



CURSO DE PROGRAMACIÓN ROBÓTICA INDUSTRIAL

COMAU C5G

2ª SESIÓN

MANUAL

Contenido

1. Cálculo automático de un Tool y un User Frame	4
1.1. Cálculo Automático de Tool	5
1.1.1. Cálculo del Tool con el método estándar – Procedimiento total.....	6
1.1.2. Verificación del Tool con el método estándar – Procedimiento Parcial	7
1.1.3. Cálculo de herramienta "método de 4 puntos" - Procedimiento total.....	9
1.1.4. Verificación de la herramienta con el método "4 puntos" – Procedimiento parcial.....	10
1.2. Cálculo Automático del UFRAME	11
2. Descripción de los distintos tipos de movimientos.....	13
2.1. Trayectoria	14
2.1.1. Evolución de la orientación durante un movimiento Lineal o Circular	15
3. Dinámica del movimiento	16
3.1. Control de la velocidad.....	16
3.1.1. Anulación de la velocidad.....	16
3.1.2. Control de la velocidad cartesiana	18
3.1.2.1. Opciones de control de la velocidad cartesiana.....	18
3.1.3. Control de la velocidad de las articulaciones	18
3.2. Aceleración y Deceleración	19
3.3. Precisión de posicionamiento	20
3.4. Movimiento Continuo	21
4. Introducción al ambiente de programación IDE	22
4.1. Programación en IDE (con TP) de un programa de movimiento	24
4.1.1. Edición de programas.....	24
4.1.1.1. Seleccione una o más líneas de texto, eliminar / restaurar, copiar	25
4.1.1.2. Mover el cursor de edición.....	26
4.1.1.3. Introducir una nueva instrucción	27
4.1.1.4. Comentar / descomentar una línea en el programa.....	28
4.1.1.5. Modificar una instrucción existente.....	28
4.1.1.6. Introducir / ver / eliminar variables	29
4.1.1.7. Importación de otros programas	30
4.1.1.8. Mostrar un programa.....	30
4.1.2. Guardado de posiciones (inserción de una instrucción de movimiento)	31
4.1.2.1. Ajuste el botón REC	31
4.1.2.2. Grabar nuevas posiciones	32
4.1.2.3. Modificar las posiciones existentes.....	32

4.1.3.	Test de la ejecución del programa	33
4.1.3.1.	Activar/Desactivar un programa	34
4.1.3.2.	Interrumpir la ejecución de una instrucción	34
4.1.3.3.	Establecer el modo de paso	34
4.1.3.4.	Mover el cursor en ejecución	35
4.1.3.5.	Insertar/Retirar un punto de ruptura.....	35
4.1.3.6.	Saltarse la ejecución de una instrucción	36
4.1.3.7.	Ejecutar una instrucción temporal.....	36
4.1.4.	Guardar el programa y cerrar el IDE	37
4.2.	Ejecución de un programa en modo automático.....	38
5.	Programa PDL2 – Conceptos básicos	39
5.1.	Estructura De un programa PDL2	39
5.2.	Atributo HOLD / NOHOLD	39
6.	Ejemplo de estructura.....	40
7.	Ciclo de vida de un programa PDL2	42
7.1.	Establecimiento y desarrollo de un programa	42
7.2.	Guardar el programa en un dispositivo de almacenamiento externo	42

1. Cálculo automático de un Tool y un User Frame

Para lograr la precisión deseada en los movimientos en el espacio cartesiano y utilizar un marco específico de referencia para grabar las posiciones, se debe indicar correctamente el tamaño de la herramienta (Tool) y definir el marco de usuario (UFrame) que se desea utilizar.

Estos cálculos se pueden realizar a través de un programa llamado TO_SET, que se activa en la programación del terminal del submenú de configuración ToolFrame.

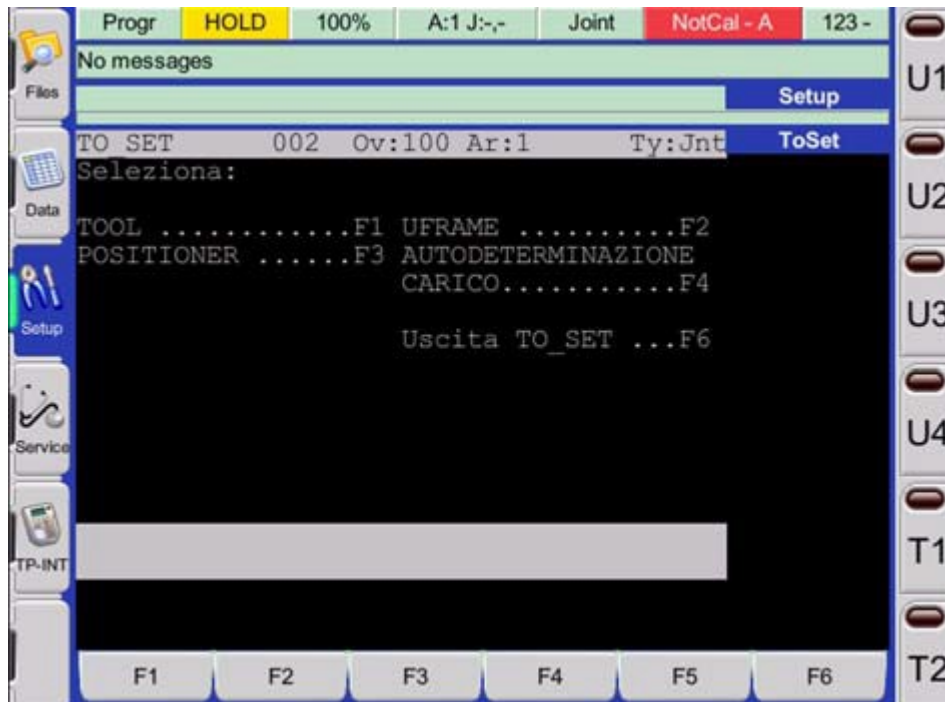
Para acceder a los procedimientos de cálculo automático de herramientas, pulse la tecla F1.

Para acceder al procedimiento de cálculo UFRAME automático, pulse la tecla F2.

Para acceder al procedimiento de POSICIONADORES (BASE beneficio opcional Cálculo automático), pulse el botón F3.

Para acceder al procedimiento de Autodeterminación (opcional), pulse la tecla F4.

Para salir del programa TO_SET, pulse la tecla F6.



1.1. Cálculo Automático de Tool

En este entorno se puede definir el tamaño de la herramienta de trabajo montada en la brida del robot. La precisión del cálculo depende del mismo robot, utilizado como un instrumento de medida.

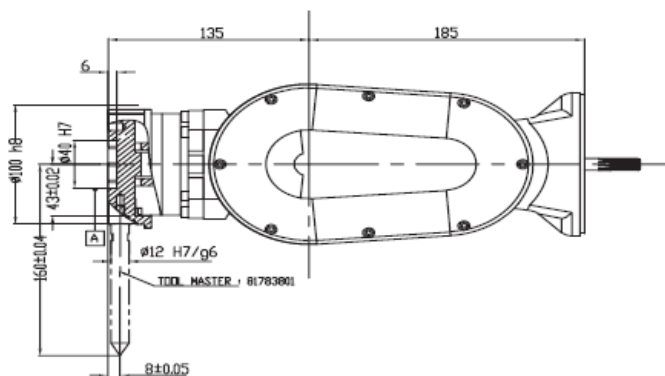
Para la ejecución del procedimiento se necesitan las siguientes herramientas:

- Calibrado de herramienta (Master Tool): Es una herramienta de calibrado X,Y,Z que tiene mediciones y notas que será montada en la brida del robot para la adquisición de la posición de referencia.

Como alternativa, también se puede utilizar un punto identificado en la práctica, cuya distancia del centro de la brida del robot debe ser conocida y precisa.

- Punto de referencia (o punto de control): Es esencial para identificar un punto físico de referencia en el entorno que rodea al robot, con el fin de llevar el TCP del robot a ese mismo punto: una vez con la herramienta principal y, posteriormente, con la herramienta que se desea medir.

Para este propósito se diseñó un cubo Punto de provisto de una punta móvil que, cuando se fija en la posición cero, se puede utilizar como punto de referencia.



El punto (o cualquier otro punto de referencia) debe estar en posición fija y estable, de fácil acceso desde la posición en la que se encuentre el robot.

Una vez seleccionado el cálculo deseado, el programa pide al usuario que introduzca el número de la herramienta para medir o verificar.

En este punto se guarda en la tabla que contiene la herramienta existente para verificar en cuál de las siguientes condiciones se encuentra:

- El número de herramienta de entrada corresponde a una posición vacía en la tabla y a continuación, se procede al cálculo de la herramienta con uno de los procedimientos disponibles.

- Si la herramienta introducida ya existe en la tabla, a continuación, el programa muestra el tamaño calculado anteriormente y el método que ha sido usado durante el cálculo.

Existen los siguientes procedimientos:

- Cálculo de la herramienta con el método estándar - Procedimiento total.
- Verificación de herramienta con el método estándar - Procedimiento parcial.
- Cálculo de herramienta "método de 4 puntos" - Procedimiento total.
- Verificación de la herramienta con el método "4 puntos" – Procedimiento parcial.

1.1.1. Cálculo del Tool con el método estándar – Procedimiento total

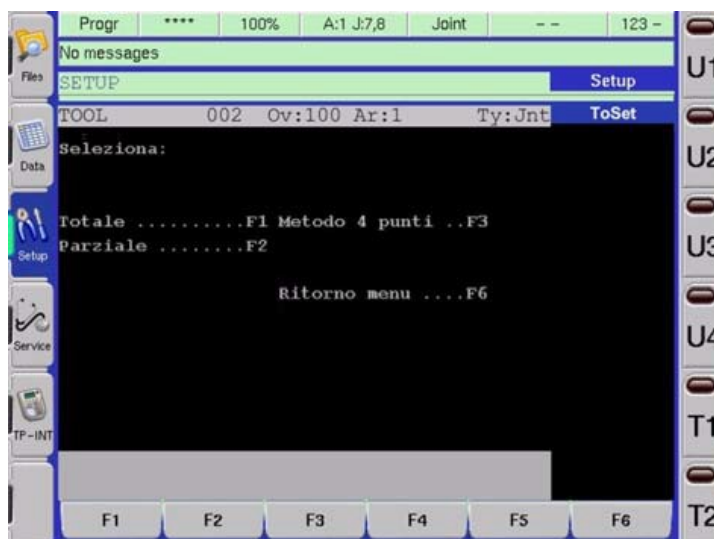
Debe realizarse:

- Si es la primera vez que se ejecuta un cálculo automático de la herramienta;
- Si el punto de referencia o el robot se ha movido.
- Si el robot ha sido recalibrado.

El procedimiento completo permite almacenar en la UD: TO_SET.VAR la posición de la muestra, y los valores de la herramienta maestra declarados. Es suficiente ejecutar una sola vez este procedimiento. Todos los cálculos posteriores de una herramienta montada en la brida del robot sólo requieren el procedimiento parcial.

a. Pulse F1 para ejecutar el procedimiento estándar total.

b. El programa muestra las mediciones de la herramienta maestra, permite editarlas (F2) o continuar (F1).



c. Llevar el TCP de la herramienta maestra a su referencia, con las nuevas medidas adecuadamente definidas.

d. Pulse F1 cuando el robot esté en el punto, para adquirir la posición.

Continúe con los pasos de la herramienta de verificación con el método estándar - Procedimiento parcial.

1.1.2. Verificación del Tool con el método estándar – Procedimiento Parcial

El procedimiento parcial permite calcular la herramienta montada en la brida del robot. Se pueden calcular las compensaciones sólo (X, Y, Z); o continuar en el cálculo de rotaciones (<A>: Euler 1; <E>: Euler 2) y, finalmente, se puede calcular la rotación alrededor de la nueva Z-herramienta (<I>: Euler 3).

Debe realizarse sólo si el procedimiento total es ya al menos una vez ha realizado.

a. Trae al punto de referencia de la herramienta el TCP a medir. El robot puede asumir cualquier posición: el operador puede posicionar el TCP en la referencia como él prefiere, de acuerdo con sus necesidades. Véase la figura de la derecha en la parte superior.

b. Pulse F1 cuando el robot está a punto de adquirir la posición.

c. El programa muestra los valores calculados de \$ HERRAMIENTA. En este momento se pueden mover los ejes en la herramienta 4, 5, 6 para asegurarse de que el nuevo TCP permanece inmóvil en la referencia (movimiento aceptable del orden de 3 mm).

c1. Si el TCP se mueve, comprobar la calibración del robot; controlar el tamaño de la herramienta maestra declarada en el Procedimiento total y controlar la posición del punto de referencia (no cambiado en comparación con el Procedimiento Total).

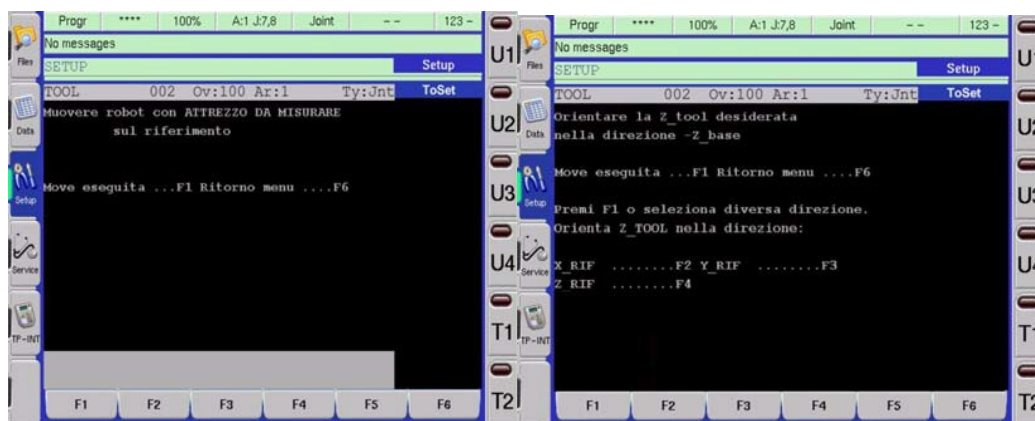
c.2 Pulse F3 para repetir el procedimiento.

d. Presione F2 para guardar los valores de desplazamiento calculados.

e. Presione F1 para continuar el procedimiento, y también para calcular las rotaciones.

f. Elegir la dirección en la que desea dirigir la herramienta Z-TOOL, paralela a uno de los ejes de accionamiento de la base del robot: teclas F2..F4, como se muestra en la figura de la derecha abajo.

g. Oriente el Z-TOOL en la dirección elegida.



- h. Pulsar F1 al final del movimiento.
- i. Compruebe si el nuevo enfoque es el que en realidad quería: si no es el deseado, pulse F3 para repetir el procedimiento.
- j. Presione F2 para guardar los valores calculados (desplazamiento y rotación).
- k. Presione F1 para continuar.
- l. Elegir la dirección en la que desea dirigir la herramienta X-TOOL, paralela a uno de los ejes de accionamiento de la base del robot. Teclas F2 y F5.
- m. Orientar el X-TOOL en la dirección elegida.
- n. Pulsar F1 al final del movimiento.
- o. Compruebe si el nuevo enfoque es el que en realidad quería: si no es el deseado, pulse F3 para repetir el procedimiento.
- p. Presione F2 para guardar los valores calculados (desplazamiento y rotación). Después de cada operación de recovery, si es que ya están presentes en la mesa de los ángulos de Euler, se ve una pantalla con los valores calculados de \$TOOL, en el que se preguntará si desea mantener esos datos o sobrescribir con los nuevos datos. Esto mantiene los ángulos de Euler en el caso de que quiera volver a calcular sólo se desplaza la herramienta.
- q. Después del procedimiento, si se intenta volver a la pantalla principal sin guardar los datos calculados, el programa presenta una pantalla en la que alerta al usuario de que falta guardar y le permite, si así lo desea, guardar los datos.

1.1.3. Cálculo de herramienta "método de 4 puntos" - Procedimiento total

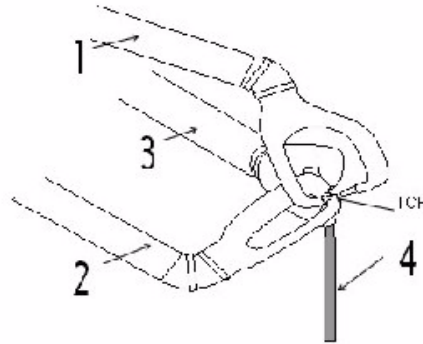
Este procedimiento tiene previsto llevar el TCP a un punto de referencia visible situado dentro del espacio de trabajo del robot, con al menos tres orientaciones diferentes y adquirir las posiciones relativas

1 Punto 3 de captura.

2 El punto 2 de captura.

3 Puntos 1 adquisición.

4 Referencia.



Los pasos del procedimiento se describen a continuación:

a. Mover el TCP al punto de referencia.

b. Pulsar F1 al final del movimiento, para guardar 1 punto.

c. Devolver el TCP al punto de referencia con una orientación suficientemente variada con respecto al punto anterior.

d. Pulsar F1 al final del movimiento, para guardar el punto 2.

e. Devolver el TCP al punto con la suficiente variación respecto a la orientación anterior.

f. Pulsar F1 al final del movimiento, para guardar el punto 3.

g. Si la medida del TCP es posible, al usuario se le proporciona una evaluación del procedimiento de adquisición (bueno, impreciso, poco fiable) con la capacidad de aceptar y proceder a la medición proporcionada del cálculo de la orientación.

h. Continuar con la adquisición de otros puntos (máximo 8 puntos), como se indica en los pasos anteriores, con la posibilidad de mover el robot en el modo de herramienta.

i. Adquisición de puntos completada: Puede continuar con el cálculo de la orientación, repetir el cálculo de X, Y, Z y volver al menú principal.

Si el usuario ha guardado 8 puntos y el algoritmo no es capaz de calcular la herramienta incluso con una medición inexacta, significa que los puntos adquiridos son inexactos o son linealmente dependientes. Y 'posible para intentar repetir puntos con una mayor orientación y, por tanto, dar más información al programa.

Cuanto más grande sea la diferencia de orientación de los puntos, mejor será el resultado.

1.1.4. Verificación de la herramienta con el método "4 puntos" – Procedimiento parcial

Este procedimiento le permite repetir una medición de la herramienta ya establecidos anteriormente con el método "4 puntos". Es muy similar a la de la herramienta de cálculo con el "método de 4 puntos" – procedimiento total, con la posibilidad adicional de mover la referencia al TCP de ejecutar el desplazamiento a las posiciones ya utilizados para la herramienta de medición. De esta manera se mejora la repetitividad de la medida, ya que realiza la verificación de la herramienta de la adquisición de los puntos con la misma orientación (menos de una ligera aproximación) de la adquisición anterior.

a. Con el pulsador en DRIVE ON, presione el botón START para mover el robot con la herramienta que se medirá hasta el primer punto almacenado. La velocidad de movimiento se visualiza en la Teach.

b. Al final del movimiento, pulse F1 para adquirir la posición.

c. Pasar a la posterior adquisición, como se describe en la herramienta de cálculo con el "método de 4 puntos" - procedimiento total. Una vez que el algoritmo pueda calcular una herramienta, aparece una pantalla que le informa de que a partir de este momento se puede mover en la herramienta de referencia.

1.2. Cálculo Automático del UFRAME

En este entorno, es posible definir un nuevo sistema de referencia en el que guardar las posiciones de trabajo. La precisión del cálculo es el mismo robot, utilizado como un instrumento de medida.

Para realizar este procedimiento se debe tener:

- Herramienta de tamaño conocido montada en la brida del robot.

Se debe utilizar la herramienta maestra calibrada, la misma utilizada para el cálculo automático de la herramienta (ver sección herramienta de cálculo automático).

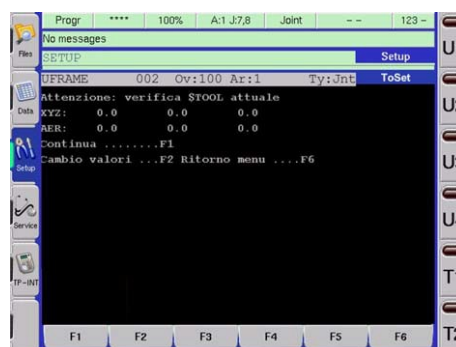
Alternativamente, también puede utilizar un punto identificado en la práctica, cuya distancia desde el centro de la brida del robot debe ser conocida y precisa.

- 3 puntos de interés (ORIGEN, XPos y XYpos) - es necesario identificar tres puntos de referencia en la pieza de trabajo:

- ORIGEN: donde establecerá el origen de la trama.
- Xpos: determina la dirección y la dirección de X-marco.
- XYpos: determina la dirección y la dirección de Y-marco.

Los pasos del procedimiento se describen a continuación:

- a. Comprobar los valores de los actuales de \$TOOL (ver figura de la derecha).



Los valores están referidos al equipo montado en la brida y serán utilizados para el cálculo de la estructura.

Los valores se muestran en el programa y se permite la modificación si es necesario. Para cambiar estos valores, pulse F2.

- b. Presione F1 para continuar.

- c. Llevar la herramienta TCP al punto de origen. El robot puede asumir cualquier posición: el operador puede posicionar el TCP en el punto de origen como él prefiera, de acuerdo con sus necesidades.

- d. Pulse F1 cuando el robot está a punto de adquirir la posición.

- e. Llevar la herramienta TCP al punto Xpos. El robot puede asumir cualquier posición: el operador puede posicionar el TCP en el punto Xpos que prefiera, de acuerdo con sus necesidades.

- f. Pulse F1 cuando el robot está a punto de adquirir la posición.

g. Llevar la herramienta TCP en el punto XYpos. El robot puede asumir cualquier posición: el operador puede posicionar el TCP en el punto XYpos que prefiera, de acuerdo con sus necesidades.

h. Pulse F1 cuando el robot está a punto de adquirir la posición.

i. El programa muestra los valores calculados de \$UFRAME (offsets XYZ y directrices AER). En este momento se puede mover en el modo UFRAME (seleccione de movimiento de página, menú COORD) para comprobar que el nuevo dato es que se desea. Si no es así, comprobar:

- La calibración del robot.
- Los valores de \$TOOL (deben consultar la herramienta montada en la brida).
- Los 3 puntos de referencia (compruebe que ha cumplido con la orden solicitada por TO_SET) realizar las correcciones y pulse F1 para repetir el procedimiento.

j. Guarde los valores calculados, en la tabla TU_FRAME (como se muestra en la figura de la derecha, pulse F2, introduzca el número de UFRAME deseado y confirmar con ENTER).

2. Descripción de los distintos tipos de movimientos

Como ya se ha explicado anteriormente, el sistema proporciona los siguientes sistemas referencia:

Sistema Mundo – Sistema de referencia en el cual se coloca el robot.

Sistema Básico – Sistema que indica la base del robot.

Sistema Usuario - Indica la pieza de trabajo.

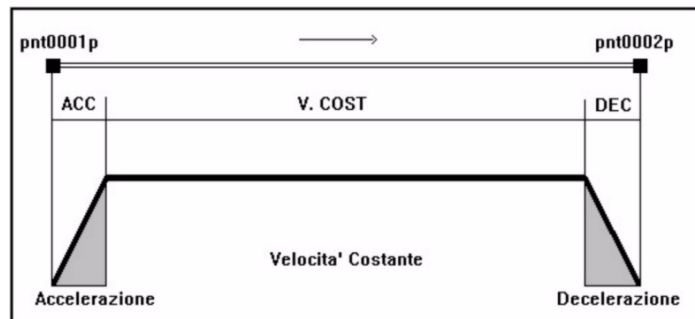
Sistema Brida- Indica la brida del robot.

Sistema TCP - Indica la punta de la herramienta.

Un movimiento se inicia con la aceleración del brazo a lo largo de una trayectoria determinada, continúa con un movimiento de velocidad constante y que termina con una desaceleración, hasta la posición final.

Un movimiento se caracteriza por:

- Trayectoria.
- Velocidad (Speed Control).
- La aceleración y deceleración.
- Parada Movimiento.
- Movimiento Continuo.



2.1. Trayectoria

Representa un movimiento del brazo robot desde una posición inicial a una posición final. Las trayectorias se dividen en:

- Trayectoria conjunta: JUNTAS.
- Trayectoria lineal: LINEAL.
- Trayectoria circular: CIRCULAR.



En el caso de las juntas de interpolación (\$ MOVE_TYPE: = Articulación en o Mover), los ángulos de cada del brazo se interpolan linealmente desde los valores iniciales hasta el final. Los ejes se desplazan a lo largo de cada uno a partir de su posición inicial y que se unen cada uno a su posición final. La ruta seguida por TCP (centro de la herramienta), aunque repetible, no es predecible.

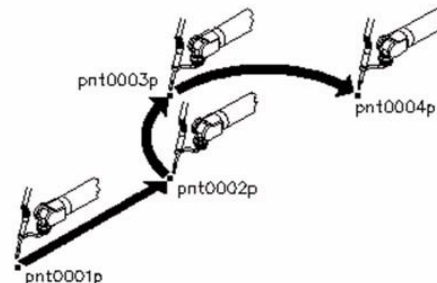


Los movimientos entre dos posiciones por medio de juntas de interpolación son siempre posibles.

Durante la interpolación lineal (\$MOVE_TYPE: = LINEAL o trasladarse a lineal), el punto central de la herramienta (TCP) se mueve a lo largo de una línea recta desde la posición inicial a la final. La orientación de la herramienta también varía desde la inicial a

la posición final en la forma definida por la \$ORNT_TYPE variable. Esta variable específica puede tomar los valores expresados por las siguientes constantes predefinidas: RS_WORLD, RS_TRAJ, EUL_WORLD, WRIST_JNT.

En el caso de la interpolación circular (\$ MOVE_TYPE: = circular o circular para MOVER), el centro de la herramienta sigue un arco desde la posición inicial a la final. Para definir el arco debe especificar una ubicación adicional, la ubicación VIA. Esta posición se utiliza sólo la ubicación de la posición; su ajuste no afecta al movimiento.



En cuanto a la interpolación lineal, el valor predeterminado variable \$ORNT_TYPE indica el tipo de cambio de orientación que se deben hacer.

2.1.1. Evolución de la orientación durante un movimiento Lineal o Circular

La orientación de la herramienta durante los movimientos lineales y circulares, va evolucionando desde el inicio hasta la posición final según el valor que posea la variable \$ ORNT_TYPE. Esta variable puede tomar los siguientes valores:

-RS_WORLD (Dos esquinas en comparación con el punto de referencia).

La evolución de la orientación se lleva a cabo mediante la interpolación lineal de los valores de dos ángulos de rotación: de deslizamiento y el giro de la herramienta. La herramienta (TOOL) es el ángulo alrededor de la perpendicular común entre el soporte de referencia de aproximación inicial y el vector de aproximación final. El giro de la herramienta está representado por el ángulo en torno al enfoque portador desde la posición inicial hasta el final. La evolución de orientación está relacionado con el marco usuario, independientemente de la trayectoria. RS_WORLD es el valor predeterminado de \$ORNT_TYPE.

-RS_TRAJ (Dos ángulos con respecto a la trayectoria)

La Interpolación de orientación se realiza con el mismo criterio de modo RS_WORLD pero el ángulo de inclinación y giro, se refiere a la trayectoria. Esto es particularmente útil durante las trayectorias circulares con ángulo mayor de 180 grados cuando se desea mantener un ángulo constante con respecto a la trayectoria. Durante los movimientos lineales la tendencia de ajuste es el mismo RS_WORLD.

-EUL_WORLD (Tres esquinas)

La evolución de orientación se lleva a cabo mediante la interpolación lineal de los valores de los tres ángulos de rotación de Euler: E1, E2 y E3.

-WRIST_JNT (Articulaciones de la muñeca)

Para la evolución del ajuste, utilizando en este caso una combinación entre la interpolación de articulaciones y la interpolación lineal. Esto permite el movimiento de la herramienta a lo largo de una línea recta, mientras que las articulaciones de la muñeca se interpolan en coordenadas diferentes. Las orientaciones iniciales y finales se utilizan como posiciones aprendidas, pero debido a las interpolaciones de las articulaciones, la evolución durante el movimiento, aunque repetible, no es predecible. Por ejemplo, el uso de EUL_WORLD RS_WORLD o, si la orientación inicial y final son iguales, la orientación de la herramienta no va a cambiar durante el movimiento. Usando WRIST_JNT, sin embargo, esto no está siempre garantizado. Sin embargo, gracias a este modo de evolución de orientación es posible obtener un movimiento más regular en la proximidad de la singularidad de la muñeca.

3. Dinámica del movimiento

3.1. Control de la velocidad

La velocidad máxima o constante de movimiento es controlada por algunas variables predefinidas, las cuales son comunes a todo el sistema, mientras que otras son específicas para el programa, otras son del brazo robot... Se utilizan dos tipos de control de la velocidad:

-Valores De velocidad absoluta o valores límite, medida en unidades de velocidad, tales como radianes o metros por segundo.

-Porcentual (Velocidad) que actúan sobre los valores absolutos de la velocidad.

3.1.1. Anulación de la velocidad

El porcentaje de los valores de velocidad de manejo que son válidos para todos los movimientos son los siguientes:

- \$ GEN_OVR permite a un operador cambiar simultáneamente la aceleración, la velocidad y la desaceleración de los programas de movimiento. Como esto afecta a la aceleración, la velocidad y la desaceleración de una manera coordinada, las trayectorias generalmente se mantienen (a menos de errores de seguimiento servo) cuando esta variable es modificada. Es común a todo el sistema y puede ser modificado por el terminal de programación. A los programas PDL2 se pueden acceder en sólo lectura.

- \$ ARM_OVR permite la edición de la aceleración, la velocidad y la desaceleración relacionada con un brazo específico. Ya que esto afecta la aceleración, la velocidad y la desaceleración de manera coordinada, las trayectorias generalmente se mantienen (a menos de errores de seguimiento servo) cuando se cambia esta variable.

Tenga en cuenta que si durante las transiciones entre los movimientos continúan poniendo énfasis en mantener una velocidad constante en lugar de dejar que la trayectoria sin cambios para variar la anulación, debe utilizar las variables \$ o \$ ARM_SPD_OVR PROG_SPD_OVR en lugar de \$ ARM_OVR (los valores de aceleración y desaceleración no se verá afectada por la reducción de estos corrección de la velocidad).

- \$ ARM_SPD_OVR permite la edición de los valores de velocidad para un brazo específica sin afectar a la aceleración y desaceleración. Esto significa que el perfil de las trayectorias de los movimientos en FLY puede cambiar si se realizan cambios en esta variable.

Existe un elemento para cada brazo y el valor predeterminado para el campo ARM_SPD_OVR es 100%.

- \$ PROG_SPD_OVR permite la edición de la velocidad para todos los movimientos de un programa sin el cambio de la aceleración y deceleración. Esto significa que el perfil de las trayectorias de los movimientos continuos puede cambiar si se realizan cambios en esta variable. Es un valor específico del programa, su valor predeterminado es 100%.



\$GEN_OVR y \$ARM_OVR tienen efecto durante el movimiento mismo. Esto significa que en caso de modificación de una de las variables durante la ejecución del movimiento, habrá una consiguiente aceleración o desaceleración del movimiento mismo. Por el contrario, un cambio en las variables \$PROG_SPD_OVR o educación \$ARM_SPD_OVR solo tendrá efecto en el próximo destino de viaje a ese curso, como parte del programa al que se refieren.

El porcentaje requerido se puede asignar cualquiera de los métodos modal nodal.

La asignación modal tiene validez definitiva (hasta la siguiente asignación de tipo modal), mientras que la asignación nodal tiene validez temporal (por sólo el movimiento que está asociada).

Ejemplo:

\$ARM_SPD_OVR: = 50 - Asignación modal: velocidad actual 50%

MOVER lineal para pnt0001p,

CON \$ ARM_SPD_OVR = 10 - Asignación nodal: Velocidad temporal del 10%

ENDMOVE

- Velocidad actual 50%

3.1.2. Control de la velocidad cartesiana

En condiciones normales de funcionamiento, la velocidad de movimiento cartesiano (lineal o circular), con cualquier modo de evolución de orientación, se controla por medio de dos variables predefinidas.

- \$LIN_SPD_LIM define el límite de velocidad de traslación lineal. Es una variable para cada brazo, cuyo valor depende del robot y no es modificable por el usuario.
- \$ROT_SPD_LIM define el límite de velocidad de rotación. Es una variable para cada brazo, cuyo valor depende del robot y no es modificable por el usuario.

El componente del movimiento que toma más tiempo para pasar de la inicial a la posición final, se moverá de acuerdo con el límite de velocidad programada.

Todos los otros componentes se moverán a una velocidad inferior a los límites programados para que todos los movimientos se inicien y concluyan juntos.

3.1.2.1. Opciones de control de la velocidad cartesiana

El proceso por el cual se determina el componente que controlará la velocidad cartesiana se conoce como pre-ingeniería, y se produce inmediatamente antes del curso real del movimiento (es decir, antes de cada movimiento).

Se puede obligar a la planificación de utilizar un componente particular del movimiento por medio de la opción predeterminada \$SPD_OPT; es una variable específica del programa (cada programa puede tener su propio valor) a la que se pueden asignar las siguientes constantes predefinidas:

- SPD_CONST Es el valor predeterminado. Mueva el brazo a una velocidad constante mientras que el valor SMS se selecciona.
- SPD_JNT Mueve el brazo a lo largo de la trayectoria cartesiana, a velocidad máxima de al menos una articulación; el centro de la herramienta no se moverá a una velocidad constante.
- SPD_LIN Mueve el centro de la herramienta sobre la base de la solicitud \$LIN_SPD, forzando al componente de rotación el moverse de una manera coordinada.

3.1.3. Control de la velocidad de las articulaciones

El porcentaje de cada articulación se determina por \$JNT_OVR que le permite editar simultáneamente el programa de la aceleración, la velocidad y deceleración.

No es un reemplazo para cada eje de cada brazo con un valor por defecto de 100%.

3.2. Aceleración y Deceleración

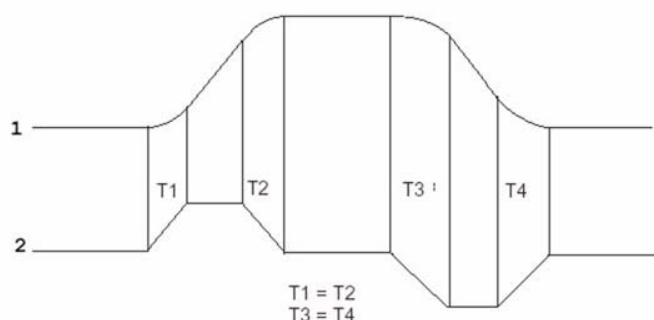
En Controller C5G, la aceleración y desaceleración están subdivididas en tres fases: una fase en la que las de aceleraciones aumentan linealmente (sacudida), una fase de aceleración constante y, finalmente, otra fase en la que la aceleración disminuye linealmente.

Actualmente, C5G fuerza el perfil de aceleración y la desaceleración de modo que son simétricas.

Esto significa que las fases de tirón constante durante la aceleración (T1 y T2) tienen la misma duración y las fases de tirón constante durante la deceleración (T3 y T4) tienen la misma duración.

Como la velocidad, la aceleración y deceleración pueden ser modificados por porcentajes variables (Override). \$GEN_OVR y \$ARM_OVR actúan no sólo en la velocidad, sino también de la aceleración y deceleración.

Sin embargo, hay algunas variables que afectan aún más la aceleración y deceleración. Estas variables son del brazo o programa específico:



Sin embargo, hay algunas variables que afectan aún más a la aceleración y deceleración. Estas variables son del brazo o programa específico:

- \$PROG_ACC_OVR permite la edición de la aceleración de todos los movimientos de un programa dado. Es un valor específico del programa con un valor por defecto de 100%.
- \$PROG_DEC_OVR permite la edición de la desaceleración de todos los movimientos de un programa dado. Es un valor específico del programa con un valor por defecto de 100%.
- \$ARM_ACC_OVR permite la edición de la aceleración de un brazo específico. Y 'un campo entero con un campo para \$ARM_DATA cada brazo. El valor por defecto es 100%.
- \$ARM_DEC_OVR permite la edición de la desaceleración de un brazo específico. Y 'un campo entero con un campo para \$ARM_DATA cada brazo. El valor por defecto es 100%.

El cambio de estas cuatro variables solamente tendrá efecto en los movimientos posteriores, no en el movimiento actual. En caso de uso de la instrucción de espera, cancelar, bloquear o desactivar, se utilizará el valor máximo de deceleración, sin tener en cuenta el porcentaje de valores de velocidad ya establecido.

3.3. Precisión de posicionamiento

Determina la precisión de posicionamiento del robot en el punto final de un movimiento antes de la siguiente operación se puede realizar.

Como variable por defecto se utiliza \$TERM_TYPE para la definición del momento en el que el movimiento tiene que ser considerado terminado; Esta definición se basa en la distancia a la que debe estar el brazo desde el punto de destino.

-COARSE - Debe ser utilizado en los movimientos cartesianas e indica que el movimiento se considera terminado cuando el TCP permanece dentro de una esfera que contiene al punto final con un radio definido por la \$TOL_COARSE, variable para un tiempo mayor que o igual a \$TOL_ABT (tiempo anti-rebote).

-END - Ha de ser utilizado en los movimientos cartesianos e indica que el movimiento se considera terminado cuando el TCP permanece en una esfera que contiene al punto final con un radio definido por el \$TOL_FINE variable por un tiempo superior o igual a \$ TOL_ABT (tiempo anti-rebote).

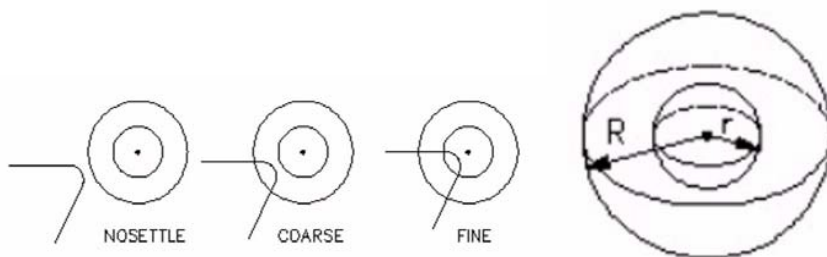
-NOSETTLE - La comprobación se efectúa de manera lógica. Es la asignación predeterminada. Representa el grado más bajo de precisión.

-JNT_COARSE - Debe ser utilizado en los movimientos de las articulaciones e indica que el movimiento se considera terminado cuando cada articulación se encuentra a una distancia (en grados) inferior a la especificada por la variable de \$TOL_JNT_COARSE del eje correspondiente.

-JNT_FINE - Debe ser utilizado en los movimientos de las articulaciones e indica que el movimiento se considera terminada cuando cada articulación se encuentra a una distancia (en grados) inferior a la especificada por la variable de \$ TOL_JNT_FINE. El predefinido variables \$TOL_COARSE y \$TOL_FINE indican los valores de tolerancia. El valor predeterminado de \$TOL_COARSE y \$ TOL_FINE depende del brazo.

La variable predefinida \$TOL_ABT (tiempo anti-rebote) es el tiempo durante el cual el brazo debe estar dentro de la tolerancia especificada antes de que el movimiento se declara acabado/terminado.

Las variables predefinidas \$TOL_JNT_COARSE y \$TOL_JNT_FINE indican los valores de tolerancia del grueso y fino, respectivamente, para cada articulación, medida en grados.



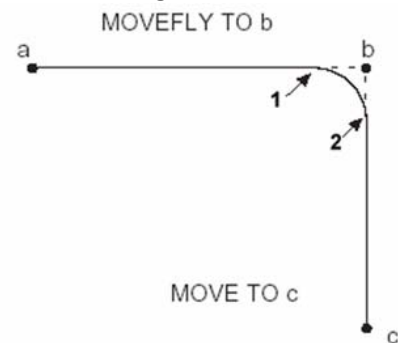
3.4. Movimiento Continuo

El movimiento continuo permite que el programa se ejecute sin causar la detención del brazo en las posiciones almacenadas.

Para indicar un movimiento continuo, se utiliza la educación MOVEFLY en lugar de dar MOVE.

Si MOVEFLY sigue otro movimiento, el brazo no se detendrá en el primer objetivo, pero se moverá desde el punto de partida del primer movimiento hasta el punto final del segundo movimiento sin detenerse en el punto común a los dos movimientos.

La variable predefinida \$FLY_TYPE determina lo que será el algoritmo que seguirá de cerca al movimiento continuo.



Este algoritmo no sólo afecta a la velocidad y la trayectoria del perfil durante el movimiento continuo, sino también la magnitud de los esfuerzos ejercidos por el brazo y en el componente o herramienta conectada a él.

El valor por defecto es FLY_NORM.

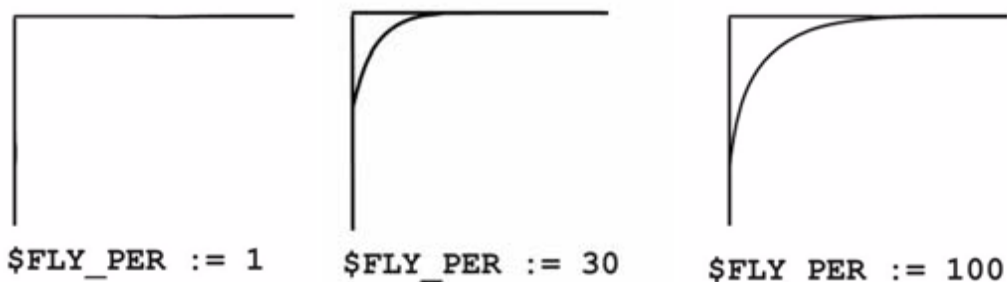
-FLY_NORM

Cuando \$FLY_TYPE se establece en FLY_NORM, MOVEFLY provoca la desaceleración del movimiento en curso que se superpone con la aceleración de la siguiente movimiento.

La variable \$FLY_PER predefinida (la única variable vinculada a FLY_NORM) se puede utilizar para reducir la marcha acercándose a la posición media guardada.

\$FLY_PER representa el porcentaje de retraso en la ejecución del movimiento en el movimiento continuo. La reducción de su valor, el movimiento tendrá que tener un mayor retraso y la trayectoria se acercará a la posición especificada en MOVEFLY (ver figuras de abajo).

Por ejemplo, si el valor de \$FLY_PER es 100%, el recorte comienza al principio de la deceleración del movimiento en la marcha. Si el valor de \$FLY_PER es 75 el recorte sólo comienza cuando el 25% del tiempo de desaceleración ha sido ejecutado (el 75% restante se une a la siguiente movimiento).



4. Introducción al ambiente de programación IDE

Los programas de control usados en C5G están escritos en el lenguaje de programación PDL2. Los métodos utilizados para la creación de tales programas pueden variar ampliamente dependiendo del programador y la aplicación.

El entorno de desarrollo principal es:

- Ambiente IDE - entorno de desarrollo que debe ser utilizado para desarrollar programas de movimiento en el Teach.

Prog (F1) - Crear Comando

Se le permite crear un nuevo programa para ser editado en el entorno IDE. Es posible que tenga las siguientes situaciones:

- El programa deseado ya existe en la memoria de ejecución: el comando no hace ningún efecto.
- El programa existe requerido sólo en forma de archivo de UD: El comando carga memoria de ejecución (ML).
- El programa requerido no existe. El comando crea un archivo .COD y lo carga en memoria.

Para abrir el programa creado en el IDE, utilice el comando Prog (F1) - IDE abierto (1 selección), o seleccionar el programa que acaba de crear y pulse ENTER.

Motion

Alarm

Prog

IDE

I/O

Appl

Progr

HOLD

100%(I)

A:1 J:-,8

Joint

--

123* --

No messages

Visualizzazione della memoria:

Nome	Attributi	Memoria	Esecuzione	Linea	Proprieta
APPL_STRTP	NoHold	Saved	Not active	0	APPL_S
b	NoHold	Saved	Suspended	6	b
esempio	Hold	Saved	Not active	0	esempio
gl_dbead_data	NoHold	Saved	Not active	0	gl_dbead
1 Apri IDE [ENTER]			Running	35	h_spot
2 Crea			Not active	0	prog_1
3 Salva			Not active	0	prog_33
4 Salva le variabili			Held	16	prog_45
5 Salva il codice			Not active	0	prog_55
6 Salva variabili con nome (As)					
7 Cancella dalla memoria di esecuzione					

1 Cancella codice e variabili

2 Cancella solo il codice

3 Cancella solo le variabili

Prog

Esecuzione

Carica...

Visualizza

POS SHIFT

Dettagli

U1

U2

U3

U4

T1

T2

4.1. Programación en IDE (con TP) de un programa de movimiento

Los pasos para el desarrollo de un programa en el entorno IDE, son los siguientes:

- a. Edición de programas
- b. Guardado de posiciones (inserción de una instrucción de movimiento)
- c. Test de la ejecución del programa
- d. Guardar el programa y cerrar el IDE

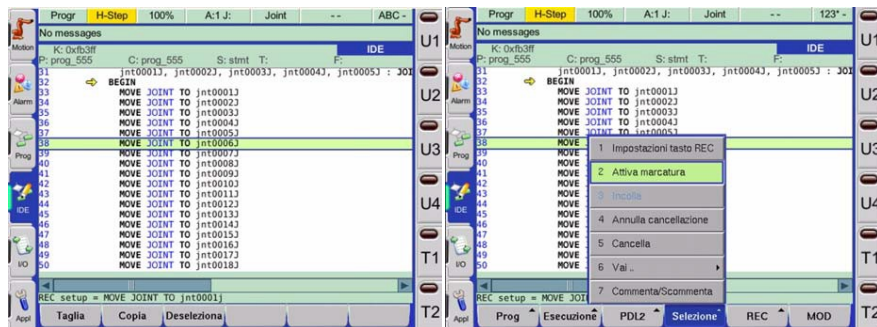
4.1.1. Edición de programas

Ahora vamos a ver cómo llevar a cabo las principales funciones de edición de un programa: se enumeran a continuación.

- Seleccionar una o más líneas de texto, eliminar / restaurar, copiar.
- Mover el cursor de edición.
- Introducir una nueva instrucción.
- Comentar / descomentar una línea en el programa.
- Modificar una instrucción existente.
- Introducir / ver / eliminar variables.
- Importación de otros programas.
- Mostrar un programa.

4.1.1.1. *Seleccione una o más líneas de texto, eliminar / restaurar, copiar*

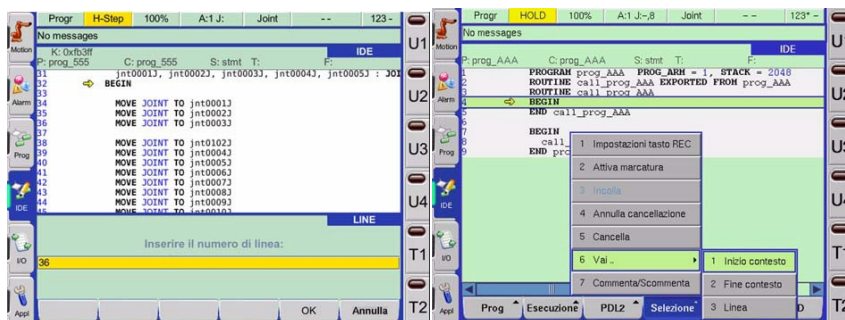
- a. Pulse la tecla de función de selección (F4) en el menú principal de la página IDE.
- b. El menú se abre por el sistema, seleccione el movimiento Habilitar marcado en la edición, use las teclas de cursor para realizar el marcado; en el menú central son 3 funciones.
 - b.1 Para cortar el texto seleccionado, utilice el comando Cortar (F1) - Trozos de las líneas seleccionadas (F1).
 - b.2 Para copiar el texto seleccionado, utilice el comando Copiar (F2) - copiar las líneas seleccionadas (F2).
 - b.3 Para cancelar el marcado, elija Seleccionar (F3) - lejos de la marca (F3).
- c. Para entrar en la parte del texto cortado o copiado, utilice la función Pegar, presente en el menú desplegable. Vd.
- d. Para restaurar la parte del texto previamente cortado (BORRAR), utilice el Undelete, presente en el menú desplegable, el texto se inserta debajo de la posición actual del cursor de edición.



4.1.1.2. Mover el cursor de edición

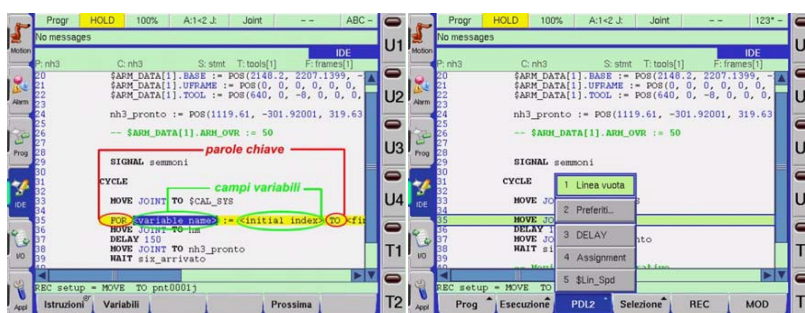
Para mover el cursor de edición, la forma más fácil es usar las teclas del cursor; pero si usted quiere moverlo a un lugar determinado, haga lo siguiente:

- a. Pulse la tecla de función de selección (F4) en el menú principal del IDE
- b. El menú se abre por el sistema, seleccione la función Ir...
- b.1 Elegir entre la instrucción, la instrucción END comenzar o indican una específica número de línea.



4.1.1.3. Introducir una nueva instrucción

- a. Pulse la tecla de función (F3) en el menú principal de la página IDE.
- b. El sistema presenta las últimas 9 instrucciones usadas con más posibilidades de elección.
 - b.1 Si aparece la instrucción deseada, seleccionarla y pulse ENTER: Esta se inserta debajo de la línea actual para EDITAR.
 - b.2 Si no está en la lista, elija 'Línea en blanco' para abrir una nueva línea de programa que se sitúa por debajo de la señalada en el EDITAR. Introduce la nueva instrucción en una de dos maneras:
 - b.2.1 Guiado de inserción, a través de la ayuda de una plantilla: pulse las Instrucciones de teclas de función (F1) y elija la clase de instrucción, pulse ENTER, y seleccione la instrucción deseada. Rellene los campos de la plantilla.
 - b.2.2 Entrada directa desde el teclado alfanumérico.
- c. Sea cual sea el método que se utiliza, cuando se acabe se debe confirmar con ENTER. El sistema prevé el control del programa de la línea insertada e informa de los errores.

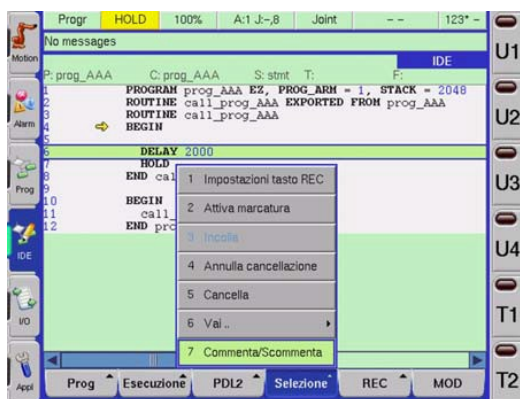


4.1.1.4. Comentar / descomentar una línea en el programa

a. Pulse la tecla de función de selección (F4).

b. El menú se abre por el sistema, seleccione la función de comentario/Descomentar.

Este botón actúa como un conmutador: cada vez que se pulsa invierte la situación actual.

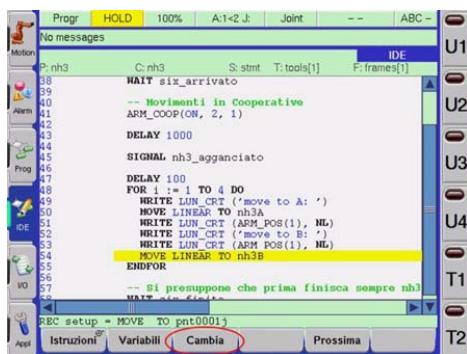


4.1.1.5. Modificar una instrucción existente

a. Coloque el cursor en EDITAR y pulse ENTER: se introduce en el modo de edición.

b. Utilizar el teclado alfanumérico o los comandos de menú central, similar a lo que se describe para la realización de las instrucciones de comando.

Caso especial: Modificación de una instrucción MOVE. Un caso especial es la modificación de declaración MOVE. En este caso también existe la función de cambio, asociado a la tecla F3, lo que le permite editar los parámetros de MOVE.



4.1.1.6. Introducir / ver / eliminar variables

Para insertar una nueva variable en la línea de programa que se está señalado por el cursor de edición, usted debe:

a. Pulsar la tecla (F1) en el menú principal de la página IDE.

b. Seleccionar la función de variables.

c. Elegir la selección 3 - Intro.

d. El sistema abre una ventana de diálogo en la que el usuario debe presentar el nombre y el tipo de la variable que desee, eligiendo entre los que figuran en la lista que aparece, o escribiendo directamente a través del teclado alfanumérico.



e. para terminar la edición de prensa ENTER.

Para ver las variables utilizadas en una línea de programa, es necesario:

a. Colocar el EDITAR de la línea de programa deseada.

b. Pulse la tecla Prog (F1) en el menú principal de la página IDE.

c. Seleccionar la función Variables desde el menú principal de la página IDE.

d. Elegir entre la selección. 1 - Véase la línea de Var

También se puede eliminar la declaración de variables que no se han utilizado:

a. Pulse la tecla Prog (F1) en el menú principal de la página IDE.

b. Seleccionar la función Variables desde el menú principal de la página IDE.

c. Elegir entre la selección 2 - Retire las variables utilizadas.

4.1.1.7. Importación de otros programas

Para importar desde otro programa se ha de realizar lo siguiente:

- a. Pulsar la tecla Prog (F1) en el menú principal de la página IDE.
- b. Seleccionar la función de importación.
- c. Dependiendo del objeto a ser importado tienen uno o dos cuadros de diálogo. Para la importación de variables y rutinas se han de tener dos cuadros de diálogo; para importar tipos no es necesario tener las dos ventanas. El usuario responde las preguntas y pulsa ENTER para confirmar.

La declaración de clases importadas son:

- Routine
- Tipo
- Variable

4.1.1.8. Mostrar un programa

Para moverse entre líneas de la página de un programa, debe utilizar las teclas de página arriba y página abajo.

4.1.2. Guardado de posiciones (inserción de una instrucción de movimiento)

En el ambiente IDE se puede asignar a las variables posicionales posición del brazo actuales, mediante la inclusión en las instrucciones de movimiento del código de programa asociado con ellos y las correspondientes declaraciones de tales variables posicionales.

Las demandas relacionadas con el aprendizaje de las posiciones, son los siguientes:

- Ajuste el botón REC.
- Grabar nuevas posiciones.
- Modificar las posiciones existentes.

También existe la posibilidad de configuración nodal (OPCIONAL).

4.1.2.1. Ajuste el botón REC

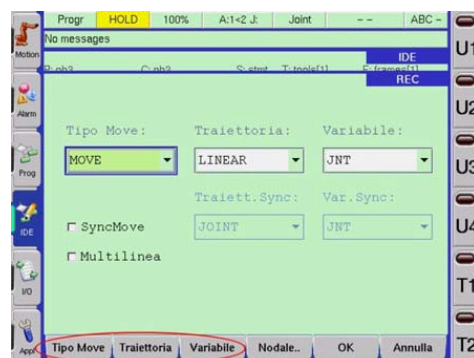
Le permite configurar las características de la medida que se inserta en el programa, cada vez que se pulsa el botón.

Al abrir la IDE, el botón REC está ajustado para articulaciones modales, si no ha sido previamente establecido. Para cambiar esta situación debe utilizar esta función REC Settings o activar esta función directamente, mientras pulsa SHIFT + REC.

a. Pulse la tecla de función de selección (F4) en el menú principal de la página IDE.

b. El menú se abre por el sistema, elija la configuración de teclas de función REC y siga las instrucciones. El usuario puede elegir:

- el tipo de variable posicional - POSICIÓN, JOINTPOS, XTNDPOS.
- Trayectoria - DEFAULT, CONJUNTA, lineal, circular.
- Continuo o no continuo...



4.1.2.2. Grabar nuevas posiciones

Una vez ajustado correctamente el botón REC, usted debe:

- a. Colocar el cursor en EDITAR en la línea de programa bajo el cual desea insertar la nueva instrucción MOVE.
- b. Mover el robot utilizando las teclas de JOG, hasta llegar a la posición que desee registrar.
- c. pulse el botón REC. Esto crea en el programa una instrucción MOVE y la declaración de variables de posición utilizado por el MOVIMIENTO misma.

4.1.2.3. Modificar las posiciones existentes

Si desea cambiar el contenido de una variable de posición que ya existe, es necesario:

- a. Mover el cursor de edición de MOVE a la línea que desea cambiar la variable de posición.
- b. Mover el robot utilizando las teclas de JOG, hasta llegar a la nueva ubicación.
- c. Pulse la tecla de función MOD (F6) en el menú principal de la página IDE - el sistema presenta un cuadro de diálogo en el que se ve una lista de las variables de posición de línea.
- d. El usuario debe seleccionar la variable en cuestión y confirmar el cambio con la tecla ENTER o cancelarlo con ESC.



4.1.3. Test de la ejecución del programa

Como se mencionó anteriormente, el IDE es un entorno de edición de programas y desarrollo integrado. Es extremadamente importante que todos los programas sean cuidadosamente revisados antes de ser ejecutado de forma automática.

Algunos comandos que afectan a la ejecución del programa actualmente abierto en el IDE se describen aquí, para que pueda tomar el desarrollo (depuración), identificar los errores y hacer mejoras.

- Activar/Desactivar un programa.
- Interrumpir la ejecución de una instrucción.
- Establecer el modo de paso.
- Mover el cursor en ejecución.
- Insertar/Retirar un punto de ruptura.
- Saltarse la ejecución de una instrucción.
- Ejecutar una instrucción temporal.

4.1.3.1. Activar/Desactivar un programa

Un programa correctamente abierto en IDE siempre está activo. Si, después de los cambios, desea iniciarlo de nuevo, usted debe:

- Pulse la tecla de función Ejecución (F2) en el menú principal de la página IDE.
- En el pop que se abre por el sistema, seleccione la función de inicialización (Reset) - Por lo tanto, el programa se desactiva y vuelve a activar.

Si simplemente desea desactivar el programa, es necesario:

- pulse la tecla de función Ejecución (F2) en el menú principal de la página IDE.
- el telón se abre por el sistema, seleccione la función de Desactivación (selección 2).

4.1.3.2. Interrumpir la ejecución de una instrucción

Si desea detener la ejecución de la instrucción actual:

- Pulse la tecla de función Ejecución (F2) en el menú principal de la página IDE.
- Un menú se abre en el sistema, seleccione la función de parada.

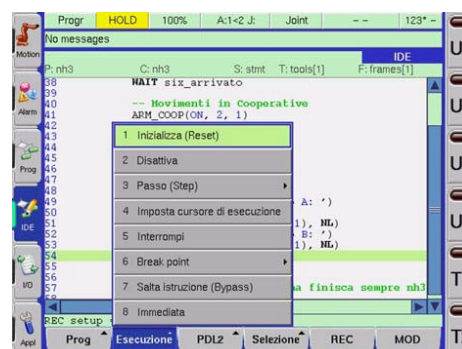
4.1.3.3. Establecer el modo de paso

Normalmente, la ejecución del programa es continua, es decir sin interrupción. El paso de ejecución es definir la hora a la que se suspenderá la ejecución.

Un nuevo paso de ejecución se inicia cada vez que se pulsa el botón START.

Para establecer el modo de ejecución de la fase del programa, es necesario:

- Presionar la función de ejecución (F2) en el menú principal de la página IDE.
- El menú se abre por el sistema, cambiar la función Step (Paso) - 3 selección.
- De la nueva lista abierta por el sistema, seleccione el modo deseado.



Los siguientes modos están disponibles:

- La instrucción individual - La ejecución se detiene en cada instrucción. Rutina protegida (cuyo cuerpo no se muestra) se realizan como si fueran una sola instrucción; la ejecución de la rutina sin protección (cuyo cuerpo es visible) se interrumpe en cada instrucción.
- Desactivado - Pone medidas de ejecución término del programa.
- En Movimiento - La ejecución se detiene en cada movimiento. No se permite en los programas no se puede sujetar.
- El Movimiento FLY - Desempeña dos o más movimientos en la marcha antes de suspender la ejecución del programa. Es similar a la opción Mover, pero la ejecución no se detiene después de la educación MOVEFLY.
- El ciclo de programa - Define cómo lanzar un solo ciclo del programa que debe incluir instrucción comenzar el ciclo o ciclo.

4.1.3.4. Mover el cursor en ejecución

Para mover la ejecución del programa a una vía diferente a la actualmente apuntada por el CURSOR, debe:

- a. Mover el puntero a EDITAR en la línea de programa que desea ejecutar.
- b. Pulsar la tecla de función Ejecución (F2) en el menú principal de la página IDE.
- c. Un menú se abre por el sistema, seleccione el cursor de ejecución fijado. 4 de selección.

4.1.3.5. Insertar/Retirar un punto de ruptura

Un punto de inflexión es un punto de interrupción de la ejecución.

Para insertar un punto de ruptura, es necesario:

- a. Colocar el cursor de edición en la línea ante la que deba insertar el punto de ruptura.
- b. Pulsar la tecla de función Ejecución (F2) en el menú principal de la página IDE.
- c. Un menú se abre por el sistema, seleccione la función de punto de ruptura.

Para eliminar un punto de interrupción, debe hacer las mismas tareas que se realizan para agregar.

4.1.3.6. Saltarse la ejecución de una instrucción

Para continuar la ejecución del programa, cuando se está a la espera para la realización de una instrucción (por ejemplo: ESPERAR A, retardo, etc.), puede utilizar el comando.

Instrucción de salto (especial):

- a. Pulse la tecla de función Ejecución (F2) en el menú principal de la página IDE.
- b. Un menú se abre por el sistema, seleccione la instrucción de salto (bypass). Selección 7.

4.1.3.7. Ejecutar una instrucción temporal.

Para ejecutar una instrucción que, sin embargo, no se incluye en el cuerpo del programa actual, es necesario:

- a. pulse la tecla de función Ejecución (F2) en el menú principal de la página IDE.
- b. Un menú se abre por el sistema, seleccione la función inmediata.
- c. En el menú central que está disponible la función de Declaración que se abre un cuadro de diálogo para la entrada de instrucción deseada.
- d. Seleccione la instrucción y pulse ENTER para confirmar su elección. También "la posibilidad de entrar directamente en la instrucción, con el teclado alfanumérico.
- e. Para confirmar la ejecución de la opción, pulse 'Aceptar'.

4.1.4. Guardar el programa y cerrar el IDE

Para guardar tanto el .COD como el .VAR del programa actualmente abierto en el IDE, debe:

- Pulsar la tecla de función Prog (F1) en el menú principal de la página IDE.
- Un menú se abre por el sistema, seleccione la función Guardar (código y var). El programa guardado permanece abierto en el IDE.

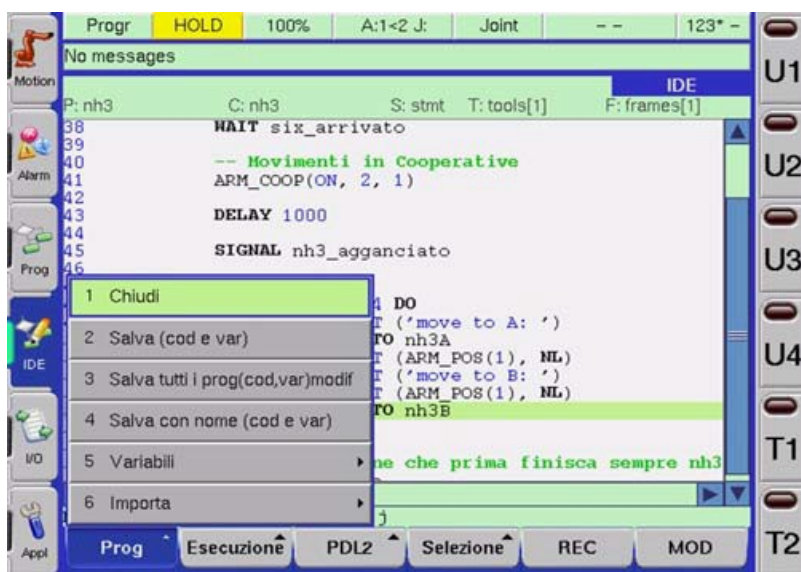
Para guardar todos los programas de medio ambiente IDE modificado:

- Pulsar la tecla de función Prog (F1) en el menú principal de la página IDE.
- Un menú se abre por el sistema, seleccione la función Guardar todo prog (bacalao y var) modificado.

Para cerrar el entorno IDE:

- Pulsar la tecla de función Prog (F1) en el menú principal de la página IDE.

- Un menú se abre por el sistema, seleccione la función Cerrar.



4.2. Ejecución de un programa en modo automático

a. Compruebe lo siguiente antes de proceder a ejecutar automáticamente los programas:

- El programa debe haber sido sometido a un control minucioso y se sabe que es capaz de funcionar correctamente.
- No debe haber nadie en la zona de trabajo y la misma debe estar libre de obstáculos.
- Todos los medios de seguridad deben estar instalados y en funcionamiento.
- Todas las demás condiciones específicas relativas a la aplicación y la instalación, se deben revisar a fondo y se considerará capaz de funcionar correctamente.

b. presione el botón HOLD (amarillo).

c. posicionar el estado del selector AUTO (o remoto).

d. establecer el porcentaje de la velocidad de movimiento (corrección de la velocidad) deseada, el modo de paso deseado, entrar en cualquiera de los puntos de quiebre.

e. Pulse el botón START (verde) para ejecutar el programa (de acuerdo con el modo de paso establecido previamente).

5. Programa PDL2 – Conceptos básicos

Los programas para la Unidad de Control C5G están escritos en lenguaje PDL2.

-Estructura De un programa PDL2.

-Atributo HOLD / NOHOLD.

5.1. Estructura De un programa PDL2

Un programa en PDL2 tiene la siguiente estructura:

```
PROGRAM name <attributes>
  <import statements>
  <constant, variable, and type declarations>
  <routine declarations>
BEGIN <CYCLE>
  <executable statements>
END name|
```

5.2. Atributo HOLD / NOHOLD

Se pueden clasificar en dos categorías principales, dependiendo de la atributo HOLD / NOHOLD asignado:

'Holdable' (es decir atributo HOLD) -Su ejecución se controla mediante el botón HOLD (amarillo) y START (verde); un programa que contiene instrucciones de movimiento debe ser del tipo "HOLD".

-Programas "NOHOLD" (es decir, con el atributo NOHOLD) - que se utilizan generalmente para el control de procesos. No pueden contener instrucciones de movimiento a pesar de que todavía pueden utilizar variables de posición, para otros fines.

6. Ejemplo de estructura

```

PROGRAM main
-- dichiarazioni di variabili locali
VAR pnt0001j, pnt0002j, pnt0003j, fuori_ingombro : JOINTPOS FOR ARM[1]
-- dichiarazioni di costanti locali
CONST num_parts=4
-- dichiarazioni di routine importate da altri programmi
ROUTINE call_l_sgr EXPORTED FROM l_sgr -- routine importata da 'l_sgr'
ROUTINE help EXPORTED FROM gest_tasti -- routine importata da 'gest_tasti'
-- dichiarazioni di routine locali al programma 'main'
ROUTINE selez -- routine locale al programma 'main'
VAR lun : INTEGER -- dichiarazione di variabile locale alla routine 'selez'
BEGIN
    .....
    help -- chiama la routine 'help', importata da 'gest_tasti'
END selez

ROUTINE times -- routine locale al programma 'main'
BEGIN
    .....
END times

-- inizio del corpo del programma 'main'
BEGIN
    .....
    CONDITION[1] :
        WHEN $DIN[5] - DO
            times -- chiamata alla routine locale 'times'
        ENDCONDITION
    .....
    call_l_sgr -- chiamata alla routine 'call_l_sgr, importata da 'l_sgr'
    .....
END main

PROGRAM l_sgr EZ, STACK = 2048
-- dichiarazioni di variabili locali al programma 'l_sgr'
VAR inizio_ciclo, pnt0003p, p1, p2, p3, p4 : POSITION

-- dichiarazione di variabili importate da 'gest_tasti'
VAR d_p, d_s_finitura, altezza_bordo : REAL EXPORTED FROM gest_tasti

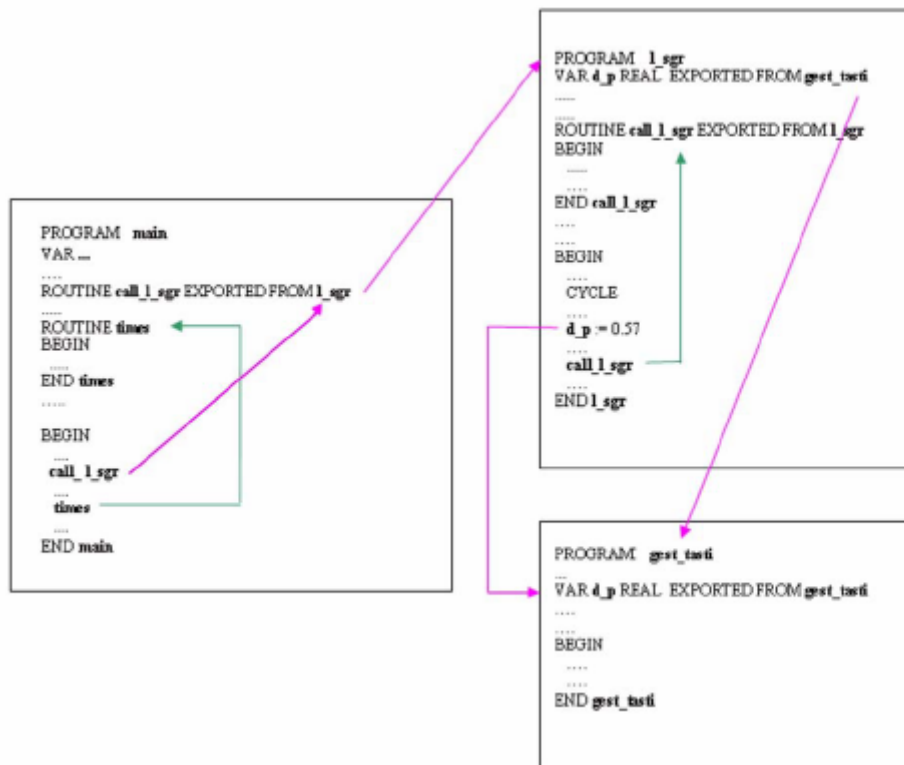
-- dichiarazione di variabile locale al programma 'l_sgr'
VAR cont : INTEGER

-- dichiaraz. di routine definita nel programma 'l_sgr' ed esportata all'esterno
ROUTINE init_motion EXPORTED FROM l_sgr
ROUTINE init_motion -- definizione della routine 'init_motion'
BEGIN
    .....
END init_motion

-- dichiaraz. di routine definita nel programma 'l_sgr' ed esportata all'esterno
ROUTINE call_l_sgr EXPORTED FROM l_sgr
ROUTINE call_l_sgr -- definizione della routine 'call_l_sgr'
BEGIN
    .....
    init_motion -- chiamata alla routine 'init_motion' definita in 'l_sgr' ed esportata all'esterno
    .....
    MOVE LINEAR TO pnt0003p
    .....
END call_l_sgr

-- corpo del programma l_sgr
BEGIN
    CYCLE
    d_p := 0.57 -- variabile importata da 'gest_tasti'
    call_l_sgr -- chiamata alla routine 'call_l_sgr'
    WAIT FOR $FDIN[21]
END l_sgr

```

7. Ciclo de vida de un programa PDL2

- Establecimiento y desarrollo de un programa.
- Guardar el programa en un dispositivo de almacenamiento externo.

7.1. Establecimiento y desarrollo de un programa

Los pasos recomendados para el desarrollo del programa son:

- Escribir el programa (utilizando un entorno de desarrollo - titulada IDE), que es escribir el código, la definición de nuevas rutinas, definir variables y aprender los valores posicionales.
- Verificar la corrección del programa al hacer las instrucciones. Si es necesario utilizar los comandos de depuración (que se encuentran en el entorno de desarrollo) que le permiten cambiar el modo de reproducción (una instrucción a la vez, de forma continua, etc.), para examinar los valores de las variables, para modificar el código de programa en sí. Los movimientos pueden ser probados, aunque a una velocidad reducida.
- Salvar a las variables (creación de un archivos .VAR) y el código (la creación de un .cod) los archivos del programa desarrollado. La copia de seguridad se realiza automáticamente en el directorio de usuario (UD :).
- Cuando el usuario está satisfecho con el funcionamiento del programa, intenta ejecutar de una manera más similar a lo que se operan en la producción (IDE o depuración de memoria).
- En este punto, el programa está listo para ser ejecutado de forma automática. Guarde última vez el código y las variables utilizando el comando apropiado ahorrar disponible en el entorno de desarrollo que está utilizando.
- Activar el programa Prog página (o comando PG), y pulse el botón START (verde).

7.2. Guardar el programa en un dispositivo de almacenamiento externo

Después de haber creado, desarrollado y guardado un programa, es necesario realizar una copia de ellos incluso en un dispositivo de almacenamiento externo (disco flash USB, PC, etc.).

Para ello, utilice el comando Enviar a (FC) disponible.

Cuando se utiliza el comando, especifique el dispositivo de destino: Unidad flash USB (XD :) u otro dispositivo (COMP :).

Si desea guardar todo el contenido de UD:, utilice el procedimiento adecuado para la copia de seguridad y restauración.