en los datos de máquina.</p> <p>(>>> 15.10.1 "Programar parámetros de movimiento específicos del eje" Página 292)</p>
setCartJerk(...)	<p>Tirón cartesiano absoluto (tipo: double; unidad: mm/s³)</p> <ul style="list-style-type: none"> ■ > 0.0 <p>Si no se indica ningún tirón, el movimiento se ejecutará con la modificación de la aceleración más rápida posible.</p> <p>Indicación: Este parámetro no se puede ajustar para los movimientos PTP.</p>
setJointJerkRel(...)	<p>Tirón relativo específico del eje (tipo: double; unidad: %)</p> <ul style="list-style-type: none"> ■ 0.0 ... 1.0 <p>Se refiere al valor máximo de la modificación de la aceleración específica del eje en los datos de máquina.</p> <p>(>>> 15.10.1 "Programar parámetros de movimiento específicos del eje" Página 292)</p>
setBlendingRel(...)	<p>Distancia de aproximación relativa (tipo: double)</p> <ul style="list-style-type: none"> ■ 0.0 ... 1.0 <p>La distancia de aproximación relativa es la distancia anterior al punto de destino del movimiento, en el que comienza la aproximación como muy pronto. Si se ajusta "0.0", el parámetro de aproximación no tendrá efecto.</p> <p>La distancia máxima (= 1.0) se corresponde siempre con la longitud del movimiento individual o para Splines, con la longitud del último segmento. Para los movimientos que no se ordenan dentro de un Spline, solo está disponible el rango entre 0 % y 50 % para la aproximación. Si en este caso se parametriza un valor superior al 50 %, la aproximación no comenzará hasta que se alcance el 50 % de la longitud del paso.</p>
setBlendingCart(...)	<p>Distancia de aproximación absoluta (tipo: double; unidad: mm)</p> <ul style="list-style-type: none"> ■ ≥ 0.0 <p>La distancia de aproximación absoluta es la distancia hasta el punto de destino del movimiento, en el que comienza la aproximación como muy pronto. Si se ajusta "0.0", el parámetro de aproximación no tendrá efecto.</p>

Método	Descripción
setBlendingOri(...)	<p>Parámetro de orientación para la aproximación (tipo: double; unidad: rad)</p> <ul style="list-style-type: none"> ■ ≥ 0.0 <p>La aproximación comienza como muy pronto cuando la diferencia absoluta del ángulo de orientación dominante con respecto a la orientación de destino no alcanza el valor ajustado aquí. Si se ajusta "0.0", el parámetro de aproximación no tendrá efecto.</p>
setOrientationType(...)	<p>Control de la orientación (tipo: Enum)</p> <ul style="list-style-type: none"> ■ Constant ■ Ignore ■ OriJoint ■ VariableOrientation (por defecto) <p>(>>> 14.9 "Control de la orientación LIN, CIRC, SPL" Página 258)</p>
setOrientationReferenceSystem(...)	<p>Solo es relevante para movimientos CIRC: Sistema de referencia del control de orientación (tipo: Enum)</p> <ul style="list-style-type: none"> ■ Base ■ Path <p>(>>> 14.9.1 "CIRC – Sistema de referencia del control de la orientación" Página 260)</p>

15.10.1 Programar parámetros de movimiento específicos del eje

Descripción	<p>Se pueden programar los siguientes parámetros de movimiento específicos del eje:</p> <ul style="list-style-type: none"> ■ Velocidad relativa setJointVelocityRel(...) ■ Aceleración relativa setJointAccelerationRel(...) ■ Tirón relativo setJointJerkRel(...) <p>Existen diferentes posibilidades para indicar estos valores relativos específicos del eje. Un valor válido para todos los ejes, diferentes valores para cada eje individual o un valor para un eje individual.</p> <p>Estas posibilidades se describen en el ejemplo de la velocidad relativa:</p> <ul style="list-style-type: none"> ■ setJointVelocityRel (Valor) Si se transmite un valor del tipo double, se aplicará la velocidad relativa para todos los ejes. ■ setJointVelocityRel (Variable de conjunto) Para asignar una velocidad relativa propia para cada eje, se transmite un conjunto double con los valores del eje correspondientes. En un conjunto se pueden definir los valores del eje de hasta 12 ejes, empezando con el eje A1. ■ setJointVelocityRel (Eje, valor) Para indicar la velocidad relativa de un eje individual, se transmite el eje como Enum del tipo JointEnum. Este Enum contiene 12 ejes (JointEnum.J1 ... JointEnum.J12).
Ejemplos	<p>Todos los ejes se desplazan con el 50 % de la velocidad máxima:</p> <pre>robot.move(ptp(getApplicationContext().getFrame("/P1")) .setJointVelocityRel(0.5));</pre>

El eje A5 se desplaza con el 50 %, todos los demás ejes se desplazan con el 20 % de la velocidad máxima:

```
double[] velRelJoints = {0.2, 0.2, 0.2, 0.2, 0.5, 0.2, 0.2, 0.2};
robot.move(ptp(getApplicationContext().getFrame("/P1"))
.setJointVelocityRel(velRelJoints));
```

El eje A4 se desplaza con el 50 % de la velocidad máxima, todos los demás ejes se desplazan con la velocidad máxima:

```
robot.move(ptp(getApplicationContext().getFrame("/P1"))
.setJointVelocityRel(JointEnum.J4, 0.5));
```

15.11 Utilizar herramientas y piezas en el programa

Una aplicación representa un esquema programado de una estación real y por ello también debe contener todos los objetos desplazables y los objetos geométricos fijos de la estación. Los objetos desplazables de la estación son, p. ej. robots, herramientas y piezas. Los objetos fijos de la estación son, p. ej. mesas de descarga o cintas transportadoras.

La unidad de control del robot y los robots se declaran y se inicializan automáticamente con la creación de la aplicación. Las herramientas y las piezas que se utilizan en la aplicación deben declararse e inicializarse por el usuario.

Las herramientas y las piezas con datos de carga y datos geométricos se crean y se gestionan en la vista **Plantillas de objetos**.

(>>> 9.3 "Gestión de objetos" Página 144)

Tipo de datos

Los tipos de datos para los objetos de una estación están predefinidos en RoboticsAPI:

Tipo de datos	Objeto
Controlador	Unidad de control del robot
LBR	Robot de estructura liviana
Tool	Herramienta
Workpiece	Pieza
GeometricObject	Objetos geométricos fijos

15.11.1 Declarar herramientas y piezas

Sintaxis

```
private Tipo de datos Objekt-Name;
```

Aclaración de la sintaxis

Elemento	Descripción
private	La palabra clave identifica variables válidas localmente. Válidas localmente significa que el campo de datos solo se puede utilizar por la clase correspondiente.
Tipo de datos	Tipo de objeto
Nombre del objeto	Nombre de la variable del objeto

Ejemplo

En la aplicación se utilizan 2 herramientas (garra, punta de guiado manual) y 1 pieza (clavija).

```
private Tool gripper;
private Tool guidingTool;
private Workpiece pen;
```

15.11.2 Inicializar herramientas y piezas

Descripción Aquí se describe cómo se inicializan las herramientas y las piezas que han sido almacenadas en las plantillas de objetos del proyecto correspondiente.

Sintaxis

```
Nombre del objeto =  
getApplicationData().createFromTemplate("Plantilla del objeto");
```

Aclaración de la sintaxis

Elemento	Descripción
<i>Nombre del objeto</i>	Nombre de la variable del objeto
<i>Plantilla del objeto</i>	Nombre de la plantilla del objeto que se indica en la vista Plantillas de objetos

Ejemplo Las siguientes herramientas y piezas han sido creadas en las plantillas de objetos:

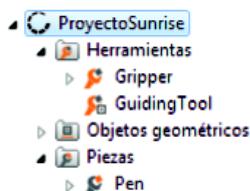


Fig. 15-9: Plantillas de objetos

```
private Tool gripper;  
private Tool guidingTool;  
private Workpiece pen;  
  
public void initialize() {  
    ...  
    gripper = getApplicationData().createFromTemplate("Gripper");  
    guidingTool = getApplicationData().createFromTemplate("GuidingTool");  
    pen = getApplicationData().createFromTemplate("Pen");  
    ...  
}
```

15.11.3 Conectar herramientas y piezas con el robot

Para que las herramientas y las piezas se puedan utilizar como objetos desplazables en instrucciones de movimiento, se deben conectar con el robot en la aplicación con el método `attachTo(...)`.

- Las herramientas están conectadas directa o indirectamente con la brida del robot.
- Las piezas están conectadas con el robot de forma indirecta, a través de una herramienta o de otras piezas.

Cuando una herramienta o una pieza está conectada con el robot mediante el método `attachTo(...)`, se tienen en cuenta los datos de carga por la unidad de control del robot. Además, se pueden utilizar todos los Frames del objeto conectado para la programación de movimientos.

(>>> 9.3.6 "Datos de carga" Página 149)

15.11.3.1 Conectar la herramienta con la brida del robot

Descripción El Frame de origen de una herramienta se conecta a través del método attachTo(...) con la brida de un robot utilizado en la aplicación. A través del método getFlange() se accede a la brida del robot.

Sintaxis Herramienta.attachTo(robot.getFlange());

Aclaración de la sintaxis

Elemento	Descripción
Herramienta	Nombre de la variable de la herramienta
Robot	Nombre del robot

Ejemplo Se conecta una punta de guiado manual con la brida del robot.

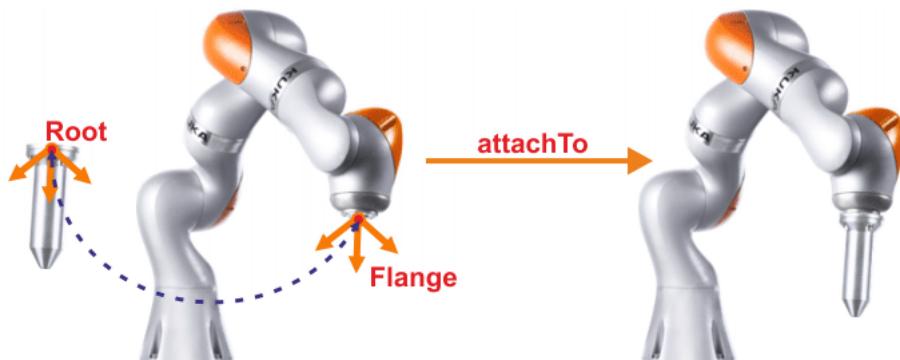


Fig. 15-10: Conectar la punta de guiado manual con la brida

```
private LBR robot;
private Tool guidingTool;
...
public void run() {
    ...
    guidingTool.attachTo(robot.getFlange());
    ...
}
```

15.11.3.2 Conectar la pieza con otros objetos

Descripción Por defecto se utiliza el Frame de origen de la pieza para conectarse con el Frame de otro objeto.

Aunque también se puede utilizar cualquier otro Frame que se haya creado para una pieza como punto de referencia para la conexión con otro objeto.

Los Frames para herramientas y piezas de trabajo se crean en la vista **Plantillas de objetos**. Para utilizar un Frame en el programa, se consulta el objeto de la herramienta o de la pieza con el método getFrame(...). Éste contiene como parámetro de entrada la ruta del Frame como cadena.

(>>> 9.3.4 "Crear Frame para herramienta o pieza" Página 146)

Sintaxis

Utilizar el Frame de origen para la conexión:

pieza.attachTo(Objeto.getFrame("Frame de destino"));

Utilizar otro Frame de referencia para la conexión:

pieza.getFrame("Frame de referencia").attachTo(Objekt.getFrame("Frame de destino"));

Aclaración de la sintaxis

Elemento	Descripción
Pieza de trabajo	Nombre de la variable de la pieza
Frame de referencia	Frame de referencia de la pieza que se utiliza para la conexión con otro objeto
Frame de destino	Frame del objeto con el que está conectado el Frame de referencia de la pieza



Después del Attach, coinciden el Frame de referencia de la pieza y el Frame de destino del objeto conectado con la misma.

Ejemplo 1

Se conecta una clavija a través de su Frame de origen con el Frame de la garra.

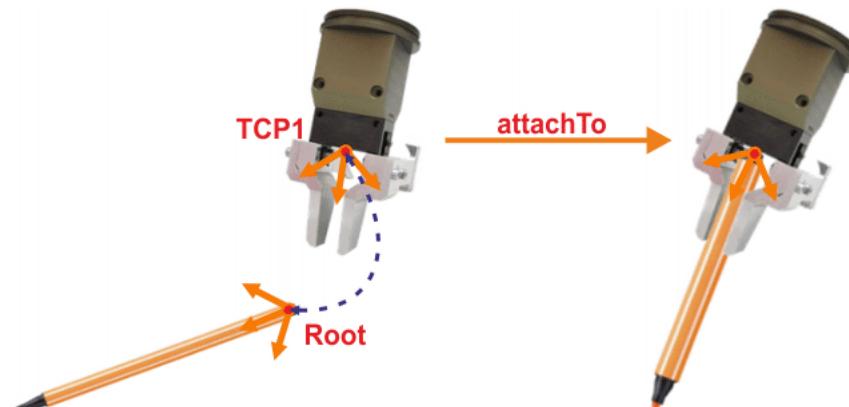


Fig. 15-11: Clavija en la garra (conexión mediante el Frame original)

```
private LBR robot;
private Tool gripper;
private Workpiece pen;
...
public void run() {
    ...
    pen.attachTo(gripper.getFrame("/TCP1"));
    ...
}
```

Ejemplo 2

En la punta de la garra hay definido un 2º Frame. Si éste se debe utilizar para agarrar la clavija, no es posible realizar la conexión a través del Frame de origen de la clavija. Para este caso se ha creado un punto de agarre en la clavija. Éste se indica como Frame de referencia para la conexión con la garra.

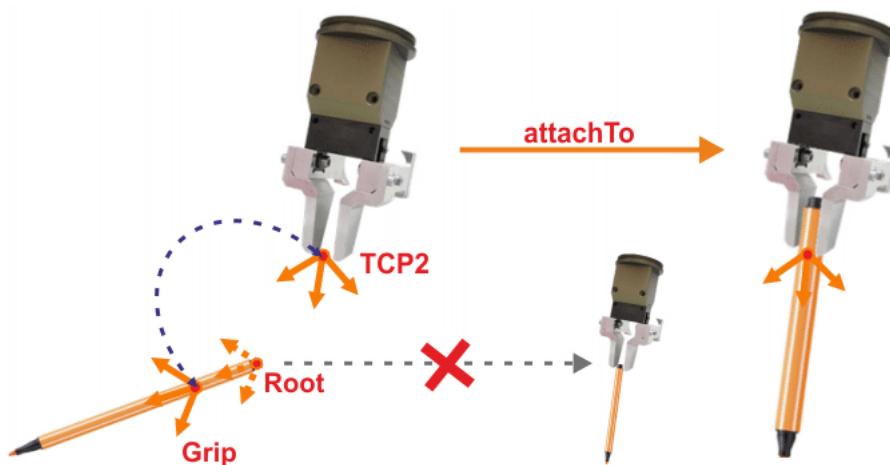


Fig. 15-12: Clavija en la garra (conexión mediante el Frame Grip)

```

private LBR robot;
private Tool gripper;
private Workpiece pen;
...
public void run() {
    ...
    pen.getFrame("/Grip").attachTo(gripper.getFrame("/TCP2"));
    ...
}

```

15.11.3.3 Desconectar conexiones

Descripción Si se va a desmontar una herramienta o se va a depositar una pieza, también se deberá desconectar la conexión en la aplicación. Para ello se utiliza el método `detach()`.

Sintaxis `Objeto.detach();`

Aclaración de la sintaxis

Elemento	Descripción
<code>Objeto</code>	Nombre de la variable del objeto

Ejemplo La conexión con la punta de guiado manual se desconecta.

```
guidingTool.detach();
```

15.11.4 Desplazar herramientas y piezas

Descripción Cualquier objeto desplazable de una estación se puede desplazar con `move(...)` y `moveAsync(...)`. El punto de referencia del movimiento depende del tipo de objeto:

- Si se desplaza un robot, el punto de referencia siempre es el punto central de la brida del robot.
- Si se desplaza una herramienta o una pieza, el punto de referencia es por defecto el Frame estándar para movimientos que se haya determinado para este objeto en la vista **Modelos de objetos**.

(>>> 9.3.5 "Establecer Frame estándar para movimientos" Página 148)

En este caso, la herramienta o la pieza se enlaza directamente con la instrucción de movimiento mediante el nombre de variable declarado en la aplicación.

- Aunque como punto de referencia para el movimiento también se puede programar cualquier otro Frame que haya sido creado para una herramienta o una pieza.

En este caso, con la ayuda del método `getFrame(...)` se debe indicar la ruta hasta el Frame del objeto que se utiliza para el movimiento (partiendo del Frame de origen del objeto).

Sintaxis

Utilizar el Frame estándar del objeto para el movimiento:

```
Objeto.move (movimiento) ;
```

Utilizar otro Frame del objeto para el movimiento:

```
Objeto.getFrame ("Frame desplazado") .move (movimiento) ;
```

Aclaración de la sintaxis

Elemento	Descripción
<i>Objeto</i>	Objeto de la estación que es desplazado Aquí se indica el nombre de variable del objeto que ha sido declarado e inicializado en la aplicación.
<i>Frame desplazado</i>	Ruta hasta el Frame del objeto que se utiliza para el movimiento
<i>Movimiento</i>	Movimiento que se ejecuta

Ejemplos

El movimiento PTP hasta el punto P1 se ejecuta con el Frame estándar de la garra.

```
gripper.attachTo(robot.getFlange());
gripper.move(ptpgetApplicationData().getFrame("/P1"));
```

El movimiento PTP hasta el punto P1 se ejecuta con otro Frame distinto al Frame estándar de la garra, aquí TCP1:

```
gripper.attachTo(robot.getFlange());
gripper.getFrame("/TCP1").move(ptpgetApplicationData().getFrame("/P1"));
```

Se agarra una clavija. El siguiente movimiento es un movimiento PTP hasta el punto P20. Este punto se ejecuta con el Frame estándar de la pieza clavija.

```
gripper.attachTo(robot.getFlange());
...
pen.attachTo(gripper.getFrame("/TCP1"));
pen.move(ptpgetApplicationData().getFrame("/P20"));
```

15.11.5 Ordenar cambios de carga en el control de seguridad

Descripción

El control de seguridad necesita los datos de carga de una pieza de trabajo para el cálculo de los momentos externos. El control de seguridad solo puede procesar los datos de carga de piezas de trabajo orientadas a las seguridad.

(>>> 9.3.8 "Piezas de trabajo destinadas a la seguridad" Página 153)

Durante un proceso pueden surgir cambios de carga debido a que el robot coge y suelta distintas piezas de trabajo. En la detección de colisión, el usuario debe comunicarle al control de seguridad qué pieza de trabajo destinada a la seguridad está actualmente activa a través del método `setSafetyWorkpiece(...)`. Para ello, esta pieza de trabajo se transfiere como parámetro de entrada.



Siempre se debe transferir una pieza de trabajo destinada a la seguridad a `setSafetyWorkpiece(...)`. Si se transfiere una pieza de trabajo no destinada a la seguridad, surge un error de excepción.

El método `setSafetyWorkpiece(...)` pertenece a la clase LBR y se puede utilizar en aplicaciones del robot o en tareas en segundo plano. La condición previa para la transferir una pieza de trabajo destinada a la seguridad al método es que se haya generado una instancia de la pieza de trabajo de las plantillas de objeto.

(>>> 15.11.2 "Inicializar herramientas y piezas" Página 294)

Para desactivar una pieza de trabajo destinada a la seguridad activa y comunicarle al control de seguridad que ya no se está sujetando una pieza de trabajo destinada a la seguridad, se le da el valor `null` al método `setSafetyWorkpiece(...)`.

A través de `setSafetyWorkpiece(...)` se ordena el cambio de carga para el control de seguridad. Para que los datos de carga de la pieza de trabajo también se tengan en cuenta en la parte no destinada a la seguridad de la unidad de control del robot, es necesario además, programar el cambio de carga con las instrucciones correspondientes.

(>>> 15.11.3.2 "Conectar la pieza con otros objetos" Página 295)

Sintaxis

`lbr.setSafetyWorkpiece(Pieza de trabajo) ;`

Aclaración de la sintaxis

Elemento	Descripción
<code>lbr</code>	Tipo: LBR Nombre del robot para el que se programa el cambio de carga
<code>Pieza</code>	Tipo: Workpiece Pieza de trabajo destinada a la seguridad cuyos datos de carga se deben transferir al control de seguridad Si ya no se debe tener en cuenta ninguna pieza de trabajo destinada a la seguridad, debe transmitirse <code>null</code> .

Ejemplo

En las plantillas de objetos se crea una herramienta destinada a la seguridad y dos piezas de trabajo destinadas a la seguridad.

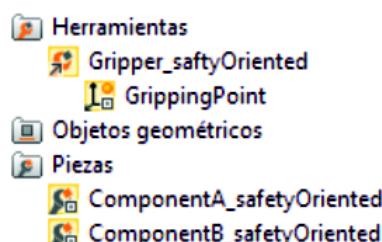


Fig. 15-13: Piezas de trabajo y herramientas (plantillas de objetos)

A la herramienta se le asigna el Frame "GrippingPoint" que sirve como punto de agarre para piezas de trabajo y está marcado como Frame estándar para movimientos.

En la aplicación se coge y se suelta la pieza de trabajo "ComponentA_safetyOriented". A continuación se coge la pieza de trabajo "ComponentB_safetyOriented". Los 3 cambios de carga deben tenerse en cuenta tanto en la parte destinada a la seguridad de la unidad de control del robot como en la no destinada a la seguridad.

```
public class ChangeOfLoadExample extends RoboticsAPIApplication {
    ...
    // safety-oriented tool and workpieces
    private Tool gripper;
```

```
private Workpiece componentA, componentB;

public void initialize() {
    ...
    // initialize safety-oriented components
    gripper = getApplicationData().
        createFromTemplate("Gripper_safetyOriented");
    componentA = getApplicationData().
        createFromTemplate("ComponentA_safetyOriented");
    componentB = getApplicationData().
        createFromTemplate("ComponentB_safetyOriented");

    // attach gripper to robot flange
    gripper.attachTo(lbr_iwa.getFlange());
}

public void run() {
    ...
    // after pick-up, attach workpiece to set load data for
    // motion control
    componentA.attachTo(gripper.getDefaultMotionFrame());
    // set load data for safety controller
    lbr_iwa.setSafetyWorkpiece(componentA);
    ...
    // after putting it down, detach workpiece to no longer
    // consider its load for motion control
    gripper.detach();
    // workpiece is no longer considered for safety
    // controller
    lbr_iwa.setSafetyWorkpiece(null);
    ...
    // pick-up of second workpiece
    componentB.attachTo(gripper.getDefaultMotionFrame());
    lbr_iwa.setSafetyWorkpiece(componentB);
    ...
}
...
}
```

15.12 Entradas/Salidas

Al exportar una configuración E/S desde WorkVisual, se crea una clase Java propia para cada grupo E/S en el proyecto Sunrise correspondiente. Cada una de estas clases Java contiene los métodos necesarios para la programación, para acceder mediante lectura a las entradas/salidas de un grupo E/S y mediante escritura a las salidas de un grupo E/S.



El código fuente de las clases Java del paquete **com.kuka.generatorAccess** no se puede modificar manualmente. Para ampliar la funcionalidad de un grupo E/S, se pueden衍生 otras clases a partir de las clases creadas o se pueden seguir utilizando objetos de estas clases, p. ej. como campos de clases propias (agregar).

Para poder utilizar las entradas/salidas de un grupo E/S en la aplicación, el usuario debe crear un campo de datos del tipo del grupo E/S e iniciarla.

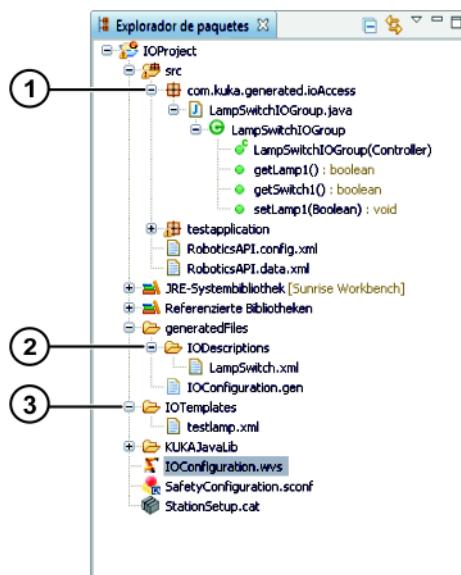


Fig. 15-14: Estructura del proyecto después de la exportación de la configuración de E/S

Pos.	Descripción
1	<p>Paquete de Java com.kuka.generated.ioAccess</p> <p>En el paquete se almacenan la clase creada para un grupo E/S y los métodos correspondientes.</p> <p>La clase de Java <i>NombreIOGroup.java</i> (aquí: <i>LampSwitchIOGroup.java</i>) contiene los siguientes elementos:</p> <ul style="list-style-type: none"> ■ Nombre de la clase del grupo E/S: <i>NombreIOGroup</i> ■ Constructor para asignar al grupo E/S la unidad de control del robot: <i>NombreIOGroup (Controlador)</i> ■ Método "get" y "set" para cada salida configurada: <i>getSalida ()</i>, <i>setSalida (Valor)</i> ■ método "get" para cada entrada configurada: <i>getEntrada ()</i>
2	<p>Carpeta generatedFiles > IDescriptions</p> <p>Los datos de un grupo E/S se guardan en un archivo XML. El archivo XML puede visualizarse pero no editarse.</p>
3	<p>Carpeta IOTemplates</p> <p>Los datos de un grupo E/S guardado como plantilla se guardan en un archivo XML. El archivo XML puede visualizarse pero no editarse.</p> <p>Una plantilla puede copiarse a otro proyecto de Sunrise para poder disponer de ella en el mismo. Entonces, la plantilla podrá importarse a WorkVisual, podrá editarse y exportarse de nuevo.</p> <p>(>>> 11.4.8 "Importar grupo E/S desde la plantilla" Página 168)</p> <p>(>>> 11.4.7 "Exportar grupo E/S como plantilla" Página 167)</p>



La carpeta **generatedFiles** se utiliza en el sistema y no se podrá utilizar para guardar archivos propios.

15.12.1 Crear el campo de datos para el grupo E/S

Descripción Mediante la declaración del campo de datos del tipo del grupo E/S, se importa automáticamente el paquete Java com.kuka.generated.ioAccess con las clases y métodos del grupo E/S.

Sintaxis

```
private Nombre de la clase Nombre de la clase;
```

Aclaración de la sintaxis

Elemento	Descripción
private	La palabra clave private identifica campos de datos que solo se pueden utilizar de la clase del entorno.
Nombre de la clase	Tipo de campo de datos Nombre de la clase del grupo E/S: ■ NombreIOGroup Nombre = Nombre del grupo E/S, definido como en WorkVisual
Nombre del campo	Nombre de campo de datos que se utiliza para la programación

Ejemplo

Para el grupo E/S "SwitchLamp" se crea el campo de datos "switchLamp".

```
public class RobotApplication extends RoboticsAPIApplication {
    ...
    private Controller controller;
    private SwitchLampIOGroup switchLamp;
    ...
}
```

15.12.2 Inicializar el campo de datos para el grupo E/S

Descripción En la inicialización del campo de datos, se indica a través del constructor de la clase la unidad de control del robot a la que están conectadas las entradas/salidas del grupo a través del bus de campo.

Sintaxis

```
Nombre del campo = new Nombre de la clase (Controlador);
```

Aclaración de la sintaxis

Elemento	Descripción
Nombre del campo	Nombre del campo de datos
new	Operador con el que se crea una nueva instancia de la clase Nombre de la clase
Nombre de la clase	Nombre de la clase del grupo E/S: ■ NombreIOGroup Nombre = Nombre del grupo E/S, definido como en WorkVisual
Controlador	Nombre del objeto que está asignado a la unidad de control del robot La asignación se realiza generalmente en el método initialize() de una aplicación del robot. (=> 15.1.2 "Estructura de una aplicación del robot" Página 269)

Ejemplo

Se inicializa el campo de datos "switchLamp".

```
public void initialize() {
    ...
}
```

```
switchLamp = new SwitchLampIOGroup(controller);
...
}
```

15.12.3 Leer entradas/salidas

Descripción Para consultar el estado de una entrada/salida, se utiliza el método get de la entrada/salida.

Sintaxis *Nombre del campo.getEntrada/salida();*

Aclaración de la sintaxis

Elemento	Descripción
<i>Nombre del campo</i>	Nombre del campo de datos
<i>Entrada/Salida</i>	Nombre de la entrada/salida (como esté definido en Work-Visual)

Ejemplo Se consulta el estado del interruptor en la entrada "Switch1" y el estado de la lámpara en la salida "Lamp1".

```
public void run() {
    ...
    switchLamp.getLamp1();
    switchLamp.getSwitch1();
    ...
}
```

15.12.4 Ajustar salidas



Las salidas se comutan en determinados casos a pesar de que existe una solicitud de parada destinada a la seguridad (p. ej. con PARADA DE EMERGENCIA ACCIONADA o una monitorización de la zona vulnerable). Por ello pueden darse movimientos inesperados de los elementos periféricos conectados (p. ej. apertura de una garra).

Se pueden dar, p. ej. los siguientes casos:

- Una tarea de segundo plano comuta la salida.
- Una función activada a través de una tecla de usuario comuta la salida.
- En caso de una solicitud de parada, las aplicaciones robot continúan en funcionamiento hasta la siguiente instrucción sincronizada de movimiento. El código que se ejecuta hasta entonces comuta la salida.

El comportamiento descrito también puede ser deseado, pero no puede causar peligro para personas o máquinas en ningún momento. Esto debe estar garantizado por la empresa constructora de instalaciones, p. ej. dejando cortando la alimentación de corriente de las salidas con potencial de peligro.

Descripción

Para modificar el valor de una salida, utilizar el método set de la salida.



Para las entradas no se encuentra disponible ningún método set. Solo se pueden leer.

Sintaxis

Nombre del campo.setSalida(valor);

Aclaración de la sintaxis

Elemento	Descripción
<i>Nombre del campo</i>	Nombre del campo de datos
<i>Salida</i>	Nombre de la salida (como esté definido en WorkVisual)
<i>Valor</i>	Valor de la salida. El tipo de datos del valor que debe transmitirse depende del tipo de salida.

Ejemplo

La lámpara en la salida "Lamp1" se conecta y pasados 2000 ms se vuelve a conectar.

```
public void run() {
    ...
    switchLamp.setLamp1(true);
    ThreadUtil.milliSleep(2000);
    switchLamp.setLamp1(false);
    ...
}
```

15.13 Consultar momentos axiales**Descripción**

En el eje del KUKA LBR iiwa se encuentra un sensor de momentos de articulaciones que mide el momento que actúa sobre el eje. Mediante el método `getMeasuredTorque()` de la clase LBR se pueden consultar y evaluar los valores de los momentos medidos en la aplicación.

A menudo los valores de medición puros no son interesantes, sino solo los momentos que actúan desde el exterior sin la parte que se genera debido al peso de la estructura y la inercia de masas durante el movimiento. Estos valores se denominan momentos externos. Se puede acceder a los momentos externos mediante el método LBR `getExternalTorque()`.

Ambas instrucciones envían como respuesta un objeto del tipo `TorqueSensorData` que contiene todos los datos del sensor de momentos de todos los objetos. Posteriormente, desde este objeto se pueden consultar opcionalmente todos los valores como conjunto mediante `getTorqueValues()` o un valor del eje individual mediante `getSingleTorqueValue(...)`.



Para la consulta de los datos de los sensores de momentos mediante Java no existe una acción en tiempo real. Esto significa que los datos suministrados por el sistema en el programa ya han sido registrados unos milisegundos antes.

Sintaxis

Consultar los datos del sensor medidos:

```
TorqueSensorData measuredData = lbr.getMeasuredTorque();
```

Consultar los datos de momentos que actúan de forma externa:

```
TorqueSensorData externalData = lbr.getExternalTorque();
```

Consultar los valores de los momentos de todos los ejes desde los datos del sensor:

```
double[] allValues = measuredData|externalData.getTorqueValues();
```

Consultar el valor del momento de un eje determinado desde los datos del sensor:

```
double singleValue =
measuredData|externalData.getSingleTorqueValues(joint);
```

Aclaración de la sintaxis

Elemento	Descripción
<i>measuredData</i>	Tipo: TorqueSensorData Variable para el valor de retorno de getMeasuredTorque(). El valor de retorno contiene los datos del sensor medidos.
<i>externalData</i>	Tipo: TorqueSensorData Variable para el valor de retorno de getExternalTorque(). El valor de retorno contiene los pares de torsión que actúan de forma externa.
<i>lbr</i>	Tipo: LBR Nombre del robot del que se consultan los datos del sensor.
<i>allValues</i>	Tipo: double[]; unidad: Nm Conjunto con todos los valores de los momentos que son consultados desde los datos del sensor.
<i>singleValue</i>	Tipo: double; unidad: Nm Valor del momento del eje que es consultado desde los datos del sensor
<i>joint</i>	Tipo: Enum del tipo JointEnum Eje cuyo valor de momento se debe consultar ■ JointEnum.J1 ... JointEnum.J12: Eje A1 ... A12

Ejemplo

En un determinado paso del proceso se consultan los momentos medidos y que actúan de forma externa en todos los ejes y se guardan en un conjunto para la evaluación posterior. El momento medido en el eje A2 se lee y se visualiza en la smartHMI.

```

TorqueSensorData measuredData = lbr.getMeasuredTorque();
TorqueSensorData externalData = lbr.getExternalTorque();

double[] measuredTorques = measuredData.getTorqueValues();
double[] externalTorques = externalData.getTorqueValues();

double torqueA2 = measuredData.getSingleTorqueValue(JointEnum.J2);
getLogger().info("Currently measured torque for joint 2 [Nm]: " +
torqueA2);

```

15.14 Leer fuerzas y momentos cartesianos

Se pueden leer las fuerzas y momentos cartesianos externos que actualmente actúan sobre la brida del robot, el TCP de una herramienta o un punto cualquiera de una pieza agarrada.

Observar los siguientes puntos:

- Los pares de torsión de los ejes se miden por los sensores de momentos.
- A partir de los pares de torsión medidos se calculan las fuerzas y momentos cartesianos.
- La fiabilidad de los valores calculados puede reducirse considerablemente en las posiciones extremas, p. ej. posiciones extendidas o singularidades.
- En la RoboticsAPI se encuentran métodos disponibles para comprobar la calidad y la validez de los valores calculados.



En caso de modificación de los datos de carga, p. ej. mediante la instrucción attachTo, la consulta se podrá realizar después del envío de una instrucción de movimiento a la unidad de control. Para ello es suficiente un movimiento de espacio cero o el envío de la instrucción de movimiento positionHold(...).



Las fuerzas y momentos cartesianos se estiman basándose en los valores de medición de los sensores de momentos de articulación. Para el cálculo es necesaria la indicación de un punto de aplicación de la fuerza. Las fuerzas y momentos calculados para el punto de aplicación indicado son físicos y solo son significativos si no actúan fuerzas externas en ningún otro punto de la estructura del robot.

15.14.1 Consultar datos de fuerza y de momentos calculados

Descripción

Para consultar las fuerzas y momentos cartesianos externos que actualmente actúan sobre un punto determinado, se utiliza el método getExternalForceTorque(...) de la clase LBR.

El método recibe un Frame como parámetro de transmisión. El Frame transmitido es el Frame de referencia para el cálculo de las fuerzas y los momentos, p. ej. la punta del palpador. El método calcula las fuerzas y los momentos externos existentes para la posición que se describe mediante el Frame.

Para realizar un cálculo físico significativo, el Frame debe describir un punto que esté unido con la brida de forma mecánicamente rígida. En la estructura de Frame, el Frame indicado también tiene que estar unido de forma estática con el Frame de la brida del robot.

Opcionalmente se puede transmitir un segundo Frame al método como parámetro. Este Frame indica la orientación de un sistema de coordenadas en el que se representan las fuerzas y los momentos.

Sintaxis

```
ForceSensorData data = lbr.getExternalForceTorque(
    measureFrame<, orientationFrame>);
```

Aclaración de la sintaxis

Elemento	Descripción
<i>data</i>	Tipo: ForceSensorData Variable para el valor de retorno de getExternalForceTorque(...). El valor de retorno incluye los datos de fuerza y de momentos calculados.
<i>lbr</i>	Tipo: LBR Nombre del robot
<i>measureFrame</i>	Tipo: AbstractFrame Frame de referencia para el que se calculan las fuerzas y los momentos que están actuando actualmente.
<i>orientationFrame</i>	Tipo: AbstractFrame Opcional: Orientación del Frame en el que se representan las fuerzas y los momentos

Ejemplos

Consulta de los datos de fuerza y de momentos en la brida del robot:

```
ForceSensorData data =
    robot.getExternalForceTorque(robot.getFlange());
```

Consulta de los datos de fuerza y de momentos en la brida del robot, referida a la orientación del sistema de coordenadas Mundo:

```
ForceSensorData data =
robot.getExternalForceTorque(robot.getFlange(),
World.Current.getRootFrame());
```

15.14.2 Consultar valores de fuerza y de momentos individuales

Descripción Los datos de fuerza y de momentos leídos con `getExternalForceTorque()` se pueden consultar de forma separada entre sí con la ayuda de los métodos `getForce()` y `getTorque(...)` de la clase `ForceSensorData`.

El resultado de estas consultas es respectivamente un vector. Con los métodos `get` de la clase `Vector` se pueden consultar individualmente los valores para cada grado de libertad.

(>>> 15.14.4 "Consultar valores individuales de un vector" Página 309)

Sintaxis Consultar el vector de fuerza:

```
Vector force = data.getForce();
```

Consultar el vector del momento:

```
Vector torque = data.getTorque();
```

Aclaración de la sintaxis

Elemento	Descripción
<code>force</code>	Tipo: Vector (<code>com.kuka.roboticsAPI.geometricModel.math</code>) Vector con las fuerzas cartesianas que actúan en la dirección X, Y y Z (unidad: N)
<code>torque</code>	Tipo: Vector (<code>com.kuka.roboticsAPI.geometricModel.math</code>) Vector con los momentos cartesianos que actúan alrededor del eje X, Y y Z (unidad: Nm)
<code>data</code>	Tipo: <code>ForceSensorData</code> Variable para el valor de retorno de <code>getExternalForceTorque(...)</code> . El valor de retorno incluye los datos de fuerza y de momentos calculados.

Ejemplo

Consulta de la fuerza cartesiana que actúa actualmente en dirección X en el Frame de la brida del robot:

```
ForceSensorData data =
robot.getExternalForceTorque(robot.getFlange());
```



```
Vector force = data.getForce();
double forceInX = force.getX();
```

15.14.3 Comprobar la fiabilidad de los valores de fuerza y de momentos calculados

Descripción Los valores de fuerza y de momentos calculados pueden desviarse en posiciones desfavorables del robot de las fuerzas y los momentos existentes realmente. Especialmente cerca de singularidades, algunos valores calculados están sometidos a una fuerte inseguridad y pueden ser no válidos. Esto se aplica para una parte de los valores calculados, en función de la posición del eje.

La calidad y la validez de los valores calculados se puede evaluar y consultarse en el programa. Para ello, se encuentran disponibles los siguientes métodos de la clase `ForceSensorData`:

- `getForceInaccuracy()`, `getTorqueInaccuracy()`

Se pueden consultar las imprecisiones de los valores de fuerza y valores de momentos calculados. El resultado de estas consultas es respectivamente un vector. Con los métodos `get` de la clase `Vector` se pueden consultar individualmente los valores para cada grado de libertad.

Dependiendo de la posición del eje, la calidad de los valores calculados para los distintos grados de libertad puede variar. A través de la consulta de valores individuales se pueden determinar aquellos grados de libertad para los que el cálculo de las fuerzas y momentos de la pose actual genera valores válidos.

(>>> 15.14.4 "Consultar valores individuales de un vector" Página 309)

- `isForceValid(...)`, `isTorqueValid(...)`

Se puede consultar si los valores de fuerza y momentos calculados son válidos. A los métodos se les transmite como parámetro respectivamente un valor límite para la máxima imprecisión permitida hasta la que el valor calculado aún es válido.

Sintaxis

Consultar la imprecisión de los valores de fuerza y momentos calculados:

```
Vector force = data.getForceInaccuracy();  
Vector torque = data.getTorqueInaccuracy();
```

Consultar la validez de los valores de fuerza y momentos calculados:

```
boolean valid = data.isForceValid(tolerance);  
boolean valid = data.isTorqueValid(tolerance);
```

Aclaración de la sintaxis

Elemento	Descripción
<code>force</code>	Tipo: Vector (com.kuka.roboticsAPI.geometricModel.math) Vector con los valores para la imprecisión con la que se han calculado las fuerzas cartesianas que actúan en la dirección X, Y y Z (unidad: N)
<code>torque</code>	Tipo: Vector (com.kuka.roboticsAPI.geometricModel.math) Vector con los valores para la imprecisión con la que se han calculado los momentos cartesianos que actúan alrededor del eje X, Y y Z (unidad: Nm)
<code>data</code>	Tipo: ForceSensorData Variable para el valor de retorno del método <code>getExternalForceTorque(...)</code> . El valor de retorno incluye los datos de fuerza y de momentos calculados.
<code>tolerance</code>	Tipo: double; unidad: N o Nm Valor límite para la máxima imprecisión permitida hasta la que los valores de fuerza y momentos calculados aún son válidos
<code>valid</code>	Tipo: boolean Variable para el valor de retorno de <code>isForceValid(...)</code> o <code>isTorqueValid(...)</code> <ul style="list-style-type: none"> ■ true: El valor de imprecisión en todas las direcciones cartesianas es menor o igual que el valor límite definido con <code>tolerance</code>. ■ false: El valor de imprecisión supera en una o varias direcciones cartesianas el valor de <code>tolerance</code>.

Ejemplo

Un bloque de instrucciones determinado solo se debe ejecutar si las fuerzas cartesianas externas, que actualmente actúan a lo largo de los ejes del sistema,

ma de coordenadas de la brida, han sido calculadas con una precisión de 20 N o mejor.

```
ForceSensoData data =
robot.getExternalForceTorque(robot.getFlange());  
  
if (data.isForceValid(20)) {
    //do something
}
```

15.14.4 Consultar valores individuales de un vector

Los métodos que consultan datos de un Frame envían generalmente un objeto de la clase Vector (paquete: com.kuka.roboticsAPI.geometricModel.math) como respuesta. Los componentes del vector se pueden consultar individualmente.

Resumen

Se encuentran disponibles los siguientes métodos de la clase Vector:

Método	Descripción
getX()	Tipo de retorno: double Consulta del componente x del vector
getY()	Tipo de retorno: double Consulta del componente y del vector
getZ()	Tipo de retorno: double Consulta del componente z del vector
get(<i>index</i>)	Tipo de retorno: double Consulta de los componentes establecidos mediante el parámetro <i>index</i> Valores de <i>index</i> (tipo: int): <ul style="list-style-type: none"> ■ 0: componente x del vector ■ 1: componente y del vector ■ 2: componente z del vector

15.15 Consulta de la posición del robot

La posición del robot específica del eje y cartesiana se puede consultar en la aplicación. Se puede consultar respectivamente la posición real y la posición nominal.

Resumen

Se encuentran disponibles los siguientes métodos de la clase Robot:

Método	Descripción
getCommandedCartesianPosition(...)	Tipo de retorno: Frame Consulta de la posición nominal cartesiana
getCommandedJointPosition()	Tipo de retorno: JointPosition Consulta de la posición nominal específica del eje
getCurrentCartesianPosition(...)	Tipo de retorno: Frame Consulta de la posición real cartesiana

Método	Descripción
getCurrentJointPosition()	Tipo de retorno: JointPosition Consulta de la posición real específica del eje
getPositionInformation(...)	Tipo de retorno: PositionInformation Consulta de la información de la posición cartesiana El valor de retorno contiene la siguiente información: <ul style="list-style-type: none"> ■ posición real específica del eje ■ posición nominal específica del eje ■ posición real cartesiana ■ posición nominal cartesiana ■ diferencia nominal-real cartesiana (rotatorio) ■ diferencia nominal-real cartesiana (translatorio)

15.15.1 Consultar la posición real o nominal específica del eje

Descripción	Para la consulta de la posición real o nominal específica del eje, primero se guarda la posición de los ejes del robot en una variable del tipo JointPosition. Seguidamente, se pueden consultar los valores del eje individualmente desde esta variable. El eje deseado se indica mediante un índice o mediante el valor JointEnum correspondiente.
Sintaxis	<p>Consultar la posición real específica del eje:</p> <pre>JointPosition position = robot.getCurrentJointPosition();</pre> <p>Consultar la posición nominal específica del eje:</p> <pre>JointPosition position = robot.getCommandedJointPosition();</pre> <p>Consultar valores del eje individualmente:</p> <pre>double value = position.get(axis);</pre>

Aclaración de la sintaxis	Elemento	Descripción
	<i>position</i>	Tipo: JointPosition Variable para el valor de retorno. El valor de retorno incluye las posiciones del eje consultadas.
	<i>robot</i>	Tipo: Robot Nombre del robot del que se consultan las posiciones del eje
	<i>value</i>	Tipo: double; unidad: rad Ángulo del eje consultado
	<i>axis</i>	<p>Tipo: int</p> <p>1. Posibilidad: Indicar el índice del eje cuyo valor del eje se consulta</p> <ul style="list-style-type: none"> ■ 0 ... 11: Eje A1 ... A12 <p>Tipo: Enum del tipo JointEnum</p> <p>2. Posibilidad: Indicar el valor JointEnum del eje cuyo valor del eje se consulta</p> <ul style="list-style-type: none"> ■ JointEnum.J1 ... JointEnum.J12: Eje A1 ... A12

Ejemplo

Primero se consulta la posición real del robot específica del eje y, seguidamente, el valor del eje A3 a través del índice del eje. El ángulo del eje A3 se indica en grados en la smartHMI.

```
JointPosition actPos = lbr.getCurrentJointPosition();
double a3 = actPos.get(2);
getLogger().info(Math.toDegrees(a3));
```

15.15.2 Consultar la posición real o nominal cartesiana**Descripción**

Se pueden consultar la posición real o nominal cartesiana de la brida del robot y cualquier otro Frame situado debajo. Es decir, cada Frame de un objeto que esté conectado con la brida del robot mediante la instrucción attachTo, es p. ej. el TCP de una herramienta o el Frame de una pieza agarrada.

El resultado de esta consulta, es decir, la posición cartesiana hace referencia por defecto al sistema de coordenadas Mundo. Opcionalmente se puede indicar otro sistema de coordenadas de referencia, con respecto al que se consulta la posición cartesiana. Éste puede ser, por ejemplo, un Frame creado en los datos de la aplicación o una base medida.

El resultado de la consulta se guarda en una variable del tipo Frame y contiene toda la información de redundancia necesaria (ángulo de redundancia, Status y Turn). Posteriormente, desde esta variable se pueden consultar la posición (X, Y, Z) y la orientación (A, B, C) del Frame, con la ayuda del método get específico del tipo.

Sintaxis

Consultar la posición real cartesiana:

```
Frame position = robot.getCurrentCartesianPosition(
frameOnFlange<, referenceFrame>);
```

Consultar la posición nominal cartesiana:

```
Frame position = robot.getCommandedCartesianPosition(
frameOnFlange<, referenceFrame>);
```

Aclaración de la sintaxis

Elemento	Descripción
<i>position</i>	Tipo: Frame Variables para el valor de retorno. El valor de retorno contiene la posición cartesiana consultada.
<i>robot</i>	Tipo: Robot Nombre del robot del que se consulta la posición cartesiana
<i>frameOn Flange</i>	Tipo: ObjectFrame Brida del robot o un Frame jerárquicamente inferior a la brida, cuya posición cartesiana se consulta
<i>reference Frame</i>	Tipo: AbstractFrame Sistema de coordenadas de referencia con respecto al que se consulta la posición cartesiana. Si no se ha indicado ningún sistema de coordenadas de referencia, la posición cartesiana hace referencia al sistema de coordenadas Mundo.

Ejemplos

Posición real cartesiana de la brida del robot referida al sistema de coordenadas Mundo:

```
Frame cmdPos = lbr.getCurrentCartesianPosition(lbr.getFlange());
```

Posición real cartesiana de TCP de una herramienta referida a una base:

```

tool.attachTo(lbr.getFlange());
...
Frame cmdPos = lbr.getCurrentCartesianPosition(tool.getFrame("/TCP"),
getApplicationData().getFrame("/Base"));

```

15.15.3 Consultar la diferencia nominal-real cartesiana

Descripción

La diferencia nominal-real cartesiana (= diferencia entre la posición programada y la posición medida) se puede consultar con el método `getPositionInformation(...)`.

El resultado de la consulta se guarda en una variable del tipo `PositionInformation`. Desde esta variable se pueden consultar diferencias nominales-reales translatorias y rotatorias separadas entre sí.

Sintaxis

Consultar información de la posición:

```
PositionInformation info = robot.getPositionInformation(
frameOnFlange<, referenceFrame>);
```

Consultar la diferencia nominal-real translatoria:

```
Vector translatoryDiff = info.getTranslationOffset();
```

Consultar la diferencia nominal-real rotatoria:

```
Rotation rotatoryDiff = info.getRotationOffset();
```



La posición real y nominal guardada en el objeto de información de posición se puede leer con los métodos ya conocidos `getCurrentCartesianPosition(...)` y `getCommandedCartesianPosition(...)`.

Aclaración de la sintaxis

Elemento	Descripción
<i>info</i>	Tipo: <code>PositionInformation</code> Variable para el valor de retorno. El valor de retorno incluye la información de posición consultada.
<i>robot</i>	Tipo: <code>Robot</code> Nombre del robot del que se consulta la información de posición
<i>frameOnFlange</i>	Tipo: <code>ObjectFrame</code> Brida del robot o un Frame jerárquicamente inferior a la brida, cuya información de posición se consulta
<i>referenceFrame</i>	Tipo: <code>AbstractFrame</code> Sistema de coordenadas de referencia relativo al que se consulta la información de posición. Si no se ha indicado ningún sistema de coordenadas de referencia, la información de posición hace referencia al sistema de coordenadas Mundo.

Elemento	Descripción
<i>translatoryDiff</i>	<p>Tipo: Vector (com.kuka.roboticsAPI.geometricModel.math)</p> <p>Diferencia nominal-real translatoria en dirección X, Y, Z (tipo: double; unidad: mm)</p> <p>Con los métodos get de la clase Vector se pueden consultar individualmente los valores de offset para cada grado de libertad.</p> <p>(>>> 15.14.4 "Consultar valores individuales de un vector" Página 309)</p>
<i>rotatoryDiff</i>	<p>Tipo: Rotation (com.kuka.roboticsAPI.geometricModel.math)</p> <p>Diferencia nominal-real del ángulo del eje A, B, C (tipo: double; unidad: rad)</p> <p>Con los métodos get de la clase Rotation - getAlphaRad(), getBetaRad, getGammaRad() - pueden consultarse individualmente los valores de offset para cada grado de libertad.</p>

Ejemplo

Lectura de la diferencia nominal-real translatoria en dirección X y de la diferencia nominal-real del ángulo de eje C.

```
tool.attachTo(lbr.getFlange());
...
PositionInformation posInf =
lbr.getPositionInformation(tool.getFrame("/TCP"),
getApplicationData().getFrame("/Base"));

Vector transDiff = posInf.getTranslationOffset();
Rotation rotDiff = posInf.getRotationOffset();

double transOffsetInX = transDiff.getX();
double rotOffsetofC = rotDiff.getGammaRad();
```

15.16 Posición HOME

La posición HOME es una posición del robot específica para aplicaciones. Se puede ajustar como nueva para una aplicación durante la inicialización.

La posición HOME está creada por defecto con los siguientes valores:

Eje	A1	A2	A3	A4	A5	A6	A7
Pos.	0°	0°	0°	0°	0°	0°	0°

15.16.1 Modificar la posición HOME

Descripción

La posición HOME en una aplicación se puede modificar con `setHomePosition(...)`. El método pertenece a la clase Robot.

Una Home Position debe cumplir las siguientes condiciones:

- Posición de salida favorable para la ejecución del programa.
- Posición de parada favorable. Por ejemplo, en la posición de parada el robot no debe ser un obstáculo.

La nueva posición HOME se puede transmitir como posición (Frame) específica de eje o cartesiana. Solo es válida en la aplicación en la que ha sido modificada. Otras aplicaciones siguen utilizando la posición HOME con los valores por defecto.

Sintaxis	<code>robot.setHomePosition(home);</code>						
Aclaración de la sintaxis	<table border="1"> <thead> <tr> <th>Elemento</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td><code>robot</code></td> <td>Tipo: Robot Nombre del robot a la que hace referencia la nueva posición HOME</td></tr> <tr> <td><code>home</code></td> <td> Tipo: JointPosition; unidad: rad 1. Posibilidad: Transmitir posiciones de los ejes del robot en la nueva posición HOME. 2. Posibilidad: Transmitir un Frame como nueva posición HOME. Indicación: El Frame debe contener toda la información de redundancia, para que las posiciones de los ejes del robot sean claras en la posición HOME. Este es el caso, por ejemplo, en un Frame programado por aprendizaje. </td></tr> </tbody> </table>	Elemento	Descripción	<code>robot</code>	Tipo: Robot Nombre del robot a la que hace referencia la nueva posición HOME	<code>home</code>	Tipo: JointPosition; unidad: rad 1. Posibilidad: Transmitir posiciones de los ejes del robot en la nueva posición HOME. 2. Posibilidad: Transmitir un Frame como nueva posición HOME. Indicación: El Frame debe contener toda la información de redundancia, para que las posiciones de los ejes del robot sean claras en la posición HOME. Este es el caso, por ejemplo, en un Frame programado por aprendizaje.
Elemento	Descripción						
<code>robot</code>	Tipo: Robot Nombre del robot a la que hace referencia la nueva posición HOME						
<code>home</code>	Tipo: JointPosition; unidad: rad 1. Posibilidad: Transmitir posiciones de los ejes del robot en la nueva posición HOME. 2. Posibilidad: Transmitir un Frame como nueva posición HOME. Indicación: El Frame debe contener toda la información de redundancia, para que las posiciones de los ejes del robot sean claras en la posición HOME. Este es el caso, por ejemplo, en un Frame programado por aprendizaje.						

Ejemplos Transmitir la posición específica del eje como posición HOME:

```
private LBR lbr;
...
JointPosition newHome = new JointPosition(0.0, 0.0, 0.0,
Math.toRadians(90), 0.0, 0.0, 0.0);
lbr.setHomePosition(newHome);
```

Transmitir el Frame programado por aprendizaje como posición HOME y desplazarse hasta el mismo con `ptpHome()`:

```
private LBR lbr;
...
ObjectFrame newHome = getApplicationData().getFrame("/Homepos");
lbr.setHomePosition(newHome);
lbr.moveAsync(ptpHome());
```

15.17 Consultar estados del sistema

El robot puede consultar diferentes estados del sistema y pueden editarse en la aplicación. La consulta de estados del sistema es necesaria principalmente para la utilización de un control superior, para que ésta pueda reaccionar ante las modificaciones del estado.

15.17.1 Consultar la posición HOME

Descripción	Para consultar la posición HOME, se encuentran disponibles los siguientes métodos de la clase Robot:
	<ul style="list-style-type: none"> ■ <code>getHomePosition()</code> Consulta de la posición HOME que está definida actualmente para el robot ■ <code>isInHome()</code> Consulta para saber si el robot se encuentra actualmente en la posición HOME

Sintaxis Consultar la posición HOME:

```
JointPosition homePos = robot.getHomePosition();
Comprobar si el robot se encuentra en la posición HOME.
```

```
boolean result = robot.isInHome();
```

Aclaración de la sintaxis

Elemento	Descripción
homePos	Tipo: JointPosition Variable para el valor de retorno de getHomePosition(). El valor de retorno contiene el ángulo del eje de la posición HOME consultada.
robot	Tipo: Robot Nombre del robot del que se consulta la posición HOME
result	Tipo: boolean Variable para el valor de retorno de isInHome(). El valor de retorno es true si el robot está en la posición HOME.

Ejemplo

Mientras que el robot aún no esté en la posición HOME, se debe ejecutar un bloque de instrucciones determinado.

```
private LBR lbr;  
...  
while (! lbr.isInHome ()) {  
    //do something  
}
```

15.17.2 Consultar el estado de ajuste

Descripción

Para consultar el estado de ajuste se encuentra disponible el método isMastered(). El método pertenece a la clase Robot.

Sintaxis

```
boolean result = robot.isMastered();
```

Aclaración de la sintaxis

Elemento	Descripción
robot	Tipo: Robot Nombre del robot cuyo estado de ajuste se consulta
result	Tipo: boolean Variable para el valor de retorno ■ true : Todos los ejes están ajustados. ■ false : Uno o más ejes están desajustados.

15.17.3 Consultar la disponibilidad de desplazamiento

Descripción

Para consultar si el robot está disponible para el desplazamiento, se encuentra disponible el método isReadyToMove(). El método pertenece a la clase Robot.

Sintaxis

```
boolean result = robot.isReadyToMove();
```

Aclaración de la sintaxis

Elemento	Descripción
<i>robot</i>	Tipo: Robot Nombre del robot cuya disponibilidad de desplazamiento se consulta
<i>result</i>	Tipo: boolean Variable para el valor de retorno <ul style="list-style-type: none"> ■ true: El robot está listo para el desplazamiento. ■ false: Una parada de seguridad está activa o los accionamientos del robot están en estado de error.



Si el valor de retorno es **true**, esto no significa necesariamente que los frenos están abiertos y que el robot se encuentra en una regulación activa.

15.17.3.1 Reaccionar a una modificación de la señal de disponibilidad de desplazamiento**Descripción**

En la RoboticsAPI se encuentra disponible un servicio de notificación de la clase Controlador, que señala la modificación de la señal de disponibilidad. Para registrarse para el servicio, se transmite un objeto IControllerStateListener al atributo Controlador en la aplicación del robot. Para ello se utiliza el método addControllerListener(...).

Para cada modificación de la señal de disponibilidad se activa el método onIsReadyToMoveChanged(...). En la estructura del método onIsReadyToMoveChanged(...) se puede programar cómo se debe reaccionar ante la modificación.

Sintaxis

```
kuka_Sunrise_Cabinet.addControllerListener(new
IControllerStateListener() {
    ...
    @Override
    public void onIsReadyToMoveChanged(Device device,
    boolean isReadyToMove) {
        // Reacción ante la modificación
    }
    ...
}) ;
```

Aclaración de la sintaxis

Elemento	Descripción
<i>kuka_Sunrise_Cabinet</i>	Tipo: Controlador Atributo Controlador de la aplicación del robot (= nombre de la unidad de control del robot en la aplicación)

15.17.4 Consultar la actividad del robot**Descripción**

Un robot está activo cuando está activo un comando de movimiento. Esto afecta tanto a las instrucciones de movimiento de la aplicación como a los comandos de desplazamiento manual.

Para consultar si el robot está activo, se encuentra disponible el método hasActiveMotionCommand(). El método pertenece a la clase Robot.

Sintaxis

```
boolean result = robot.hasActiveMotionCommand();
```

Aclaración de la sintaxis

Elemento	Descripción
<i>robot</i>	Tipo: Robot Nombre del robot cuya actividad se consulta
<i>result</i>	Tipo: boolean Variable para el valor de retorno <ul style="list-style-type: none"> ■ true: Una instrucción de movimiento está activa ■ false: No está activa ninguna instrucción de movimiento.



La consulta no aporta información relativa a si el robot está actualmente en movimiento o no.

Si el valor de retorno es **false**, esto no significa necesariamente que el robot está detenido.

Puede suceder, por ejemplo, que la actividad de robot se consulta justo después de una instrucción de movimiento sincronizada con condición de cancelación. Si la condición de cancelación se da, la consulta ofrece el valor **false** si el robot se frena pero está en movimiento.

Si el valor de retorno es **true**, esto no significa necesariamente que el robot está en movimiento. La consulta ofrece el valor **true** si, por ejemplo, un robot con regulación de posición ejecuta la instrucción de movimiento position-Hold(...) pero está detenido.

15.17.5 Consultar y evaluar señales de seguridad**Resumen**

En una aplicación de robot puede consultarse y evaluarse el estado de las siguientes señales de seguridad.

- Modo de servicio
- PARADA DE EMERGENCIA local
- PARADA DE EMERGENCIA externa
- Protección del operario
- Solicitud de parada (parada de seguridad)

Condición previa

Para poder evaluar una PARADA DE EMERGENCIA o una protección del operario:

- En la configuración de seguridad, en la fila correspondiente de la tabla de PSM está seleccionada la categoría adecuada.
 - Categoría **PARADA DE EMERGENCIA local** para PARADA DE EMERGENCIA local
 - Categoría **PARADA DE EMERGENCIA externa** para PARADA DE EMERGENCIA externa
 - Categoría **Protección del operario** para la protección del operario
- La reacción configurada es una parada de seguridad (ninguna salida).

15.17.5.1 Consultar el estado de las señales de seguridad**Descripción**

El estado actual de las distintas señales de seguridad se consulta primero con el método `getSafetyState()` y se resume en un objeto del tipo `ISafetyState`.

Posteriormente, desde este objeto se puede consultar el estado de señales de seguridad individuales con la ayuda de métodos específicos. Los posibles estados de las señales son Enums del tipo de retorno correspondiente.

Sintaxis `ISafetyState currentState = robot.getSafetyState();`

Aclaración de la sintaxis

Elemento	Descripción
<code>currentState</code>	Tipo: ISafetyState Variables para el valor de retorno. El valor de retorno contiene el estado de las señales de seguridad en el momento de la consulta.
<code>robot</code>	Tipo: LBR Nombre del robot del que se consulta el estado de las señales de seguridad

Resumen

Para consultar individualmente las señales de seguridad de ISafetyState, se encuentran disponibles los siguientes métodos:

Método	Descripción
<code>getOperationMode()</code>	Tipo de retorno: OperationMode (paquete com.kuka.robotics-API.deviceModel) Consulta del modo de servicio ajustado actualmente Valores Enum del tipo de retorno: <ul style="list-style-type: none"> ■ T1, T2, AUT ■ KRF: Modo de servicio KRF
<code>getEmergencyStopInt()</code>	Tipo de retorno: EmergencyStop Consulta de si está activa una PARADA DE EMERGENCIA local Valores Enum del tipo de retorno: <ul style="list-style-type: none"> ■ ACTIVE: La PARADA DE EMERGENCIA local está activa. ■ INACTIVE: La PARADA DE EMERGENCIA local no está activa. ■ NOT_CONFIGURED: No es relevante, ya que siempre está configurada una PARADA DE EMERGENCIA local.
<code>getEmergencyStopEx()</code>	Tipo de retorno: EmergencyStop Consulta de si está activa una PARADA DE EMERGENCIA externa Valores Enum del tipo de retorno: <ul style="list-style-type: none"> ■ ACTIVE: La PARADA DE EMERGENCIA externa está activa. ■ INACTIVE: La PARADA DE EMERGENCIA externa no está activa. ■ NOT_CONFIGURED: No está configurada ninguna PARADA DE EMERGENCIA externa.

Método	Descripción
getOperatorSafetyState()	<p>Tipo de retorno: OperatorSafety</p> <p>Consulta de la señal de protección del operario</p> <p>Valores Enum del tipo de retorno:</p> <ul style="list-style-type: none"> ■ OPERATOR_SAFETY_OPEN: La protección del operario se ha vulnerado (p. ej. la puerta de protección está abierta). ■ OPERATOR_SAFETY_CLOSED: La protección del operario no se ha vulnerado. ■ NOT_CONFIGURED: No está configurada ninguna protección del operario.
getSafetyStopSignal()	<p>Tipo de retorno: SafetyStopType</p> <p>Consulta de si una parada de seguridad es relevante</p> <p>Valores Enum del tipo de retorno:</p> <ul style="list-style-type: none"> ■ NOSTOP: No está activa ninguna parada de seguridad. ■ STOP0: Está activa una parada de seguridad 0 o una parada de seguridad 1. ■ STOP1: Está activa una parada de seguridad 1 (sobre la trayectoria). ■ STOP2: El valor no se emite actualmente.

Ejemplo

Se consulta si está activa una parada de seguridad. Si es el caso, se comprobará a continuación la protección del operario. Si la protección del operario se ha vulnerado, se emitirá un mensaje en la smartHMI.

```
ISafetyState safetyState = robot.getSafetyState();

SafetyStopType safetyStop = safetyState.getSafetyStopSignal();

if(safetyStop != SafetyStopType.NOSTOP) {
    OperatorSafety operatorSafety =
    safetyState.getOperatorSafetyState();
    if (operatorSafety == OperatorSafety.OPERATOR_SAFETY_OPEN)
        getLogger().warn("The safety gate is open!");
}
```

15.17.5.2 Reaccionar ante la modificación del estado de las señales de seguridad**Descripción**

En la RoboticsAPI se encuentra disponible un servicio de notificación de la clase Controlador, que señaliza modificaciones del estado de las señales de seguridad. Este servicio permite al usuario reaccionar directamente ante la modificación del estado de una señal.

Para registrarse para el servicio, se transmite un objeto ISunriseControllerStateListener al atributo Controlador en la aplicación del robot. Para ello se utiliza el método addControllerListener(...).

Para cada modificación del estado de una señal de seguridad se activa el método onSafetyStateChanged(...). En la estructura del método onSafetyStateChanged(...) se puede programar cómo se debe reaccionar ante la modificación del estado.

Sintaxis

```
kuka_Sunrise_Cabinet.addControllerListener(new
ISunriseControllerStateListener() {
    ...
    @Override
```

```
public void onSafetyStateChanged(Device device,
SunriseSafetyState safetyState) {
    // Reacción ante una modificación del estado
}
});
```

Aclaración de la sintaxis

Elemento	Descripción
<code>kuka_Sunrise_Cabinet</code>	Tipo: Controlador Atributo Controlador de la aplicación del robot (= nombre de la unidad de control del robot en la aplicación)

Ejemplo

Si cambia el estado de una señal de seguridad, se comprobará la protección del operario a través del método `onSafetyStateChanged(...)`. Si la protección del operario se ha vulnerado, se emitirá un mensaje en la smartHMI.

```
kuka_Sunrise_Cabinet.addControllerListener(new
ISunriseControllerStateListener() {
    ...
    @Override
    public void onSafetyStateChanged(Device device,
        SunriseSafetyState safetyState) {
        OperatorSafety operatorSafety =
        safetyState.getOperatorSafetyState();
        if (operatorSafety == OperatorSafety.OPERATOR_SAFETY_OPEN)
            getLogger().warn("The saftey gate is open!");
    }
});
```

15.18 Modificar y consultar el modo de ejecución del programa

Descripción

Mediante los métodos `setExecutionMode(...)` y `getExecutionMode()` del `SunriseExecutionService` puede modificarse y consultarse el tipo de flujo de programa. El controlador consulta al propio `SunriseExecutionService`.

Preparación

1. Crear una variable del tipo `SunriseExecutionService`.
2. Consultar `SunriseExecutionService` con la ayuda del método `getExecutionService()` del controlador y guardar en la variable.

Sintaxis

Modificar el tipo de flujo de programa:

```
service.setExecutionMode(ExecutionMode.newMode);
```

Consultar el tipo de flujo de programa actual:

```
currentMode = service.getExecutionMode();
```

Aclaración de la sintaxis

Elemento	Descripción
Servicio	Tipo: SunriseExecutionService Variable para el valor de retorno (contiene el SunriseExecutionService consultado por el controlador)
newMode	Tipo: Enum del tipo ExecutionMode Nuevo modo de ejecución del programa <ul style="list-style-type: none"> ■ ExecutionMode.Step: Modo Step (flujo de programa con parada después de cada instrucción de movimiento) ■ ExecutionMode.Continuous: Modo estándar (flujo de programa continuo sin paradas)
currentMode	Tipo: ExecutionMode Variable para el valor de retorno (contiene el modo de ejecución del programa consultado por SunriseExecutionService)

Ejemplo

El SunriseExecutionService es consultado por el controlador se guarda en la variable "serv".

```
private SunriseExecutionService serv;
...
public void initialize() {
    controller = getController("Controller_LBR5");
    robot = (LBR) getRobot(controller, "LBR5_7kg");
    serv = (SunriseExecutionService) controller.getExecutionService();
    ...
}
```

Primero se comuta al modo Step y, seguidamente, de nuevo al modo estándar.

```
public void run() {
    ...
    serv.setExecutionMode(ExecutionMode.Step);
    ...
    serv.setExecutionMode(ExecutionMode.Continuous);
    ...
}
```

Se consulta el modo de ejecución del programa actual.

```
public void run() {
    ...
    ExecutionMode currentMode;
    currentMode = serv.getExecutionMode();
    ...
}
```

15.19 Modificar y consultar el Override

La interfaz IApplicationOverrideControl ofrece métodos con los que se puede consultar o modificar el Override actual de la aplicación. Para ello, es necesario acceder, en un primer paso, a la interfaz IApplicationControl a través del método getApplicationControl().

Se distinguen los siguientes tipos de Override:

- Override manual: Override manualmente ajustable por el operario a través del smartPAD
(>>> 6.14.3 "Ajustar el Override manual" Página 94)
- Override de aplicación: Override programado, fijado por la aplicación
- Override efectivo: Producto del Override manual y el Override de aplicación
Override efectivo = Override manual * Override de aplicación

Resumen

Métodos para la consulta del Override actual:

Método	Descripción
getApplicationOverride()	Tipo de retorno: double Consulta del Override de aplicación
getManualOverride()	Tipo de retorno: double Consulta del Override manual
getEffectiveOverride()	Tipo de retorno: double Consulta del Override efectivo

Métodos para la modificación del Override:

Método	Descripción
setApplicationOverride(...)	Ajusta el Override de aplicación al valor indicado (tipo: double) ■ 0 ... 1
clipApplicationOverride(...)	Reduce el Override de aplicación al valor indicado (tipo: double) ■ 0 ... 1 Si se indica un valor que es superior al valor actualmente programado para el Override de aplicación, se ignora la instrucción clipApplicationOverride(...).
clipManualOverride(...)	Reduce el Override manual al valor indicado (tipo: double) ■ 0 ... 1 Si se indica un valor que es superior al valor actualmente programado para el Override manual, se ignora la instrucción clipManualOverride(...).

Ejemplo

```
getApplicationControl().setApplicationOverride(0.5);
...
double actualOverride =
getApplicationControl().getEffectiveOverride();
```

15.19.1 Reaccionar a un cambio de Override

Descripción

Una aplicación puede solicitar que se le comunique cuando se modifica un Override. Para ello, es necesario registrar y definir un Listener del tipo IApplicationOverrideListener.

En cada cambio de un Override se activa el método overrideChanged(...). En la estructura del método overrideChanged(...) se puede programar cómo se debe reaccionar ante el cambio.

Sintaxis

Definir Listener:

```
IApplicationOverrideListener overrideListener =
new IApplicationOverrideListener() {
```

```

@Override
public void overrideChanged(double effectiveOverride,
    double manualOverride, double applicationOverride) {
    // Reacción ante el cambio de Override
}
;
;

```

Registrar Listener:

```
getApplicationControl() .
addOverrideListener (overrideListener) ;
```

Dar de baja el Listener:

```
getApplicationControl() .
removeOverrideListener (overrideListener) ;
```

Aclaración de la sintaxis

Elemento	Descripción
override Listener	Tipo: IApplicationOverrideListener Nombre del Listener

15.20 Condiciones

A menudo en las aplicaciones se deben controlar valores y en caso de que se superen o no se alcancen los límites definidos, se debe activar una reacción. Las posibles fuentes para estos valores son, por ejemplo, los sensores del robot o las entradas configuradas. También se puede controlar al progreso de un movimiento. Las posibles reacciones son la cancelación de un movimiento en ejecución o la ejecución de una rutina de tratamiento.

15.20.1 Condiciones en la RoboticsAPI

Descripción Una condición puede tener 2 estados: Se cumple (estado TRUE) o no se cumple (estado FALSE). Para la definición de una condición se formula una expresión. En esta expresión, p. ej. se comparan las mediciones efectuadas por el sistema con un valor límite permitido. El resultado de la evaluación de la expresión define el estado de la condición.

Debido a que se utilizan distintos datos del sistema para la formulación de condiciones, existen diferentes tipos de condiciones. Cada uno de estos tipos de condiciones está disponible en la RoboticsAPI como clase propia. Pertenecen al paquete com.kuka.roboticsAPI.conditionModel e implementan la interfaz ICondition.

Resumen Están disponibles los siguientes tipos de condiciones:

Tipo de datos	Descripción
JointTorqueCondition	La condición de los momentos de los ejes se cumple si el momento medido en un eje se encuentra fuera de un rango de valores definido. (>>> 15.20.3 "Condición de los momentos axiales" Página 325)
ForceCondition	La condición de fuerza se cumple si la fuerza cartesiana que actúa sobre un Frame debajo de la brida del robot (p. ej en el TCP) supera un valor establecido. (>>> 15.20.4 "Condición de fuerza" Página 326)

Tipo de datos	Descripción
ForceComponentCondition	<p>La condición de componentes de fuerza se cumple cuando la fuerza cartesiana que actúa a lo largo de un Frame debajo de la brida del robot (p. ej. a lo largo del TCP), se encuentra fuera de una zona delimitada.</p> <p>(>>> 15.20.5 "Condición de componentes de Fuerza" Página 331)</p>
MotionPathCondition	<p>La condición referida a la trayectoria se cumple si se ha alcanzado una distancia definida en la trayectoria planificada, partiendo desde el punto inicial o de destino del movimiento. Adicionalmente se puede definir un desplazamiento temporal que debe cumplirse.</p> <p>(>>> 15.20.6 "Condición referida a la trayectoria" Página 333)</p>
BooleanIOCondition	<p>La condición para señales booleanas se cumple si una entrada digital booleana posee un estado determinado.</p> <p>(>>> 15.20.7 "Condición para señales booleanas" Página 335)</p>
IORangeCondition	<p>La condición para el rango de valores de una señal se cumple si el valor de una entrada analógica o digital se encuentra dentro de un rango definido.</p> <p>(>>> 15.20.8 "Condición para el rango de valores de una señal" Página 336)</p>

Campos de aplicación

- cancelación de movimientos
Un movimiento se cancela en el momento en que se produce una incidencia determinada. La incidencia se produce si la condición ya tiene el estado TRUE antes del inicio del movimiento o si cambia al estado TRUE durante el movimiento.
(>>> 15.21 "Condiciones de cancelación para instrucciones de movimiento" Página 336)
- Acciones de comutación referidas a la trayectoria (Trigger)
Una acción se activa en el momento en que se produce una incidencia determinada. La incidencia se produce si la condición ya tiene el estado TRUE antes del inicio del movimiento o si cambia al estado TRUE durante el movimiento.
(>>> 15.22 "Acciones de comutación referidas a la trayectoria (Trigger)" Página 340)
- Control de procesos (Monitoring)
El estado de una condición se comprueba cíclicamente con la ayuda de un Listener. Si cambia el estado de la condición, se podrá reaccionar ante ello.
(>>> 15.23 "Control de procesos (Monitoring)" Página 344)
- Espera con bloqueo de la condición
Una aplicación se detendrá hasta que se cumpla una condición determinada o haya finalizado un tiempo de espera determinado.
(>>> 15.24 "Espera con bloqueo de la condición" Página 349)

15.20.2 Condiciones complejas

Las condiciones se pueden enlazar de forma lógica entre sí, de forma que sea posible establecer condiciones completas. Los operadores lógicos necesarios para ello están disponibles como métodos ICondition. El objeto ICondition que activa se vincula con una o varias condiciones transmitiéndolas como parámetros.

Las instrucciones de vinculación pueden ejecutarse varias veces consecutivas para fijar e intercalar las operaciones. La evaluación depende del orden de la activación.

Operadores

Operador	Descripción/sintaxis
NOT	Inversión del objeto ICondition que activa ICondition invert();
XOR	Vinculación de disyunción exclusiva del objeto ICondition que activa con otra condición ICondition xor(ICondition other); other: otra condición
AND	Vinculación Y del objeto ICondition que activa con una o varias condiciones adicionales ICondition and(ICondition other1, ICondition other2, ...); other1, other2, ...: otras condiciones
OR	Vinculación O del objeto ICondition que activa con una o varias condiciones adicionales ICondition or(ICondition other1, ICondition other2, ...); other1, other2, ...: otras condiciones

Ejemplo

```
JointTorqueCondition condA = ...;
JointTorqueCondition condB = ...;
JointTorqueCondition condC = ...;
JointTorqueCondition condD = ...;

ICondition combil, combi2, combi3, combi4;

// NOT A
combil = condA.invert();

// A AND B AND C
combi2 = condA.and(condB, condC);

// (A OR B) AND C
combi3 = condA.or(condB).and(condC);

// (A OR B) AND (C OR D)
combi4 = condA.or(condB).and(condC.or(condD));
```

15.20.3 Condición de los momentos axiales

Descripción

Con la condición de los momentos de los ejes se puede comprobar si el momento externo calculado en un eje se encuentra fuera de un rango de valores definido.

(>>> 15.13 "Consultar momentos axiales" Página 304)



En la programación se deben indicar correctamente los datos de carga. Solo así se podrá utilizar correctamente la condición de los momentos axiales.

Sintaxis del constructor

JointTorqueCondition(JointEnum joint, double minTorque, double maxTorque)

Elemento	Descripción
<i>joint</i>	Eje cuyo valor de momento se comprueba ■ JointEnum.J1 ... JointEnum.J12: Eje A1 ... A12
<i>minTorque</i>	Valor límite inferior para el momento del eje (unidad: Nm) La condición se habrá cumplido si el momento del eje es menor o igual que <i>minTorque</i> .
<i>maxTorque</i>	Valor límite superior para el momento del eje (unidad: Nm) La condición se habrá cumplido si el momento del eje es mayor o igual que <i>maxTorque</i> .

Para la determinación del valor límite inferior y superior para el momento del eje se debe aplicar los siguientes: *minTorque* \leq *maxTorque*.

Ejemplo

```
JointTorqueCondition torqueCondJ3 =
new JointTorqueCondition(JointEnum.J3, -2.5, 4.0);
```

La condición se habrá cumplido si en el eje A3 se ha medido un valor de momento \leq -2,5 Nm o \geq 4,0 Nm.

15.20.4 Condición de fuerza

Descripción

Con la condición de fuerza se puede comprobar si una fuerza cartesiana que actúa sobre un Frame por debajo de la brida del robot supera algún valor límite definido.

Por ejemplo, se puede reaccionar sobre la fuerza que surge cuando el robot ejerce presión sobre una superficie mediante una herramienta montada en la brida. Para la condición de fuerza, se tienen en cuenta las proyecciones del vector de fuerza que actúa sobre un Frame por debajo de la brida. La posición de este Frame se define por el punto de aplicación de la fuerza (en este caso, la punta de la herramienta). La orientación del Frame debe corresponderse con la orientación de la superficie.

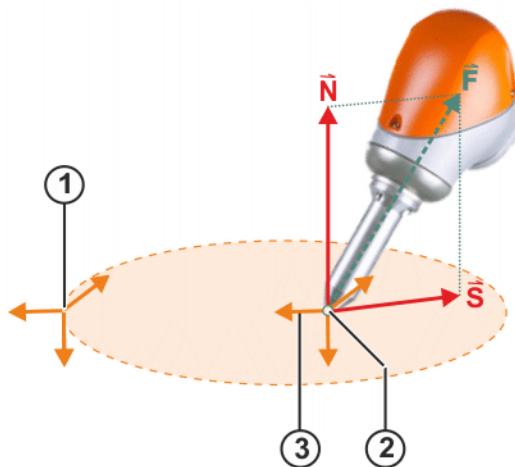


Fig. 15-15: Vectores de fuerza

- 1 Frame que indica la orientación del Frame de referencia (en este caso: la orientación de la superficie)

- 2 Punto de aplicación de la fuerza, en este caso, la punta de la herramienta
- 3 Frame de referencia por debajo de la brida sobre el que se proyecta el vector de fuerza. La posición del Frame se corresponde con el punto de aplicación de la fuerza. La orientación se corresponde con la orientación de la superficie.

Los siguientes vectores de fuerza son relevantes:

■ **Fuerza normal N:**

Proyección de la fuerza aplicada en la normal de la superficie (= vector que se encuentra dispuesto verticalmente sobre la superficie). De allí resulta la parte de la fuerza aplicada que actúa en vertical sobre la superficie. Por ejemplo, a través de la fuerza normal se ejerce presión para encajar un componente.

■ **Esfuerzo de corte S:**

Proyección de la fuerza aplicada sobre la superficie. De ahí resulta la parte de la fuerza aplicada que actúa en paralelo sobre la superficie. El esfuerzo de corte se produce, p. ej., mediante la fricción.



En la programación se deben indicar correctamente los datos de carga. Solo así se podrá aplicar correctamente la condición de fuerza.



Cerca de posiciones singulares, la estimación de fuerza no puede proporcionar valores correctos. Se recomienda no utilizar la condición de fuerza para semejantes configuraciones de ejes. Como alternativa puede utilizarse la condición del momento del eje, o bien la posición del eje puede adaptarse mediante la redundancia de forma que ya no haya ninguna posición singular más.

Métodos

Las condiciones de fuerza son del tipo de datos ForceCondition. ForceCondition contiene los siguientes métodos estáticos para la programación de condiciones:

- `createSpatialForceCondition(...)`: Condición para la fuerza cartesiana desde todas las direcciones
- `createNormalForceCondition(...)`: Condición para fuerza normal
- `createShearForceCondition(...)`: Condición para la fuerza de corte

Para la formulación de la condición se define un Frame por debajo del sistema de coordenadas de brida (p. ej. la punta de una herramienta) como sistema de referencia. Se determinan las fuerzas que actúan con respecto a este Frame. La orientación del sistema de referencia puede definirse opcionalmente a través de un Frame de orientación. Así puede definirse, por ejemplo, la posición de la superficie sobre la cual se aplica la fuerza.

Introduciendo un valor límite se define a partir de qué importe de la fuerza se cumple la condición.

La fuerza cartesiana se calcula a partir de los valores de los sensores de momentos de articulación. Dependiendo de la configuración del eje variará la fiabilidad de los valores de fuerza calculados. Si también se debe tener en cuenta la calidad del cálculo de la fuerza, se puede indicar un valor para la máxima imprecisión permitida. Si la imprecisión calculada por el sistema supera este valor, también se habrá cumplido la condición de fuerza.

15.20.4.1 Condición para la fuerza cartesiana desde todas las direcciones

Descripción A través del método estático `createSpatialForceCondition(...)` se puede definir una condición que se aplique independientemente de la dirección desde la que actúa la fuerza cartesiana sobre un Frame por debajo de la brida.

Sintaxis

```
ForceCondition.createSpatialForceCondition(
AbstractFrame measureFrame<, AbstractFrame orientationFrame<,
double threshold<, double tolerance>)
```

Elemento	Descripción
<i>measure Frame</i>	Frame debajo de la brida de robot sobre el que se determina la fuerza que actúa. A través de este parámetro se define la posición del punto de aplicación de fuerza.
<i>orientation Frame</i>	Opcional. A través de este parámetro se establece la orientación del sistema de referencia. Si no se indica el parámetro <i>orientationFrame</i> , <i>measureFrame</i> establece la orientación del sistema de referencia.
<i>threshold</i>	Valor de la fuerza que puede actuar como máximo sobre el sistema de referencia (unidad: N). <ul style="list-style-type: none"> ■ ≥ 0.0 La condición se cumple si el valor de la fuerza que actúa desde cualquier dirección sobre el sistema de referencia supera el valor aquí indicado.
<i>tolerance</i>	Opcional. Imprecisión máxima admisible de los valores determinados. <ul style="list-style-type: none"> ■ > 0.0 Por defecto: 10.0 La condición se cumple si la imprecisión del cálculo de la fuerza es mayor o igual que el valor aquí indicado. Si el parámetro no se indica, se emplea automáticamente el valor por defecto.

Ejemplo

En el momento en que el valor de la fuerza que actúa desde cualquier dirección sobre el TCP de una herramienta, supera 30 N, se habrá cumplido la condición.

```
Tool gripper = ...;
gripper.attachTo(lbr.getFlange());

ForceCondition spatialForce_tcp =
ForceCondition.createSpatialForceCondition(
    gripper.getFrame("/TCP"),
    30.0

);
```

15.20.4.2 Condición para fuerza normal

Descripción A través del método estático `createNormalForceCondition(...)` se puede formular una condición para la fuerza normal. Se analiza el porcentaje de la fuerza generada que actúa a lo largo de un eje definible de un Frame debajo de la brida (p. ej. a lo largo del eje del TCP). Este eje se establece generalmente de forma que se encuentre en posición vertical sobre la superficie en la que se aplica la fuerza (patrones de superficie).

Sintaxis

```
ForceCondition.createNormalForceCondition(AbstractFrame
    measureFrame<, AbstractFrame orientationFrame>, CoordinateAxis
    direction, double threshold<, double tolerance>)
```

Elemento	Descripción
<i>measureFrame</i>	Frame debajo de la brida de robot sobre el que se determina la fuerza que actúa. A través de este parámetro se define la posición del punto de aplicación de fuerza.
<i>orientationFrame</i>	Opcional. A través de este parámetro se establece la orientación del sistema de referencia. Si no se indica el parámetro <i>orientationFrame</i> , <i>measureFrame</i> establece la orientación del sistema de referencia.
<i>direction</i>	Eje de coordenadas del sistema de referencia. Se comprueba la parte de fuerza que actúa a lo largo del eje indicado aquí. <ul style="list-style-type: none"> ■ X, Y, Z
<i>threshold</i>	Valor de la fuerza que puede actuar como máximo a lo largo del sistema de referencia (unidad: N). <ul style="list-style-type: none"> ■ ≥ 0.0 La condición se cumple si el valor de la fuerza supera al valor indicado aquí.
<i>tolerance</i>	Opcional. Imprecisión máxima admisible de los valores determinados. <ul style="list-style-type: none"> ■ > 0.0 Por defecto: 10.0 La condición se cumple si la imprecisión del cálculo de la fuerza es mayor o igual que el valor aquí indicado. Si el parámetro no se indica, se emplea automáticamente el valor por defecto.

Ejemplo

Una garra montada en la brida presiona sobre una placa de mesa. Se desea reaccionar a la parte de la fuerza aplicada en el TCP de la garra que actúa de forma vertical sobre la placa de la mesa. Por ello, el sistema de referencia se determina de tal manera que su eje Z transcurra a lo largo de la normal de superficie de la placa de la mesa.

En cuanto la fuerza normal supera el valor de 45 N, se habrá cumplido la condición. Además, la condición se considerará como cumplida si la imprecisión de los datos calculados supera un valor de 8.

```
Tool gripper = ...;
gripper.attachTo(lbr.getFlange());

SpatialObject table = ...;

ForceCondition normalForce_z =
ForceCondition.createNormalForceCondition(
    gripper.getFrame("/TCP"),
    table.getFrame("/Tabletop"),
    CoordinateAxis.Z,
    45.0,
    8.0
);
```

15.20.4.3 Condición para la fuerza de corte

Descripción A través del método estático `createShearForceCondition(...)` se puede formular una condición para la fuerza de corte. Se tiene en cuenta la parte de la fuerza generada que actúa en paralelo a una superficie. La posición de la superficie se determina indicando el eje que se encuentra en vertical sobre ella.

Sintaxis

```
ForceCondition.createShearForceCondition(AbstractFrame
measureFrame<, AbstractFrame orientationFrame>, CoordinateAxis
normalDirection, double threshold<, double tolerance>)
```

Elemento	Descripción
<i>measureFrame</i>	Frame debajo de la brida de robot sobre el que se determina la fuerza que actúa. A través de este parámetro se define la posición del punto de aplicación de fuerza.
<i>orientationFrame</i>	Opcional. A través de este parámetro se establece la orientación del sistema de referencia. Si no se indica el parámetro <i>orientationFrame</i> , <i>measureFrame</i> establece la orientación del sistema de referencia.
<i>normalDirection</i>	Eje de coordenadas del sistema de referencia. El eje indicado aquí define la normal de superficie de un nivel. Se comprueba la parte de fuerza que actúa de forma paralela a este nivel. <ul style="list-style-type: none"> ■ X, Y, Z
<i>threshold</i>	Valor de fuerza que puede actuar como máximo de forma paralela al nivel del sistema de referencia definido por su normal de superficie (unidad: N). <ul style="list-style-type: none"> ■ ≥ 0.0 La condición se cumple si el valor de la fuerza supera al valor indicado aquí.
<i>tolerance</i>	Opcional. Imprecisión máxima admisible de los valores determinados. <ul style="list-style-type: none"> ■ > 0.0 Por defecto: 10.0 La condición se cumple si la imprecisión del cálculo de la fuerza es mayor o igual que el valor aquí indicado. Si el parámetro no se indica, se emplea automáticamente el valor por defecto.

Ejemplo

Una garra montada en la brida presiona sobre una placa de mesa. Se desea medir la fuerza en el TCP de la garra con la orientación de la placa de mesa. Se tiene en cuenta la fuerza de corte que actúa en paralelo al plano XY del punto de medición definido por el TCP y la posición de la mesa.

Para definir el plano XY debe indicarse como parámetro el eje que se encuentra perpendicular a este plano. Este es el eje Z.

En cuanto la fuerza de corte supera el valor de 25 N, se habrá cumplido la condición. Además, la condición se considerará como cumplida si la imprecisión de los datos calculados supera un valor de 5.

```
Tool gripper = ...;
gripper.attachTo(lbr.getFlange());
```

```

SpatialObject table = ...;

ForceCondition shearForce_xyPlane =
ForceCondition.createShearForceCondition(
    gripper.getFrame("/TCP"),
    table.getFrame("/Tabletop"),
    CoordinateAxis.Z,
    25.0,
    5.0

);

```

15.20.5 Condición de componentes de Fuerza

Descripción	Mediante la condición de componentes de fuerza se puede comprobar si la fuerza cartesiana que actúa en dirección X, Y o Z sobre un Frame por debajo de la brida del robot (p. ej. sobre el TCP) se encuentra fuera de un sector definido.
--------------------	---



En la programación se deben indicar correctamente los datos de carga. Solo así se podrá utilizar correctamente la condición de componentes de fuerza.



Cerca de posiciones singulares, la estimación de fuerza no puede proporcionar valores correctos. Se recomienda no utilizar la condición de componentes de fuerza para este tipo de configuraciones de ejes. Como alternativa puede utilizarse la condición de momentos de eje, o bien adaptar la posición de eje mediante la redundancia de forma que ya no haya singularidad.

La condición de componentes de fuerza pertenece a la clase ForceComponentCondition. Para la condición de componentes de fuerza se define un Frame por debajo del sistema de coordenadas debridas como sistema de referencia. En este Frame, p. ej., la punta de una herramienta, se determina la fuerza. La orientación del sistema de referencia puede definirse opcionalmente a través de un Frame de orientación.

La dirección desde la que se comprueba la fuerza se establece con uno de los ejes de coordenadas del sistema de referencia. La condición de componentes de fuerza se cumple si la fuerza cartesiana a lo largo del eje de coordenadas determinado del sistema de referencia se encuentra fuera de una gama de valores definible.

La fuerza cartesiana se calcula a partir de los valores de los sensores de momentos de articulación. Dependiendo de la configuración del eje variará la fiabilidad de los valores de fuerza calculados. Si también se debe tener en cuenta la calidad del cálculo de la fuerza, se puede indicar un valor para la máxima imprecisión permitida. Si la imprecisión determinada por el sistema supera este valor, también se habrá cumplido la condición de componentes de fuerza.

Sintaxis del constructor	La clase ForceComponentCondition cuenta con varios constructores que se distinguen en el número de los parámetros de entrada:
---------------------------------	---

```

ForceComponentCondition(AbstractFrame measureFrame
<, AbstractFrame orientationFrame>, CoordinateAxis coordinateAxis,
double min, double max<, double tolerance>)

```

Elemento	Descripción
<i>measureFrame</i>	Frame debajo de la brida de robot sobre el que se determina la fuerza que actúa. A través de este parámetro se define la posición del punto de aplicación de fuerza.
<i>orientationFrame</i>	Opcional. A través de este parámetro se establece la orientación del sistema de referencia. Si no se indica el parámetro <i>orientationFrame</i> , <i>measureFrame</i> establece la orientación del sistema de referencia.
<i>coordinateAxis</i>	Eje de coordenadas del Frame sobre el que se determina la fuerza que actúa. Establece la dirección desde la que se comprueba la fuerza que actúa. <ul style="list-style-type: none"> ■ X, Y, Z
<i>mín</i>	Límite inferior de la gama de valores para la fuerza que actúa a lo largo del eje de coordenadas del sistema de referencia (unidad: N). La condición de componentes de fuerza se cumple si el valor de la fuerza no alcanza al valor indicado aquí.
<i>máx.</i>	Límite superior de la gama de valores para la fuerza que actúa a lo largo del eje de coordenadas del sistema de referencia (unidad: N). La condición de componentes de fuerza se cumple si la fuerza supera al valor indicado aquí. Nota: El valor límite superior debe ser mayor que el valor límite inferior: <i>max > min</i> .
<i>tolerance</i>	Opcional. Imprecisión máxima admisible de los valores determinados. <ul style="list-style-type: none"> ■ > 0.0 Por defecto: 10.0 La condición de componentes de fuerza se cumple si la imprecisión del cálculo de la fuerza es mayor o igual que el valor aquí indicado. Si el parámetro no se indica, se emplea automáticamente el valor por defecto.

Ejemplo

Un proceso de ensamblado se ejecuta de manera óptima con una fuerza entre 20 N y 25 N. Se desea definir una condición de componentes de fuerza que se habrá cumplido, cuando la fuerza que actúa sobre el extremo libre de una pieza de trabajo sujetada en dirección Z, se encuentra entre 20 N y 25 N.

Para ello, primero se define una condición de componentes de fuerza que posee el estado FALSE en esta gama de valores. A continuación se obtiene el resultado deseado mediante inversión.

```
Workpiece peg = ...;
peg.attachTo(...);

ForceComponentCondition assemblyForce_inverted = new
ForceComponentCondition(
    peg.getFrame("/Assembly"),
    CoordinateAxis.Z,
    20.0,
    25.0
);
```

```
ForceComponentCondition assemblyForce = (ForceComponentCondition)
assemblyForce_inverted.invert();
```

15.20.6 Condición referida a la trayectoria

Descripción

Las condiciones referidas a la trayectoria se utilizan siempre en relación con una instrucción de movimiento. Sirven como criterio de cancelación o como Trigger para las acciones de conmutación referidas a la trayectoria.

La condición define el punto en la trayectoria planificada (punto de conexión) en el que se debe cancelar un movimiento o se debe activar una acción deseada. Cuando se alcanza el punto de conexión, la condición se considerará como cumplida.



El proceso de frenado o la acción definida solo se lleva a cabo al alcanzar el punto de conexión. De esta forma, si se emplea la condición referida a la trayectoria como condición de terminación, el robot no se detiene directamente en el punto de conexión, sino después de este.

El punto de conexión se puede establecer mediante un desplazamiento espacial y/o temporal. El desplazamiento puede referirse opcionalmente al punto de inicio o al punto de destino de un movimiento.



Si se define un desplazamiento temporal, una modificación del Override afecta al punto de conexión. Por ello, la acción conectada con una condición referida a la trayectoria solo se activa en el punto de conexión definido con un Override efectivo del 100 % y en los modos de servicio T2 o Automático.

Las condiciones referidas a la trayectoria son del tipo de datos MotionPathCondition.

Sintaxis del constructor

La clase MotionPathCondition posee el siguiente constructor:

```
MotionPathCondition(ReferenceType reference, double distance,  
long delay)
```

Métodos estáticos

Un objeto MotionPathCondition también se puede crear mediante uno de los siguientes métodos estáticos:

```
MotionPathCondition.createFromDelay(ReferenceType reference, long delay)
```

```
MotionPathCondition.createFromDistance(ReferenceType reference, double distance)
```

Elemento	Descripción
<i>reference</i>	<p>Tipo de datos: com.kuka.roboticsAPI.conditionModel.ReferenceType</p> <p>Punto de referencia para la condición</p> <ul style="list-style-type: none"> ■ ReferenceType.START: Punto de inicio ■ ReferenceType.DEST: Punto de destino
<i>distance</i>	<p>Desplazamiento espacial con respecto al punto de referencia de la condición</p> <p>En el caso de movimientos CP, <i>distance</i> indica la distancia cartesiana entre el punto de conexión y el punto de referencia (= distancia de la trayectoria que une el punto de conexión con el punto de referencia), no la unión más corta entre estos puntos. (Unidad: mm)</p> <p>En movimientos PTP con <i>distance</i> no se indica una distancia cartesiana, sino un parámetro de trayectoria sin unidad.</p> <ul style="list-style-type: none"> ■ Valor negativo: Desplazamiento en sentido contrario a la dirección de movimiento ■ Valor positivo: Desplazamiento en la dirección de movimiento <p>(>>> "Desplazamiento máximo" Página 334)</p>
<i>delay</i>	<p>Desplazamiento temporal con respecto al punto de la trayectoria definido mediante <i>distance</i>. O si <i>distance</i> no está definido, con respecto al punto de referencia de la condición. (Unidad: ms)</p> <ul style="list-style-type: none"> ■ Valor negativo: Desplazamiento en sentido contrario a la dirección de movimiento ■ Valor positivo: Desplazamiento en la dirección de movimiento <p>Las fases en las que la aplicación está pausada no se incluyen en la medición de tiempo.</p> <p>(>>> "Desplazamiento máximo" Página 334)</p>

Desplazamiento máximo

El punto de conexión solo se puede desplazar hasta límites determinados. Los límites son válidos para el desplazamiento total que se obtiene del desplazamiento en el espacio y temporal.

- Desplazamiento negativo máximo hasta el punto de inicio del movimiento
- Desplazamiento positivo máximo hasta el punto de destino del movimiento

Las siguientes parametrizaciones no se deben utilizar, ya que provocan forzosamente un desplazamiento más allá de los límites permitidos y un error de tiempo de ejecución:

Combinación de valores	Efecto
<i>reference = ReferenceType.START</i> <i>distance < 0</i>	El punto de conexión se encuentra antes del inicio del movimiento.
<i>reference = ReferenceType.START</i> <i>distance = 0</i> <i>delay < 0</i>	

Combinación de valores	Efecto
reference = ReferenceType.DEST distance > 0	El punto de conexión se encuentra después del final del movimiento.
reference = ReferenceType.DEST distance = 0 delay > 0	

Si se ha utilizado una combinación de valores válida, puede ocurrir igualmente que el punto de conexión sea desplazado más allá de los límites permitidos. En estos casos, el comportamiento adecuado es el siguiente:

- Una condición que se ha cumplido antes del inicio del movimiento activa el movimiento en el punto de inicio.
- Una condición que se ha cumplido después del final del movimiento, nunca se activa.

Ejemplo

Para una aplicación de pegamentos se debe formular una condición referida a la trayectoria. El canal de pegado debe terminar a 5 cm del punto de destino del movimiento. Para que el flujo de pegamento pueda finalizarse a tiempo, la condición debe cumplirse 700 ms antes de alcanzar esta distancia hasta el destino.

```
MotionPathCondition glueStop = new
MotionPathCondition(ReferenceType.DEST, -50.0, -700);
```

15.20.7 Condición para señales booleanas

Descripción

Con la condición para señales booleanas se pueden comprobar entradas digitales de 1 bit. La condición se cumple si una entrada booleana posee un estado determinado.

Las condiciones para señales booleanas son del tipo de datos BooleanIO-Condition.

Sintaxis del constructor

```
BooleanIOCondition(AbstractIO booleanSignal, boolean booleanIO-
Value)
```

Elemento	Descripción
boolean Signal	Señal de entrada booleana que se comprueba
boolean IOValue	Estado de la señal de entrada en la que se cumple la condición <ul style="list-style-type: none"> ■ true, false

Ejemplo

Mediante un interruptor se envía una señal de entrada digital booleana. Para poder reaccionar ante la señal en una aplicación, debe formularse una condición de señal booleana. La condición debe cumplirse en cuanto existe un nivel HIGH (estado TRUE) al pulsar el interruptor.

```
SwitchesIOGroup switches = new SwitchesIOGroup(...);
AbstractIO switch_1 = switches.getInput("Switch1");

BooleanIOCondition switch1_active = new BooleanIOCondition(switch_1,
true);
```

15.20.8 Condición para el rango de valores de una señal

Descripción Con la condición para el rango de valores de una señal se puede comprobar el valor de una entrada digital o analógica. La condición se cumple si el valor de la señal se encuentra dentro de un rango definido.

Las condiciones para los rangos de valores son del tipo de datos IORangeCondition.

Sintaxis del constructor

```
IORangeCondition(AbstractIO signal, Number minValue, Number maxValue)
```

Elemento	Descripción
<i>signal</i>	Señal analógica o digital que se comprueba
<i>minValue</i>	Límite inferior del rango de valores en el que se cumple la condición El valor que envía la señal debe ser mayor o igual que el <i>minValue</i> .
<i>maxValue</i>	Límite superior del rango de valores en el que se cumple la condición El valor que envía la señal debe ser menor o igual que el <i>maxValue</i> .

Ejemplo Mediante un sensor de temperatura se envía una señal de entrada analógica, cuyo valor puede estar en un rango entre 0°C y 2000°C. En cuanto se supera el umbral de 35°C, se debe cumplir una condición para el control de la señal del sensor.

```
SensorIOGroup sensors = new SensorIOGroup(...);
AbstractIO temperatureSensor =
sensors.getInput("TemperatureSensor2");

IORangeCondition tempHigher35 = new
IORangeCondition(temperatureSensor, 35.0, 2000.0);
```

15.21 Condiciones de cancelación para instrucciones de movimiento

Para determinados procesos es necesario que un movimiento planificado no se ejecute hasta el final, sino que se cancela si se producen incidencias definibles. En los procesos de ensamblado, el robot se debe detener, por ejemplo, cuando se alcanza un umbral de fuerza.

15.21.1 Establecer la condición de cancelación

Descripción Las condiciones de cancelación son condiciones que deben provocar la cancelación de un movimiento. Una condición de cancelación se cumple si ya tiene el estado TRUE antes del inicio del movimiento o si cambia al estado TRUE durante el movimiento.

Las condiciones están definidas como objetos del tipo ICondition. Los tipos de condiciones disponibles pertenecen al paquete com.kuka.roboticsAPI.conditionModel.

Aquí se puede encontrar un resumen de los tipos de condiciones disponibles:
(>>> 15.20.1 "Condiciones en la RoboticsAPI" Página 323)

Para establecer una condición de cancelación para un movimiento, se transmite a la instrucción de movimiento un objeto del tipo de condición deseado mediante el método de movimiento breakWhen(...).

`breakWhen(...)` se puede activar varias veces para la programación de una instrucción de movimiento, para establecer diferentes condiciones de cancelación para un movimiento. En este caso, las condiciones de cancelación individuales se enlazan con un O lógico.

Los siguientes puntos se deben tener en cuenta para la programación de condiciones de cancelación:

- Para un bloque Spline solo se pueden programar condiciones de cancelación para el bloque Spline completo. Las condiciones de cancelación para segmentos Spline individuales no están permitidas.
- Si se produce una condición de cancelación que está definida para un movimiento dentro de un MotionBatch, dicho movimiento se cancelará y seguidamente se ejecutará la siguiente instrucción de movimiento en el Batch. Si se produce una condición de cancelación para el definida para el MotionBatch completo, se cancelará el MotionBatch completo.
- Una condición de cancelación provoca la cancelación del movimiento que se está ejecutando actualmente. Los movimientos siguientes se ejecutarán inmediatamente después del movimiento cancelado, siempre que no se haya programado una estrategia de reacción adecuada en la aplicación.
- En los movimientos de posicionamiento aproximado, el arco de aproximación pertenece a la trayectoria del movimiento siguiente. Por ello, sobre el arco de aproximación solo actúan las condiciones de cancelación para el movimiento siguiente.
- Si la condición de cancelación se produce en un movimiento de posicionamiento aproximado poco antes de alcanzar el punto de posicionamiento aproximado y, como consecuencia, no se detiene hasta el arco de aproximación, el robot se acelerará de nuevo al alcanzar el arco de aproximación para ejecutar el movimiento siguiente.

Sintaxis

```
motion.breakWhen( condition_1<, condition_2, ... > );
```

Aclaración de la sintaxis

Elemento	Descripción
<code>motion</code>	Tipo: Motion Movimiento para el que se debe establecer una condición de cancelación. Ejemplo: <ul style="list-style-type: none">■ <code>ptpgetApplicationData().getFrame("/P1")</code>
<code>condition</code>	Tipo: ICondition Objeto ICondition parametrizado que describe una condición de cancelación

Ejemplo

Un movimiento LIN se cancela si el momento en el eje A3 es menor o igual que -12 Nm o mayor o igual que 0 Nm.

```
JointTorqueCondition cond_1 = new JointTorqueCondition(JointEnum.J3,
-12.0, 0.0);
robot.move(lin(getApplicationData().getFrame("/P10"))
.breakWhen(cond_1));
```

15.21.2 Evaluar condiciones de cancelación

Descripción

Si se han fijado condiciones de cancelación para una instrucción de movimiento, se pueden consultar distintas informaciones relativas a la cancelación de un movimiento. Para ello, la instrucción de movimiento se guarda de forma transitoria en una variable IMotionContainer. Desde esta variable, a través del

método `getFiredBreakConditionInfo()` se puede consultar un objeto del tipo `IFiredConditionInfo` que contenga la información para la cancelación del movimiento. Si durante el movimiento no se produce ninguna condición de cancelación, `getFiredBreakConditionInfo()` enviará `null` como respuesta.

Sintaxis

```
IMotionContainer motionCmd = motion.breakWhen(...);  
IFiredConditionInfo firedCondInfo =  
    motionCmd.getFiredBreakConditionInfo();
```

Aclaración de la sintaxis

Elemento	Descripción
<code>motion</code>	Instrucciones de movimiento Ejemplo: ■ <code>lbr.move(ptpgetApplicationData().getFrame("/P1"))</code>
<code>motionCmd</code>	Tipo: <code>IMotionContainer</code> Memoria intermedia para la instrucción de movimiento
<code>firedCondInfo</code>	Tipo: <code>IFiredConditionInfo</code> Información sobre la cancelación del movimiento

Resumen

La interfaz `IFiredConditionInfo` ofrece los siguientes métodos:

Método	Descripción
<code>getFiredCondition()</code>	Tipo de retorno: <code>ICondition</code> Consulta de la condición que ha provocado la cancelación de un movimiento
<code>getPositionInfo()</code>	Tipo de retorno: <code>PositionInformation</code> Consulta de la posición del robot en el momento de la activación de la condición de cancelación
<code>getStoppedMotion()</code>	Tipo de retorno: <code>IMotion</code> Consulta de qué segmento de un bloque Spline o qué movimiento de un MotionBatch se ha cancelado

15.21.2.1 Consultar condición de cancelación**Descripción**

La condición que ha causado la cancelación de un movimiento, se puede consultar mediante el método `getFiredCondition()`. El valor de retorno es del tipo `ICondition` y se puede comparar con las condiciones de cancelación transmitidas mediante el método `equals(...)`.

La consulta resulta especialmente útil, cuando por la activación repetida del método `breakWhen(...)` se han definido varias condiciones de cancelación para un movimiento.

Sintaxis

```
ICondition firedCondition = firedCondInfo.getFiredCondition();
```

Aclaración de la sintaxis

Elemento	Descripción
<code>firedCondition</code>	Tipo: <code>ICondition</code> Variable para el valor de retorno. La variable contiene la condición que ha provocado la cancelación del movimiento.
<code>firedCondInfo</code>	Tipo: <code>IFiredConditionInfo</code> Información sobre la cancelación del movimiento

Ejemplo

Se generan las condiciones de cancelación "cond1" y "cond2".

```
ICondition cond1;
ICondition cond2;
cond1 = new ...;
cond2 = new ...;
```

Las condiciones de cancelación "cond1" y "cond2" se transmiten a un movimiento LIN con breakWhen(...). Mediante la variable "motionCmd" del tipo IMotionContainer se puede evaluar la instrucción de movimiento.

```
IMotionContainer motionCmd =
lbr.move(lingetApplicationData().getFrame("P10")).breakWhen(cond1).b
reakWhen(cond2);
```

La información relativa a la cancelación del movimiento se le consulta a "motionCmd". Si la información consultada no es igual a null, el movimiento se ha cancelado. Solo en este caso se consulta la condición de cancelación que se ha activado.

```
IFiredConditionInfo firedInfo = motionCmd.getFiredConditionInfo();
if(firedInfo != null){

    ICondition firedCond = firedInfo.getFiredCondition();
    if(firedCond.equals(cond1)){
        ...
    }
    ...
}
```

15.21.2 Consultar la posición del robot en el momento de cancelación

Descripción La posición del robot en el momento de la activación de la condición de cancelación se puede consultar con el método getPositionInfo().

Mediante el valor de retorno del tipo Información de posición se puede acceder a la siguiente información de posición:

- Posición real específica del eje
- Posición real cartesiana
- Posición nominal específica del eje
- Posición nominal cartesiana
- Diferencia nominal-real (translatorio)
- Diferencia nominal-real (rotatorio)

Sintaxis Información de posición firedPosInfo =
firedCondInfo.getPositionInfo();

Aclaración de la sintaxis	Elemento	Descripción
	firedPosInfo	Tipo: PositionInformation Variable para el valor de retorno. El valor de retorno contiene la información de posición en el momento de la activación de la condición de cancelación.
	firedCondInfo	Tipo: IFiredConditionInfo Información sobre la cancelación del movimiento

Ejemplo La posición real cartesiana del robot al producirse la condición de cancelación se consulta con el método getCurrentCartesianPosition().

```
PositionInformation firedPosInfo = firedInfo.getPositionInfo();
Frame firedCurrPos = firedPosInfo.getCurrentCartesianPosition();
```

15.21.2.3 Consultar movimiento interrumpido (bloque Spline, MotionBatch)

Descripción	Las condiciones de terminación pueden estar definidas para un bloque Spline o MotionBatch completo. Si se da una condición de terminación, se interrumpe el bloque Spline o MotionBatch completo.						
	A través del método getStoppedMotion() se puede consultar qué segmento Spline o qué movimiento de un MotionBatch se ha interrumpido. El valor de retorno es del tipo IMotion.						
Sintaxis	<pre>IMotion stoppedMotion = firedCondInfo.getStoppedMotion();</pre>						
Aclaración de la sintaxis	<table border="1"> <thead> <tr> <th>Elemento</th><th>Descripción</th></tr> </thead> <tbody> <tr> <td><i>stoppedMotion</i></td><td>Tipo: IMotion Variable para el valor de retorno. La variable contiene el movimiento cancelado.</td></tr> <tr> <td><i>firedCondInfo</i></td><td>Tipo: IFiredConditionInfo Información sobre la cancelación del movimiento</td></tr> </tbody> </table>	Elemento	Descripción	<i>stoppedMotion</i>	Tipo: IMotion Variable para el valor de retorno. La variable contiene el movimiento cancelado.	<i>firedCondInfo</i>	Tipo: IFiredConditionInfo Información sobre la cancelación del movimiento
Elemento	Descripción						
<i>stoppedMotion</i>	Tipo: IMotion Variable para el valor de retorno. La variable contiene el movimiento cancelado.						
<i>firedCondInfo</i>	Tipo: IFiredConditionInfo Información sobre la cancelación del movimiento						

Ejemplo Consulta en el ejemplo de un bloque Spline:

```
ICondition stopCondition = new ...;
...
Spline splineMotion = new Spline(
    spl(getApplicationData().getFrame("/P1")),
    circ(getApplicationData().getFrame("/P2"),
       getApplicationData().getFrame("/P3")),
    spl(getApplicationData().getFrame("/P4")).setCartVelocity(150),
    lin(getApplicationData().getFrame("/P5"))
).setCartVelocity(250).breakWhen(stopCondition);

IMotionContainer splineCont = robot.move(splineMotion);

IFiredConditionInfo firedInfoSpline =
splineCont.getFiredConditionInfo();
if(firedInfoSpline != null) {
    IMotion stoppedMotion = firedInfoSpline.getStoppedMotion();
    ...
}
```

15.22 Acciones de conmutación referidas a la trayectoria (Trigger)

Un Trigger es una incidencia que se utiliza como activador para acciones referidas a la trayectoria y definidas por el usuario. Si durante la ejecución de un movimiento se produce una incidencia determinada, se activará la acción. La acción se ejecuta de forma paralela al movimiento del robot. Por ejemplo, para un movimiento de aproximación se debe abrir la garra a tiempo, de forma que esté abierta cuando se haya alcanzado una posición de descarga libre para la pieza transportada.

15.22.1 Programar un Trigger

Descripción	Las incidencias que activan una acción de conmutación referida a la trayectoria se denominan como Trigger. Las incidencias se definen con la ayuda de condiciones. Una incidencia se produce si la condición definida ya tiene el estado TRUE antes del inicio del movimiento o si cambia al estado TRUE durante el movimiento.
--------------------	---

Las condiciones están definidas como objetos del tipo ICondition. Los tipos de condiciones disponibles pertenecen al paquete com.kuka.roboticsAPI.conditionModel.

Aquí se puede encontrar un resumen de los tipos de condiciones disponibles:
 (>>> 15.20.1 "Condiciones en la RoboticsAPI" Página 323)

Para programar un Trigger se entrega a una instrucción de movimiento con el método Motion triggerWhen(...) un objeto del tipo de condición deseado y un objeto ITriggerAction que describe la acción que se va a ejecutar.

triggerWhen(...) se puede activar varias veces para la programación de una instrucción de movimiento, para establecer diferentes Trigger para un movimiento. La ejecución de las acciones de conmutación correspondientes solo depende de si se produce la incidencia activadora y no es influenciada por el orden de la activación con triggerWhen(...).



Mientras se está realizando una acción, el evento que la está causando no puede provocar nuevamente esta acción. El trigger no está activo hasta que no haya finalizado el método triggerWhen(...). Se puede consultar el número de eventos que no se pudieron realizar durante la ejecución del método.

(>>> 15.22.3 "Evaluar información del Trigger" Página 343)

Sintaxis

`motion.triggerWhen(condition, action);`

Aclaración de la sintaxis

Elemento	Descripción
<code>motion</code>	<p>Tipo: Motion</p> <p>Movimiento para el que se debe establecer un Trigger.</p> <p>Ejemplo:</p> <ul style="list-style-type: none"> ■ <code>ptpgetApplicationData().getFrame("/P1")</code>
<code>condition</code>	<p>Tipo: ICondition</p> <p>Objeto ICondition parametrizado que describe la condición para el Trigger</p>
<code>action</code>	<p>Tipo: ITriggerAction</p> <p>Objeto ITriggerAction que describe la acción que se va a ejecutar</p> <p>(>>> 15.22.2 "Programar acción de conmutación referida a la trayectoria" Página 341)</p>

15.22.2 Programar acción de conmutación referida a la trayectoria

Descripción

La acción que debe ejecutarse referida a la trayectoria al producirse una incidencia, se establece mediante un objeto ITriggerAction. ITriggerAction es una interfaz del paquete com.kuka.roboticsAPI.conditionModel. Esta interfaz no ofrece ningún método actualmente.

Para la programación de las acciones se puede utilizar la interfaz ICallbackAction derivada de ITriggerAction. La interfaz posee el método onTriggerFired(...). En la estructura del método de onTriggerFired(...), se puede programar la acción que debe ejecutarse al activarse el Trigger.

Un objeto ICallbackAction se puede utilizar en el número deseado de Triggers.



El método `onTriggerFired(...)` no se activa en tiempo real. Por ello, no se puede garantizar un comportamiento temporal concreto. Puede suceder que la acción se ejecute con retardo.

Sintaxis

```
ICallbackAction action = new ICallbackAction() {
    @Override
    public void onTriggerFired(IFiredTriggerInfo
        triggerInformation) {
        // Acción que se ejecuta
    }
};
```

Aclaración de la sintaxis

Elemento	Descripción
<i>action</i>	Tipo: ICallbackAction Objeto ICallbackAction que describe la acción que se transmite con <code>triggerWhen(...)</code>
<code>onTriggerFired(...)</code>	Método cuya ejecución se activa por el Trigger
<code>triggerInformation</code>	Tipo: IFiredTriggerInfo Contiene información sobre el Trigger activador (>>> 15.22.3 "Evaluar información del Trigger" Página 343)

Ejemplo

Durante el movimiento hacia el punto "P1", la salida "DO1" se conmuta siempre que la entrada "DI1" sea TRUE.

```
//set trigger action
ICallbackAction toggleOut_1 = new ICallbackAction() {

    @Override
    public void onTriggerFired(IFiredTriggerInfo triggerInformation) {
        //toggle output state when trigger fired
        if(IOs.getDO1())
        {
            IOs.setDO1(false);
        }
        else
        {
            IOs.setDO1(true);
        }

    }
};

//set trigger condition
BooleanIOCondition buttonPressed = new
BooleanIOCondition(IOs.getInput("DI1"), true);

//motion with trigger
robot.move(ptp(P1)).triggerWhen(buttonPressed, toggleOut_1));
robot.move(ptp(P2));
```

15.22.3 Evaluar información del Trigger

Al activar un Trigger se activa el método `onTriggerFired(...)`. Al método `onTriggerFired(...)` se le transmite el objeto `triggerInformation` del tipo `IFiredTriggerInfo`, que contiene información variada sobre el Trigger activado. Esta información del Trigger se puede consultar.

Resumen Se encuentran disponibles los siguientes métodos de la clase `IFiredTriggerInfo`:

Método	Descripción
<code>getFiredCondition()</code>	Tipo de retorno: <code>ICondition</code> Consulta de la condición que ha activado al Trigger
<code>getMissedEvents()</code>	Tipo de retorno: <code>int</code> Consulta del número de veces que aún se ha producido la incidencia que ha activado al Trigger, mientras que se ejecuta la acción activada por el Trigger Nota: Mientras se está realizando una acción, el evento que la está causando no puede provocar nuevamente esta acción.
<code>getMotionContainer()</code>	Tipo de retorno: <code>IMotionContainer</code> Consulta de la instrucción de movimiento con cuya ejecución se ha activado el Trigger
<code>getPositionInformation()</code>	Tipo de retorno: <code>PositionInformation</code> Consulta de la información de posición en el momento de la activación del Trigger El valor de retorno contiene la siguiente información de posición: <ul style="list-style-type: none"> ■ Posición real específica del eje ■ Posición real cartesiana ■ Posición nominal específica del eje ■ Posición nominal cartesiana ■ Diferencia nominal-real (translatorio) ■ Diferencia nominal-real (rotatorio)
<code>getTriggerTime()</code>	Tipo de retorno: <code>java.util.Date</code> Consulta del momento en el que se ha activado el Trigger

Para consultar la información de posición recibida con `getPositionInformation()`, se encuentran disponibles los siguientes métodos de la clase `Information de posición`:

Método	Descripción
<code>getCommandedCartesianPosition()</code>	Tipo de retorno: <code>Frame</code> Consulta de la posición nominal cartesiana en el momento del Trigger
<code>getCommandedJointPosition()</code>	Tipo de retorno: <code>JointPosition</code> Consulta de la posición nominal específica del eje en el momento del Trigger

Método	Descripción
getCurrentCartesianPosition()	Tipo de retorno: Frame Consulta de la posición real cartesiana en el momento del Trigger
getCurrentJointPosition()	Tipo de retorno: JointPosition Consulta de la posición real específica del eje en el momento del Trigger

Ejemplo 1

Cuando se activa el Trigger, se indican el momento de activación y la condición de activación en la smartHMI.

```
BooleanIOCondition in1 = new BooleanIOCondition(_input_1, true);

ICallbackAction ica = new ICallbackAction() {

    @Override
    public void onTriggerFired(IFiredTriggerInfo triggerInformation) {
        getLogger().info("TriggerTime: " +
        triggerInformation.getTriggerTime().toString());
        getLogger().info("TriggerCondition: " +
        triggerInformation.getFiredCondition().toString());
    }
};

robot.move(P1).triggerWhen(in1, ica));
```

Ejemplo 2

Se consultan la posición del robot específica del eje y cartesiana en el momento del Trigger.

```
BooleanIOCondition in1 = new BooleanIOCondition(_input_1, true);

ICallbackAction ica = new ICallbackAction() {

    @Override
    public void onTriggerFired(IFiredTriggerInfo triggerInformation) {
        PositionInformation posInfo =
        triggerInformation.getPositionInformation();
        posInfo.getCommandedCartesianPosition();
        posInfo.getCommandedJointPosition();
        posInfo.getCurrentCartesianPosition();
        posInfo.getCommandedJointPosition();
    }
};

robot.move(P1).triggerWhen(in1, ica));
```

15.23 Control de procesos (Monitoring)

Monitoring significa controlar un proceso con la ayuda de un Listener para poder reaccionar ante determinadas incidencias durante la ejecución de una aplicación.

Estas incidencias son modificaciones de estado de condiciones definidas. El Listener controla el estado de la condición. Si cambia el estado de la condición, se informará al Listener y se activará la rutina de tratamiento establecida como reacción.

El Listener no es informado durante la ejecución de una rutina de tratamiento si surgen más eventos. Si se ha finalizado la rutina de tratamiento, estos sucesos solo se le comunican al Listener y se tratan si se ha seleccionado el tipo de comunicación adecuado.

(>>> 15.23.3 "Registrar el Listener para la notificación en caso de modificación del estado" Página 347)

15.23.1 Listener para el control de condiciones

Para el control de una condición se encuentran disponibles diferentes interfaces Listener del paquete com.kuka.roboticsAPI.conditionModel. Los Listener se diferencian por tipos en que son informados respectivamente con una determinada modificación del estado de una condición controlada.

Cada tipo de Listener declara un método que se ejecuta cuando el Listener es informado. En la estructura del método de este método se puede programar la rutina de tratamiento deseada.

Tipo de datos	Descripción
IRisingEdgeListener	<p>Notificación cuando se cumple la condición controlada (flanco ascendente, modificación del estado FALSE > TRUE).</p> <p>Método para la rutina de tratamiento:</p> <ul style="list-style-type: none"> ■ onRisingEdge(...)
IFallingEdgeListener	<p>Notificación cuando deja de cumplirse la condición controlada (flanco descendente, modificación del estado TRUE > FALSE).</p> <p>Método para la rutina de tratamiento:</p> <ul style="list-style-type: none"> ■ onFallingEdge(...)
IAnyEdgeListener	<p>Notificación con cada modificación del estado de la condición (flanco ascendente o descendente, modificación del estado FALSE > TRUE o TRUE > FALSE).</p> <p>Método para la rutina de tratamiento:</p> <ul style="list-style-type: none"> ■ onAnyEdge(...)

Resumen

Para poder reaccionar ante la modificación del estado de una condición, son necesarios los siguientes pasos durante la programación:

Paso	Descripción
1	<p>Crear un objeto de Listener para el control de la condición.</p> <p>(>>> 15.23.2 "Crear objeto de Listener para el control de una condición" Página 346)</p>
2	Programar la rutina de tratamiento deseada en el método Listener.
3	<p>Registrar el Listener para la notificación en caso de modificación del estado de una condición.</p> <p>(>>> 15.23.3 "Registrar el Listener para la notificación en caso de modificación del estado" Página 347)</p>
4	<p>Si no se ha realizado ya mediante el método seleccionado para el registro, activar el servicio de notificación para el Listener.</p> <p>(>>> 15.23.4 "Activar o desactivar el servicio de notificación para el Listener" Página 348)</p>

15.23.2 Crear objeto de Listener para el control de una condición

Descripción

Aquí se describe la sintaxis de un objeto de Listener en el ejemplo del Listener IAnyEdgeListener. El método de Listener onAnyEdge(...) declarado automáticamente en la creación del objeto posee parámetros de entrada. Estos parámetros de entrada contienen información sobre la incidencia que ha activado el método y se pueden consultar y evaluar.

Los objetos de Listener de los otros tipos de Listener se generan del mismo modo y tienen una estructura análoga.

Sintaxis

```
IAnyEdgeListener condListener = new IAnyEdgeListener() {
    @Override
    public void onAnyEdge(ConditionObserver conditionObserver, Date time, int missedEvents, boolean conditionValue) {
        // Reacción ante una modificación del estado
    }
};
```

Aclaración de la sintaxis

Elemento	Descripción
condListener	Tipo: IAnyEdgeListener Nombre del objeto de Listener
Parámetros de entrada del método Listener:	
condition Observer	Tipo: ConditionObserver Objeto que ha informado al listener
time	Tipo: Date Fecha y hora en la que el Listener ha sido informado
missed Events	Tipo: int Número de los cambios de estado que no se han tratado. Posibles causas de los eventos no tratados: <ul style="list-style-type: none"> ■ El servicio de notificación estaba desactivado cuando se produjo la incidencia de activación. ■ La rutina de tratamiento se ha ejecutado justo en el momento en que se ha producido de nuevo la incidencia de activación. Estos eventos pueden tratarse empleando el tipo de notificación NotificationType.MissedEvents . (>>> "NotificationType" Página 347)
condition Value	Tipo: boolean Solo está disponible con el método de Listener onAnyEdge(...). Indica el flanco por el que ha sido activado el método. <ul style="list-style-type: none"> ■ true: Flanco ascendente (modificación de estado FALSE > TRUE) ■ false: Flanco descendente (modificación de estado TRUE > FALSE)

15.23.3 Registrar el Listener para la notificación en caso de modificación del estado

Descripción Para registrar un Listener para la notificación en caso de modificación del estado de una condición, es necesario un objeto del tipo ConditionObserver.

Para la generación de un objeto del tipo ConditionObserver, primero se deberá consultar al ObserverManager de la aplicación sobre el método getObserverManager(). La clase ObserverManager ofrece diferentes métodos para generar el objeto necesario.

- `createAndEnableConditionObserver(...)`
El servicio de notificación para el Listener está activo de forma inmediata.

- `createConditionObserver(...)`

El servicio de notificación para el Listener no está activo de forma inmediata, sino que primero se debe activar.

(**>>> 15.23.4 "Activar o desactivar el servicio de notificación para el Listener"** Página 348)

Los parámetros transmitidos respectivamente son idénticos para ambos métodos.

Sintaxis

```
ConditionObserver myObserver =
getObserverManager().createAndEnableConditionObserver
(condition, notificationType, listener)
```

Aclaración de la sintaxis

Elemento	Descripción
<i>myObserver</i>	Tipo: ConditionObserver Objeto que controla la condición definida
<i>condition</i>	Tipo: ICondition Condición que se controla
<i>notification Type</i>	Tipo: Enum del tipo NotificationType Tipo de notificación Determina en qué incidencias se debe informar al Listener para ejecutar la rutina de tratamiento deseada. (>>> "NotificationType" Página 347)
<i>listener</i>	Tipo: IRisingEdgeListener, IFallingEdgeListener o IAnyEdgeListener Objeto del Listener que se registra

NotificationType

El Enum del tipo NotificationType posee los siguientes valores:

Valor	Descripción
EdgesOnly	El Listener solo es informado en caso de un cambio de flanco (de acuerdo con el tipo de Listener utilizado).
OnEnable	El Listener es informado en caso de un cambio de flanco (de acuerdo con el tipo de Listener utilizado). Adicionalmente se comprueba el estado de la condición controlada en la activación del Listener. En función del tipo de Listener, éste será informado si se producen las siguientes incidencias: <ul style="list-style-type: none"> ■ IRisingEdgeListener: Solo si se cumple la condición en la activación ■ IFallingEdgeListener: Solo si no se cumple la condición en la activación ■ IAnyEdgeListener: Si la condición se cumple o no se cumple en la activación
MissedEvents	El Listener es informado en caso de un cambio de flanco (de acuerdo con el tipo de Listener utilizado). Adicionalmente se informa al Listener después de la ejecución de la rutina de tratamiento si se han atendido las incidencias activadas. Esto significa: Si el cambio de flanco activador se produce de nuevo durante la ejecución de la rutina de tratamiento, también se volverá a informar al Listener y la rutina de tratamiento se ejecutará una segunda vez.
All	Combinación de OnEnable y MissedEvents El Listener es informado cuando se producen todas las incidencias que se describen en OnEnable y MissedEvents.

15.23.4 Activar o desactivar el servicio de notificación para el Listener

Descripción Los métodos para activar o desactivar el servicio de notificación pertenecen a la clase ConditionObserver.
Solo es necesario activar el servicio de notificación si se ha utilizado el método createConditionObserver(...) para el registro del Listener.

Sintaxis Activar el servicio de notificación:

myObserver.enable()

Desactivar el servicio de notificación:

myObserver.disable()

Aclaración de la sintaxis

Elemento	Descripción
<i>myObserver</i>	Tipo: ConditionObserver Objeto que controla la condición definida

15.23.5 Ejemplo de programación de control

Para controlar una condición de fuerza, se define un Listener del tipo IRisingEdgeListener. En cuanto en el TCP se supera una fuerza de 35 N, esto se considera una colisión. Se avisa al Listener y se enciende una luz de advertencia.

Como tipo de notificación se ha ajustado **NotificationType.MissedEvents**. Si la fuerza admisible en el TCP se supera repetidas veces mientras la luz de advertencia está encendida, se informa al Listener sin demora.

```
ForceCondition collision = ForceCondition
.createSpatialForceCondition(tool.getDefaultMotionFrame(), 35);

IRisingEdgeListener collisionListener = new IRisingEdgeListener() {

    @Override
    public void onRisingEdge(ConditionObserver conditionObserver,
    Date time, int missedEvents) {

        signals.setWarningLED(true);

    }
});

ConditionObserver collisionObserver = getObserverManager()
.createConditionObserver(collision, NotificationType.MissedEvents,
collisionListener);
collisionObserver.enable();
```

15.24 Espera con bloqueo de la condición

Descripción

Con `waitFor(...)` se detiene una aplicación hasta que se cumple una condición determinada o ha finalizado un tiempo de espera determinado. Después, la aplicación continúa.



Si se procesa la instrucción pueden surgir tiempos de espera. No se puede garantizar que se cumpla exactamente el tiempo de espera programado.

`waitFor(...)` debe acceder al `ObserverManager` de la aplicación. Se activa con `getObserverManager()`.

Se soportan todos los tipos de condiciones excepto `MotionPathCondition`.

Aquí se puede encontrar un resumen de los tipos de condiciones disponibles:
[\(>>> 15.20.1 "Condiciones en la RoboticsAPI" Página 323\)](#)

Sintaxis

Espera con bloqueo sin limitación temporal:

```
getObserverManager().waitFor(condition)
```

Espera con bloqueo con limitación temporal:

```
boolean result = getObserverManager().waitFor(condition, timeout,
timeUnit)
```

Aclaración de la sintaxis

Elemento	Descripción
<i>condition</i>	Tipo: <code>ICondition</code> Condición a la que se espera. Si al activar <code>waitFor(...)</code> la condición ya se ha cumplido, la aplicación continúa de forma inmediata.
<i>timeout</i>	Tipo: <code>long</code> Máximo tiempo de espera: Si la condición no se cumple en el tiempo de espera definido, también continuará la aplicación sin que se cumpla la condición.

Elemento	Descripción
<i>timeUnit</i>	<p>Tipo: Enum del tipo TimeUnit</p> <p>Unidad del tiempo de espera indicado. El Enum se incluye por defecto en las bibliotecas de Java.</p>
<i>result</i>	<p>Tipo: boolean</p> <p>Variable para el valor de retorno de <code>waitFor(...)</code>. El valor de retorno es true si la condición se cumple dentro del tiempo de espera indicado.</p> <p>Nota: Si no se define ningún tiempo de espera, <code>waitFor(...)</code> no envía ningún valor de retorno.</p>

Ejemplo

En la aplicación se debe esperar a una señal de entrada booleana. Para ello, la aplicación debe estar bloqueada como máximo 30 segundos. Si la señal de entrada no se envía en este tiempo, a continuación se deberá ejecutar una determinada rutina de tratamiento.

```
SwitchIOGroup inputs = new SwitchIOGroup(kuka_Sunrise_Cabinet);
Input input = inputs.getInput ("Input");

BooleanIOCondition inputCondition = new BooleanIOCondition(input,
true);

boolean result = getObserverManager().waitFor(inputCondition, 30,
TimeUnit.SECONDS);

if(!result){
    //do something
}
else{
    //continue program
}
```

15.25 Registrar y evaluar datos

Durante la ejecución de una aplicación se pueden registrar determinados datos, p. ej., fuerzas y momentos externos, para una evaluación posterior. Para la programación del registro de datos está disponible la clase DataRecorder (paquete: com.kuka.roboticsAPI.sensorModel).

Los datos registrados se guardan en un archivo y se almacenan en la unidad de control del robot en el directorio C:\KRC\Roboter\Log\DataRecorder.

El nombre del archivo se establece con el objeto DataRecorder que se crea. Si se produce un error durante el registro, el nombre del archivo comenzará con "FaultyDataRecorder...".

El archivo se puede abrir con un editor de textos o puede leerse en una tabla Excel.

15.25.1 Crear objeto para el registro de datos

Descripción

Para el registro de datos primero se debe crear y parametrizar un objeto del tipo DataRecorder. Si para ello se utiliza un constructor estándar, estarán ajustados los siguientes parámetros por defecto:

- El nombre del archivo en el que se guardan los datos registrados se crea automáticamente y contiene una ID especificada de forma interna por el sistema: `DataRecorderID.log`

- No se encuentra establecida ninguna duración del registro. Se registrarán datos hasta que el buffer (actualmente 16 MB) esté lleno o se haya alcanzado el número máximo de juego de datos (actualmente 30 000).
- La tasa de registro, es decir, distancia temporal mínima entre 2 registros, es de 1 ms.

Sintaxis del constructor

La clase DataRecorder posee los siguientes constructores:

`DataRecorder()` (constructor estándar)

`DataRecorder(String fileName, long timeout, TimeUnit timeUnit, int sampleRate)`

Aclaración de la sintaxis

Elemento	Descripción
<code>fileName</code>	Nombre del archivo (con extensión) con el que se guardan los datos registrados Ejemplo: "Registro_1.log"
<code>timeout</code>	Duración del registro <ul style="list-style-type: none"> ■ -1: No se encuentra establecida ninguna duración del registro. ■ ≥ 1 Por defecto: -1 La unidad de tiempo se establece con <code>timeUnit</code> .
<code>timeUnit</code>	Unidad de tiempo para la duración del registro Ejemplo: <code>TimeUnit.SECONDS</code> El Enum del tipo <code>TimeUnit</code> se incluye por defecto en las bibliotecas de Java.
<code>sampleRate</code>	Tasa de registro (unidad: ms) <ul style="list-style-type: none"> ■ ≥ 1 Por defecto: 1



La clase DataRecorder ofrece métodos `set` con los que se pueden adaptar los valores de parámetros, especialmente en caso de utilización del constructor estándar.

- `setFileName(...), setSampleRate(...), setTimeout(..., ...)`

En `setTimeout(..., ...)` se ajusta con el primer parámetro la duración del registro y con el segundo parámetro la unidad de tiempo correspondiente.

Ejemplo 1

Se deben registrar datos durante 5 s en un ciclo de 100 ms y escribirse en el archivo Recording_1.log.

```
DataRecorder rec_1 = new DataRecorder("Recording_1.log", 5,
TimeUnit.SECONDS, 100);
```

Ejemplo 2

El objeto DataRecorder se crea con el constructor estándar. De este modo solo se establece que en el ciclo de 1 ms se registren datos durante un tiempo indeterminado. Los datos registrados se deben escribir en el archivo Recording_2.log. Para ello se establece el nombre del archivo con el método `set` correspondiente.

```
DataRecorder rec_2 = new DataRecorder();
rec_2.setFileName("Recording_2.log");
```

15.25.2 Establecer datos para el registro

Los datos que deben registrarse se añaden con la ayuda del operador del punto y con los métodos add correspondientes en el objeto DataRecorder creado para ello. Es posible el registro simultáneo de diferentes datos.

Resumen

Se encuentran disponibles los siguientes métodos add de la clase DataRecorder:

Método	Descripción
addInternalJointTorque(...)	Tipo de retorno: DataRecorder Registro de los momentos del eje medidos del robot transmitido como parámetro (tipo: Robot)
addExternalJointTorque(...)	Tipo de retorno: DataRecorder Registro de los momentos del eje externos (sin modelo) del robot transmitido como parámetro (tipo: Robot)
addCartesianForce(...)	Tipo de retorno: DataRecorder Registro de las fuerzas cartesianas a lo largo del eje X, Y y Z del Frame transmitido como parámetro (unidad: N). La varianza de las fuerzas cartesianas también se registra. Opcionalmente se puede transmitir un segundo Frame como parámetro, para establecer la orientación para la medición de fuerza. Si no se indica ningún Frame propio para la orientación, se deberá transmitir <i>cero</i> .
addCartesianTorque(...)	Tipo de retorno: DataRecorder Registro de los momentos cartesianos a lo largo del eje X, Y y Z del Frame transmitido como parámetro (unidad: Nm). La varianza de los momentos cartesianos también se registra. Opcionalmente se puede transmitir un segundo Frame como parámetro, para establecer la orientación para la medición del momento. Si no se indica ningún Frame propio para la orientación, se deberá transmitir <i>cero</i> .
	Parámetro: <ul style="list-style-type: none"> ■ <code>AbstractFrame measureFrame</code> Frame conectado con la brida del robot, p. ej. el TCP de una herramienta. Define la posición del punto de medición. ■ <code>AbstractFrame orientationFrame</code> Define la orientación del punto de medición. Indicación: Deben transmitirse siempre los dos parámetros. La orientación puede ser <i>cero</i> .
addCommandedJointPosition(...)	Tipo de retorno: DataRecorder Registro de la posición nominal específica del eje del robot transmitido como parámetro (tipo: Robot). Como segundo parámetro se debe transmitir la unidad en la que se registran los ángulos del eje (Enum del tipo: AngleUnit).
addCurrentJointPosition(...)	Tipo de retorno: DataRecorder Registro de la posición real específica del eje del robot transmitido como parámetro (tipo: Robot). Como segundo parámetro se debe transmitir la unidad en la que se registran los ángulos del eje (Enum del tipo: AngleUnit).

Método	Descripción
	<p>Parámetro:</p> <ul style="list-style-type: none"> ■ Robot <i>robot</i> ■ AngleUnit <i>angleUnit</i> <ul style="list-style-type: none"> ■ AngleUnit.Degree: Ángulo del eje en grados ■ AngleUnit.Radian: Ángulo del eje en radianes
addCommandedCartesianPositionXYZ(...)	<p>Tipo de retorno: DataRecorder</p> <p>Registro de la posición nominal cartesiana (parte translatoria)</p> <p>Se transmiten como parámetros el punto de medición y el sistema de coordenadas de referencia, en relación con el que se registra la posición.</p>
addCurrentCartesianPositionXYZ(...)	<p>Tipo de retorno: DataRecorder</p> <p>Registro de la posición real cartesiana (parte translatoria)</p> <p>Se transmiten como parámetros el punto de medición y el sistema de coordenadas de referencia, en relación con el que se registra la posición.</p>

Ejemplo

Para un LBR iiwa se deben registrar los siguientes datos con la ayuda de un objeto DataRecorder:

- Momentos del eje que se miden en el robot
- Fuerza en el TCP de una garra montada en el robot con la orientación de un Frame base

```
private LBR lbr_iiwa;
private Tool gripper;
...
gripper.attachTo(lbr_iiwa.getFlange());

DataRecorder rec = new DataRecorder();
rec.addInternalJointTorque(lbr_iiwa);
rec.addCartesianForce(gripper.getFrame("TCP"),
getApplicationData().getFrame("/Base"));
```

15.25.3 Iniciar el registro de datos

El registro de datos se puede iniciar de forma independiente con respecto al movimiento (en cualquier lugar de la aplicación), o de forma sincronizada con el movimiento, activado por un Trigger.

independiente con respecto al movimiento

Antes del inicio del registro independiente con respecto al movimiento, se deberá activar el objeto DataRecorder mediante el método enable(). El registro se inicia mediante el método startRecording().

Cuando el registro ha finalizado, se desactiva automáticamente el objeto DataRecorder. Si se deben registrar datos una vez más con el mismo objeto DataRecorder, se deberá activar de nuevo el objeto DataRecorder.



No pueden estar activados al mismo tiempo varios objetos DataRecorder.

sincronizado a través de Trigger

Para un Trigger se debe formular una condición del tipo ICondition que si se cumple, activará el Trigger y una acción que se ejecutará al activarse el Trigger.

(>>> 15.22.1 "Programar un Trigger" Página 340)

La acción a ejecutar es el inicio del registro de datos. Para ello se debe transmitir un objeto del tipo StartRecordingAction. Al crear el objeto, se deberá indicar el objeto DataRecorder que se utiliza para el registro de datos.

Sintaxis del constructor:

```
StartRecordingAction(DataRecorder recorder)
```

A continuación, el objeto ICondition y el objeto StartRecordingAction se enlazan a una instrucción de movimiento con triggerWhen(...).

Ejemplo 1

El registro de datos se debe iniciar cuando el robot ha ejecutado el movimiento de desplazamiento hasta una posición previa. El objeto DataRecorder se activa antes de que se realice el desplazamiento hasta la posición previa, para reducir el retardo temporal que se produce con el inicio del registro.

```
private LBR lbr_iiwa;  
...  
DataRecorder rec = new DataRecorder();  
...  
rec.enable();  
...  
lbr_iiwa.move(lingetApplicationData().getFrame("/Pre-position"));  
rec.startRecording();
```

Ejemplo 2

El registro de datos debe comenzar 2 s después del inicio de un movimiento. Para ello se parametriza un objeto MotionPathCondition.

```
private LBR lbr_iiwa;  
...  
DataRecorder rec = new DataRecorder();  
...  
StartRecordingAction startAct = new StartRecordingAction(rec);  
MotionPathCondition startCond = new  
MotionPathCondition(ReferenceType.START, 0.0, 2000);  
lbr_iiwa.move(lingetApplicationData().getFrame("/  
Destination")).triggerWhen(startCond, startAct));
```

15.25.4 Finalizar el registro de datos

El registro de datos se puede finalizar de forma independiente con respecto al movimiento (en cualquier lugar de la aplicación), o de forma sincronizada con el movimiento, activado por un Trigger.

Además, el registro finaliza automáticamente si se finaliza la aplicación o se ha finalizado la duración del registro establecida en el objeto DataRecorder utilizado.

independiente con respecto al movimiento	El registro se puede finalizar en cualquier momento mediante el método stopRecording().
sincronizado a través de Trigger	<p>Para un Trigger se debe formular una condición del tipo ICondition que si se cumple, activará el Trigger y una acción que se ejecutará al activarse el Trigger.</p> <p>(>>> 15.22.1 "Programar un Trigger" Página 340)</p> <p>La acción a ejecutar es la parada del registro de datos. Para ello se debe transmitir un objeto del tipo StopRecordingAction. Al crear el objeto, se deberá indicar el objeto DataRecorder que se utiliza para el registro de datos.</p> <p>Sintaxis del constructor:</p> <pre>StopRecordingAction(DataRecorder recorder)</pre> <p>A continuación, el objeto ICondition y el objeto StopRecordingAction se enlazan a una instrucción de movimiento con triggerWhen(...).</p>

15.25.5 Consultar estados del objeto DataRecorder

Resumen Se encuentran disponibles los siguientes métodos de la clase DataRecorder:

Método	Descripción
isEnabled()	<p>Tipo de retorno: boolean</p> <p>Se consulta si está activado el objeto DataRecorder (= true).</p>
isRecording()	<p>Tipo de retorno: boolean</p> <p>Se consulta si el registro de datos se está ejecutando (= true).</p>
isFileAvailable()	<p>Tipo de retorno: boolean</p> <p>Se consulta si el archivo con los datos registrados ya está guardado en la unidad de control del robot y disponible para la evaluación (= true).</p>
awaitFileAvailable(...)	<p>Tipo de retorno: boolean</p> <p>Bloquea la aplicación activada o la tarea en segundo plano como máximo hasta que haya finalizado el tiempo de bloqueo definido o hasta que el archivo con los datos registrados esté guardado en la unidad de control del robot y esté disponible para la evaluación (= true).</p> <p>La instrucción de bloqueo reenvía false cuando el archivo no está disponible dentro del máximo tiempo de bloqueo.</p> <p>Sintaxis:</p> <ul style="list-style-type: none"> ■ <code>awaitFileAvailable(long timeout, java.util.concurrent.TimeUnit timeUnit)</code> <p>Parámetros:</p> <ul style="list-style-type: none"> ■ <i>timeout</i>: máximo tiempo de bloqueo ■ <i>timeUnit</i>: unidad de tiempo para el máximo tiempo de bloqueo

15.25.6 Programa de ejemplo para el registro de datos

Durante un proceso de montaje se deben registrar los momentos que actúan de forma externa sobre los ejes de un LBR iiwa y además las fuerzas cartesianas que actúan sobre el TCP de una garra montada en la brida del robot. Los datos se deben registrar cada 10 ms.

El registro debe iniciarse de forma sincronizada con el movimiento si la fuerza que actúa sobre el TCP de la garra desde cualquier dirección supera 20 N. Cuando el proceso de montaje ha finalizado, también debe finalizar el registro.

A continuación se debe evaluar el archivo si está disponible como máximo después de 5 s.

```
private LBR lbr_iwa;
private Tool gripper;
...
gripper.attachTo(lbr_iwa.getFlange());
...
DataRecorder rec = new DataRecorder();
rec.setFileName("Recording.log");
rec.setSampleRate(10);

rec.addExternalJointTorque(lbr_iwa);
rec.addCartesianForce(gripper.getFrame("/TCP"), null);

StartRecordingAction startAction = new StartRecordingAction(rec);
ForceCondition startCondition =
ForceCondition.createSpatialForceCondition(gripper.getFrame("/TCP"),
20.0);

lbr_iwa.move(ptpgetApplicationData().getFrame("/StartPosition")));
lbr_iwa.move(lingetApplicationData().getFrame("/")
MountingPosition")).triggerWhen(startCondition, startAction));
lbr_iwa.move(lingetApplicationData().getFrame("/DonePosition")));

rec.stopRecording();

if (rec.awaitFileEnable(5, TimeUnit.SECONDS)) {
// Evaluation of the file if available
}
```

15.26 Definir teclas de usuario

Descripción

Las 4 teclas de usuario en el smartPAD pueden ocuparse libremente con funciones. Para ello, se pueden definir diferentes barras de teclas de usuario en el código fuente de las aplicaciones del robot o de las tareas en segundo plano.

Mediante la barra de teclas de usuario se asigna a las teclas de usuario su función correspondiente. No todas las teclas de usuario de la barra se tienen que ocupar con una función, aunque al menos se debe ocupar una. Adicionalmente se asigna un elemento gráfico o textual al campo situado en el margen de la pantalla de la smartHMI junto a la tecla de usuario, para aclarar la función de una tecla de usuario.

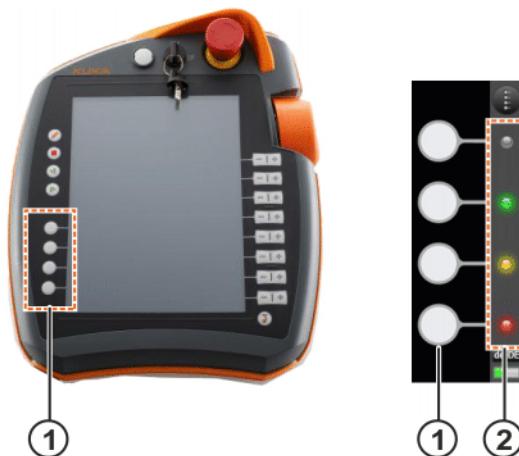


Fig. 15-16: Teclas de usuario en el smartPAD (ejemplo)

1 Teclas de usuario

2 Barra con iconos LED

El operario tiene a su disposición todas las barras de teclas de usuario que estén definidas en la aplicación del robot en ejecución o en la tarea en segundo plano en ejecución. Por ejemplo, se puede ofrecer una barra de teclas de usuario para el mando de una garra, mientras que mediante otra barra se utilizan las teclas de usuario para seleccionar diferentes secciones del programa.

Las barras de teclas de usuario están disponibles hasta que haya finalizado la aplicación del robot o la tarea en segundo plano que las ha creado.

Resumen

Para programar una barra de teclas de usuario son necesarios los siguientes pasos:

Paso	Descripción
1	Crear una barra de teclas de usuario. (>>> 15.26.1 "Crear una barra de teclas de usuario" Página 358)
2	Añadir las teclas de usuario a la barra (al menos una). (>>> 15.26.2 "Añadir teclas de usuario a la barra" Página 358)
3	Establecer la función que debe ejecutarse al accionar la tecla de usuario. (>>> 15.26.3 "Establecer la función de una tecla de usuario" Página 360)
4	Asignar al menos un elemento gráfico o textual respectivamente al campo situado en el margen de la pantalla de la smartHMI junto a la tecla de usuario. (>>> 15.26.4 "Rotulación y diseño gráfico de la barra de teclas de usuario" Página 362)
5	Para las teclas de usuario que pueden activar funciones que van unidas a un riesgo: Definir un mensaje de advertencia que se mostrará al accionar la teclas de usuario antes de que se pueda activar la función. (>>> 15.26.5 "Identificar teclas de usuario críticas para la seguridad" Página 365)
6	Publicar la barra de teclas de usuario. (>>> 15.26.6 "Publicar la barra de teclas de usuario" Página 366)

15.26.1 Crear una barra de teclas de usuario

Descripción	Para crear una barra de teclas de usuario son necesarios los siguientes métodos:						
	<ul style="list-style-type: none"> ■ <code>getApplicationUI()</code> Con este método se puede acceder desde una aplicación del robot o una tarea en segundo plano a la interfaz para la interfaz de usuario smartHMI. Tipo de retorno: ITaskUI ■ <code>createUserKeyBar(...)</code> Con este método se crea la barra de teclas de usuario. Pertenece a la interfaz ITaskUI. 						
Sintaxis	<pre>IUserKeyBar keybar = getApplicationUI().createUserKeyBar("name");</pre>						
Aclaración de la sintaxis	<table border="1"> <thead> <tr> <th>Elemento</th><th>Descripción</th></tr> </thead> <tbody> <tr> <td><code>keybar</code></td><td>Tipo: IUserKeyBar Nombre de la barra de teclas de usuario creada con <code>createUserKeyBar(...)</code></td></tr> <tr> <td><code>name</code></td><td>Tipo: String Nombre con el que se muestra la barra de teclas de usuario en la smartHMI (>>> Fig. 6-16) El número de caracteres que pueden mostrarse está limitada:<ul style="list-style-type: none"> ■ Se recomiendan como máximo de 12 a 15 caracteres. </td></tr> </tbody> </table>	Elemento	Descripción	<code>keybar</code>	Tipo: IUserKeyBar Nombre de la barra de teclas de usuario creada con <code>createUserKeyBar(...)</code>	<code>name</code>	Tipo: String Nombre con el que se muestra la barra de teclas de usuario en la smartHMI (>>> Fig. 6-16) El número de caracteres que pueden mostrarse está limitada: <ul style="list-style-type: none"> ■ Se recomiendan como máximo de 12 a 15 caracteres.
Elemento	Descripción						
<code>keybar</code>	Tipo: IUserKeyBar Nombre de la barra de teclas de usuario creada con <code>createUserKeyBar(...)</code>						
<code>name</code>	Tipo: String Nombre con el que se muestra la barra de teclas de usuario en la smartHMI (>>> Fig. 6-16) El número de caracteres que pueden mostrarse está limitada: <ul style="list-style-type: none"> ■ Se recomiendan como máximo de 12 a 15 caracteres. 						

Ejemplo

Se crea una barra de teclas de usuario para el mando de una garra.

```
IUserKeyBar gripperBar =  
getApplicationUI().createUserKeyBar("Gripper");
```

15.26.2 Añadir teclas de usuario a la barra

Descripción	Una barra de teclas de usuario recién creada en principio no tiene teclas de usuario. Las teclas de usuario que deben utilizarse se deben añadir a la barra. La interfaz IUserKeyBar ofrece para ello los siguientes métodos:
	<ul style="list-style-type: none"> ■ <code>addUserKey(...)</code> Añade una tecla de usuario individual a la barra. ■ <code>addDoubleUserKey(...)</code> Agrupa 2 teclas de usuario contiguas para formar una tecla doble y la añade a la barra. Los campos correspondientes en el margen de la pantalla de la smartHMI también se agrupan para formar un campo común más grande.
	Al añadir una tecla de usuario a una barra, se establece la función que se ejecutará al accionar la tecla de usuario (p. ej. abrir una garra, modificación de un parámetro, etc.). En función de la programación, se puede aplicar la acción de pulsar y soltar la tecla de usuario como accionamiento y se puede enlazar con una función.
	Una barra de teclas de usuario debe contener al menos una tecla de usuario. Cada tecla de usuario tiene asignado un número único. Este número se transmite al añadir una tecla de usuario.

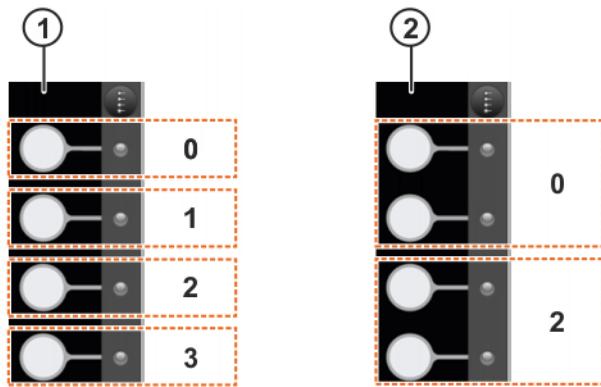


Fig. 15-17: Numeración de las teclas de usuario

1 Teclas individuales

2 Teclas dobles

Sintaxis

Añadir tecla individual

`IUserKey key = keybar.addUserKey(slot, listener, ignoreEvents);`

Añadir teclas dobles:

`IUserKey doubleKey = keybar.addDoubleUserKey(slot, listener, ignoreEvents);`**Aclaración de la sintaxis**

Elemento	Descripción
<code>keybar</code>	Tipo: <code>IUserKeyBar</code> Nombre de la barra de teclas de usuario a la que se añade una tecla de usuario
<code>key</code>	Tipo: <code>IUserKey</code> Nombre de la tecla individual que se añade a la barra
<code>doubleKey</code>	Tipo: <code>IUserKey</code> Nombre de la tecla doble que se añade a la barra
<code>slot</code>	Tipo: <code>int</code> Nombre de la tecla de usuario que se añade Teclas individuales: ■ 0 ... 3 Teclas dobles. ■ 0, 2

Elemento	Descripción
<i>listener</i>	<p>Tipo: IUserKeyListener</p> <p>Nombre del objeto de Listener con el que se ha establecido la función que se ejecutará al accionar la tecla de usuario</p> <p>(>>> 15.26.3 "Establecer la función de una tecla de usuario" Página 360)</p>
<i>ignoreEvents</i>	<p>Tipo: boolean</p> <p>Establece si debe reaccionarse cuando la tecla de usuario se pulsa de nuevo durante la ejecución de la función de la tecla</p> <ul style="list-style-type: none"> ■ true: Si la tecla se pulsa durante la ejecución de la función, no tendrá ningún efecto. ■ false: Se cuentan las veces que se pulsa la tecla durante la ejecución de la función. La función se repite el número de veces correspondiente.

Ejemplo

A las teclas de usuario se asignan las siguientes funciones para el mando de una garra:

- La tecla de usuario superior se utiliza para abrir y la tecla de usuario que está debajo, para cerrar la garra.
- Las dos teclas de usuario inferiores se agrupan para formar una tecla doble. Con ella se debe poder aumentar o reducir la velocidad de la garra.
- Las funciones para abrir y cerrar la garra se deben activar de nuevo cuando haya finalizado la función correspondiente.

```
IUserKeyBar gripperBar =
getApplicationUI().createUserKeyBar("Gripper");

IUserKeyListener openGripperListener = ...;
IUserKeyListener closeGripperListener = ...;
IUserKeyListener gripperVelocityListener = ...;

IUserKey openKey = gripperBar.addUserKey(0,
    openGripperListener, true);
IUserKey closeKey = gripperBar.addUserKey(1,
    closeGripperListener, true);
IUserKey velocityKey = gripperBar.addDoubleUserKey(2,
    gripperVelocityListener, false);
```

15.26.3 Establecer la función de una tecla de usuario**Descripción**

Para establecer la función que debe ejecutarse al pulsar una tecla de usuario, se deberá crear un objeto de Listener del tipo IUserKeyListener. Con la creación del objeto se declara automáticamente el método onKeyEvent(...).

El método Listener onKeyEvent(...) se acciona en los siguientes eventos:

- La tecla de usuario se pulsa.
- La tecla de usuario se suelta.



Aunque se empleen Listeners distintos, solo se puede ejecutar un OnKeyEvent(...) a la vez. Si, por ejemplo, el operario acciona el OnKeyEvent(...) de la tecla de usuario 2 mientras todavía se está ejecutando el OnKeyEvent(...) de la tecla de usuario 1, el segundo OnKeyEvent(...) no se inicia hasta que no haya finalizado el primero.

Sintaxis

```
IUserKeyListener listener = new IUserKeyListener() {
    @Override
    public void onKeyEvent(IUserKey key, UserKeyEvent event) {
        // Reacción ante una incidencia
    }
};
```

Aclaración de la sintaxis

Elemento	Descripción
listener	Tipo: IUserKeyListener Nombre del objeto de Listener
Parámetros de entrada del método Listener onKeyEvent(...):	
key	Tipo: IUserKey Tecla de usuario que se ha pulsado Mediante el parámetro se puede acceder directamente a la tecla de usuario para, por ejemplo, modificar la rotulación o el diseño gráfico correspondiente. Además, se puede determinar la tecla de usuario que se ha pulsado, especialmente en caso de utilización de la misma reacción para diferentes teclas de usuario.
event	Tipo: Enum del tipo UserKeyEvent Incidencia que ha activado el método Listener onKeyEvent(...) Valores Enum para teclas individuales: <ul style="list-style-type: none"> ■ UserKeyEvent.KeyDown: La tecla se ha pulsado. ■ UserKeyEvent.KeyUp: La tecla se ha soltado. Valores Enum para teclas dobles: <ul style="list-style-type: none"> ■ UserKeyEvent.FirstKeyDown: Se ha pulsado la tecla superior de las dos teclas. ■ UserKeyEvent.SecondKeyDown: Se ha pulsado la tecla inferior de las dos teclas. ■ UserKeyEvent.FirstKeyUp: Se ha soltado la tecla superior de las dos teclas. ■ UserKeyEvent.SecondKeyUp: Se ha soltado la tecla inferior de las dos teclas.

Ejemplo

La barra de teclas de usuario para el mando de la garra se amplia en un método a través del cual se puede adaptar la velocidad de la garra. Para ello, se emplean las dos teclas de usuario inferiores que se agruparon para formar una tecla doble.

Para ajustar la velocidad, se declara el atributo `velocity` que indica la velocidad actual en relación a la velocidad máxima (rango de valores: 0,1 ... 1,0). Al accionar la tecla de usuario superior, el valor aumenta 0,1, al accionar la inferior, el valor se reduce 0,1.

```
final double velocity = 0.1;
...
IUserKeyBar gripperBar = ...;
...
IUserKeyListener gripperVelocityListener = new IUserKeyListener() {
    @Override
    public void onKeyEvent(IUserKey key, IUserKeyEvent event) {
```

```

        if(event == UserKeyEvent.FirstKeyDown && velocity <= 0.9){
            velocity = velocity + 0.1;
        }
        else if(event == UserKeyEvent.SecondKeyDown && velocity >= 0.2){
            velocity = velocity - 0.1;
        }
    };
    ...
 IUserKey velocityKey = gripperBar.addDoubleUserKey(2,
    gripperVelocityListener, false);

```

15.26.4 Rotulación y diseño gráfico de la barra de teclas de usuario

Descripción

Se debe asignar al menos un elemento gráfico o textual respectivamente al campo situado en el margen de la pantalla de la smartHMI junto a la tecla de usuario. Como elementos gráficos están disponibles iconos LED de distintos colores y tamaños. Estos elementos pueden adaptarse durante el tiempo de funcionamiento de la aplicación de robot o de la tarea en segundo plano.

Para colocar de forma única cada uno de los elementos, el campo junto a las teclas de usuario se divide en una cuadrícula con 3x3 puestos. Esto también se aplica a las teclas de usuario que se han agrupado en una tecla doble, de forma que la cuadrícula en este caso se extiende a los dos campos.

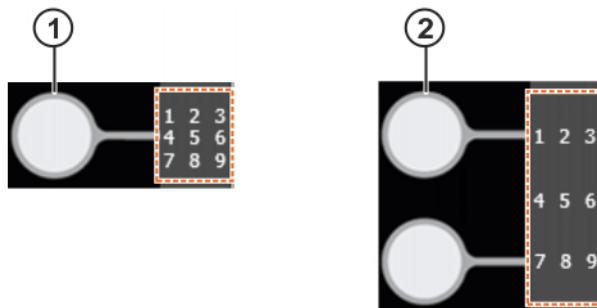


Fig. 15-18: División de la cuadrícula

1 Teclas individuales

2 Teclas dobles

En cada puesto puede colocarse un elemento como máximo. Este puesto se determina mediante los valores del Enum UserKeyAlignment. Si se asigna un nuevo elemento a un puesto que ya está ocupado, se borra el existente.

UserKey-Alignment

N.º de posición	Valor
1	UserKeyAlignment.TopLeft
2	UserKeyAlignment.TopMiddle
3	UserKeyAlignment.TopRight
4	UserKeyAlignment.MiddleLeft
5	UserKeyAlignment.Middle
6	UserKeyAlignment.MiddleRight
7	UserKeyAlignment.BottomLeft
8	UserKeyAlignment.BottomMiddle
9	UserKeyAlignment.BottomRight

15.26.4.1 Asignar elemento textual

Descripción En cada posición de la cuadrícula se puede asignar un elemento textual. Para ello se utiliza el método setText(...). El método pertenece a la interfaz IUserKey.

Sintaxis `key.setText (position, "text") ;`

Aclaración de la sintaxis

Elemento	Descripción
<code>key</code>	Tipo: IUserKey Tecla de usuario a la que se asigna un elemento textual
<code>position</code>	Tipo: Enum del tipo UserKeyAlignment Posición del elemento (posición de la cuadrícula) (>>> "UserKeyAlignment" Página 362)
<code>text</code>	Tipo: String Texto que se desea indicar A menudo, la longitud del texto supera el tamaño de la posición de la cuadrícula ya con 2 o pocos caracteres más. En este caso, la zona de visualización del texto se amplía, pero a pesar de ello solo se puede representar de forma práctica un número limitado de caracteres. Esto depende de los elementos de texto de las posiciones de la cuadrícula contiguas y de los caracteres utilizados.

Ejemplo

Se amplía la barra de teclas de usuario para el mando de una garra. Se desea indicar de forma duradera una rotulación adecuada junto a cada una de las teclas de usuario.

- Rotulación para las teclas de usuario para abrir y cerrar la garra: OPEN y CLOSE
 - Rotulación para las teclas de usuario para aumentar y reducir la velocidad de la garra: Símbolo de más y símbolo de menos
- Además se desea que se indique la velocidad actual y que se actualice en caso de cambio.

```
final double velocity = 0.1;
...
IUserKeyBar gripperBar = ...;
...
IUserKeyListener gripperVelocityListener = new IUserKeyListener() {
    @Override
    public void onKeyEvent(IUserKey key, IUserKeyEvent event) {
        if(event == UserKeyEvent.FirstKeyDown && velocity <= 0.9){
            velocity = velocity + 0.1;
        }
        else if(event == UserKeyEvent.SecondKeyDown && velocity >= 0.2){
            velocity = velocity - 0.1;
        }
        // La siguiente linea formatea la indicación de la velocidad
        // Indicación de los 3 primeros caracteres
        String value = String.valueOf(velocity).substring(0, 3);
        key.setText(UserKeyAlignment.Middle, value);
    }
};
IUserKey openKey = ...;
openKey.setText(UserKeyAlignment.TopLeft, "OPEN");
```

```
IUserKey closeKey = ...;
closeKey.setText(UserKeyAlignment.TopLeft, "CLOSE");

IUserKey velocityKey = ...;
velocityKey.setText(UserKeyAlignment.TopMiddle, "+");
velocityKey.setText(UserKeyAlignment.Middle,
    Double.toString(velocity));
velocityKey.setText(UserKeyAlignment.BottomMiddle, "-");
```

15.26.4.2 Asignar un icono de LED

Descripción

En cada posición de la cuadrícula se puede asignar un icono de LED. Para ello se utiliza el método `setLED(...)`. El método pertenece a la interfaz `IUserKey`.

Sintaxis

```
key.setLED (position, led, size);
```

Aclaración de la sintaxis

Elemento	Descripción
<code>key</code>	Tipo: <code>IUserKey</code> Tecla de usuario a la que se asigna un elemento gráfico
<code>position</code>	Tipo: Enum del tipo <code>UserKeyAlignment</code> Posición del elemento (posición de la cuadrícula) (>>> "UserKeyAlignment" Página 362)
<code>led</code>	Tipo: Enum del tipo <code>UserKeyLED</code> Color del icono de LED <ul style="list-style-type: none"> ■ UserKeyLED.Grey: Gris ■ UserKeyLED.Green: Verde ■ UserKeyLED.Yellow: Amarillo ■ UserKeyLED.Red: Rojo
<code>size</code>	Tipo: Enum del tipo <code>UserKeyLEDSize</code> Tamaño del icono de LED <ul style="list-style-type: none"> ■ UserKeyLEDSize.Small: Pequeño ■ UserKeyLEDSize.Normal: Grande

Ejemplo

Se amplia la barra de teclas de usuario para el mando de una garra. A las teclas de usuario para abrir y cerrar la garra se le debe asignar un icono de LED de tamaño pequeño a cada una.

Mientras la garra se abre o se cierre los iconos de LED deben indicarse de color verde. Si la garra está parada, los iconos de LED deben indicarse de color gris.

```
IUserKeyBar gripperBar =
getApplicationUI().createUserKeyBar("Gripper");

IUserKeyListener openGripperListener = new IUserKeyListener() {
@Override
public void onKeyEvent(IUserKey key, UserKeyEvent event) {
    key.setLED(UserKeyAlignment.BottomMiddle, UserKeyLED.Green,
        UserKeyLEDSize.Small);
    openGripper(); // Método para abrir la garra
    key.setLED(UserKeyAlignment.BottomMiddle, UserKeyLED.Grey,
        UserKeyLEDSize.Small);
}
};
```

```

IUserKeyListener openGripperListener = new IUserKeyListener() {
    @Override
    public void onKeyEvent(IUserKey key, UserKeyEvent event) {
        key.setLED(UserKeyAlignment.BottomMiddle, UserKeyLED.Green,
                   UserKeyLEDSize.Small);
        closeGripper(); // Método para cerrar la garra
        key.setLED(UserKeyAlignment.BottomMiddle, UserKeyLED.Grey,
                   UserKeyLEDSize.Small);
    }
};

IUserKeyListener gripperVelocityListener = ...;
...
IUserKey openKey = ...;
openKey.setText...;
openKey.setLED(UserKeyAlignment.BottomMiddle, UserKeyLED.Grey,
               UserKeyLEDSize.Small);

IUserKey closeKey = ...;
closeKey.setText...;
closeKey.setLED(UserKeyAlignment.BottomMiddle, UserKeyLED.Grey,
               UserKeyLEDSize.Small);

IUserKey velocityKey = ...;

```

15.26.5 Identificar teclas de usuario críticas para la seguridad

Descripción Las teclas de usuario pueden activar funciones que conllevan un riesgo. Para evitar que se produzcan daños por la activación involuntaria de este tipo de teclas de usuario, se pueden proveer con un mensaje de advertencia y, de este modo, identificarse como críticas para la seguridad. Para ello se utiliza el método setCriticalText(...). El método pertenece a la interfaz IUserKey.

Si el operario pulsa una de las teclas de usuario identificadas como críticas para la seguridad, se mostrará el mensaje definido con setCriticalText(...) en una ventana con el nombre **Operación crítica** en la smartHMI. Posteriormente, la tecla de usuario está desactivada durante aprox. 5 s. Al finalizar este tiempo, el operario puede activar la función deseada pulsando de nuevo la tecla de usuario en un período de 5 s.

Si la tecla de usuario no se pulsa durante este tiempo o se toca una zona fuera de la ventana **Operación crítica**, se cerrará la ventana y la tecla de usuario se restablecerá al estado anterior.

Sintaxis `key.setCriticalText ("text");`

Aclaración de la sintaxis

Elemento	Descripción
<code>key</code>	Tipo: IUserKey Tecla de usuario que se provee con un mensaje de advertencia
<code>text</code>	Tipo: String Texto del mensaje que se muestra al pulsar la tecla de usuario

Ejemplo

Se amplia la barra de teclas de usuario para el mando de una garra. Si se acciona la tecla de usuario para abrir la garra, debe indicarse un mensaje de advertencia. En dicho mensaje se le recuerda al operario que antes de abrir la garra descarte que puedan surgir daños por la caída de piezas.

```
IUserKeyBar gripperBar =
getApplicationUI().createUserKeyBar("Gripper");
...
IUserKey openKey = ...;
openKey.setText...;
openKey.setLED...;
openKey.setCriticalText("La garra se abre si se vuelve a accionar.
Asegúrese de que no surge ningún daño por la caída de piezas.");
```

15.26.6 Publicar la barra de teclas de usuario

Descripción

Si una barra de teclas de usuario dispone de todas las teclas de usuario y las funcionalidades necesarias, se debe publicar con el método publish(). Posteriormente estará disponible para el operario en el smartPAD.

Si una barra de teclas de usuario se ha publicado, no se podrán añadir más teclas de usuario en un momento posterior en el flujo de programa. Es decir, no es posible añadir posteriormente una tecla de usuario que haya quedado sin ocupar y no se puede proveer con una función. En contraposición, sí es posible modificar posteriormente el rótulo o el gráfico que se indica en la smartHMI al lado de la tecla de usuario.

Sintaxis

```
keybar.publish();
```

Aclaración de la sintaxis

Elemento	Descripción
keybar	Tipo: IUserKeyBar Nombre de la barra de teclas de usuario creada con createUserKeyBar(...).

Ejemplo

Se publica la barra de teclas de usuario creada para el mando de una garra.

```
IUserKeyBar gripperBar =
getApplicationUI().createUserKeyBar("Gripper");
...
gripperBar.publish();
```

15.27 Programación de mensajes

15.27.1 Programar mensajes de usuario

Descripción

Se pueden programar mensajes de indicación, de aviso y de error que se visualizarán durante la ejecución de la aplicación en la smartHMI y se escriben en el archivo Log de la aplicación. Además se pueden programar mensajes que no se visualizan en la smartHMI, sino que solo se escriben en el archivo Log.



Se recomienda visualizar en la smartHMI solo mensajes que sean imprescindibles. El uso demasiado intensivo de la visualización de mensajes puede tener un efecto negativo en el tiempo ejecución de la aplicación y en el manejo de la smartHMI.



Se recomienda solo emplear las instrucciones aquí indicadas para la emisión de mensajes y no utilizar otras funcionalidades de logging como p. ej. las instrucciones Java System.out.println(...) o System.err.println(...). En caso de utilizar estas instrucciones, no se puede garantizar que el mensaje se visualizará en la smartHMI.

Sintaxis

Mensaje de observación:

```
getLogger().info ("texto del mensaje");
```

Mensaje de advertencia:

```
getLogger().warn ("texto del mensaje");
```

Mensaje de error:

```
getLogger().error ("texto del mensaje");
```

Mensaje que solo se escribe en el archivo Log:

```
getLogger().fine ("texto del mensaje");
```

Aclaración de la sintaxis

Elemento	Descripción
Texto del mensaje	Texto que se debe visualizar en la smartHMI y/o escribir en el archivo Log

Ejemplo

Después de que el robot haya alcanzado un punto de destino, se debe visualizar un mensaje de indicación. Si el movimiento ha finalizado debido a una colisión, en su lugar se visualizará una indicación de advertencia.

```
IMotionContainer motion = lbr.move(lin(getFrame ("/P20"))
.breakWhen(collision));

if(motion.getFiredBreakConditionInfo() == null) {
    getLogger().info("End point reached.");
}
else{
    getLogger().warn("Motion canceled after collision!");
}
```

15.27.2 Programar cuadros de diálogo

Descripción

En una aplicación se pueden programar cuadros de diálogo. Estos cuadros de diálogo se muestran en una ventana de diálogo durante la ejecución de la aplicación en la smartHMI y requieren una reacción del usuario.

Mediante el método `displayModalDialog(...)` se pueden programar diferentes tipos de diálogos. En la smartHMI se muestran los siguientes símbolos en función del tipo:

Símbolo	Tipo
	INFORMATION Diálogo con una indicación que debe tener en cuenta el usuario
	QUESTION Diálogo con una pregunta que el usuario debe responder
	WARNING Diálogo con una advertencia que debe tener en cuenta el usuario
	ERROR Diálogo con un mensaje de error que debe tener en cuenta el usuario

El usuario selecciona la respuesta con mediante un botón cuya rotulación se define por el programador. Es posible definir hasta 12 botones.

La aplicación o la tarea en segundo plano desde la que se activa el diálogo se detiene hasta que el usuario reaccione. Se puede conseguir que el resto del flujo de programa dependa del botón seleccionada por el usuario. El método

`displayModalDialog(...)` envía de vuelta el índice del botón que el usuario ha seleccionado en la smartHMI. El índice comienza con "0" (= índice del primer botón).

Sintaxis

```
getApplicationUI().displayModalDialog (Tipo de diálogo, "Texto de diálogo", "Botón_1"<, ... "Botón_12">)
```

Aclaración de la sintaxis

Elemento	Descripción
<i>Tipo de diálogo</i>	Tipo: Enum del tipo ApplicationDialogType <ul style="list-style-type: none"> ■ INFORMATION: Se muestra el diálogo con el símbolo para una indicación. ■ QUESTION: Se muestra el diálogo con el símbolo para una pregunta. ■ WARNING: Se muestra el diálogo con el símbolo para una advertencia. ■ ERROR: Se muestra el diálogo con el símbolo para un error.
<i>Texto de diálogo</i>	Tipo: String Texto que se muestra en la smartHMI en la ventana de diálogo.
<i>Botón_1 ... Botón_12</i>	Tipo: String Rotulación del botón 1 ... 12 (se puede ver en la smartHMI de izquierda a derecha)

Ejemplo

El siguiente cuadro de diálogo del tipo QUESTION se debe mostrar en la smartHMI:

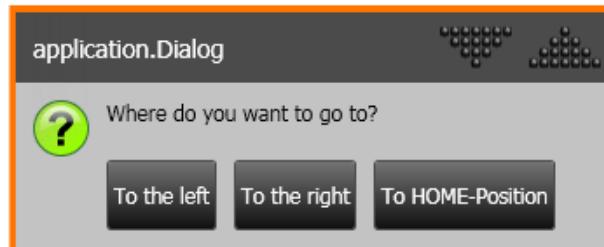


Fig. 15-19: Ejemplo de un cuadro de diálogo

```
int direction = getApplicationUI().displayModalDialog(
    ApplicationDialogType.QUESTION, "Where do you want to go to?", "To the left", "To the right", "To HOME-Position");

switch (direction) {
    case 0:
        lbr.move(ptp(getApplicationData().getFrame("/Left")));
        break;
    case 1:
        lbr.move(ptp(getApplicationData().getFrame("/Right")));
        break;
    case 2:
        lbr.move(ptpHome());
        break;
}
```

15.28 Control de la ejecución del programa

15.28.1 Pausar la aplicación

Descripción	Mediante el método halt() se puede pausar una aplicación. Con halt() se pausa el movimiento que se está ejecutando actualmente y el estado de la aplicación en el smartHMI cambia a Movimiento pausado . halt() provoca una parada con bloqueo del subproceso activando. Si de forma paralela hay más subprocessos en progreso, estos se siguen ejecutando. La ejecución de aplicación solo se detiene si halt() se activa en el subproceso de aplicación. Por ello se recomienda no activar halt() en las rutinas de tratamiento para acciones de conmutación referidas a la trayectoria o en rutinas de tratamiento para el control de procesos. En su lugar se recomienda utilizar el método pause() en estas rutinas de tratamiento.
	(>> 15.28.2 "Pausar la ejecución del movimiento" Página 369) La continuación del movimiento y del subproceso pausado solo es posible mediante la tecla de arranque en el smartPAD. Pulsando la tecla de arranque continuará el movimiento pausado. El subprocesso pausado continúa con la instrucción siguiente a halt() en el código fuente.

Sintaxis `getApplicationControl().halt();`

15.28.2 Pausar la ejecución del movimiento

Descripción	La ejecución del movimiento se puede pausar con el método pause(). El comportamiento se corresponde con el pausado de la aplicación mediante la smartPAD. Con pause() se pausa el movimiento que se está ejecutando actualmente y el estado de la aplicación en el smartHMI cambia a Movimiento pausado . pause() no provoca una espera con bloqueo. La aplicación se sigue ejecutando hasta alcanzar una instrucción de movimiento sincronizada. La continuación de la ejecución del movimiento solo es posible mediante la tecla de arranque en la smartPAD.
Sintaxis	<code>getApplicationControl().pause();</code>

15.28.3 Bucle for

Descripción	El bucle for, también denominado bucle de conteo, repite el bloque de instrucciones hasta que se cumpla una condición definida. Para ello se define un contador que se aumenta o se reduce con un valor constante después de cada ejecución del bucle. Al principio de una ejecución del bucle se comprueba si la condición definida aún se cumple. Esta condición se formula generalmente mediante la comparación del contador con un valor límite. Si la condición ya no se cumple, dejará de ejecutarse el bucle y continuará la ejecución del programa después del bucle. El bucle for se utiliza generalmente cuando se conoce el número de veces que debe ejecutarse un bucle. Los bucles for se pueden intercalar. (>> 15.28.8 "Ejemplos de bucles intercalados" Página 375)
Sintaxis	<code>for (int Contador = Valor inicial; Condición; Instrucción de conteo) { Instrucción_1; }</code>

```
<...
Instrucción_n; >
}
```

Aclaración de la sintaxis

Elemento	Descripción
<i>Contador</i>	Contador para el número de ejecuciones del bucle. Al contador se le asigna un valor inicial. Después de cada ejecución del bucle, el contador se aumenta o se reduce con un valor constante.
<i>Valor inicial</i>	Valor inicial del contador
<i>Condición</i>	Condición para la ejecución de una ejecución del bucle Generalmente, el contador se compara con un valor límite. El resultado de la comparación es del tipo boolean. El bucle finaliza en cuanto la comparación indica FALSE y, por lo tanto, la condición ya no se cumple.
<i>Instrucción de conteo</i>	La instrucción de conteo determina el valor según el que cambia el contador en cada ejecución del bucle. La anchura de paso y la dirección de conteo se pueden indicar de diferentes modos. Ejemplos: <ul style="list-style-type: none"> ■ <i>Valor inicial ++ --</i>: El valor inicial se aumenta o se reduce con el valor 1 después de cada ejecución. ■ <i>Valor inicial + - anchura de paso</i>: El valor inicial se aumenta o se reduce con la anchura de paso indicada después de cada ejecución del bucle.

Ejemplo

```
for (int i = 0; i < 10; i++) {
    getLogger().info(i);
}
```

El valor de la variable *i* se aumenta con el valor de 1 después de cada ejecución. El valor actual de *i* se indica en la smartHMI con cada recorrido. El bucle se recorre en total 10 veces. En este caso se indican los valores de 0 a 9.

15.28.4 Bucle while

Descripción

El bucle while repite un bloque de instrucciones hasta que se cumpla una condición determinada. También se denomina bucle finito, ya que la condición se comprueba antes de cada ejecución del bucle.

Cuando ya no se cumple la condición, el bloque de instrucciones del bucle no se ejecuta otra vez y el programa se reanuda después del bucle. Si la condición no se cumple desde el principio, el bloque de instrucciones no se ejecuta ninguna vez.

El bucle while se utiliza generalmente cuando se desconoce cuántas veces se debe ejecutar un bucle. P. ej., porque se calcula la condición de repetición o es una señal determinada.

Los bucles while se pueden intercalar. ([>>> 15.28.8 "Ejemplos de bucles intercalados"](#) Página 375)

Sintaxis

```
while (condición de repetición) {
    Instrucción_1;
    ...
}
```

Instrucción_n; >

}

Aclaración de la sintaxis

Elemento	Descripción
<i>Condición de repetición</i>	<p>Tipo: boolean</p> <p>Possibilidades:</p> <ul style="list-style-type: none"> ■ Variable del tipo boolean ■ Combinación, por ejemplo una comparación, con resultado de tipo boolean

Ejemplo 1

```
while(input1 == true) {
    getLogger().info("Input 1 is TRUE.");
}
getLogger().info("Input 1 is FALSE.");
```

Antes de la ejecución del bucle se comprueba si hay una señal de entrada. Si este es el caso, el bucle se ejecutará una y otra vez y en la smartHMI se emitirá que la entrada es true. Cuando la señal de entrada se ha restablecido, el bucle (ya) no se ejecutará y se emitirá que la entrada es false.

Ejemplo 2

```
int w = 0;
Random num = new Random();

while (w <= 21) {
    w = w + (num.nextInt(6) + 1);
}
```

En cada ejecución del bucle, el valor de las variables `w` aumenta un número aleatorio entre 1 y 6. Mientras la suma de todos los números aleatorios sea inferior a 21, se ejecutará el bucle. No puede predecirse el número exacto de ejecuciones. Puede ocurrir que el bucle finalice tras 4 ejecuciones (3 x 6 y 1 x 3) o después de 21 ejecuciones (21 x 1).

15.28.5 Bucle do-while

Descripción

El bucle do-while repite un bloque de instrucciones hasta que se cumpla una condición determinada. También se denomina bucle creciente, ya que la condición se comprueba después de la ejecución del bucle.

El bloque de instrucciones se ejecuta al menos una vez. Cuando se cumple la condición, finaliza el bucle y el programa continúa.

El bucle do-while se utiliza generalmente si un bucle se debe ejecutar al menos una vez, pero se desconoce cuántas veces en total. P. ej., porque se calcula la condición de cancelación o es una señal determinada.

Los bucles do-while se pueden intercalar. ([>>> 15.28.8 "Ejemplos de bucles intercalados"](#) Página 375)

Sintaxis

```
do {
    Instrucción_1;
    <...
    Instrucción_n; >
} while (condición de cancelación);
```

Aclaración de la sintaxis

Elemento	Descripción
<i>Condición de cancelación</i>	<p>Tipo: boolean</p> <p>Possibilidades:</p> <ul style="list-style-type: none"> ■ Variable del tipo boolean ■ Combinación, por ejemplo una comparación, con resultado de tipo boolean

Ejemplo

```
int num;

do {
    num = (int) (Math.random() * 6 + 1);
} while (num != 6);
```

Se lanza el dado para sacar cifras aleatorias entre 1 y 6 hasta que salga un 6. El dado se debe lanzar al menos una vez.

15.28.6 Ramificación if-else

Descripción

La ramificación if-else también se denomina ramificación condicionada. En función de una condición se ejecuta el primer bloque de instrucciones (bloque if) o el segundo bloque de instrucciones (bloque else).

El bloque else se ejecuta cuando no se cumple la condición if. El bloque else puede faltar. En este caso no se ejecutan más instrucciones, por si no se cumple la condición if.

Después del bloque if con `else if`, existe la posibilidad de comprobar otras condiciones y de enlazarlas con instrucciones. En el momento en que se cumple una de estas condiciones y se ejecuta la instrucción correspondiente, se dejarán de comprobar las ramificaciones siguientes.

Se pueden intercalar varias instrucciones if entre sí.

Sintaxis

```
if (condición_1) {
    Instrucción_1;
    <...
    Instrucción_n;>
}

<else if (condición_2) {
    Instrucción_1;
    <...
    Instrucción_n;>
}>

<else {
    Instrucción_1;
    <...
    Instrucción_n;>
}>
```

Aclaración de la sintaxis

Elemento	Descripción
Condición	<p>Tipo: boolean</p> <p>Possibilidades:</p> <ul style="list-style-type: none"> ■ Variable del tipo boolean ■ Combinación, por ejemplo una comparación, con resultado de tipo boolean

Ejemplo 1

Ramificación if sin else

```
int a;
int b;

if (a == 17) {
    b = 1;
}
```

Si la variable `a` tiene el valor 17, se asignará a la variable `b` el valor 1.

Ejemplo 2

Ramificación if dentro de un bucle for, sin else

```
for(int a = 1; a <= 10; a++) {
    if(a == 3) {
        a = a + 5;
    }
    getLogger().info(a);
}
```

El bucle se recorre 5 veces. Si la variable `a` tiene el valor 3, se asignará el valor de `a` una vez aumentado por 5.

En la smartHMI se indican los valores 1, 2, 8, 9 y 10.

Ejemplo 3

Ramificación if-else con else if

```
double velAct = 0.0;
double velDesired = 130.0;

...

if (velAct < velDesired) {
    accelerating();
}
else if (velAct > velDesired) {
    braking();
}
else {
    testrun();
}
```

En un programa se desea realizar una prueba para un vehículo. Esta prueba solo será significativa con una velocidad de consigna determinada.

Con la consulta if se comprueba si la velocidad real `velAct` es menor que la velocidad de consigna `velDesired`. Si es el caso, se acelera el vehículo. Si no es el caso, se continua con else if.

Con la consulta if-else se comprueba si la velocidad real `velAct` es mayor que la velocidad de consigna `velDesired`. Si es el caso, se frena el vehículo. Si no es el caso, se ejecuta el bloque else con la prueba.

15.28.7 Ramificación switch

Descripción

La ramificación switch también se denomina ramificación múltiple. En principio una ramificación switch es como una ramificación if intercalada varias veces.

En un bloque switch se pueden ejecutar distintos bloques case que se identifican por marcas case (marcas de salto). Dependiendo del resultado de una expresión, se selecciona y ejecuta el bloque case correspondiente. El programa salta a la marca case y se reanuda a partir de ese punto.

La palabra clave `break` al final de un bloque case significa que se abandona el bloque case. Si no aparece `break` al final de un bloque de instrucciones, se ejecutan también todas las instrucciones siguientes (no solo las instrucciones con marcas case) hasta que bien aparezca una marca de `break` o se hayan ejecutado todas las instrucciones.

Opcionalmente se puede programar un bloque default. Si no se cumple ninguna condición para el salto a una marca case, se ejecuta el bloque default.

Sintaxis

```
switch (Expresión) {
    case Constante_1:
        Instrucción_1;
        <...
        Instrucción_n;
        < break;>
        <...
    case Constante_n:
        Instrucción_1;
        <...
        Instrucción_n;
        < break;>
        < default:
            Instrucción_1;
            <...
            Instrucción_n;
            < break;>
        }
}
```

Aclaración de la sintaxis

Elemento	Descripción
<i>Expresión</i>	Tipo: int, byte, short, char, enum
<i>Constantes</i>	<p>Tipo: int, byte, short, char, enum</p> <p>El tipo de datos de la constante debe coincidir con el tipo de datos de la expresión.</p> <p>Nota: Las constantes del tipo char deben indicarse con ', p. ej. case 'a'</p>

Ejemplo

Ramificación switch con instrucción break y default:

```
int a, b;
switch (a) {
    case 1:
```

```

    b = 10;
case 2:
case 3:
    b = 20;
    break;
case 4:
    b = 30
    break;
default:
    b = 40;
}

```

Si la variable `a` tiene el valor 1, el programa salta a la marca `case 1`. A la variable `b` se le asigna el valor 10.

El programa continúa. Con la marca `case 2` no se realiza ninguna instrucción. El programa cambia a la marca `case 3`. A la variable `b` se le asigna el valor 20. Si la variable `a` tiene el valor 2 o 3, a la variable `b` también se le asigna el valor 20. Con la instrucción `break` se abandona el bloque `switch`.

Si la variable `a` tiene el valor 4 al principio, a la variable `b` se le asigna el valor 30.

Con la instrucción `break` se abandona el bloque `switch`.

Si la variable `a` tiene otro valor (p. ej 5), a la variable `b` se le asigna el valor 40. La ejecución del bloque `switch` se finaliza.

15.28.8 Ejemplos de bucles intercalados

Primero se recorre el bucle exterior hasta llegar al bucle interior. Seguidamente, el interior se recorre completamente. A continuación, se recorre el bucle exterior hasta el final y se comprueba si es necesario recorrer de nuevo el bucle exterior. Si es el caso, se recorrerá una vez más el bucle interior.

Los bucles se pueden intercalar con la profundidad deseada. Los bucles interiores se ejecutan siempre con la misma frecuencia con la que se recorre el bucle exterior.

Bucle for-in-for

```

for (int i = 1; i < 4; i++) {
    getLogger().info(i + ".Cycle begins");

    for (int k = 10; k > 0; k--) {
        getLogger().info("..." + k);
    }
}

```

El bucle exterior determina que el bucle interior se recorrerá 3 veces. El contador del bucle exterior comienza con el valor `i = 1`.

Después de que se haya indicado en la smartHMI que comienza el 1º recorrido, se inicia el contador del bucle interior con el valor `k = 10`. El valor de la variable `k` se reduce por el valor de 1 después de cada recorrido. El valor actual de `k` se indica en la smartHMI con cada recorrido. Si la variable `k` tiene el valor 1, el bucle interior se recorrerá por última vez.

Seguidamente finaliza el bucle exterior y el valor de la variable `i` se aumenta con el valor de 1. Se inicia el 2º recorrido.

Bucle for-in-while

```

int sum = 0;
int round = 1;
int diceRoll = 0;
Random num = new Random();

```

```

while (sum < 21) {
    round++;

    for (int i = 1; i <= 3; i++) {
        diceRoll = (num.nextInt(6) + 1);
        if (diceRoll % 2 == 0)
            sum += diceRoll;
    }
}

```

En un juego de dados se aplican las siguientes reglas:

- La suma total de todos los dados debe dar como resultado como mínimo 21 (consulta con el bucle while).
- En cada ronda se tiran los dados 3 veces (bucle for).
- Solo se suman las cifras pares (2, 4 y 6) (consulta if con módulo).

15.29 Continuar con una aplicación pausada en el modo automático (recovery)

Descripción

Si se desea continuar con una aplicación de robot pausada en el modo automático, es necesario que el control de orden superior sea capaz de detectar si el robot se sigue encontrando en la trayectoria programada. Si ya no está en la trayectoria, p. ej. después de una parada fuera de la trayectoria o porque ha sido desplazado manualmente durante la pausa, debe ser posible devolver el robot a su posición de forma automatizada con una estrategia adecuada.

La estrategia de retorno solo se puede aplicar si es seguro que en el camino de vuelta a la trayectoria puede haber una colisión. Si no está garantizada esta circunstancia, el operario tiene que retornar el robot a su posición manualmente.

Para el retorno a posición automática, RoboticsAPI facilita la interfaz IRecovery. A la interfaz se puede acceder desde aplicaciones robot y tareas de segundo plano:

- **IRecovery getRecovery()**

Resumen

La interfaz IRecovery ofrece métodos para consultar si es necesario retornar un robot a su posición para reanudar una aplicación pausada y qué estrategia de retorno se debe aplicar.

Método	Descripción
isRecoveryRequired()	<p>Tipo de retorno: boolean</p> <p>Comprueba para una aplicación pausada si uno o varios robots empleados en la misma necesitan ser retornados a su posición.</p> <p>true: Como mínimo un robot debe ser retornado a su posición para que se pueda reanudar la aplicación.</p> <p>false: La aplicación se puede reanudar inmediatamente.</p>

Método	Descripción
isRecoveryRequired(...)	<p>Tipo de retorno: boolean</p> <p>Comprueba para una aplicación pausada si es necesario retornar un robot a su posición. El robot se transfiere como parámetro (tipo: Robot).</p> <p>true: El robot debe ser retornado a su posición para que se pueda reanudar la aplicación.</p> <p>false: La aplicación se puede reanudar inmediatamente.</p>
getRecoveryStrategy(...)	<p>Tipo de retorno: RecoveryStrategy</p> <p>Consulta de la estrategia que se emplea para devolver un robot en concreto a su trayectoria. El robot se transfiere como parámetro (tipo: Robot).</p> <ul style="list-style-type: none"> ■ PTPRecoveryStrategy: El robot se retorna a su posición mediante un movimiento PTP. Se realiza el desplazamiento con un 20 % de la máxima velocidad del eje posible y con el Override efectivo. Actualmente no hay disponibles más estrategias. <p>En los siguientes casos el método resulta en <code>cero</code>:</p> <ul style="list-style-type: none"> ■ No es necesaria una estrategia de retorno o no hay ninguna disponible. ■ La aplicación no está pausada.

PTPRecoveryStrategy La clase PTPRecoveryStrategy ofrece métodos get con los que se pueden consultar características del movimiento PTP. Mediante estos métodos se puede evaluar si la estrategia de retorno se puede ejecutar en el modo automático.

Método	Descripción
getStartPosition()	<p>Tipo de retorno: JointPosition</p> <p>Consulta de la posición inicial del movimiento PTP (= posición de ejes desde la cual se realiza el retorno a posición)</p> <p>La posición inicial es la posición nominal actualmente ajustada y no la posición real medida en el momento.</p>
getMotion()	<p>Tipo de retorno: PTP</p> <p>Consulta del movimiento PTP que se ejecuta en la realización de la estrategia</p> <p>Del objeto móvil retornado se pueden consultar más informaciones:</p> <ul style="list-style-type: none"> ■ <code>getDestination()</code>: Posición meta del movimiento PTP (= posición de eje en la que se ha abandonado la trayectoria) ■ <code>getMode()</code>: Modo de regulación del movimiento que ha sido interrumpido

Control externo La unidad de control del robot debe comunicarle al control de orden superior si es necesario realizar un retorno a posición. El control de orden superior solo puede permitir la realización de la estrategia de retorno si es posible llevarla a cabo sin ningún riesgo. En caso contrario solo se pueden realizar retornos a posición de forma manual.

Se dispone de las siguientes señales de sistema:

- Salida AutExt_AppReadyToStart

Con esta salida, la unidad de control del robot le comunica al control de orden superior si la aplicación se puede reanudar.

- a. Si `isRecoveryRequired(...)` aporta el valor **false** (= no es necesario un retorno a posición), la salida se puede ajustar a TRUE.
- b. Si `getRecoveryStrategy(...)` aporta el valor **null** (= no hay disponible ninguna estrategia de retorno), es necesario ajustar la salida a FALSE.
- c. Si la evaluación de la estrategia de retorno indica que se puede ejecutar en el modo automático, la salida se puede ajustar a TRUE.
Si no es el caso, la salida debe ajustarse a FALSE.

- Entrada App_Start

A través de un flanco ascendente de esta entrada el control de orden superior le comunica a la unidad de control del robot que la aplicación se debe reanudar. (requisito: `AutExt_AppReadyToStart` indica TRUE)

El control de orden superior debe enviar la señal de arranque `App_Start` 2 veces:

1. Señal de arranque para el retorno a posición
2. Señal de arranque para la reanudación de la aplicación

15.30 Tratamiento de fallos

15.30.1 Tratamiento de instrucciones de movimiento fallidas

Las instrucciones de movimiento que se transmiten a la unidad de control del robot pueden fallar por distintos motivos, p. ej.:

- El punto de destino está fuera de un campo de trabajo
- El punto de destino no se puede alcanzar con la configuración de ejes actual
- el Frame empleado no está disponible en los datos de aplicación

Por defecto, una instrucción de movimiento fallida lleva a la cancelación de la aplicación. Para evitar la cancelación de la aplicación en caso de error, se pueden establecer rutinas de tratamiento.

Dependiendo del caso de error existen las siguientes posibilidades de tratamiento:

- El tratamiento de instrucciones de movimiento sincrónicas fallidas se realiza mediante un bloque try-catch
- El tratamiento de instrucciones de movimiento asíncronas fallidas se realiza mediante un administrador de eventos

15.30.2 Tratamiento de instrucciones de desplazamiento sincronizadas y fallidas

Descripción

Las instrucciones de desplazamiento sincronizadas (`.move(...);`) se envían paso a paso a la unidad de control en tiempo real y se procesan. La ejecución posterior del programa está cancelada hasta que se haya ejecutado el movimiento. Seguidamente se envía la siguiente instrucción.

Mediante un bloque try-catch se pueden procesar errores de tiempo de ejecución previsibles o excepciones sin tener que interrumpir la aplicación.

Dentro de un bloque try-catch se acciona un método fijado para tratar errores. Al activar la palabra clave `try` se debe intentar procesar la instrucción indicada. Si durante la ejecución surge un error, se inicia la rutina de tratamiento correspondiente en el bloque catch.

Sintaxis

```

try{
    // Código durante cuyo procesamiento podría surgir un error de tiempo de ejecución
}
catch (Exception e) {
    // Código para el tratamiento del error de tiempo ejecución
}
< finally{
    // Tratamiento final (opcional)
}>

```

Aclaración de la sintaxis

Elemento	Descripción
try{...}	El bloque try contiene un código que podría llevar a un error de tiempo de ejecución. Si surge un error, se interrumpe la ejecución del bloque try y se ejecuta el bloque catch.
catch (...){...}	El bloque catch contiene el código para el tratamiento del error del tiempo de ejecución. Solo se accede al bloque catch si en el bloque try aparece un error.
Exception e	A través del tipo de datos de error (aquí: Exception) se puede fijar el tipo de error que se debe tratar en el bloque catch. El tipo de error Exception es la clase más elevada de la mayoría de los tipos de datos de error. Pero también es posible profundizar más en errores específicos. A través del parámetro e se puede consultar información relativa a errores que han surgido. Para el tratamiento de instrucciones de desplazamiento fallidas, importa sobre todo el tipo de datos de error CommandInvalidException (paquete: com.kuka.robotics-API.executionModel). Aparece, entre otros casos, si no es posible alcanzar el punto de destino del desplazamiento.
finally {...}	El bloque finally es opcional. Aquí se puede fijar un tratamiento final que se debe aplicar en cualquier caso. Independientemente de si ha surgido un error en el bloque try.

Ejemplo

Un robot realiza un movimiento con regulación de impedancia con una rigidez muy reducida. Por ello, no está garantizado que alcance la posición de destino. A continuación debe desplazarse de forma relativa 50 cm en dirección Z positiva del sistema de coordenadas de la brida. Si el robot después del movimiento con regulación de impedancia se encuentra en una posición poco favorable, no se puede realizar el movimiento lineal y surge un error de tiempo de ejecución. Para evitar en este caso una interrupción de la aplicación, se programa el movimiento lineal crítico en un bloque try-catch. Si la planificación de desplazamiento falla, el robot se debe desplazar a un punto intermedio antes de continuar con el desarrollo de la aplicación.

```

CartesianImpedanceControlMode softMode = new
CartesianImpedanceControlMode();

softMode.parametrize(CartDOF.ALL).setStiffness(10.0);
exampleRobot.move(ptp(getFrame("/Start"))

```

```

.setMode(softMode).setJointVelocityRel(0.3));

try{
    getLogger().info("1: Intentos de realizar movimientos lineales");
    exampleRobot.move(linRel(0.0, 0.0, 500.0)
        .setJointVelocityRel(0.5));
}

catch(CommandInvalidException e){
    getLogger().info("2: Bewegung nicht möglich");
    exampleRobot.move(ptp(getFrame("/Punto intermedio"))
        .setJointVelocityRel(0.5));
}

finally{
    getLogger().info(
        "3: Se realiza el tratamiento final en el bloque finally");
}

getLogger().info("4: Seguir con el programa");

```

15.30.3 Tratamiento de instrucciones de desplazamiento asíncrónicas y fallidas

Descripción

En el caso de instrucciones de movimiento realizados de forma asíncrona (.moveAsync(...);), justo después de emitir la instrucción de movimiento se ejecuta la siguiente línea de programa.

Para reaccionar a una instrucción de movimiento asíncrona fallida, se emplea un administrador de eventos.

Este administrador de eventos es un objeto del tipo IErrorHandler y define el método handleError(...). Durante la ejecución del método handleError(...) se bloquea la transmisión de más instrucciones de movimiento a la unidad de control en tiempo real, la aplicación se detiene.

Con handleError(...) se determina la rutina de tratamiento. A través de los parámetros de entrada del método se puede acceder a información relativa a la instrucción de movimiento fallida. El método devuelve un parámetro del tipo ErrorHandlingAction. A través de este parámetro se selecciona la reacción final al error.

Se puede elegir entre las siguientes reacciones:

- La aplicación se finaliza con un error.
- La instrucción de movimiento se pausa y debe ser reanudada por parte del operario mediante la tecla de inicio del smartPAD.
- El error se ignora y la aplicación continúa.

Es necesario registrar al administrador de eventos definido antes de que se pueda utilizar en la aplicación. Para ello se utiliza el método getApplicationControl().registerMoveAsyncErrorHandler(...). El método pertenece a la interfaz IApplicationControl.

Sintaxis

Definir el administrador de eventos:

```
IErrorHandler errorHandler = new IErrorHandler() {
    @Override
    public ErrorHandlingAction handleError
        (Device device, IMotionContainer failedContainer,
        List<IMotionContainer> canceledContainers) {
```

```
// código que se ejecuta en caso de error.

return ErrorHandlingAction.reaction;
}

};
```

Registrar el administrador de eventos:

```
getApplicationControl().registerMoveAsyncErrorHandler(errorHandler);
```

Aclaración de la sintaxis

Elemento	Descripción
<code>errorHandler</code>	Tipo: IErrorHandler Nombre del administrador de eventos que es el responsable del tratamiento de instrucciones de movimiento asincrónicos fallidos.
Parámetros de entrada del método <code>handleError(...)</code> :	
<code>device</code>	Tipo: Dispositivo A través del parámetro se puede acceder al robot para el que se había dado la instrucción de movimiento fallida.
<code>failed Container</code>	Tipo: IMotionContainer A través del parámetro se puede acceder a la instrucción de movimiento fallido.
<code>canceled Containers</code>	Tipo: List<IMotionContainer> A través del parámetro se puede acceder a una lista de todas las instrucciones de movimiento borradas. Se incluyen todas las instrucciones de movimiento que en el momento de activar el método <code>handleError(...)</code> ya habían sido enviadas a la unidad de control en tiempo real.
<code>reaction</code>	Tipo: Enum del tipo ErrorHandlingAction Valor de retorno del método <code>handleError(...)</code> a través del cual se fija la reacción final al error. <ul style="list-style-type: none"> ■ ErrorHandlingAction.EndApplication: La aplicación se finaliza con un error. ■ ErrorHandlingAction.PauseMotion: La instrucción de movimiento se detiene hasta que el operario reanuda la aplicación a través del smartPAD. ■ ErrorHandlingAction.Ignore: El error se ignora y la aplicación continúa.

Ejemplo

En una aplicación se desean ejecutar varias instrucciones de movimiento asincrónicas una detrás de otra. A través del registro de un administrador de eventos del tipo IErrorHandler se fija una rutina de tratamiento mediante el método `handleError(...)` para el caso de que una de las instrucciones de movimiento asincrónicas falle:

- En la smartHMI se indica qué instrucción de movimiento ha fallado.
- En la smartHMI se indica qué instrucciones de movimiento ya no se ejecutan.

El método `handleError(...)` se finaliza con el retorno del valor `ErrorHandlingAction.Ignore`.

```
public void initialize(){
    kuka_Sunrise_Cabinet_1 = getController("kuka_Sunrise_Cabinet_1");
    robot = (LBR) getRobot(kuka_Sunrise_Cabinet_1, "LBR_iwa_14_R820_1");
```

```
IErrorHandler errorHandler = new IErrorHandler()
{
    @Override
    public ErrorHandlingAction handleError(Device device,
                                            IMotionContainer failedContainer,
                                            List<IMotionContainer> canceledContainers) {
        getLogger().warn("La siguiente instrucción de movimiento ha
fallado: " + failedContainer.toString());
        getLogger().info("Las siguientes instrucciones de movimiento
no se ejecutan: ");
        for(int i = 0; i < canceledContainers.size(); i++){
            getLogger().info(canceledContainers.get(i).toString());
        }
        return ErrorHandlingAction.Ignore
    }
};

getApplicationControl().
registerMoveAsyncErrorHandler(errorHandler);
}

public void run(){
...
    robot.moveAsync(ptp(getFrame("/P1")));
    robot.moveAsync(ptp(getFrame("/P2")));

    robot.moveAsync(lin(getFrame("/P3")));

    robot.moveAsync(ptp(getFrame("/P4")));
    robot.moveAsync(ptp(getFrame("/P5")));
    robot.moveAsync(ptp(getFrame("/P6")));

    robot.moveAsync(ptp(getFrame("/P7")));
    robot.moveAsync(ptp(getFrame("/P8")));
    robot.moveAsync(ptp(getFrame("/P9")));
...
}
```

Para explicar el comportamiento del sistema se supone que el movimiento lineal a P3 no se puede planear. De esta forma se activa el método handleError(...). En ese momento, en nuestro ejemplo el robot se encuentra en el punto de destino P2.

Si al mismo tiempo se encuentran por ejemplo, las instrucciones de movimiento a P4, P5, P6 ya se encuentran en la unidad de control en tiempo real, estas instrucciones de movimiento se borran y ya no se ejecutan.

Al activar el método handleError(...) se bloquea el envío de instrucciones de movimiento a la unidad de control en tiempo real. En este caso la aplicación se detiene antes de la instrucción de movimiento a P7. Si el método handleError(...) se finaliza con el retorno del valor ErrorHandlingAction.Ignore, la aplicación continúa. En ese caso el robot se desplaza desde la posición actual P2 a P7.

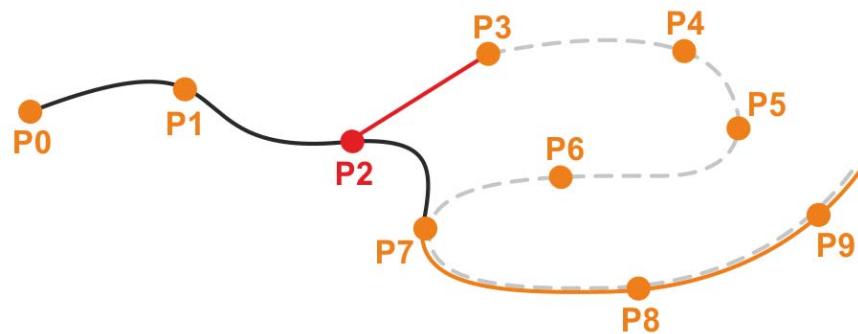


Fig. 15-20: Movimiento a P3 fallido (ejemplo curso de trayectoria)

16 Tareas en segundo plano

16.1 Utilización de tareas en segundo plano

Tareas

Las tareas en segundo plano se utilizan para ejecutar acciones en segundo plano de forma paralela a la aplicación del robot en ejecución. Se pueden ejecutar varias tareas en segundo plano de forma paralela e independiente entre sí.

Las tareas en segundo plano se pueden utilizar para las siguientes tareas:

- Controlar dispositivos periféricos. Ejemplos: Control de dispositivos de protección; control de un circuito de refrigeración.

Por esta razón no es necesario un control de orden superior, p. ej. un PLC para las aplicaciones menores, ya que la unidad de control del robot se puede encargar de este tipo de tareas.



Las salidas que se comutan a través de una tarea de segundo plano, se comutan independientemente de si actualmente se está ejecutando una aplicación robot o de si esta aplicación robot se detiene, por ejemplo por una PARADA DE EMERGENCIA o falta de confirmación.

- Consultar información sobre el robot y la estación



ADVERTENCIA Las tareas en segundo plano no se pueden utilizar para mover el robot. Esta es la tarea de la aplicación del robot. La activación de comandos de movimiento desde una tarea en segundo plano puede provocar un comportamiento no especificado del robot y, como consecuencia, provocar daños materiales y personales.

Propiedades

Las tareas en segundo plano forman parte del proyecto Sunrise. Se crean en Sunrise.Workbench y se transmiten a la unidad de control del robot con la sincronización del proyecto.

(>>> 5.5 "Crear una nueva tarea en segundo plano" Página 55)

Existen 2 tipos de tareas en segundo plano que se diferencian en cuanto a su duración:

- Tareas cíclicas en segundo plano
Se ejecutan cíclicamente. El comportamiento cíclico se puede adaptar por el programador en función de la acción que debe cumplir la tarea.
- Tareas en segundo plano no cíclicas
Se ejecutan una vez.

Las tareas en segundo plano del tipo de inicio **Automático** se inician automáticamente después de la sincronización de un proyecto y después del arranque de la unidad de control del robot. Las tareas en segundo plano del tipo de inicio **Manual** se deben iniciar manualmente a través del smartPAD.

16.2 Tarea en segundo plano cíclica

Estructura

```

(1) package backgroundTask;
(2) import java.util.concurrent.TimeUnit;
(3) public class BackgroundTaskCyclic extends RoboticsAPICyclicBackgroundTask {
    private Controller kuka_Sunrise_Cabinet_1;
(4)
(5)     public void initialize() {
        kuka_Sunrise_Cabinet_1 = getController("KUKA_Sunrise_Cabinet_1");
        initializeCyclic(0, 500, TimeUnit.MILLISECONDS,
                         CycleBehavior.BestEffort);
    }
(6)     public void runCyclic() {
    }
}

```

Fig. 16-1: Estructura de una tarea en segundo plano cíclica

Pos.	Descripción
1	Línea con el nombre del paquete en el que se encuentra la tarea
2	Zona de importación La zona contiene las clases importadas que son necesarias para la programación de las tareas.
3	Encabezamiento de la tarea La tarea en segundo plano cíclica es una subclase de RoboticsAPICyclicBackgroundTask.
4	Zona de declaración Al crear una tarea se crea automáticamente una instancia de la clase Controlador. A través del campo de datos, p. ej., se puede acceder a los robots disponibles para consultar los datos de posiciones y de momentos.
5	Método initialize() Aquí se asignan valores iniciales a los campos de datos creados en la zona de declaración. Al crear la tarea se activa automáticamente el método initializeCyclic(...), con el que se establece el comportamiento cíclico de la tarea. (>>> "Inicialización" Página 386) Indicación: El método no se puede borrar o renombrarse.
6	Método runCyclic() Aquí se programa el código que debe ejecutarse cíclicamente. Indicación: El método no se puede borrar o renombrarse.

Inicialización

Con initializeCyclic(...) se inicia el comportamiento cíclico de la tarea en segundo plano. Los parámetros que se transmiten al método están previamente ocupados con valores por defecto y pueden ser modificados por el programador.

```
initializeCyclic(long initialDelay, long period, TimeUnit timeUnit,
                 CycleBehavior behavior);
```

Elemento	Descripción
<i>initialDelay</i>	Retardo temporal con el que se inicia la tarea cíclica secundaria Por defecto: 0 ms La unidad de tiempo se establece con <i>timeUnit</i> .
<i>period</i>	Duración del periodo (= intervalo de tiempo entre 2 activaciones de runCyclic()) Por defecto: 500 ms La unidad de tiempo se establece con <i>timeUnit</i> .
<i>timeUnit</i>	Unidad de tiempo de <i>initialDelay</i> y <i>period</i> Por defecto: TimeUnit.MILLISECONDS El Enum del tipo TimeUnit se incluye por defecto en las bibliotecas de Java.
<i>behavior</i>	Comportamiento en caso de exceder el tiempo Se establece cómo debe comportarse la tarea en segundo plano cuando se excede la duración del periodo establecida con <i>period</i> a través del tiempo de ejecución de runCyclic(). <ul style="list-style-type: none"> ■ CycleBehavior.BestEffort runCyclic() se ejecuta por completo y a continuación se activa de nuevo. ■ CycleBehavior.Strict Se cancela la ejecución de la tarea en segundo plano y se emite una CycleExceededException. Por defecto: CycleBehavior.BestEffort

16.3 Tarea en segundo plano no cíclica

Estructura

```

(1) package backgroundTask;
(2) import com.kuka.roboticsAPI.applicationModel.tasks.RoboticsAPIBackgroundTask;
(3) public class BackgroundTask extends RoboticsAPIBackgroundTask {
    private Controller kuka_Sunrise_Cabinet_1;
(4)
(5)     public void initialize() {
        kuka_Sunrise_Cabinet_1 = getController("KUKA_Sunrise_Cabinet_1");
    }
(6)     public void run() {
    }
}

```

Fig. 16-2: Estructura de una tarea en segundo plano no cíclica

Pos.	Descripción
1	Línea con el nombre del paquete en el que se encuentra la tarea
2	Zona de importación La zona contiene las clases importadas que son necesarias para la programación de las tareas.
3	Encabezamiento de la tarea La tarea en segundo plano no cíclica es una subclase de RoboticsAPIBackgroundTask.

Pos.	Descripción
4	Zona de declaración Al crear una tarea se crea automáticamente una instancia de la clase Controlador. A través del campo de datos, p. ej., se puede acceder a los robots disponibles para consultar los datos de posiciones y de momentos.
5	Método initialize() Aquí se asignan valores iniciales a los campos de datos creados en la zona de declaración. Indicación: El método no se puede borrar o renombrarse.
6	Método run() Aquí se programa el código que debe ejecutarse una vez. El tiempo de ejecución no está limitado. Indicación: El método no se puede borrar o renombrarse.

17 Programación con robot flexible

17.1 Sensores y regulación

Un robot industrial estándar se puede utilizar regulado en función de la posición si no cuenta con elementos auxiliares adicionales. El objetivo de la regulación de posición consiste en mantener siempre al mínimo la diferencia entre la posición predefinida del robot y la posición real del robot.

Además de los sensores de posición para la determinación de la posición actual de la articulación, el KUKA LBR iiwa posee sensores de momentos de articulación en cada eje con los que se mide el momento actual en la unión articulada. Estos datos permiten, de forma adicional a la regulación de posición, la utilización de un regulador de impedancia que permite realizar un comportamiento flexible del robot. El modelo subyacente es un sistema virtual de amortiguador por resorte con valores ajustables para la rigidez y la amortiguación. Además se pueden conectar fuerzas y oscilaciones de fuerza de forma adicional.

Mediante los sensores especiales y los mecanismos del regulador disponibles, el KUKA LBR iiwa es sensible y preciso. Puede reaccionar de un modo muy rápido ante las fuerzas de proceso que puedan producirse y está especialmente indicado para diferentes tareas de ensamblaje, así como para la colaboración con el hombre.

17.2 Resumen de reguladores disponibles

El KUKA LBR iiwa puede utilizarse con diferentes reguladores. La RoboticsAPI pone a disposición una clase propia para cada tipo de regulación en el paquete com.kuka.roboticsAPI.motionModel.controlModeModel. La clase superior común es AbstractMotionControlMode.

Regulador	Descripción
Regulador de posición	<p>Tipo de datos: PositionControlMode</p> <p>El objetivo de la regulación de posición consiste en recorrer la trayectoria especificada, si es posible, con una posición exacta y sin desviación de la trayectoria. Las influencias externas, por ejemplo obstáculos o fuerzas de proceso que puedan producirse, no se tendrán en cuenta por defecto.</p>
Regulador de impedancia cartesiano	<p>Tipo de datos: CartesianImpedanceControlMode</p> <p>El regulador de impedancia cartesiano se basa en el modelo de un sistema virtual de amortiguador por resorte con valores ajustables para la rigidez y la amortiguación. Este resorte se fija entre la posición nominal y real del TCP. De este modo el robot reacciona de forma flexible ante las influencias externas.</p>
Regulador de impedancia cartesiano con oscilación de fuerza conectada adicionalmente	<p>Tipo de datos: CartesianSinelImpedanceControlMode</p> <p>Forma especial del regulador de impedancia cartesiano. De forma adicional al comportamiento flexible, se pueden conectar adicionalmente fuerzas nominales constantes y oscilaciones de fuerza senoidales. Con este regulador se pueden realizar, por ejemplo, desplazamientos de búsqueda dependientes de la fuerza y movimientos de vibración para procesos de ensamblado.</p>

17.3 Utilizar el regulador en la aplicación del robot

Descripción	En la aplicación del robot, el regulador se ajusta individualmente para cada instrucción de movimiento. Para ello son necesarios los siguientes pasos por defecto:
Procedimiento	<ol style="list-style-type: none"> 1. Crear el objeto del regulador del tipo de datos del regulador. 2. Parametrizar el objeto del regulador para establecer el comportamiento del regulador. 3. Ajustar el regulador como parámetro de movimiento para la instrucción de movimiento.

17.3.1 Crear un objeto del regulador

Descripción	Para poder utilizar un regulador, primero se deberá crear e inicializar una variable del tipo de datos deseado del regulador. El objeto del regulador se crea por defecto con el constructor estándar.
--------------------	--

Sintaxis	<i>Modo de regulación controlMode;</i> <i>controlMode = new Modo de regulación () ;</i>
-----------------	--

Aclaración de la sintaxis	Elemento	Descripción
	<i>Modo de regulación</i>	Tipo de datos del regulador. Subclase de AbstractMotionControlMode.
	<i>controlMode</i>	Nombre del objeto del regulador

Ejemplo	Crear un regulador de impedancia cartesiano:
	<pre>CartesianImpedanceControlMode cartImpCtrlMode; cartImpCtrlMode = new CartesianImpedanceControlMode();</pre>

17.3.2 Determinar parámetros del regulador

Los parámetros ajustables dependen del tipo de regulador utilizado. Las distintas clases de regulador en RoboticsAPI ponen métodos set y get propios a disposición de cada parámetro.

(>>> 17.5.2 "Parametrización del regulador de impedancia" Página 394)

(>>> 17.6.3 "Parametrización del regulador de impedancia con oscilación de fuerza conectada adicionalmente" Página 402)

17.3.3 Transferir el objeto del regulador como parámetro de movimiento

Descripción	El objeto del regulador se transfiere como parámetro a un movimiento con la instrucción <i>setMode(...)</i> . Si a un movimiento no se le transfiere ningún objeto del regulador como parámetro, el movimiento se ejecuta automáticamente regulado en función de la posición.
--------------------	---



Movimientos que utilizan el regulador de impedancia cartesiano no pueden contener posiciones que se encuentren en las inmediaciones de posiciones singulares.

Sintaxis	<i>movableObject.move (motion.setMode (controlMode)) ;</i>
-----------------	--

Aclaración de la sintaxis

Elemento	Descripción
<i>motion</i>	Tipo: Motion Movimiento que se ha de ejecutar
<i>controlMode</i>	Tipo: Subclase de AbstractMotionControlMode Nombre del objeto del regulador

17.4 Regulador de posición

En la regulación de posición, los motores se accionan de forma que la posición actual del robot coincida en todo momento, hasta una diferencia mínima, con la posición nominal que se especifica por el control. El regulador de posición está especialmente indicado, cuando es necesario un posicionamiento preciso.

El regulador de posición se representa mediante la clase PositionControlMode. El tipo de datos no posee parámetros ajustables para la adaptación del robot.

Si el modo del regulador de una movimiento no se indica de forma explícita, se utilizará el regulador de posición.

17.5 Regulador de impedancia cartesiano

El regulador de impedancia cartesiano se representa mediante la clase CartesianImpedanceControlMode .

La regulación de impedancia hace referencia por defecto al sistema de coordenadas con el que se ejecuta el comando de movimiento.

Ejemplos:

- `robot.move(...);`
La regulación de impedancia hace referencia al sistema de coordenadas de la brida del robot.
- `gripper.move(...);`
La regulación de impedancia hace referencia al sistema de coordenadas de la herramienta de la garra utilizado actualmente o al Frame estándar establecido para la garra para movimientos.
- `gripper.getFrame("/TipCenter").move(...);`
La regulación de impedancia hace referencia al sistema de coordenadas de la herramienta que se fija por el Frame "TipCenter" en la garra.

Comportamiento del robot

En la regulación de impedancia, el robot se comporta de forma flexible. Es preciso y puede reaccionar ante influencias externas como obstáculos o fuerzas de proceso. El robot puede desviarse de la trayectoria planificada debido a la actuación de una fuerza externa.

El modelo subyacente son resortes y amortiguadores virtuales que se fijan mediante la diferencia entre la posición medida actualmente y la posición ordenada del TCP. Las propiedades de los resortes se describen mediante rígideces y las de los amortiguadores mediante amortiguaciones. Estos parámetros se pueden ajustar individualmente para cada dimensión translatoria y rotatoria.



Si el robot se desplaza con regulación de impedancia, no se puede garantizar la configuración del robot programada, p. ej. el estado.

17.5.1 Cálculo de las fuerzas según la ley de elasticidad

Si la posición medida y ordenada del robot coinciden, los resortes virtuales estarán tensados. Debido a que el robot se comporta de forma flexible si actúa una fuerza externa o debido a un comando de desplazamiento se produce una desviación entre la posición nominal y real del robot. Como consecuencia se produce una desviación de los resortes virtuales, lo que provoca una fuerza según la ley de elasticidad.

La fuerza F resultante se puede calcular según la ley de elasticidad con la rigidez del resorte C ajustada y la desviación Δx :

$$F = C * \Delta x$$

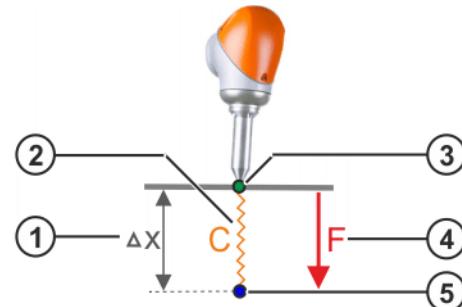


Fig. 17-1: Resorte virtual con rigidez del resorte C

- | | |
|-------------------------|-------------------------|
| 1 Desviación Δx | 4 Fuerza F resultante |
| 2 Resorte virtual | 5 Posición de consigna |
| 3 Posición real | |

Si el robot se encuentra en una resistencia, ejercerá la fuerza calculada. Si se encuentra en un espacio libre, se desplazará hasta la posición ordenada, produciéndose aquí también desviaciones de la trayectoria debido a las fuerzas de fricción internas que actúan en las uniones articuladas, cuya extensión depende de la rigidez del resorte ajustada. Las rigideces mayores provocan desviaciones más reducidas.

Si el robot se encuentra ya en la posición ordenada y se ejerce una fuerza externa sobre el sistema, se desviará de acuerdo con dicha fuerza hasta que las fuerzas de la regulación de elasticidad compensen las fuerzas externas.

Ejemplos

La fuerza que se ejerce con el contacto depende de la diferencia entre la posición nominal y real y de la rigidez ajustada.

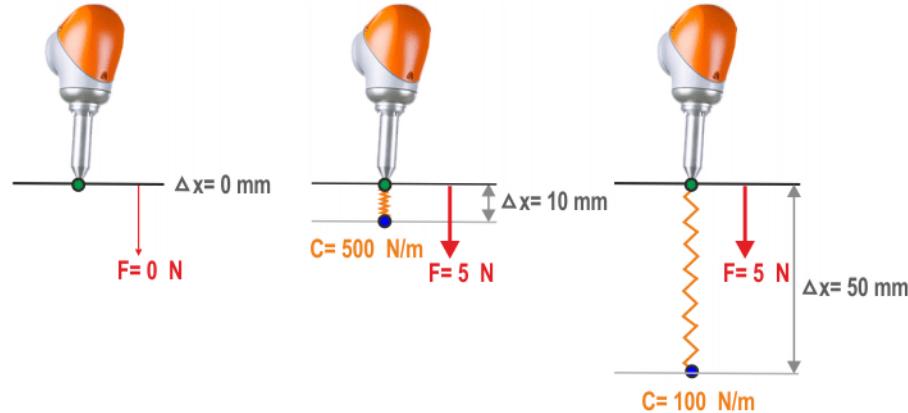


Fig. 17-2: Fuerza ejercida en caso de contacto

Como se muestra en la figura ([>>> Fig. 17-2](#)), con una diferencia de posición grande y una rigidez reducida se puede generar la misma fuerza que con una diferencia de posición reducida y una rigidez más alta. Si la fuerza se genera

por un movimiento en una situación de contacto, con la misma velocidad cartesiana se diferenciará el tiempo que resulta necesario para alcanzar la fuerza.

Si se utilizan rigideces más altas, se podrá alcanzar antes una fuerza deseada, ya que solo es necesaria una diferencia de posición reducida. Debido a que la posición nominal se alcanza rápidamente, se puede generar un tirón de este modo.



Fig. 17-3: Fuerza en el transcurso del tiempo (rigidez elevada, diferencia de posición reducida)

Con una diferencia de posición grande y una rigidez reducida, la fuerza se genera de un modo más lento. Esto se puede utilizar, p. ej., cuando el robot se desplaza hasta hacer contacto y las cargas se deben reducir para el contacto.

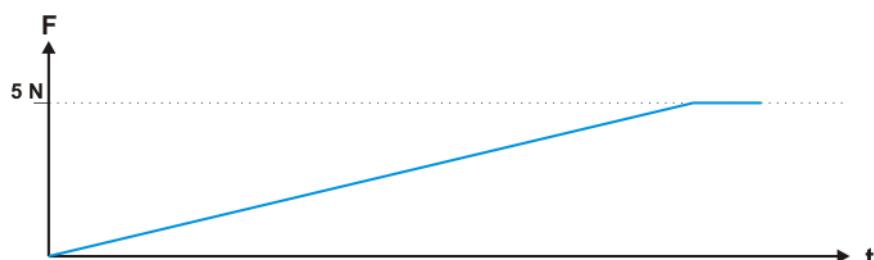


Fig. 17-4: Fuerza en el transcurso del tiempo (rigidez reducida, diferencia de posición grande)

Las desviaciones nominal-real en varias direcciones provocan la desviación de todos los resortes virtuales afectados. El valor y la dirección de la fuerza total se obtienen de la suma de los vectores de las fuerzas individuales para cada dirección.

La desviación de Δx en la dirección x y de Δy en la dirección y provocan la fuerza F_x en dirección x y la fuerza F_y en dirección y. Mediante la suma de vectores se obtiene la fuerza total de F_{res} .

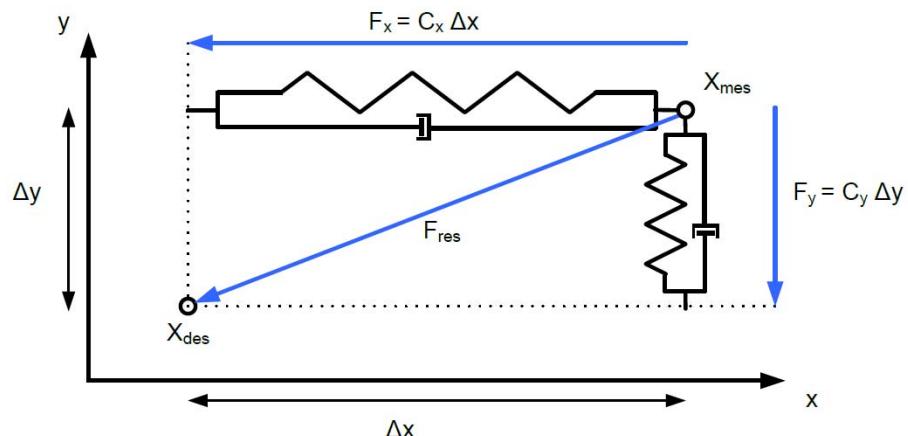


Fig. 17-5: Fuerza total en caso de desviación en 2 direcciones

17.5.2 Parametrización del regulador de impedancia

En la regulación de impedancia, el robot se comporta como un resorte. Las propiedades de este resorte se describen a través de diferentes parámetros. De ellos resulta el comportamiento del robot.

Con un regulador de impedancia cartesiano se pueden conectar fuerzas para todos los grados de libertad cartesianos. Mediante las fuerzas que actúan en torno a un eje, se genera un par de torsión. Por ello, para los grados de libertad rotatorios no se indica la fuerza conectada adicionalmente, sino el momento conectado adicionalmente. A modo de simplificación, los siguientes términos "fuerza" y "oscilación de fuerza" y los términos "momento" y "oscilación del momento" incluyen los grados de libertad rotatorios.

⚠ ADVERTENCIA

En la regulación de impedancia, los parámetros seleccionados incorrectamente (p. ej. datos de carga defectuosos, herramienta incorrecta) o la información incorrecta (p. ej. sobre sensores de momentos defectuosos) se pueden interpretar como fuerzas externas y pueden provocar movimientos inesperados del robot.

Las siguientes propiedades del regulador se pueden determinar individualmente para cada grado de libertad cartesiano:

- Rigidez
- Amortiguación
- Adicional a la fuerza que actúa en el resorte

Las siguientes propiedades del regulador se pueden determinar independientemente del grado de libertad:

- Rigidez del grado de libertad de redundancia
- Amortiguación del grado de libertad de redundancia
- Limitación de la fuerza máxima en el TCP
- Máxima velocidad cartesiana
- Máxima desviación cartesiana de la trayectoria

17.5.2.1 Representación de los grados de libertad cartesianos

En la RoboticsAPI se representan los grados de libertad del regulador de impedancia cartesiano mediante el Enum CartDOF (paquete com.kuka.robotics-API.geometricModel). Con los valores de este Enum se puede describir

individualmente cada grado de libertad o la combinación de varios grados de libertad.

Valor del Enum	Descripción
CartDOF.X	Grado de libertad translatorio en dirección X
CartDOF.Y	Grado de libertad translatorio en dirección Y
CartDOF.Z	Grado de libertad translatorio en dirección Z
CartDOF.TRANS	Combinación de los grados de libertad translatorios en las direcciones X, Y y Z
CartDOF.A	Grado de libertad rotatorio alrededor del eje Z
CartDOF.B	Grado de libertad rotatorio alrededor del eje Y
CartDOF.C	Grado de libertad rotatorio alrededor del eje X
CartDOF.ROT	Combinación de los grados de libertad rotatorios alrededor del eje Z, Y y X
CartDOF.ALL	Combinación de todos los grados de libertad cartesianos

17.5.2.2 Determinar parámetros del regulador para grados de libertad individuales

Descripción Algunos parámetros del regulador de impedancia cartesiano se pueden definir individualmente para cada grado de libertad cartesiano.

Durante la programación, primero se indican los grados de libertad cartesianos para los que debe aplicarse el parámetro del regulador. Para ello se utiliza el método parametrize(...) de los tipos de datos del regulador. Para el establecimiento de los grados de libertad, se transmiten uno o varios parámetros del tipo CartDOF a este método.

A continuación, se activa el método set del parámetro del regulador deseado a través del operador del punto. Este parámetro del regulador se ajusta para todos los grados de libertad indicados en parametrize(...), de acuerdo con el valor indicado como parámetro de entrada en el método set.

Sintaxis

```
controlMode.parametrize(CartDOF.degreeOfFreedom_1
<, CartDOF.degreeOfFreedom_2,...>).setParameter(value);
```

Aclaración de la sintaxis

Elemento	Descripción
controlMode	Tipo: CartesianImpedanceControlMode Nombre del objeto del regulador
degreeOfFreedom_1, degreeOfFreedom_2, ...	Tipo: CartDOF Enumeración de los grados de libertad que deben describirse
setParameter(value)	Método para ajustar un parámetro del regulador Para cada <i>Parámetro</i> ajustable se encuentra disponible un método propio (<i>value</i> = valor del parámetro).

Ejemplo

Se debe ejecutar un movimiento LIN con regulación de impedancia hacia un punto definido. El regulador de impedancia cartesiano está configurado de forma que el TCP utilizado actualmente, aquí el Frame de la brida del robot, sea flexible en dirección Z.

```
CartesianImpedanceControlMode cartImpCtrlMode = new
CartesianImpedanceControlMode();

cartImpCtrlMode.parametrize(CartDOF.X,
CartDOF.Y).setStiffness(3000.0);
```

```
cartImpCtrlMode.parametrize(CartDOF.Z).setStiffness(1.0);
cartImpCtrlMode.parametrize(CartDOF.ROT).setStiffness(300.0);
cartImpCtrlMode.parametrize(CartDOF.ALL).setDamping(0.7);

lbr.move(lingetApplicationData().getFrame("/")
P1")).setCartVelocity(800).setMode(cartImpCtrlMode));
```

17.5.2.3 Parámetros del regulador específicos del grado de libertad

Resumen

Los siguientes métodos están disponibles para los parámetros específicos del grado de libertad del regulador de impedancia cartesiano:

Método	Descripción
setStiffness(...)	<p>Rigidez del resorte (tipo: double)</p> <p>La rigidez del resorte determina en qué medida cede el robot cuando actúa una fuerza y en qué medida se desvía de la trayectoria planificada.</p> <p>Grados de libertad translatorios (unidad: N/m):</p> <ul style="list-style-type: none">■ 0.0 ... 5000.0 <p>Por defecto: 2000.0</p> <p>Grados de libertad rotatorios (unidad: Nm/rad):</p> <ul style="list-style-type: none">■ 0.0 ... 300.0 <p>Por defecto: 200.0</p> <p>Nota: Si no se establece ninguna rigidez del resorte para el grado de libertad, se utilizará el valor por defecto para este grado de libertad.</p>

Método	Descripción
setDamping(...)	<p>Amortiguación por resorte (tipo: double)</p> <p>La amortiguación por resorte determina en qué medida oscilan los resortes virtuales tras la desviación.</p> <p>Para todos los grados de libertad (sin unidad: medida de amortiguación de Lehr):</p> <ul style="list-style-type: none"> ■ 0.1 ... 1.0 <p>Por defecto: 0.7</p> <p>Nota: Si no se establece ninguna amortiguación por resorte para el grado de libertad, se utilizará el valor por defecto para este grado de libertad.</p>
setAdditionalControl-Force(...)	<p>Adicional de la fuerza que actúa en el resorte (tipo: double)</p> <p>Mediante la fuerza adicional se genera una fuerza cartesiana en el TCP. Dicha fuerza actúa de forma adicional a las fuerzas que se generan debido a la rigidez del resorte.</p> <p>Grados de libertad translatorios (unidad: N):</p> <ul style="list-style-type: none"> ■ Posibilidad de valores negativos y positivos. <p>Por defecto: 0.0</p> <p>Grados de libertad rotatorios (unidad: Nm):</p> <ul style="list-style-type: none"> ■ Posibilidad de valores negativos y positivos. <p>Por defecto: 0.0</p> <p>Nota: Si no se establece ninguna fuerza adicional para el grado de libertad, se utilizará el valor por defecto para este grado de libertad.</p> <p>Nota: La fuerza se activa sin retardo. Si la fuerza que debe activarse es demasiado grande, se puede producir la sobrecarga del robot y la cancelación del programa. La clase <code>CartesianSinelImpedanceControlMode</code> dispone de la posibilidad de activar fuerzas con retardo temporal.</p>

17.5.2.4 Parámetros del regulador independientes del grado de libertad

Algunos ajustes se aplican independientemente de los grados de libertad cartesianos. Los métodos de ajuste con los que se establecen estos parámetros del regulador pertenecen a la clase `CartesianImpedanceControlMode` y se activan directamente en el objeto del regulador.

Resumen

Los siguientes métodos están disponibles para los parámetros independientes del grado de libertad del regulador de impedancia cartesiano:

Método	Descripción
setNullSpaceStiffness(...)	<p>Rigidez del resorte del grado de libertad de la redundancia (tipo: double; unidad: Nm/rad)</p> <p>La rigidez del resorte determina en qué medida cede el robot cuando actúa una fuerza y en qué medida se desvía de la trayectoria planificada.</p> <ul style="list-style-type: none"> ■ ≥ 0.0 <p>Indicación: Si no se establece ninguna rigidez del resorte para el grado de libertad de la redundancia, se utilizará un valor por defecto para este grado de libertad.</p>
setNullSpaceDamping(...)	<p>Amortiguación por resorte del grado de libertad de la redundancia (tipo: double)</p> <p>La amortiguación por resorte determina en qué medida oscilan los resortes virtuales tras la desviación.</p> <ul style="list-style-type: none"> ■ 0.3 ... 1.0 <p>Indicación: Si no se establece ninguna amortiguación por resorte para el grado de libertad de la redundancia, se utilizará un valor por defecto para este grado de libertad.</p>
setMaxControlForce(...)	<p>Limitación de la fuerza máxima en el TCP</p> <p>Se limita la fuerza que se genera como máximo en el TCP por los resortes virtuales. De este modo, también está definida la fuerza que es necesaria como máximo para desviar el resorte virtual. Adicionalmente se establece si el movimiento debe cancelarse en caso de superarse la máxima fuerza en el TCP.</p> <p>Sintaxis:</p> <ul style="list-style-type: none"> ■ <code>setMaxControlForce (maxForceX, maxForceY, maxForceZ, maxTorqueA, maxTorqueB, maxTorqueC, addStopCondition)</code> <p>Aclaración de la sintaxis:</p> <ul style="list-style-type: none"> ■ <i>maxForceXY1Z</i>: Máxima fuerza en el TCP en la dirección cartesiana correspondiente (tipo: double, unidad: N) <ul style="list-style-type: none"> ■ ≥ 0.0 ■ <i>maxTorqueA1B1C</i>: Máximo momento en el TCP en la dirección rotatoria correspondiente (tipo: double, unidad Nm) <ul style="list-style-type: none"> ■ ≥ 0.0 ■ <i>addStopCondition</i>: Cancelación del movimiento en caso de superación de la máxima fuerza en el TCP (tipo: boolean) <ul style="list-style-type: none"> ■ true: Se cancela el movimiento. ■ false: No se cancela el movimiento.

Método	Descripción
setMaxCartesianVelocity(...)	<p>Máxima velocidad cartesiana</p> <p>El movimiento se cancela cuando se supera el límite de velocidad establecido.</p> <p>Sintaxis:</p> <ul style="list-style-type: none"> ■ <code>setMaxCartesianVelocity(maxVelocityX, maxVelocityY, maxVelocityZ, maxVelocityA, maxVelocityB, maxVelocityC)</code> <p>Aclaración de la sintaxis:</p> <ul style="list-style-type: none"> ■ <i>maxVelocityXYZ</i>: Máxima velocidad translatoria permitida en el TCP en la dirección cartesiana correspondiente (tipo: double, unidad: mm/s) <ul style="list-style-type: none"> ■ ≥ 0.0 ■ <i>maxVelocityAIBIC</i>: Máxima velocidad rotatoria permitida en el TCP en la dirección rotatoria correspondiente (tipo: double, unidad: rad/s) <ul style="list-style-type: none"> ■ ≥ 0.0
setMaxPathDeviation(...)	<p>Máxima desviación cartesiana de la trayectoria</p> <p>Establece la desviación cartesiana de la trayectoria con respecto a la posición nominal planificada actualmente, que puede producirse como máximo en caso de un movimiento flexible. El movimiento se cancela cuando se supera la máxima desviación de la trayectoria establecida.</p> <p>Sintaxis:</p> <ul style="list-style-type: none"> ■ <code>setMaxPathDeviation(maxDeviationX, maxDeviationY, maxDeviationZ, maxDeviationA, maxDeviationB, maxDeviationC)</code> <p>Aclaración de la sintaxis:</p> <ul style="list-style-type: none"> ■ <i>maxDeviationXYZ</i>: Máxima desviación de la trayectoria permitida en el TCP en la dirección cartesiana correspondiente (tipo: double, unidad: mm) <ul style="list-style-type: none"> ■ ≥ 0.0 ■ <i>maxDeviationAIBIC</i>: Máxima desviación rotatoria permitida en el TCP en la dirección rotatoria correspondiente (tipo: double, unidad: rad) <ul style="list-style-type: none"> ■ ≥ 0.0

Ejemplo 1

Un robot con regulación de impedancia debe comportarse de forma flexible en un grado de libertad redundante para poder reaccionar ante posibles obstáculos durante el movimiento. Para ello se parametrizan la rigidez y la amortiguación del grado de libertad redundante para el regulador de impedancia.

```
CartesianImpedanceControlMode mode = new
CartesianImpedanceControlMode();

mode.setNullSpaceStiffness(10.0);
mode.setNullSpaceDamping(0.7);
```

Ejemplo 2

Un robot debe pasar de forma flexible a lo largo de una placa de mesa. Para ello se parametriza un regulador de impedancia cartesiano. En la dirección Z del sistema de coordenadas de la herramienta en el TCP se ajusta una rigidez alta. Además se debe generar una fuerza adicional de 20 N. El movimiento se cancela cuando se supera un límite de fuerza de 50 N en dirección Z. En el nivel XY se ajusta una rigidez baja. No obstante, la desviación cartesiana en dirección X e Y no puede ser superior a 1 cm. Para todos los demás parámetros se indican valores mayores adecuados.

```
CartesianImpedanceControlMode mode = new  
CartesianImpedanceControlMode();  
  
mode.parametrize(CartDOF.Z).setStiffness(3000.0);  
mode.parametrize(CartDOF.Z).setAdditionalControlForce(20.0);  
mode.setMaxControlForce(100.0, 100.0, 50.0, 20.0, 20.0, 20.0, true);  
  
mode.parametrize(CartDOF.X, CartDOF.Y).setStiffness(10.0);  
mode.setMaxPathDeviation(10.0, 10.0, 50.0, 2.0, 2.0, 2.0);
```

17.6 Regulador de impedancia cartesiano con oscilación de fuerza conectada adicionalmente

El regulador de impedancia cartesiano con oscilación de fuerza conectada es una forma especial del regulador de impedancia cartesiano. La conexión de fuerza es posible individualmente para cada grado de libertad cartesiano.

Las oscilaciones de fuerza en torno a un eje generan oscilaciones del par de torsión. Mediante la conexión de oscilaciones del momento, se pueden generar oscilaciones giratorias.

La conexión de fuerzas constantes o senoidales provoca el movimiento del robot. Mediante la combinación de las oscilaciones en los grados de libertad individuales, se pueden generar diferentes patrones de movimiento.

Con la ayuda de oscilaciones conectadas, p. ej., se pueden realizar movimientos oscilantes flexibles para desplazamientos de búsqueda y vibraciones en la herramienta para procesos de ensamblado.

El regulador de impedancia cartesiano con oscilación de fuerza conectada se representa mediante la clase `CartesianSinelImpedanceControlMode`.

Comportamiento del robot

Con esta forma de regulación de impedancia, el robot se aleja de forma precisa de la trayectoria planificada mediante la conexión de una fuerza. El nuevo recorrido de la trayectoria se determina por gran número de diferentes parámetros.

De forma adicional a la rigidez y a la amortiguación, también se pueden definir otros parámetros, p. ej. la frecuencia y la amplitud. La velocidad programada del robot también tiene una función esencial para el recorrido concreto de la trayectoria.



La conexión de fuerzas adicionales afecta notablemente al movimiento del robot y a las fuerzas que ejerce el robot. Por ejemplo, con una rigidez reducida y fuerzas conectadas elevadas, puede darse una aceleración fuerte del robot.

Por lo tanto, si se trabaja con conexiones de fuerza, hay que realizar la parametrización con mucho cuidado. Por ejemplo, empezando con la conexión de fuerzas reducidas y acercándose poco a poco a los valores de fuerza adecuados. Además, el movimiento que surge de la fuerza conectada siempre debe probarse primero en el modo de servicio T1.

17.6.1 Conexión de una oscilación de fuerza simple

Mediante la conexión de una oscilación de fuerza simple se desvía el punto de trabajo de la trayectoria planificada (= curso de la trayectoria sin oscilaciones conectadas) y en su lugar se guía en una trayectoria ondulada desde el punto inicial hasta el punto final del movimiento.

Ejemplo

El robot ejecuta un movimiento relativo en la dirección Y del sistema de coordenadas de la herramienta en el TCP. Para ello se conecta una oscilación de

fuerza senoidal en dirección X. El resultado es una trayectoria ondulada en el nivel XY del sistema de coordenadas.

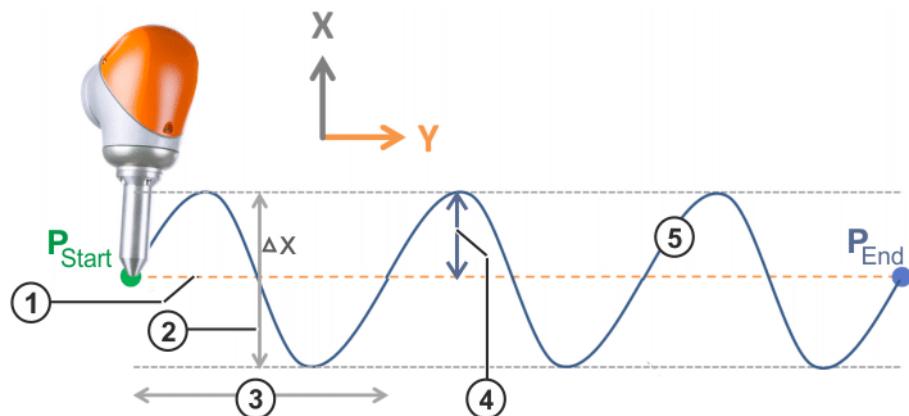


Fig. 17-6: Conexión de una oscilación de fuerza simple

- | | | | |
|---|-----------------------|---|-------------------|
| 1 | Trayectoria original | 4 | Amplitud |
| 2 | Desviación Δx | 5 | Nueva trayectoria |
| 3 | Longitud de onda | | |

La máxima desviación Δx es la variación de la trayectoria original en la dirección X positiva y negativa. La máxima desviación se determina mediante la rigidez y la amplitud que están definidas para el regulador de impedancia en la dirección X cartesiana, p. ej.:

- Rígidez cartesiana: $C = 500 \text{ N/m}$
- Amplitud: $F = 5 \text{ N}$

La máxima desviación resulta de la ley de elasticidad:

$$\Delta x = F / C = 5 \text{ N} / (500 \text{ N/m}) = 1 / (100 \text{ 1/m}) = 1 \text{ cm}$$

Mediante la longitud de onda se puede determinar el número de oscilaciones que el robot debe ejecutar entre el punto inicial y el punto final del movimiento. La longitud de onda se determina mediante la frecuencia que está definida para el regulador de impedancia con oscilación de fuerza conectada y a través de la velocidad programada del robot.

La longitud de onda λ se calcula del siguiente modo:

$$\lambda = c / f = \text{velocidad del robot} / \text{frecuencia}$$

17.6.2 Conexión de oscilaciones de fuerza superiores (figuras de Lissajous)

Las figuras de Lissajous se generan cuando una oscilación de fuerza senoidal se conecta en 2 direcciones cartesianas diferentes. Mediante la superposición de las dos oscilaciones se pueden generar las formas más variadas para el recorrido de la trayectoria. El recorrido exacto de la trayectoria depende de un gran número de parámetros.

Aplicación

Con la superposición de 2 oscilaciones de fuerza senoidales que se diferencian en cuanto a la frecuencia, se pueden generar vibraciones en el TCP. Si, por ejemplo, durante un proceso de montaje se producen tensiones y atascamientos, se podrán eliminar mediante las vibraciones generadas de este modo.

Ejemplo

Se conecta respectivamente una oscilación de fuerza senoidal en dirección X y en dirección Y del sistema de coordenadas de la herramienta en el TCP. Las máximas desviaciones Δx y Δy se determinan mediante la rigidez y la ampli-

tud que están definidas para el regulador de impedancia en la dirección X e Y cartesianas.

De forma adicional a los parámetros ya conocidos del regulador de impedancia, el desfase entre las dos oscilaciones no tiene una importancia esencial para el recorrido de la trayectoria.

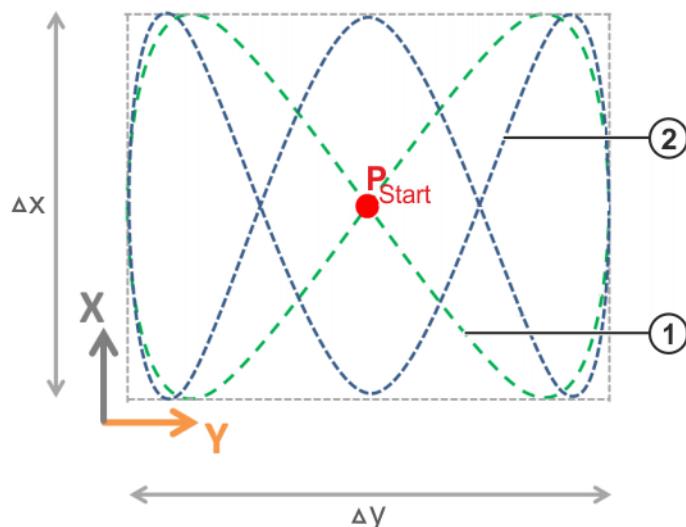


Fig. 17-7: Curso de trayectoria en una figura de Lissajous

- 1 Recorrido de la trayectoria sin desfase (relación de frecuencia X:Y = 2:1)
- 2 Recorrido de la trayectoria con desfase (relación de frecuencia X:Y = 3:1)

La forma del recorrido de la trayectoria se determina principalmente por la relación de las dos frecuencias y del desfase entre las dos oscilaciones. La figura resultante siempre es simétrica con respecto a los ejes y a los puntos. A partir de la amplitud de fuerza y la rigidez ajustadas para una dirección de oscilación se obtiene su amplitud de posición. La relación de las dos amplitudes determina la relación de la anchura con respecto a la altura de la figura.

17.6.3 Parametrización del regulador de impedancia con oscilación de fuerza conectada adicionalmente

El regulador de impedancia cartesiano con oscilación de fuerza conectada es una forma especial del regulador de impedancia estándar.

Con un regulador de impedancia cartesiano con oscilación de fuerza conectada adicionalmente se pueden conectar adicionalmente fuerzas para todos los grados de libertad cartesianos. Mediante las fuerzas que actúan en torno a un eje, se genera un par de torsión. Por ello, para los grados de libertad rotatorios no se indica la fuerza conectada adicionalmente, sino el momento conectado adicionalmente. A modo de simplificación, los siguientes términos "fuerza" y "oscilación de fuerza" y los términos "momento" y "oscilación del momento" incluyen los grados de libertad rotatorios.

ADVERTENCIA

En la regulación de impedancia, los parámetros seleccionados incorrectamente (p. ej. datos de carga defectuosos, herramienta incorrecta) o la información incorrecta (p. ej. sobre sensores de momentos defectuosos) se pueden interpretar como fuerzas externas y pueden provocar movimientos inesperados del robot.

El regulador de impedancia cartesiano con oscilación de fuerza conectada adicionalmente se parametriza de forma análoga al regulador de impedancia

estándar. Los parámetros del regulador específicos para el grado de libertad e independientes del grado de libertad descritos para el regulador de impedancia estándar, se pueden utilizar igualmente para el regulador de impedancia con oscilación de fuerza conectada adicionalmente.

(>>> 17.5.2 "Parametrización del regulador de impedancia" Página 394)



Excepción: El método `setAdditionalControlForce(...)` de la clase `CartesianImpedanceControlMode` para conectar adicionalmente una fuerza que se debe aplicar de forma adicional al resorte, está disponible para la clase `CartesianSineImpedanceControlMode`, aunque no se debe utilizar.

Para conectar adicionalmente fuerzas constantes, se encuentra disponible el método `setBias(...)` en la clase `CartesianSineImpedanceControlMode`.

Las siguientes propiedades adicionales del regulador se pueden determinar individualmente para cada grado de libertad cartesiano:

- Amplitud de la oscilación de fuerza
- Frecuencia de la oscilación de fuerza
- Desfase de la oscilación de fuerza
- Fuerza constante superior
- Limitación de fuerza de la oscilación de fuerza
- Limitación de la desviación mediante la oscilación de fuerza

Las siguientes propiedades adicionales del regulador se pueden determinar de forma independiente con respecto del grado de libertad:

- Tiempo de subida de la oscilación de fuerza
- Tiempo de parada de la oscilación de fuerza
- Tiempo de disminución de la oscilación de fuerza
- Duración total de la oscilación de fuerza

17.6.3.1 Parámetros del regulador específicos del grado de libertad

Resumen

Los siguientes métodos están disponibles para los parámetros específicos del grado de libertad del regulador de impedancia cartesiano con oscilación de fuerza conectada:

Método	Descripción
setAmplitude(...)	<p>Amplitud de la oscilación de fuerza (tipo: double) La amplitud y la rigidez determinan la amplitud de posición. Grados de libertad translatorios (unidad: N):</p> <ul style="list-style-type: none"> ■ ≥ 0.0 Por defecto: 0.0 <p>Grados de libertad rotatorios (unidad: Nm):</p> <ul style="list-style-type: none"> ■ ≥ 0.0 Por defecto: 0.0 <p>Nota: Si no se establece ninguna amplitud para el grado de libertad, se utilizará el valor por defecto para este grado de libertad.</p>
setFrequency(...)	<p>Frecuencia de la oscilación de fuerza (tipo: double; unidad: Hz) La frecuencia y la velocidad cartesiana determinan la longitud de onda de la oscilación de fuerza.</p> <ul style="list-style-type: none"> ■ 0.0 ... 15.0 Por defecto: 0.0 <p>Nota: Si no se establece ninguna frecuencia para el grado de libertad, se utilizará el valor por defecto para este grado de libertad.</p>
setPhaseDeg(...)	<p>Desfase de la oscilación de fuerza al principio de la conexión de fuerza (tipo: double; unidad: °)</p> <ul style="list-style-type: none"> ■ ≥ 0.0 Por defecto: 0.0 <p>Nota: Si no se establece ningún desfase para el grado de libertad, se utilizará el valor por defecto para este grado de libertad.</p>
setBias(...)	<p>Fuerza constante conectada (tipo: double) Mediante setBias(...) se puede conectar una fuerza constante aparte de la oscilación de fuerza conectada. Esta fuerza se suma a la fuerza que surge de la rigidez del muelle y la oscilación de fuerza establecida. Si se conecta una fuerza constante sin oscilación de fuerza adicional, se obtiene una curva de fuerza que asciende dependiendo del tiempo de ascenso ajustado mediante setRiseTime(...) y que después de mantenerse constante. setRiseTime(...) es uno de los parámetros de regulación independientes del grado de libertad (>>> 17.6.3.1 "Parámetros del regulador específicos del grado de libertad" Página 403). Si una fuerza constante se conecta adicionalmente a una oscilación de fuerza, la oscilación de fuerza se desplaza en la dirección definida. Grados de libertad translatorios (unidad: N):</p> <ul style="list-style-type: none"> ■ Posibilidad de valores negativos y positivos. Por defecto: 0.0 <p>Grados de libertad rotatorios (unidad: Nm):</p> <ul style="list-style-type: none"> ■ Posibilidad de valores negativos y positivos. Por defecto: 0.0 <p>Nota: Si no se conecta ninguna fuerza constante adicional para el grado de libertad, se utilizará el valor por defecto para este grado de libertad.</p>

Método	Descripción
setForceLimit(...)	<p>Limitación de fuerza de la oscilación de fuerza (tipo: double)</p> <p>Establece el valor límite que no puede superar la fuerza total, es decir, la suma de la amplitud de la oscilación de fuerza y la fuerza constante conectada adicionalmente. Si la fuerza total supera el valor límite, se reducirá la fuerza conectada al valor límite.</p> <p>Grados de libertad translatorios (unidad: N):</p> <ul style="list-style-type: none"> ■ ≥ 0.0 Por defecto: No limitada. <p>Grados de libertad rotatorios (unidad: Nm):</p> <ul style="list-style-type: none"> ■ ≥ 0.0 Por defecto: No limitada. <p>Nota: Si no se establece ninguna limitación de fuerza para el grado de libertad, se utilizará el valor por defecto para este grado de libertad.</p>
setPositionLimit(...)	<p>Máxima desviación a través de la oscilación de fuerza (tipo: double)</p> <p>En caso de superación de la máxima desviación permitida, se desconectará la fuerza. En cuanto el robot se encuentra de nuevo en la zona permitida, se vuelve a conectar la fuerza.</p> <p>Grados de libertad translatorios (unidad: mm):</p> <ul style="list-style-type: none"> ■ ≥ 0.0 Por defecto: No limitada. <p>Grados de libertad rotatorios (unidad: rad):</p> <ul style="list-style-type: none"> ■ ≥ 0.0 Por defecto: No limitada. <p>Nota: Si no se establece ninguna máxima desviación para el grado de libertad, se utilizará el valor por defecto para este grado de libertad.</p>

Ejemplo

Durante un proceso de ensamblaje se debe ejecutar una oscilación alrededor del eje Z del sistema de coordenadas de la herramienta en el TCP. Para ello se utiliza el regulador de impedancia cartesiano con oscilación de fuerza conectada. Con una rigidez de 10 Nm/rad y una amplitud de 15 Nm, la amplitud de posición es de aprox. 1.5 rad. La frecuencia se ajusta a 5 Hz. Para ejercer una fuerza de presión adicional en la dirección del movimiento, se genera una fuerza constante de von 5 N en dirección Z, que se superpone a la oscilación de fuerza conectada alrededor del eje Z.

```
CartesianSineImpedanceControlMode sineMode = new
CartesianSineImpedanceControlMode();

sineMode.parametrize(CartDOF.Z).setStiffness(4000.0);
sineMode.parametrize(CartDOF.Z).setBias(5.0);

sineMode.parametrize(CartDOF.A).setStiffness(10.0);
sineMode.parametrize(CartDOF.A).setAmplitude(15.0);
sineMode.parametrize(CartDOF.A).setFrequency(5.0);

tool.getFrame("/TCP").move(linRel(0.0, 0.0,
10.0).setCartVelocity(10.0).sineMode(sineMode));
```

17.6.3.2 Parámetros del regulador independientes del grado de libertad

Algunos ajustes se aplican independientemente de los grados de libertad cartesianos. Los métodos de ajuste con los que se establecen estos parámetros

del regulador pertenecen a la clase `CartesianSinelImpedanceControlMode` y se activan directamente en el objeto del regulador.

Resumen

Los siguientes métodos están disponibles para los parámetros independientes del grado de libertad del regulador de impedancia cartesiano con oscilación de fuerza conectada:

Método	Descripción
<code>setTotalTime(...)</code>	Duración total de la oscilación de fuerza (tipo: double; unidad: s) <code>>>></code> "Duración total de la oscilación de fuerza" Página 406) <ul style="list-style-type: none"> ■ ≥ 0.0 <p>Por defecto: Ilimitado</p>
<code>setRiseTime(...)</code>	Tiempo de subida de la oscilación de fuerza (tipo: double; unidad: s) <ul style="list-style-type: none"> ■ ≥ 0.0 <p>Por defecto: 0.0</p> <p>Indicación: Si no se establece ningún tiempo de subida, se utilizará el valor por defecto. Esto significa que la amplitud aumenta sin transición y repentinamente hasta el valor establecido. Si la fuerza que debe activarse es demasiado grande, se puede producir la sobrecarga del robot y la cancelación del programa.</p>
<code>setHoldTime(...)</code>	Tiempo de parada de la oscilación de fuerza (tipo: double; unidad: s) <ul style="list-style-type: none"> ■ ≥ 0.0 <p>Por defecto: Ilimitado</p> <p>Indicación: Si no se establece ningún tiempo de parada, se utilizará el valor por defecto. Esto significa que la oscilación de fuerza conectada finaliza con el movimiento correspondiente.</p>
<code>setFallTime(...)</code>	Tiempo de disminución de la oscilación de fuerza (tipo: double; unidad: s) <ul style="list-style-type: none"> ■ ≥ 0.0 <p>Por defecto: 0.0</p> <p>Indicación: Si no se establece ningún tiempo de disminución, se utilizará el valor por defecto. Esto significa que la amplitud disminuye sin transición y repentinamente hasta cero. Si la disminución de fuerza es demasiado grande, se puede producir la sobrecarga del robot y la cancelación del programa.</p>
<code>setStayActiveUntil-PatternFinished(...)</code>	Comportamiento en caso de superación de la duración del movimiento (tipo: boolean) <p>En caso de que la oscilación de fuerza dure más tiempo que el movimiento, puede establecerse si la oscilación debe cancelarse o debe continuar después del final del movimiento.</p> <ul style="list-style-type: none"> ■ true: La oscilación continúa después del final del movimiento. ■ false: La oscilación se cancela con el final del movimiento. <p>Por defecto: false</p> <p>Indicación: Si no se establece el comportamiento en caso de superación de la duración del movimiento, se utilizará el valor por defecto.</p>

Duración total de la oscilación de fuerza

La duración total es la suma del tiempo de subida, el tiempo de parada y el tiempo de disminución de la oscilación de fuerza:

- Tiempo de subida

Tiempo en el que se genera la amplitud de la oscilación de fuerza

- Tiempo de parada
Tiempo en el que la oscilación de fuerza se ejecuta con la amplitud establecida
- Tiempo de disminución
Tiempo en el que la amplitud de la oscilación de fuerza vuelve a disminuir hasta cero

El tiempo de subida, el tiempo de parada y el tiempo de disminución de la oscilación de fuerza se pueden establecer individualmente o indirectamente, definiendo la duración total de la oscilación de fuerza.

Si la duración total se define mediante `setTotalTime(...)`, el tiempo de subida y de disminución se establecen automáticamente.

Cálculo:

- Tiempo de subida = tiempo de disminución = $(1/\text{frecuencia}) * 0,5$
- Entre las frecuencias definidas para la oscilación de fuerza (referidas a todos los grados de libertad) se utiliza para el cálculo aquella frecuencia, que ofrezca como resultado el tiempo de subida y de disminución más amplios posibles.
- Si se conectan exclusivamente fuerzas constantes, la frecuencia de todos los grados de libertad será de 0,0 Hz. El tiempo de subida y de disminución se ajusta a 0,0 s.
- Si la suma del tiempo de subida y de disminución calculada de este modo supera la duración total establecida, el tiempo de subida y de disminución se ajustarán respectivamente al 25 % y el tiempo de parada se ajustará al 50 % del tiempo total.

Si la duración completa de la oscilación de fuerza es más corta que la duración del movimiento correspondiente, finalizará la oscilación de fuerza antes del final del movimiento. El comportamiento en caso de superación de la duración del movimiento se establece mediante `setStayActiveUntilPatternFinished(...)`.

17.7 Métodos estáticos para el regulador de impedancia con oscilación de fuerza superior.

Resumen

El regulador de impedancia cartesiano con oscilación de fuerza conectada también se puede configurar a través del método de la clase `CartesianSineImpedanceControlMode`. Esto facilita la programación, especialmente en las figuras de Lissajous, ya que el usuario solo debe indicar pocos parámetros. Los restantes parámetros importantes para la implementación se calculan y se ajustan automáticamente. Para todos los demás parámetros se utilizan valores por defecto. Los ajustes adicionales se realiza, como se ha descrito, mediante la función `parametrize(...)` y los métodos `set` del `CartesianSineImpedanceControlMode`.

- `createDesiredForce(...)`: Método estático para fuerza constante
- `createSinePattern(...)`: Método estático para oscilaciones de fuerza simples
- `createLissajousPattern(...)`: Método estático para figuras de Lissajous
- `createSpiralPattern(...)`: Método estático para espirales

Indicación de planos cartesianos

En comparación con las oscilaciones simples, en las figuras de Lissajous y las espirales no se transmite ningún grado de libertad individual, sino el plano en el que debe transcurrir la trayectoria. El plano se indica mediante el Enum `CartPlane` (paquete `com.kuka.roboticsAPI.geometricModel`).

Valor del Enum	Descripción
CartPlane.XY	Curso de la trayectoria en el plano XY
CartPlane.XZ	Curso de la trayectoria en el plano XZ
CartPlane.YZ	Curso de la trayectoria en el plano YZ

17.7.1 Conectar fuerza constante

Descripción	Con el método <code>createDesiredForce (...)</code> , en una dirección cartesiana se conecta una fuerza constante no variable en el tiempo.										
Sintaxis	<code>controlMode = CartesianSineImpedanceControlMode.createDesiredForce (CartDOF.degreeOfFreedom, force, stiffness) ;</code>										
Aclaración de la sintaxis	<table border="1"> <thead> <tr> <th>Elemento</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td><i>controlMode</i></td> <td>Tipo: <code>CartesianSineImpedanceControlMode</code> Nombre del objeto del regulador</td></tr> <tr> <td><i>degreeOfFreedom</i></td> <td>Tipo: <code>CartDOF</code> Grado de libertad para que el que ha de conectarse la fuerza constante</td></tr> <tr> <td><i>force</i></td> <td>Tipo: <code>double</code> Valor de la fuerza constante conectada. Corresponde a la activación de <code>setBias(...)</code> para el grado de libertad indicado. Grados de libertad translatorios (unidad: N): ■ ≥ 0.0 Grados de libertad rotatorios (unidad: Nm): ■ ≥ 0.0</td></tr> <tr> <td><i>stiffness</i></td> <td>Tipo: <code>double</code> Valor de rigidez para el grado de libertad indicado Grados de libertad translatorios (unidad: N/m): ■ 0.0 ... 5000.0 Grados de libertad rotatorios (unidad: Nm/rad): ■ 0.0 ... 300.0</td></tr> </tbody> </table>	Elemento	Descripción	<i>controlMode</i>	Tipo: <code>CartesianSineImpedanceControlMode</code> Nombre del objeto del regulador	<i>degreeOfFreedom</i>	Tipo: <code>CartDOF</code> Grado de libertad para que el que ha de conectarse la fuerza constante	<i>force</i>	Tipo: <code>double</code> Valor de la fuerza constante conectada. Corresponde a la activación de <code>setBias(...)</code> para el grado de libertad indicado. Grados de libertad translatorios (unidad: N): ■ ≥ 0.0 Grados de libertad rotatorios (unidad: Nm): ■ ≥ 0.0	<i>stiffness</i>	Tipo: <code>double</code> Valor de rigidez para el grado de libertad indicado Grados de libertad translatorios (unidad: N/m): ■ 0.0 ... 5000.0 Grados de libertad rotatorios (unidad: Nm/rad): ■ 0.0 ... 300.0
Elemento	Descripción										
<i>controlMode</i>	Tipo: <code>CartesianSineImpedanceControlMode</code> Nombre del objeto del regulador										
<i>degreeOfFreedom</i>	Tipo: <code>CartDOF</code> Grado de libertad para que el que ha de conectarse la fuerza constante										
<i>force</i>	Tipo: <code>double</code> Valor de la fuerza constante conectada. Corresponde a la activación de <code>setBias(...)</code> para el grado de libertad indicado. Grados de libertad translatorios (unidad: N): ■ ≥ 0.0 Grados de libertad rotatorios (unidad: Nm): ■ ≥ 0.0										
<i>stiffness</i>	Tipo: <code>double</code> Valor de rigidez para el grado de libertad indicado Grados de libertad translatorios (unidad: N/m): ■ 0.0 ... 5000.0 Grados de libertad rotatorios (unidad: Nm/rad): ■ 0.0 ... 300.0										

17.7.2 Conectar la oscilación de fuerza simple

Descripción	Con el método <code>createSinePattern(...)</code> se conecta una oscilación de fuerza simple en una dirección cartesiana.
Sintaxis	<code>controlMode = CartesianSineImpedanceControlMode.createSinePattern (CartDOF.degreeOfFreedom, frequency, amplitude, stiffness) ;</code>

Aclaración de la sintaxis

Elemento	Descripción
<i>controlMode</i>	Tipo: CartesianSineImpedanceControlMode Nombre del objeto del regulador
<i>degreeOfFreedom</i>	Tipo: CartDOF Grado de libertad para que el que ha de conectarse la oscilación de fuerza
<i>frequency</i>	Tipo: double Frecuencia de la oscilación (unidad: Hz) ■ 0.0 ... 15.0
<i>amplitude</i>	Tipo: double Amplitud de la oscilación que se conecta en la dirección del grado de libertad indicado Grados de libertad translatorios (unidad: N): ■ ≥ 0.0 Grados de libertad rotatorios (unidad: Nm): ■ ≥ 0.0
<i>stiffness</i>	Tipo: double Valor de rigidez para el grado de libertad indicado Grados de libertad translatorios (unidad: N/m): ■ 0.0 ... 5000.0 Grados de libertad rotatorios (unidad: Nm/rad): ■ 0.0 ... 300.0

Ejemplo

Desde la posición actual se debe ejecutar un movimiento relativo de 15 cm en la dirección Y. El movimiento debe transcurrir en una trayectoria ondulada con una desviación de aprox. 10 cm (se obtiene de la amplitud y la rigidez) y una frecuencia de 2 Hz la dirección X.

```
CartesianSineImpedanceControlMode sineMode;

sineMode =
CartesianSineImpedanceControlMode.createSinePattern(CartDOF.X, 2.0,
50.0, 500.0);

lbr.move(linRel(0.0, 150.0,
0.0).setCartVelocity(100).setMode(sineMode));
```

17.7.3 Conectar curva de Lissajous

Descripción

Con el método `createLissajousPattern(...)` se genera una curva bidimensional en un plano. El plano se transmite como valor del tipo `CartPlane`. El resto de parámetros transmitidos se refieren al primer grado de libertad del plano indicado (ejemplo: para `CartPlane.XY`, los datos se refieren a `CartDOF.X`).

Los parámetros del segundo grado de libertad del plano se calculan de forma que el resultado sea una figura de Lissajous con las siguientes propiedades:

- Relación de amplitud 1. Grado de libertad: 2. Grado de libertad: 1 : 1
- Relación de frecuencia 1. Grado de libertad: 2. Grado de libertad: 1 : 0.4
- Desfase entre 1. y 2. Grado de libertad: $\frac{1}{2} \pi$

Sintaxis

```
controlMode = CartesianSineImpedanceControlMode.createLissajousPattern(CartPlane.plane, frequency, amplitude,  
stiffness);
```

Aclaración de la sintaxis

Elemento	Descripción
<i>controlMode</i>	Tipo: CartesianSineImpedanceControlMode Nombre del objeto del regulador
<i>plane</i>	Tipo: Enum del tipo CartPlane Plano en el que ha de conectarse la curva de Lissajous
<i>frequency</i>	Tipo: double Frecuencia de la oscilación para el primer grado de libertad del plano indicado (unidad: Hz) <ul style="list-style-type: none"> ■ 0.0 ... 15.0 <p>La frecuencia para el segundo grado de libertad se calcula de la siguiente forma:</p> <ul style="list-style-type: none"> ■ <i>frequency</i> * 0.4
<i>amplitude</i>	Tipo: double Amplitud de la oscilación para los dos grados de libertad del plano indicado (unidad: N) <ul style="list-style-type: none"> ■ ≥ 0.0
<i>stiffness</i>	Tipo: double Valor de rigidez para los dos grados de libertad del plano indicado (unidad: N/m) <ul style="list-style-type: none"> ■ 0.0 ... 5000.0

Ejemplo

En la brida del robot debe generarse una oscilación en forma de figura de Lissajour con una relación de frecuencia X: Y de 1: 0.4 y un desfase en Y de $\pi/2$. Curso de trayectoria con desfase (= línea azul ([>>>](#) Fig. 17-7)).

```
CartesianSineImpedanceControlMode lissajousMode;  
  
lissajousMode =  
CartesianSineImpedanceControlMode.createLissajousPattern(CartPlane.XY  
, 10.0, 50.0, 500.0);  
  
lbr.move(linRel(0.0, 150.0,  
0.0).setCartVelocity(100).setMode(lissajousMode));
```

17.7.4 Conectar la oscilación de fuerza en espiral**Descripción**

Con el método `createSpiralPattern(...)` se genera una oscilación de fuerza en espiral en un nivel.

La trayectoria de la fuerza se genera a través de la superposición de 2 oscilaciones senoidales. Las oscilaciones están desajustadas $\pi/2$ (90°) entre sí. Las amplitudes de las oscilaciones ascienden continuamente hasta el valor fijado y a continuación vuelven a bajar a cero. De esta forma resulta una forma espiral que se extiende hasta el valor de amplitud fijado y que a continuación se vuelve a contraer.

En el movimiento de robot resultante, el TCP se desplaza a lo largo de esta espiral. La extensión cartesiana de la espiral depende de los valores fijados para la rigidez y la amplitud y de los obstáculos existentes.

El nivel en el que se desea conectar la oscilación en espiral, se transfiere como valor del tipo CartPlane. Los valores fijados para los parámetros de rigidez, frecuencia y amplitud son idénticos para ambos grados de libertad del nivel.

De forma adicional se transfiere un valor para el tiempo total de la oscilación de fuerza. Una mitad de este tiempo se emplea para la subida de la oscilación y la otra, para el descenso.

Tiempo de subida = tiempo total / 2

Tiempo de parada = 0

Tiempo de descenso = tiempo total / 2

Sintaxis

```
controlMode = CartesianSineImpedanceControlMode.createSpiralPattern(CartPlane.plane, frequency, amplitude, stiffness, totalTime);
```

Aclaración de la sintaxis

Elemento	Descripción
<i>controlMode</i>	Tipo: CartesianSineImpedanceControlMode Nombre del objeto del regulador
<i>plane</i>	Tipo: Enum del tipo CartPlane Nivel en el que ha de conectarse la oscilación en espiral
<i>frequency</i>	Tipo: double Frecuencia de la oscilación para los dos grados de libertad del nivel indicado (unidad: Hz) ■ 0.0 ... 15.0
<i>amplitude</i>	Tipo: double Amplitud de la oscilación para los dos grados de libertad del plano indicado (unidad: N) ■ ≥ 0.0
<i>stiffness</i>	Tipo: double Valor de rigidez para los dos grados de libertad del plano indicado (unidad: N/m) ■ 0.0 ... 5000.0
<i>totalTime</i>	Tipo: double Tiempo total de la oscilación en espiral. Una mitad de este tiempo se emplea para la subida de la oscilación y la otra, para el descenso (unidad: s). ■ ≥ 0.0

Ejemplo

En la posición actual de la brida del robot se desea conectar una oscilación de fuerza en espiral en el nivel XY del sistema de coordenadas de la brida. La fuerza debe ascender en espiral hasta un valor máximo de 100 N. La curva de fuerza debe girar una vez por segundo en torno al punto de inicio de la espiral (frecuencia de la oscilación de fuerza: 1.0 Hz). Se desea que la espiral de fuerza ascienda y descienda en un plazo de 10 segundos.

```
CartesianSineImpedanceControlMode spiralMode;
spiralMode =
CartesianSineImpedanceControlMode.createSpiralPattern(CartPlane.XY,
1.0, 100, 500, 10);
lbr.move(positionHold(spiralMode, 10, TimeUnit.SECONDS));
```

El número de giros se calcula dividiendo el tiempo total entre el tiempo necesario para un giro (t_{periodo}). El tiempo para un giro se corresponde con la duración del periodo de una oscilación, p. ej.:

- Frecuencia de la oscilación de fuerza: $f = 1,0 \text{ Hz}$
- Tiempo total: $t = 10 \text{ s}$

El número de giros se calcula de la siguiente manera:

$$\text{Número de giros} = \text{tiempo total} / t_{\text{periodo}} = 10 \text{ s} / 1 \text{ s} = 10$$

$$t_{\text{periodo}} = 1 / f = 1 / 1,0 \text{ Hz} = 1 \text{ s}$$

La máxima desviación resulta de la ley de elasticidad:

$$\Delta x = F / C = 100 \text{ N} / (500 \text{ N/m}) = 0,2 \text{ m} = 20 \text{ cm}$$

17.8 Mantener la posición con regulación

Descripción

Con la ayuda de la instrucción de movimiento positionHold(...), el robot puede mantener su posición nominal cartesiana durante un período de tiempo ajustable y, al mismo tiempo, permanecer regulado.

Si el robot se opera con regulación de elasticidad, puede alejarse de su posición nominal. El hecho de si el robot se aleja de la posición nominal cartesiana actual (=posición en el inicio de la instrucción positionHold(...)), a qué distancia y en qué dirección, depende de los parámetros ajustados del regulador y de las fuerzas resultantes. Además, el robot regulado por elasticidad se puede alejar de su posición nominal debido a fuerzas externas.

Sintaxis

```
object.move(positionHold(controlMode, time, unit));
```

Aclaración de la sintaxis

Elemento	Descripción
<i>controlMode</i>	Tipo: Subclase de AbstractMotionControlMode Nombre del objeto del regulador
<i>time</i>	Tipo: long Indica el tiempo durante el que se debe mantener el <i>controlMode</i> indicado. El valor debe ser ≥ 0 . Un valor < 0 significa sin fin.
<i>unit</i>	Tipo: Enum del tipo TimeUnit Establece la unidad del tiempo indicado. El Enum se incluye por defecto en las bibliotecas de Java.

Ejemplo

El robot debe mantenerse durante 10 segundos en su posición actual. Durante este tiempo, el robot está comutado de forma suave en la dirección cartesiana X.

```
CartesianImpedanceControlMode controlMode = new
CartesianImpedanceControlMode();

controlMode.parametrize(CartDOF.X).setStiffness(1000.0);
controlMode.parametrize(CartDOF.ALL).setDamping(0.7);

lbr.move(positionHold(controlMode, 10, TimeUnit.SECONDS));
```

18 Diagnóstico

18.1 Visualizar errores de bus de campo



Para realizar un análisis de errores preciso se utiliza WorkVisual. En la documentación **WorkVisual** puede consultarse más información sobre el diagnóstico del bus de campo con WorkVisual.

18.1.1 Errores generales de bus de campo

Descripción	El estado general de errores de los buses de campo conectados se puede mostrar en la vista de la estación mediante el cuadro KUKA_Sunrise_Cabinet . Abriendo el nivel Buses de campo se pueden mostrar más detalles.
Procedimiento	<ol style="list-style-type: none">1. Abrir la vista de la estación.2. Seleccionar el cuadro KUKA_Sunrise_Cabinet. La indicación de estado del cuadro Buses de campo muestra el estado colectivo de todos los buses de campo conectados a la unidad de control.3. Seleccionar el cuadro Buses de campo. Se abre la vista en detalla con información sobre el error relativa a los buses de campo conectados actualmente.

18.1.2 Estado de error de E/S y de grupos E/S

Descripción	En la barra de navegación del smartHMI, la indicación de estado de la zona Grupos E/S muestra el estado de los grupos E/S configurados. La indicación inferior representa el estado total de todos los grupos E/S y la superior, el estado del grupo E/S seleccionado.
Procedimiento	<ul style="list-style-type: none">■ Seleccionar el grupo E/S en la barra de navegación en Grupos E/S. Se abre la vista detallada del grupo E/S. Están identificadas todas las entradas/salidas erróneas del grupo seleccionado.

18.2 Visualizar el protocolo

En la smartHMI se puede visualizar un protocolo de las incidencias y de las modificaciones del estado de un sistema.

Procedimiento	<ol style="list-style-type: none">1. Abrir la vista de la estación o la vista del robot.2. Seleccionar el cuadro Protocolo. Se abre la vista Protocolo. Si la vista se abre a través de la vista del robot, por defecto solo se mostrarán las entradas del protocolo que afecten al robot seleccionado en la barra de navegación.
----------------------	--

18.2.1 Vista Protocolo

Resumen

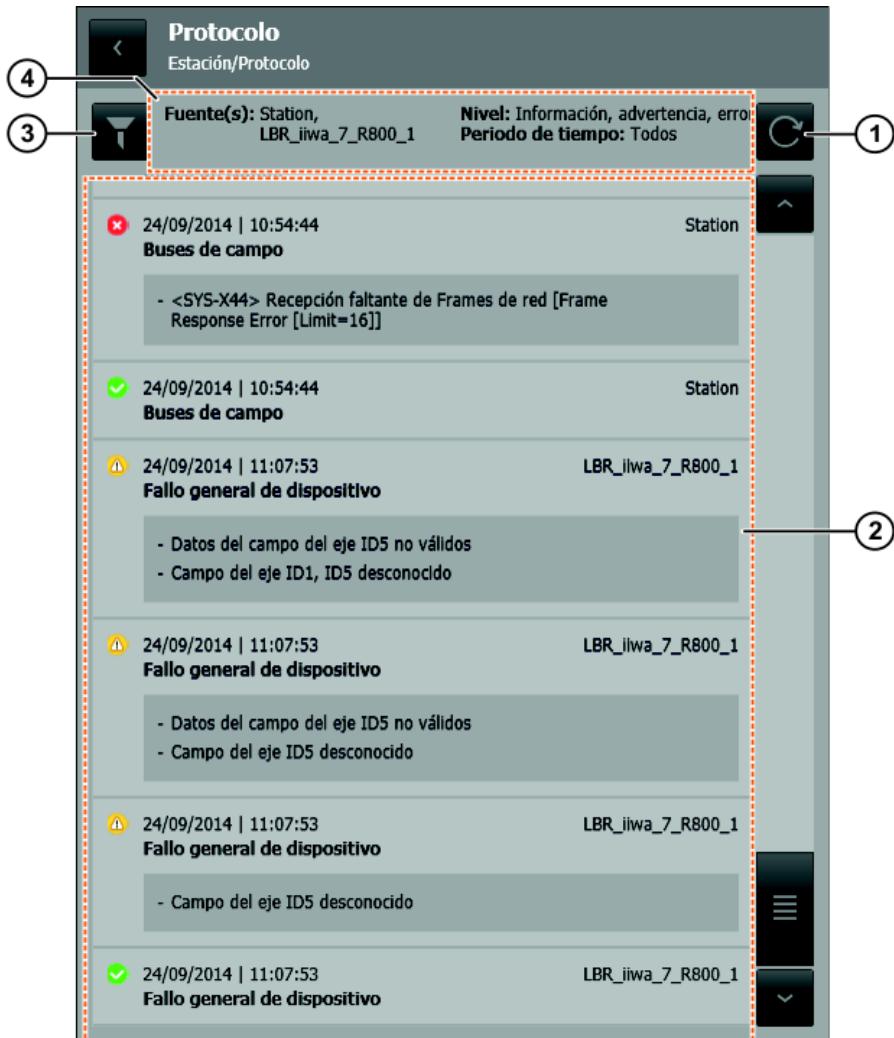


Fig. 18-1: Vista del protocolo

Pos.	Descripción
1	Botón Actualizar Actualiza las entradas del protocolo indicadas. Por defecto, después de la actualización se mostrará la entrada más actual en la posición superior. Si está activo un filtro de tiempo, se mostrará la entrada más antigua en la posición superior.
2	Lista de las entradas del protocolo (>>> "Incidencia Log" Página 414)
3	Botón Ajustes de filtrado Abre la ventana Ajustes de filtrado en la que se pueden filtrar las entradas del protocolo según diferentes criterios.
4	Indicación Ajustes de filtrado Aquí se visualizan los filtros activos actualmente.

Incidencia Log

Las entradas del protocolo contiene diferente información sobre la incidencia Log correspondiente

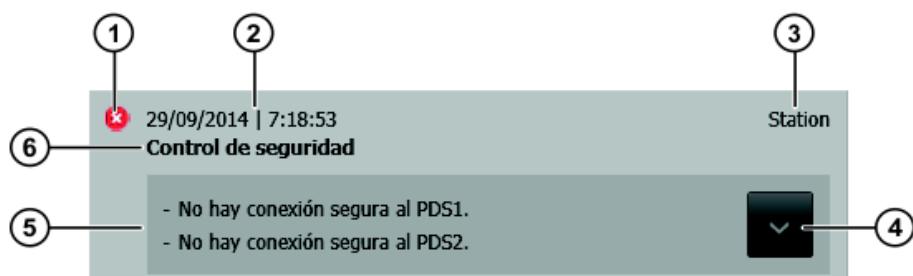


Fig. 18-2: Información sobre la incidencia Log

Pos.	Descripción
1	Nivel Log de la incidencia (>>> "Nivel Log" Página 415)
2	Fecha y hora de la incidencia Log (hora del sistema de la unidad de control del robot)
3	Origen de la incidencia Log (robot o estación)
4	Botón para maximizar/minimizar la visualización detallada El botón solo está disponible cuando existen más de 2 síntomas para una incidencia.
5	Síntomas de la incidencia Log (visualización detallada) Por defecto se muestran hasta 2 síntomas por incidencia.
6	Categoría o breve descripción de la incidencia Log

Nivel Log

El nivel Log de una incidencia se representa mediante los siguientes símbolos:

Símbolo	Descripción
✗	Error Incidencia crítica que tiene como consecuencia un estado de error del sistema
⚠	Advertencia Incidencia crítica que puede provocar un error
✓	Información Incidencia no crítica o información sobre la modificación del estado

18.2.2 Filtrar entradas del protocolo

Condición previa ■ La vista **Protocolo** está abierta.

- Procedimiento**
- Tocar el botón **Ajustes de filtrado**. Se abre la ventana **Ajustes de filtrado**.
 - Seleccionar los filtros deseados mediante los botones correspondientes.
 - Tocar el botón **Ajustes de filtrado** o una zona fuera de la ventana.
- La ventana **Ajustes de filtrado** se cierra y los filtros seleccionados se activan.



Cerrando la vista **Protocolo** se restablecen los filtros. Si la vista se abre de nuevo, estarán de nuevo activos los ajustes por defecto.

Descripción

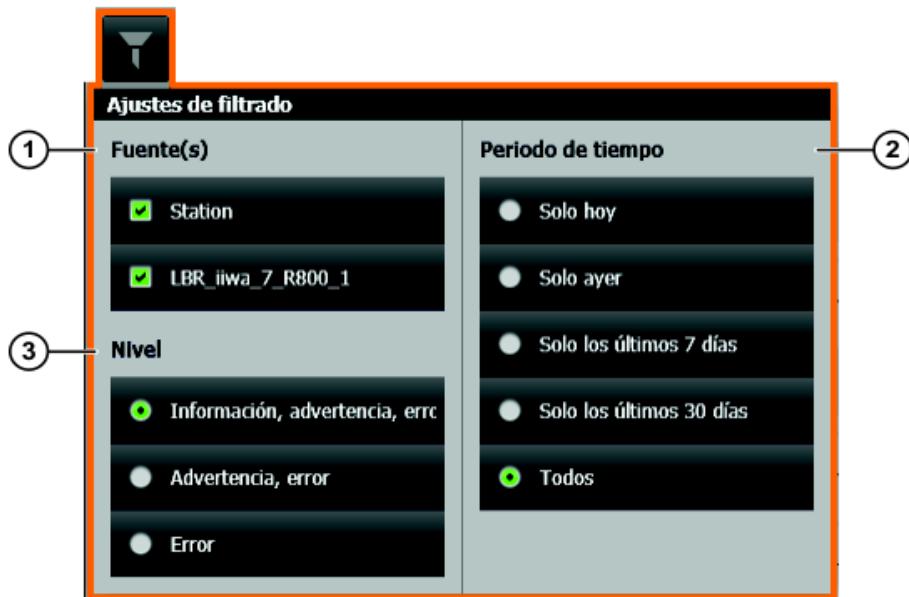


Fig. 18-3: Ventana Ajustes de filtrado

Pos.	Descripción
1	<p>Filtro Fuente(s)</p> <p>Las entradas del protocolo se pueden filtrar por las fuentes que han causado la incidencia Log.</p> <ul style="list-style-type: none"> ■ Estación: Se muestran todas las entradas del protocolo que afectan a la estación y a las entradas/salidas de buses de campo. ■ Robot: Solo se muestran las entradas del protocolo que afectan al robot seleccionado en la barra de navegación, aquí un LBR iiwa 7 R800. <p>Por defecto para el protocolo a través de la vista de la estación: Las dos fuentes están seleccionadas.</p> <p>Por defecto para el protocolo a través de la vista del robot: La fuente es el robot seleccionado en la barra de navegación.</p>
2	<p>Filtro Periodo de tiempo</p> <p>Se puede activar un filtro de tiempo de forma que solo se muestren las entradas del protocolo de un período de tiempo determinado.</p> <p>Por defecto: Todos (sin filtro de tiempo activo)</p>
3	<p>Filtro Nivel</p> <p>Las entradas del protocolo se pueden filtrar según su nivel Log.</p> <p>Por defecto: Información, advertencia, error (sin filtro activo después de nivel Log)</p>

18.3 Indicación de mensajes de error (vista de aplicación)

Si durante la ejecución de una aplicación surgen errores, los mensajes de error correspondientes se indican en la smartHMI.

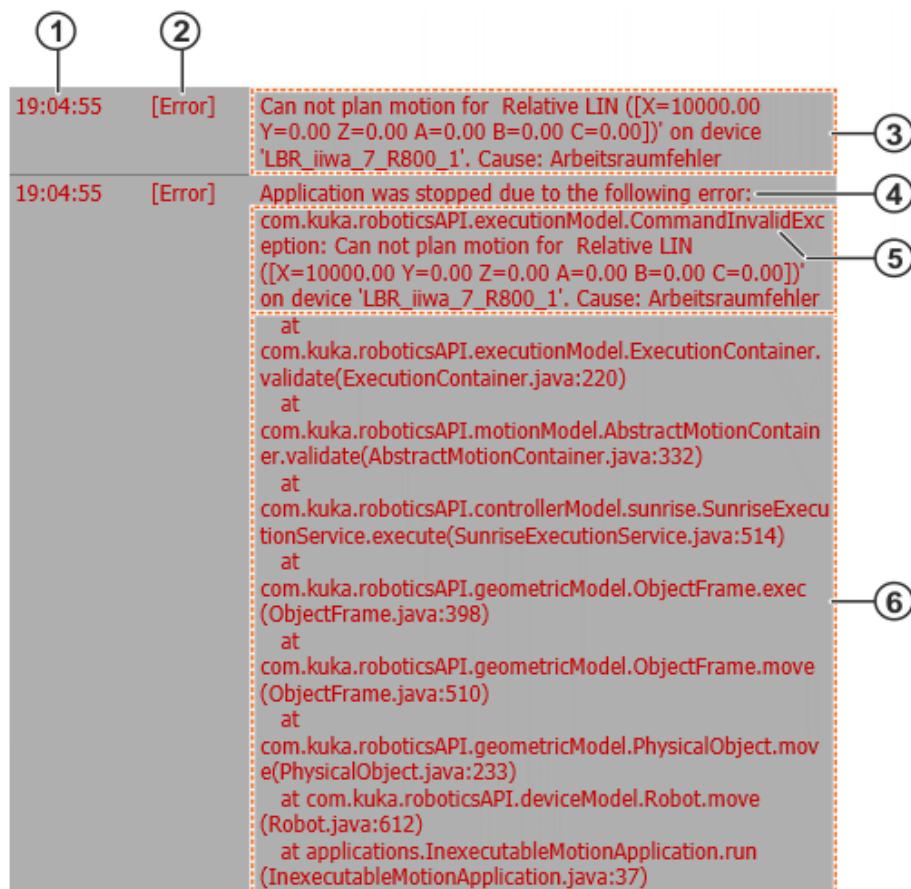


Fig. 18-4: Estructura de mensaje de error (ejemplo)

Pos.	Descripción
1	Marca de tiempo Momento en el que surgió el error.
2	Nivel Nivel Log del mensaje. Los errores tienen el nivel Log Error .
3	Mensaje de error
4	Información en caso de cancelación de la aplicación, p. ej. después de un error de tiempo de ejecución

Pos.	Descripción
5	<p>Tipo de error</p> <p>Los errores se clasifican como clases Java. El nombre de la clase, así como el paquete correspondiente se indican. Justo después aparece el mensaje de error (véase pos. 3).</p>
6	<p>Stacktrace</p> <p>Las activaciones de método que llevaron al error se indican en orden ascendente. Los métodos se indican con su denominador completo. Además, se indica el número de la línea de programa en la que apareció el error.</p> <p>Mediante el Stacktrace se puede determinar en qué punto del programa se ha activado el método que finalmente provocó el error.</p> <p>Ejemplo leído de abajo a arriba:</p> <ul style="list-style-type: none"> ■ Origen del fallo: Método run() de la aplicación InexecutableMotion.java, línea 37 ■ En la línea 37 de la aplicación se ha activado el método move(...) de la clase Robot. En el código fuente de la clase robot.java el error apareció en la línea 612, durante la activación del método move(...) de la clase PhysicalObject. ■ ... ■ El error real apareció en la línea 220 en el código fuente de la clase ExecutionContainer.java, durante la activación del método validate(...).

En muchas ocasiones un error es la consecuencia de un encadenamiento de errores previos. En este caso se indica toda la cadena de errores en orden descendiente.

```

19:07:38 [Error] Application was stopped due to the following error:
java.lang.RuntimeException: Es ist ein Fehler aufgetreten!
  at
applications.EmbeddedExceptionApplication.getNextPosition(EmbeddedExceptionApplication.java:46)
  at applications.EmbeddedExceptionApplication.run(EmbeddedExceptionApplication.java:38)
Caused by: java.lang.Exception: Fehler bei der Berechnung
  at
applications.Utils.calculateValue(Utils.java:8)
  at
applications.EmbeddedExceptionApplication.getNextPosition(EmbeddedExceptionApplication.java:43)
... 1 more

```

Fig. 18-5: Indicación de cadena de fallos (ejemplo)

Pos.	Descripción
1	<p>Errores consecutivos</p> <p>Aquí se indica el último eslabón de la cadena de errores. En el ejemplo, este último eslabón es un error del tipo <code>RuntimeException</code> que apareció durante la ejecución del método <code>run()</code> en la línea 38 de la aplicación <code>EmbeddedExceptionApplication.java</code>.</p>
2	<p>Errores causantes</p> <p>La indicación de los errores causantes siempre se introduce de la siguiente manera:</p> <ul style="list-style-type: none"> ■ Caused by: <i>Tipo de error</i> <p>En el ejemplo, el error causante es del tipo <code>Exception</code> y apareció durante la activación del método <code>calculateValue(...)</code> de la clase <code>Utils</code>. De esta forma se indica toda la cadena de errores hasta la causa real del error.</p>

18.4 Recopilar información de diagnóstico para el análisis de errores de KUKA

El KUKA Customer Support necesita datos de diagnóstico de la unidad de control del robot para realizar el análisis de errores.

Para ello se crea un archivo ZIP con el nombre **KRCDiag**, que puede archivarse en la unidad de control del robot en `D:\DiagnosisPackages` o en una memoria USB conectada a la unidad de control del robot. El paquete de diagnóstico **KRCDiag** contiene los datos que necesita el KUKA Customer Support para analizar un error. Para ello, se incluye información sobre recursos del sistema, datos de la máquina y mucho más.

También se puede acceder a la información de diagnóstico a través de `Sunrise.Workbench`. Para ello, se carga un paquete de diagnóstico existente de la unidad de control del robot o se crea un paquete nuevo.



Los proyectos y las aplicaciones no se aceptan en el paquete de diagnóstico. Se recomienda transmitir estos datos por separado, ya que pueden contener información importante para la búsqueda de errores.



Recomendación: Si es posible, la información de diagnóstico solo se debe recopilar si el robot está parado.



Si falla la recopilación de información de diagnóstico con la aplicación en ejecución, detener deseleccionar la aplicación y reiniciar el proceso de diagnóstico.

18.4.1 Crear un paquete de diagnóstico mediante smartHMI

Descripción Con este procedimiento se puede crear el paquete de diagnóstico **KRCDiag** y archivarse en la unidad de control del robot en `D:\DiagnosisPackages` o en una memoria USB.

- Procedimiento**
1. Para el archivado en una memoria USB: Conectar la memoria USB a la unidad de control del robot y esperar hasta que el LED en la memoria USB esté iluminado de forma permanente.
 2. En el menú principal seleccionar **Diagnóstico > Crear un paquete de diagnóstico** y el lugar de almacenamiento deseado:
 - **Disco duro**

■ Memoria USB

Se compila la información de diagnóstico. El progreso se visualiza en una ventana. Cuando el proceso ha finalizado, también se indica en la ventana. A continuación, la ventana se vuelve a ocultar automáticamente.

18.4.2 Crear un paquete de diagnóstico mediante smartPAD

Descripción	Este procedimiento no utiliza opciones de menú, sino teclas en el smartPAD. Por ello, también se puede utilizar si la smartHMI no está disponible. El paquete de diagnóstico KRCDiag se crea y se archiva en la unidad de control del robot D:\DiagnosisPackages.
--------------------	---



La secuencia de teclas descrita en el procedimiento se debe ejecutar en 2 segundos.

Procedimiento

1. Pulsar y mantener pulsada la tecla de Menú principal.
2. Pulsar 2 veces la tecla del teclado-.
3. Soltar la tecla de Menú principal.

Se compila la información de diagnóstico. El progreso se visualiza en una ventana. Cuando el proceso ha finalizado, también se indica en la ventana. A continuación, la ventana se vuelve a ocultar automáticamente.

18.4.3 Crear un paquete de diagnóstico mediante Sunrise.Workbench

Condición previa	■ Conexión de red a la unidad de control de robot
Procedimiento	<ol style="list-style-type: none">1. En el Explorador de paquetes hacer clic con el botón derecho del ratón sobre el proyecto y seleccionar en el menú contextual Sunrise > Crear paquete de diagnóstico. Se abre el asistente para crear el paquete de diagnóstico.2. Seleccionar Abrir... y navegar hasta el directorio en el que debe crearse el paquete de diagnóstico KRCDiag. En caso necesario, crear una carpeta para el paquete de diagnóstico a través de Crear carpeta nueva. Confirmar con OK.3. Hacer clic en Siguiente. El paquete de diagnóstico se crea en la carpeta indicada.4. A través de Abrir el directorio de destino en el explorador de Windows se puede navegar hasta la carpeta en el que se ha creado el paquete de diagnóstico, p. ej. para enviarlo directamente por correo electrónico.5. Hacer clic en Finalizar. Se cierra el asistente.



Los proyectos y las aplicaciones no se aceptan en el paquete de diagnóstico. Se recomienda transmitir estos datos por separado, ya que pueden contener información importante para la búsqueda de errores.

18.4.4 Cargar paquetes de diagnóstico existentes desde la unidad de control del robot

Condición previa	■ Conexión de red a la unidad de control de robot
Procedimiento	<ol style="list-style-type: none">1. En el Explorador de paquetes hacer clic con el botón derecho del ratón sobre el proyecto y seleccionar en el menú contextual Sunrise > Crear paquete de diagnóstico. Se abre el asistente para crear el paquete de diagnóstico.

2. Seleccionar **Abrir...** y navegar hasta el directorio al que debe copiarse el paquete de diagnóstico **KRCDiag**. En caso necesario, crear una carpeta para el paquete de diagnóstico a través de **Crear carpeta nueva**. Confirmar con **OK**.
3. Activar el botón de radio **Cargar los paquetes de diagnóstico existentes desde el control** y marcar el paquete de diagnóstico deseado.
4. Hacer clic en **Siguiente**. El paquete de diagnóstico se copia en la carpeta indicada.
Si la carpeta ya contiene un paquete de diagnóstico con el mismo nombre de archivo, se mostrará un cuadro de diálogo. El proceso de copia se puede cancelar.
5. A través de **Abrir el directorio de destino en el explorador de Windows** se puede navegar hasta la carpeta en el que se ha copiado el paquete de diagnóstico, p. ej. para enviarlo directamente por correo electrónico.
6. Hacer clic en **Finalizar**. Se cierra el asistente.

19 Servicio KUKA

19.1 Requerimiento de soporte técnico

Introducción Esta documentación ofrece información para el servicio y el manejo y también constituye una ayuda en caso de reparación de averías. Para más preguntas dirigirse a la sucursal local.

Información Para poder atender cualquier consulta es necesario tener a disposición la siguiente información:

- Tipo y número de serie del manipulador
- Tipo y número de serie de la unidad de control
- Tipo y número de serie de la unidad lineal (si existe)
- Tipo y número de serie de la alimentación de energía (si existe)
- Versión del software del sistema
- Software opcional o modificaciones
- Paquete de diagnóstico **KrcDiag**

Adicionalmente, para KUKA Sunrise: Proyectos existentes, aplicaciones incluidas

Para versiones del KUKA System Software anteriores a V8: Archivo del software (**KrcDiag** aún no está disponible aquí.)

- aplicación existente
- Ejes adicionales existentes
- Descripción del problema, duración y frecuencia de la avería

19.2 KUKA Customer Support

Disponibilidad El servicio de atención al cliente de KUKA se encuentra disponible en muchos países. Estamos a su entera disposición para resolver cualquiera de sus preguntas.

Argentina Ruben Costantini S.A. (agencia)
Luis Angel Huergo 13 20
Parque Industrial
2400 San Francisco (CBA)
Argentina
Tel. +54 3564 421033
Fax +54 3564 428877
ventas@costantini-sa.com

Australia KUKA Robotics Australia Pty Ltd
45 Fennell Street
Port Melbourne VIC 3207
Australia
Tel. +61 3 9939 9656
info@kuka-robotics.com.au
www.kuka-robotics.com.au

Bélgica	KUKA Automatisering + Robots N.V. Centrum Zuid 1031 3530 Houthalen Bélgica Tel. +32 11 516160 Fax +32 11 526794 info@kuka.be www.kuka.be
Brasil	KUKA Roboter do Brasil Ltda. Travessa Claudio Armando, nº 171 Bloco 5 - Galpões 51/52 Bairro Assunção CEP 09861-7630 São Bernardo do Campo - SP Brasil Tel. +55 11 4942-8299 Fax +55 11 2201-7883 info@kuka-roboter.com.br www.kuka-roboter.com.br
Chile	Robotec S.A. (agencia) Santiago de Chile Chile Tel. +56 2 331-5951 Fax +56 2 331-5952 robotec@robotec.cl www.robotec.cl
China	KUKA Robotics China Co., Ltd. No. 889 Kungang Road Xiaokunshan Town Songjiang District 201614 Shanghai P. R. China Tel. +86 21 5707 2688 Fax +86 21 5707 2603 info@kuka-robotics.cn www.kuka-robotics.com
Alemania	KUKA Roboter GmbH Zugspitzstr. 140 86165 Augsburg Alemania Tel. +49 821 797-4000 Fax +49 821 797-1616 info@kuka-roboter.de www.kuka-roboter.de

Francia	KUKA Automatisme + Robotique SAS Techvallée 6, Avenue du Parc 91140 Villebon S/Yvette Francia Tel. +33 1 6931660-0 Fax +33 1 6931660-1 commercial@kuka.fr www.kuka.fr
India	KUKA Robotics India Pvt. Ltd. Office Number-7, German Centre, Level 12, Building No. - 9B DLF Cyber City Phase III 122 002 Gurgaon Haryana India Tel. +91 124 4635774 Fax +91 124 4635773 info@kuka.in www.kuka.in
Italia	KUKA Roboter Italia S.p.A. Via Pavia 9/a - int.6 10098 Rivoli (TO) Italia Tel. +39 011 959-5013 Fax +39 011 959-5141 kuka@kuka.it www.kuka.it
Japón	KUKA Robotics Japón K.K. YBP Technical Center 134 Godo-cho, Hodogaya-ku Yokohama, Kanagawa 240 0005 Japón Tel. +81 45 744 7691 Fax +81 45 744 7696 info@kuka.co.jp
Canadá	KUKA Robotics Canada Ltd. 6710 Maritz Drive - Unit 4 Mississauga L5W 0A1 Ontario Canadá Tel. +1 905 670-8600 Fax +1 905 670-8604 info@kukarobotics.com www.kuka-robotics.com/canada

Corea	KUKA Robotics Korea Co. Ltd. RIT Center 306, Gyeonggi Technopark 1271-11 Sa 3-dong, Sangnok-gu Ansan City, Gyeonggi Do 426-901 Corea Tel. +82 31 501-1451 Fax +82 31 501-1461 info@kukakorea.com
Malasia	KUKA Robot Automation (M) Sdn Bhd South East Asia Regional Office No. 7, Jalan TPP 6/6 Taman Perindustrian Puchong 47100 Puchong Selangor Malasia Tel. +60 (03) 8063-1792 Fax +60 (03) 8060-7386 info@kuka.com.my
México	KUKA de México S. de R.L. de C.V. Progreso #8 Col. Centro Industrial Puente de Vigas Tlalnepantla de Baz 54020 Estado de México México Tel. +52 55 5203-8407 Fax +52 55 5203-8148 info@kuka.com.mx www.kuka-robotics.com/mexico
Noruega	KUKA Sveiseanlegg + Roboter Sentrumsvegen 5 2867 Hov Noruega Tel. +47 61 18 91 30 Fax +47 61 18 62 00 info@kuka.no
Austria	KUKA Roboter CEE GmbH Gruberstraße 2-4 4020 Linz Austria Tel. +43 7 32 78 47 52 Fax +43 7 32 79 38 80 office@kuka-roboter.at www.kuka.at

Polonia	KUKA Roboter Austria GmbH Spółka z ograniczoną odpowiedzialnością Oddział w Polsce Ul. Porcelanowa 10 40-246 Katowice Polonia Tel. +48 327 30 32 13 or -14 Fax +48 327 30 32 26 ServicePL@kuka-roboter.de
Portugal	KUKA Sistemas de Automatización S.A. Rua do Alto da Guerra nº 50 Armazém 04 2910 011 Setúbal Portugal Tel. +351 265 729780 Fax +351 265 729782 kuka@mail.telepac.pt
Rusia	KUKA Robotics RUS Werbnaia ul. 8A 107143 Moskau Rusia Tel. +7 495 781-31-20 Fax +7 495 781-31-19 info@kuka-robotics.ru www.kuka-robotics.ru
Suecia	KUKA Svetsanläggningar + Robotar AB A. Odhners gata 15 421 30 Västra Frölunda Suecia Tel. +46 31 7266-200 Fax +46 31 7266-201 info@kuka.se
Suiza	KUKA Roboter Schweiz AG Industriestr. 9 5432 Neuenhof Suiza Tel. +41 44 74490-90 Fax +41 44 74490-91 info@kuka-roboter.ch www.kuka-roboter.ch

España

KUKA Robots IBÉRICA, S.A.
Pol. Industrial
Torrent de la Pastera
Carrer del Bages s/n
08800 Vilanova i la Geltrú (Barcelona)
España
Tel. +34 93 8142-353
Fax +34 93 8142-950
Comercial@kuka-e.com
www.kuka-e.com

Sudáfrica

Jendamark Automation LTD (Agentur)
76a York Road
North End
6000 Port Elizabeth
Sudáfrica
Tel. +27 41 391 4700
Fax +27 41 373 3869
www.jendamark.co.za

Taiwán

KUKA Robot Automation Taiwan Co., Ltd.
No. 249 Pujong Road
Jungli City, Taoyuan County 320
Taiwan, R. O. C.
Tel. +886 3 4331988
Fax +886 3 4331948
info@kuka.com.tw
www.kuka.com.tw

Tailandia

KUKA Robot Automation (M) Sdn Bhd
Thailand Office
c/o Maccall System Co. Ltd.
49/9-10 Soi Kingkaew 30 Kingkaew Road
Tt. Rachatheva, A. Bangpli
Samutprakarn
10540 Thailand
Tel. +66 2 7502737
Fax +66 2 6612355
atika@ji-net.com
www.kuka-roboter.de

Chequia

KUKA Roboter Austria GmbH
Organisation Tschechien und Slowakei
Sezemická 2757/2
193 00 Praha
Horní Počernice
República Checa
Tel. +420 22 62 12 27 2
Fax +420 22 62 12 27 0
support@kuka.cz

Hungría KUKA Robotics Hungaria Kft.
Fö út 140
2335 Taksony
Hungría
Tel. +36 24 501609
Fax +36 24 477031
info@kuka-robotics.hu

EE. UU. KUKA Robotics Corporation
51870 Shelby Parkway
Shelby Township
48315-1787
Michigan
EE. UU.
Tel. +1 866 873-5852
Fax +1 866 329-5852
info@kukarobotics.com
www.kukarobotics.com

Reino Unido KUKA Automation + Robotics
Hereward Rise
Halesowen
B62 8AN
Reino Unido
Tel. +44 121 585-0800
Fax +44 121 585-0900
sales@kuka.co.uk

Índice

Números

2004/108/CE 42
2006/42/CE 42
89/336/CEE 42
95/16/CE 42

A

Accesorios 19, 23
Acciones de conmutación referidas a la trayectoria 324, 340
Actividad del robot, consultar 316
addCartesianForce(...) 352
addCartesianTorque(...) 352
addCommandedCartesianPositionXYZ(...) 353
addCommandedJointPosition(...) 352
addControllerListener(...) 316, 319
addCurrentCartesianPositionXYZ(...) 353
addCurrentJointPosition(...) 352
addDoubleUserKey(...) 358
addExternalJointTorque(...) 352
addInternalJointTorque(...) 352
addUserKey(...) 358
Ajuste 105
Ajuste, borrar 105
Ajustes de filtrado 414
Almacenamiento 41
AMF 16
ANSI/RIA R.15.06-2012 43
API 16
Aplicación de la prueba de frenos, plantilla 122
Aplicación de robot, seleccionar 90
Aplicación del robot, nueva 54
Aplicación, pausar 369
App_Enable 174
App_Start 174, 378
areDataValid() 127
Arrancar, programa 95
attachTo(...) 294, 295
AUT 25
AutExt_Active 174
AutExt_AppReadyToStart 174, 378
Autocompletar 270
Automático 25
Averías 37
awaitFileAvailable(...) 355

B

Barra de menús 48
Barra de navegación 68
Barra de teclas de usuario, crear 358
Barras de herramientas 48, 50
Base para desplazamiento manual 140
base, medición 112
Bloque Spline CP 247
Bloque Spline CP, crear 286
Bloque Spline JP 247
Bloque Spline JP, crear 287
Bloqueo de distintos dispositivos separadores

de protección 30
BooleanIOCondition 324
BrakeState (Enum) 133
BrakeTest (clase) 126, 129
BrakeTestResult (clase) 132
breakWhen(...) 336, 338
Brida de medios Touch 205, 206
Bucle creciente 371
Bucle de conteo 369
Bucle do-while 371
Bucle finito 370
Bucle for 369
Bucle while 370
Bucle, intercalar 375
Bus, conectar 169

C

Cables de unión 19, 23
Campo de trabajo 28, 209, 211
Campo del eje 25, 215
Campos de trabajo cartesianos, definir 211
Campos de trabajo, cambiar 57
Caracteres 278
Categoría de parada 1 (sobre la trayectoria) 26
Categoría de parada 0 26
Categoría de parada 1 26
Cese del servicio 41
CIB-SR 16
CIRC 283
CIRC, tipo de movimiento 248
Circuito E/S (ventana) 169
Circuito ES (ventana) 170
clipApplicationOverride(...) 322
clipManualOverride(...) 322
Compatibilidad electromagnética (CEM) 43
Componentes software 20
Concepto de seguridad 180
Condiciones 323
Condiciones complejas 324
Condiciones de cancelación para movimientos 336
Condiciones de cancelación, evaluar 337
Condición de componentes de fuerza 331
Condición de fuerza 326
Condición de los momentos axiales 325
Condición para el rango de valores de una señal 336
Condición para señales booleanas 335
Condición referida a la trayectoria 333
Conectar, entradas/salidas 171
Conectar, unidad de control del robot 66
Conexión USB 65
Configuración de E/S, abrir 164
Configuración de E/S, exportar 171
Configuración de E/S, nueva 164
Configuración de la estación 159
Configuración de la estación, abrir 159
Configuración de seguridad 179

- Configuración de seguridad, abrir 190
Configuración de seguridad, activar 200
Configuración de seguridad, convertir 161
Configuración de seguridad, desactivar 201
Configuración de seguridad, restaurar 201
Conocimientos necesarios 15
Consulta, posición del robot 309
Continuous Path 247
Contraseña, modificar 201
Control de fuerza TCP 222
Control de la ejecución del programa 369
Control de la orientación 292
Control de la orientación LIN, CIRC, SPL 258
Control de parada 219
Control de procesos 324, 344
Control del campo del eje 221
Control externo 173
Control, orientación de herramienta 217
Controles de velocidad 207
`createAndEnableConditionObserver(...)` 347
`createConditionObserver(...)` 347
`createDesiredForce(...)` 407
`createLissajousPattern(...)` 407
`createNormalForceCondition(...)` 327, 328
`createShearForceCondition(...)` 327, 330
`createSinePattern(...)` 407
`createSpatialForceCondition(...)` 327, 328
`createSpiralPattern(...)` 407
`createUserKeyBar(...)` 358
Cuadros de diálogo, programar 367
Cursos de formación 15
Curva de Lissajous, conectar 409
- D**
- `DataRecorder` 350
Datos de carga 149
Datos de carga de la herramienta, determinar 114
Datos de carga, introducir 149
Datos de la aplicación (vista) 49
Datos, registrar y evaluar 350
Declaración de conformidad 24
Declaración de conformidad de la CE 24
Declaración de montaje 23, 24
`DefaultApp_Error` 174
Definir la herramienta destinada a la seguridad 151
Definir piezas de trabajo destinadas a la seguridad 154
Depurar (perspectiva) 49
Desajustar 105
Desbloquear, robot 82
Desconectar, unidad de control del robot 66
Descripción del producto 19
Deseleccionar (botón) 91
Desplazamiento cartesiano 78
Desplazamiento de forma cartesiana 81
Desplazamiento específico de los ejes 78
Desplazamiento específico del eje 80
Desplazar, forma manual, robot 78
`detach()` 297
- Detección de colisión 221
Determinación del valor del momento 125
Diagnóstico 413
Diferencia nominal-real cartesiana, consultar 312
Direcciones IP 51
Dirección de montaje 52, 77
Dirección IP, mostrar 102
Directiva CEM 24
Directiva de baja tensión 24
Directiva relativa a las máquinas 42
Directiva sobre compatibilidad electromagnética 42
`displayModalDialog(...)` 367
Disponibilidad de desplazamiento, consultar 315
Dispositivo de mando manual 19, 23
Dispositivo de PARADA DE EMERGENCIA 29, 31
Dispositivo de validación 29, 30
Dispositivo de validación, externo 29, 31
Dispositivo externo de PARADA DE EMERGENCIA 29
Dispositivos de seguridad, externos 35
Distancia de frenado 25
Distancia de parada 25, 29, 211
Distancia de reacción 25
Documentación, robot industrial 15
- E**
- E/S Sunrise, borrar 167
E/S Sunrise, crear 164
E/S Sunrise, modificar 167
Editor de Java 269
Editor de Java, abrir 269
Ejecución del movimiento, pausar 369
Ejecución del programa 90
Ejecutar la prueba de frenos 135
Eliminación de residuos 41
`EN 60204-1 + A1` 43
`EN 61000-6-2` 43
`EN 61000-6-4 + A1` 43
`EN 614-1` 43
`EN ISO 10218-1` 43
`EN ISO 12100` 43
`EN ISO 13849-1` 42
`EN ISO 13849-2` 42
`EN ISO 13850` 42
`enable()`, `DataRecorder` 353
Entrada rápida, Java 271
Entradas del protocolo, filtrar 415
Entradas/salidas, visualizar 100
`equals(...)` 338
Equipamiento de protección 34
Errores de bus de campo, visualizar 413
Escáner de virus, instalar 162
Esfuerzo de corte 327
`ESM 16`
Espera con bloqueo 349
Estado 263
Estado de la señal, consultar 317
Estado de la señal, reaccionar ante una mo-

- dificación 319
 Estado ESM, borrar 198
 Estado ESM, nuevo 196
 Estados de ajuste, consultar 315
 Estados del sistema, consultar 314
 Estructura de una instrucción de movimiento 280
 Estructura, aplicación del robot 269
 Event-driven Safety Monitoring 184
 Exception 16
 Explorador de paquetes (vista) 48
 Explotador 25, 27
- F**
- ForceComponentCondition 324
 - ForceCondition 323
 - Frame 16
 - Frame de herramienta, crear 146
 - Frame de pieza, crear 146
 - Frame estándar para movimientos 148
 - Frame, crear 140
 - Frames, borrar 142
 - Frames, desplazar 89, 141
 - Frames, programar por aprendizaje 86
 - Frames, visualizar 85
 - Freno defectuoso 36
 - Freno, defectuoso 121, 135
 - Frenos de retención, abrir 84
 - FSoE 16
 - Fuerza constante, conectar 408
 - Fuerza normal 327
 - Funciones de seguridad 24
 - Funciones de seguridad, configurar 192, 195
 - Funciones destinadas a la seguridad 29
 - Funciones no destinadas a la seguridad 33
 - Función de seguridad, nueva para ESM 198
 - Función de seguridad, nueva para PSM 194
- G**
- Gestión de Frames 139
 - Gestión de objetos 144
 - Gestión del proyecto 139
 - getAlphaRad() 313
 - getApplicationData().createFromTemplate() 294
 - getApplicationData().getFrame() 144
 - getApplicationOverride() 322
 - getApplicationUI() 358
 - getAxis() 132
 - getBetaRad() 313
 - getBrakeIndex() 132
 - getCommandedCartesianPosition(...) 309
 - getCommandedCartesianPosition() 343
 - getCommandedJointPosition() 309, 343
 - getCurrentCartesianPosition() 309, 344
 - getCurrentJointPosition() 310, 344
 - getEffectiveOverride() 322
 - getEmergencyStopEx() 318
 - getEmergencyStopInt() 318
 - getExecutionMode() 320
 - getExternalForceTorque(...) 306, 307
 - getExternalTorque() 304
 - getFiredBreakConditionInfo() 337
 - getFiredCondition() 338, 343
 - getFlange() 295
 - getForce() 307
 - getForceInaccuracy() 308
 - getFrame(...) 295, 298
 - getFriction() 132
 - getGammaRad() 313
 - getGravity() 132
 - getHomePosition() 314
 - getLogLevel() 133
 - getManualOverride() 322
 - getMaxAbsTorqueValues() 127
 - getMaxBrakeHoldingTorque() 132
 - getMeasuredBrakeHoldingTorque() 132
 - getMeasuredTorque() 304
 - getMinBrakeHoldingTorque() 132
 - getMissedEvents() 343
 - getMotion() 377
 - getMotionContainer() 343
 - getMotorHoldingTorque() 132
 - getMotorIndex() 132
 - getMotorMaximalTorque() 133
 - getObserverManager() 347, 349
 - getOperationMode() 318
 - getOperatorSafetyState() 319
 - getPositionInfo() 338, 339
 - getPositionInformation(...) 310
 - getPositionInformation() 312, 343
 - getRecovery() 376
 - getRecoveryStrategy(...) 377
 - getRotationOffset() 312
 - getSafetyState() 317
 - getSafetyStopSignal() 319
 - getSingleMaxAbsTorqueValue(...) 127
 - getSingleTorqueValue(...) 304
 - getStartPosition() 377
 - getStartTimestamp() 127
 - getState() 133
 - getStoppedMotion() 338, 340
 - getStopTimestamp() 128
 - getTestedTorque() 133
 - getTimestamp() 133
 - getTorque() 307
 - getTorqueInaccuracy() 308
 - getTorqueValues() 304
 - getTranslationOffset() 312
 - getTriggerTime() 343
 - Grupo destinatario 15
 - Grupo E/S, borrar 167
 - Grupo E/S, crear 166
 - Grupo E/S, editar 167
 - Grupo E/S, exportar plantilla 167
 - Grupo E/S, importar plantilla 168
 - Guiado manual, programar 288
 - Guiado manual, tipo de movimiento 255
- H**
- halt() 369
 - handGuiding() 255, 288
 - Hardware 45
 - hasActiveMotionCommand() 316

Herramienta de aplicación 79
Herramienta destinada a la seguridad 150
Herramienta, crear 146
Herramienta, desconectar 203
Herramienta, medir 106
Herramientas, declarar 293
Herramientas, inicializar 294
HOV 68, 80

I

IAnyEdgeListener 345
IApplicationOverrideControl (interfaz) 321
ICallbackAction, interfaz 341
ICondition, interfaz 323
IControllerStateListener 316
Identificaciones 35
IFallingEdgeListener 345
Indicación de estados 69
Información de diagnóstico, recopilar 419
Información de redundancia 143, 262
Información del Trigger, evaluar 343
Información Javadoc, visualizar 273
Información sobre la versión, RoboticsAPI 280
Iniciar, System Software 66
initialize() 270, 386, 388
initializeCyclic(...) 386
Instalación 159
Instalación, KUKA Sunrise.Workbench 45
Integrador de la instalación 26
Integrador de sistemas 26
Integrador del sistema 24, 27
Interfaz de usuario, KUKA smartHMI 67
Interfaz de usuario, Sunrise.Workbench 47
Interruptor de final de carrera de software 34
Introducción 15
IORangeCondition 324
IRecovery, interfaz 376
IRisingEdgeListener 345
isEnabled() 355
isFileAvailable() 355
isForceValid(...) 308
isInHome() 314
isMastered() 315
isReadyToMove() 315
isRecording() 355
isRecoveryRequired(...) 377
isRecoveryRequired() 376
isTorqueMeasured() 128
isTorqueValid(...) 308
ISunriseControllerStateListener 319
ITriggerAction, interfaz 341
IUserKeyBar, interfaz 358

J

Javadoc 16
Javadoc (vista) 49
Joint Path 247
JointTorqueCondition 323
JRE 16

K

KRCDiag 419
KRF 17, 26, 82
KUKA Customer Support 423
KUKA RoboticsAPI 17
KUKA smartHMI 17, 67
KUKA smartPAD 17, 26, 36, 63
KUKA Sunrise Cabinet 17, 19
KUKA Sunrise.OS 17
KUKA PSM, tabla PSM 190

L

Ley de elasticidad 392
LIN 282
LIN REL 283
LIN, tipo de movimiento 248
Límite del eje 215

M

main() 270
Manejo, KUKA smartPAD 63
Manejo, KUKA Sunrise.Workbench 47
Manipulador 19, 23, 26, 29
Mantenimiento 40
Marca CE 24
Marcar Frame como base 140
Marcas 16
Mecanismo de PSM 192
Mecanismo ESM 195
Medición 106
medición, base 112
Medidas generales de seguridad 36
Medir, herramienta 106
Memoria de trabajo 45
Memoria del disco duro 45
Mensajes de usuario, programar 366
Menú principal, activar 73
Modificar y consultar el Override 321
Modo de aplicación 89
Modo de ejecución del programa, modificar y consultar 320
Modo de ejecución del programa, seleccionar 93
Modo de guiado manual 255, 288
Modo de servicio automático 40
Modo de servicio manual 39
Modo de servicio, cambiar 75
Modo paso a paso 34
Momentos axiales, consultar 304
Momentos, específicos del eje 100
Monitoring 324, 344
MotionBatch 284
MotionPathCondition 324
move(...) 280, 297, 378
moveAsync(...) 281, 297, 380
Movimiento circular 283
Movimiento CP 247
Movimiento de espacio cero 82
Movimiento JP 247
Movimiento lineal 282, 283
Movimiento punto a punto 281
MRK 17

- Método ABC 2 puntos 109
 Método ABC mundial 111
 método de 3 puntos 112
 Método XYZ 4 puntos 107
 Métodos set 290
 Métodos, extraer 272
- N**
- Navegador Javadoc, estructura 275
 Normal de la superficie 327
 Normas y prescripciones aplicadas 42
 Normativa sobre construcción de máquinas 24
 NotificationType, Enum 347
 Número de serie, visualizar 103
- O**
- Objeto del regulador, crear 390
 Observaciones 15
 Observaciones de seguridad 15
 Observaciones sobre responsabilidades 23
 ObserverManager 347, 349
 onIsReadyToMoveChanged(...) 316
 onKeyEvent(...), IUserKeyListener 360
 onSafetyStateChanged(...) 319
 onTriggerFired(...) 341
 Opciones 19, 23
 Operadores 325
 Orientación de herramienta 217
 Oscilación de fuerza en espiral, conectar 410
 Oscilación de fuerza simple, conectar 408
 Override 68, 80, 92, 94, 322
 Override de aplicación 94, 322
 Override efectivo 94, 95, 322
 Override manual 68, 80, 92, 94, 322
- P**
- Pantalla táctil 63
 Paquete de diagnóstico, cargar desde la unidad de control del robot 420
 Paquete de diagnóstico, crear 419, 420
 Paquete de idioma, instalar 46, 161
 Paquete de Java, nuevo 54
 PARADA DE EMERGENCIA 64
 PARADA DE EMERGENCIA, externa 29, 205
 PARADA DE EMERGENCIA, externo 31
 Parada de seguridad 26
 Parada de seguridad 1 26
 Parada de seguridad 1 (sobre la trayectoria) 26
 Parada de seguridad 0 26
 Parada de seguridad, externa 29, 31
 Parada de servicio externa segura 32
 Parada de servicio segura 219
 Parada de servicio segura, externa 29
 Parámetros de movimiento 290
 Parámetros del regulador, determinar 390
 Performance Level 24
 Permanent Safety Monitoring 182
 Personal 27
 Perspectivas, seleccionar 48
 Perspectivas, visualizar 49
 pieza, crear 146
- Piezas de trabajo destinadas a la seguridad 153
 piezas, declarar 293
 piezas, inicializar 294
 Placa de características 65
 Plantilla, para proyecto Sunrise 51
 Plantillas 271
 Plantillas de objetos (vista) 49
 Plantillas, específicas de usuario 272
 PLC 17
 Point to Point 247
 Posicionamiento aproximado 256
 Posición cartesiana, consultar 311
 Posición de pánico 30
 Posición del robot, consulta 309
 Posición específica del eje, consultar 310
 Posición HOME 313
 Posición HOME, consultar 314
 Posición HOME, modificar 313
 Posición real, cartesiana 99
 Posición real, específica 98
 positionHold(...) 412
 Procesador 45
 PROFINET 17
 PROFIsafe 17
 Programa, arrancar de forma automática 95
 Programa, arrancar de forma manual 95
 Programación 269
 Programación (perspectiva) 49
 Programación de mensajes 366
 Programación de movimiento, principios 247
 Propiedades (vista) 49
 Protección del operario 29, 30
 Protocolo, vista 414
 Protocolo, visualizar 413
 Proyecto de Java, nuevo 58
 Proyecto Sunrise, nuevo 51
 Proyecto, cargar desde la unidad de control del robot 158
 Proyecto, sincronizar 155
 Proyecto, transferir a la unidad de control del robot 156
 Proyectos Java, referenciar 59
 Proyectos, archivar 57
 Proyectos, cargar en la zona de trabajo 57
 Prueba de frenos 119
 Prueba de frenos, consultar el resultado 133
 Prueba de frenos, evaluar 131
 Prueba de frenos, iniciar la ejecución 130
 Prueba de frenos, interfaz de programación 126
 Prueba de frenos, modificar la posición de salida 125
 Prueba de frenos, resultado (indicación) 136
 Prueba de funcionamiento 38
 PSM 17
 PTP 281
 PTP, tipo de movimiento 247
 PTPRecoveryStrategy (clase) 377
 Puesta en servicio 37, 105
 Pulsador de validación 30, 65, 66
 Punto auxiliar 283
 Punto de posicionamiento aproximado 256

- Punto intermedio 248
- R**
- Ramificación condicionada 372
 - Ramificación if-else 372
 - Ramificación múltiple 374
 - Ramificación switch 374
 - Reacciones de parada destinadas a la seguridad 32
 - Reacciones de parada, destinadas a la seguridad 32
 - Reanudación del servicio 37, 105
 - Reanudar control de seguridad 83
 - Recepción de seguridad, resumen 230
 - Referenciación de posición 227
 - Referenciación de posición y del momento 227
 - Referenciación del momento 228
 - Referenciación, eliminar 59
 - Regulador de impedancia cartesiano 389, 391, 400
 - Regulador de posición 389, 391
 - Reguladores, resumen 389
 - Renombrar, variable 270
 - Reparación 40
 - Requerimiento de soporte técnico 423
 - Requisitos del sistema PC 45
 - Restablecer (botón) 91
 - Resumen, recepción de seguridad 230
 - Retardo conexión, para funciones de seguridad 220
 - Robot industrial 23
 - Robot, retroceder 95
 - RoboticsAPI 17
 - RoboticsAPI, visualizar la versión 280
 - run() 270, 388
 - runCyclic() 386
- S**
- SafetyConfiguration.sconf (archivo) 54, 190
 - Salida, modificar 100
 - Segmento Spline 249
 - Seguridad 23
 - Seguridad de las máquinas 42, 43
 - Seguridad de máquinas 42, 43
 - Seguridad, condiciones marco legales 23
 - Selección de modos de servicio 33
 - Servicio, KUKA Roboter 423
 - Servidor de aplicaciones 16
 - setAdditionalControlForce(...) 397
 - setAmplitude(...) 404
 - setApplicationOverride(...) 322
 - setBias(...) 404
 - setBlendingCart(...) 291
 - setBlendingOri(...) 292
 - setBlendingRel(...) 291
 - setCartAcceleration(...) 291
 - setCartJerk(...) 291
 - setCartVelocity(...) 290
 - setCriticalText(...), IUserKey 365
 - setDamping(...) 397
 - setExecutionMode(...) 320
- setFallTime(...) 406
 - setForceLimit(...) 405
 - setFrequency(...) 404
 - setHoldTime(...) 406
 - setHomePosition(...) 313
 - setJointAccelerationRel(...) 291, 292
 - setJointJerkRel(...) 291, 292
 - setJointVelocityRel(...) 290, 292
 - setLED(...), IUserKey 364
 - setMaxCartesianVelocity(...) 399
 - setMaxControlForce(...) 398
 - setMaxPathDeviation(...) 399
 - setNullSpaceDamping(...) 398
 - setNullSpaceStiffness(...) 398
 - setOrientationReferenceSystem(...) 260, 292
 - setOrientationType(...) 258, 292
 - setPhaseDeg(...) 404
 - setPositionLimit(...) 405
 - setRiseTime(...) 406
 - setSafetyWorkpiece(...) 298
 - setStayActiveUntilPatternFinished(...) 406
 - setStiffness(...) 396
 - setText(...), IUserKey 363
 - setTotalTime(...) 406
 - Señal de disponibilidad de desplazamiento, reaccionar ante la modificación 316
 - Señales de seguridad, consultar 317
 - Señales de seguridad, evaluar 317
 - Sincronización, proyecto 155
 - Single Point of Control 41
 - Singularidad 259
 - Singularidades 265
 - Sistema de coordenadas base 77
 - Sistema de coordenadas de base 112
 - Sistema de coordenadas de herramientas 106
 - Sistema de coordenadas de la brida 77
 - Sistema de coordenadas de la herramienta 77
 - Sistema de coordenadas del pie del robot 77
 - Sistema de coordenadas Mundo 76
 - Sistema de coordenadas, para teclas de desplazamiento 68
 - Sistemas de coordenadas 76
 - smartHMI 17, 67
 - smartPAD 17, 26, 36, 63
 - Sobrecarga 36
 - Software 19, 23, 45
 - Space Mouse 64
 - SPL, tipo de movimiento 249
 - Spline, tipo de movimiento 249
 - SPOC 41
 - startEvaluation() 126
 - startRecording() 353
 - StartRecordingAction 354
 - Station_Error 174
 - StationSetup.cat (archivo) 54, 159
 - Step (botón) 92
 - stopEvaluation() 126
 - stopRecording() 355
 - StopRecordingAction 355
 - Suma de control, configuración de seguridad 200

Sunrise.Workbench, desinstalar 45
 Sunrise.Workbench, iniciar 47
 Sunrise.Workbench, instalar 45
 Sunrise.Workbench, interfaz de usuario 47
 SunriseExecutionService 320
 System Software, instalar 160

T

T1 27
 T2 27
 Tabla ESM 190
 Tabla PSM, KUKA PSM 190
 Tabla PSM, usuario PSM 190
 Tarea en segundo plano cíclica 386
 Tarea en segundo plano no cíclica 387
 Tarea en segundo plano, nueva 55
 Tareas (vista) 49
 Tareas en segundo plano 385
 Tarjeta gráfica 45
 TCP 17, 106, 146
 Tecla de arranque 64, 65
 Tecla de arranque-hacia atrás 64
 Tecla de inicio 65
 Tecla de menú principal 64
 Tecla de PARADA 64
 Tecla del teclado 64
 Teclado 70
 Teclas de desplazamiento 64, 80, 81
 Teclas de usuario 64
 Teclas de usuario, activar 96
 Teclas de usuario, definir 356
 Terminología utilizada 16
 Terminología, utilizada 16
 Tipo de desplazamiento 68
 Tipo de robot, visualizar 103
 Tipos de datos 278
 Tipos de letra 278
 Tipos de movimiento 247
 Tool Center Point 106
 TorqueEvaluator (clase) 123, 124, 126
 TorqueStatistic (clase) 124, 126, 127
 Trabajos de limpieza 41
 Trabajos de mantenimiento 41
 Transporte 37
 Tratamiento de fallos 378
 Tratamiento de instrucciones de movimiento fallidas 378
 Trigger 324, 340
 Trigger, programar 340
 triggerWhen(...) 341
 Turn 264
 Términos, seguridad 25

U

Unidad de control del robot 23
 Unidad de control del robot, conectar/
 desconectar 66
 UserKeyAlignment (Enum) 362
 Uso conforme a lo previsto 23
 Usuario 25, 27
 Usuario PSM, tabla PSM 190

Utilización conforme a los fines previstos 21
 Utilización, distinta al uso previsto 23
 Utilización, indebida 23

V

Variable, renombrar 270
 Variables 279
 Velocidad 80
 Ventana de mensajes 92
 Versión de software, mostrar 102
 Vida útil 25
 Vista de la estación 70
 Vista del robot 72
 Vista general del sistema de robot 19
 Vista, protocolo 414
 Vistas, disponer 49

W

waitFor(...) 349

Z

Zona de peligro 25
 Zona de protección 28, 209, 213
 Zona de seguridad 26, 29
 Zona de trabajo 25, 29, 215
 Zona de trabajo, cambiar 56
 Zona de trabajo, nueva 56
 Zona de trabajo, Sunrise.Workbench 56
 Zonas de control 208
 Zonas de control específicas del eje, definir 215
 Zonas de protección cartesianas, definir 213
 Ángulo de redundancia 263

