

Funciones Modo abierto

Al poner el sistema en modo abierto desde la HMI, el sistema conmuta las conexiones a la bornera. Las conexiones que conmuta son:

- Entradas de los drivers de los motores
- Conexiones de los encoders
- Salida digital del sistema de prevención de colisiones
- Terminales de lectura de los sensores:
 - Temperatura y humedad
 - Nivel de corriente

Funciones Modo Cerrado

El usuario puede acceder a las siguientes funciones desde los lenguajes C++, Python, MATLAB y LabVIEW. Al final de ejecutar alguna de las funciones que mueven el robot, el sistema puede entregar un listado con todos los valores que tuvieron todos los sensores a lo largo de la ejecución de dicha instrucción, así como los parámetros involucrados en el control.

Todos los movimientos tendrán un perfil de velocidad que permita ir del reposo a la velocidad final, para terminar en el reposo nuevamente. La velocidad final aplica para el movimiento de cada una de las juntas y esta expresada en porcentaje.

- ***set_joint_angle(int joint, float angle, float speed)***: esta función permite darle un valor a la junta seleccionada en grados con una velocidad máxima determinada por *speed*. El sistema moverá dicha junta al valor seleccionado usando los parámetros internos del PID del sistema.
- ***set_position(float xf, float yf, float zf, float phi/roll, float theta/pitch, float psi/yaw, int frame, float speed, float aux_x, float aux_y, float aux_z)***: esta función permite colocar el efector final activo en una posición y orientación determinada en el espacio por los parámetros *xf, yf, zf, phi/roll, theta/pitch, psi/yaw*, medida desde el marco *frame*. La velocidad del movimiento está determinada por *speed*. La convención de orientación (ángulos de Euler o RPY) y el tipo de interpolación (*joint, linear* o ***circular***) que se usará para colocar el efector final se definen desde la función *config()*. Sino se especifican los parámetros de orientación, el sistema llegará a la posición ingresada con una orientación libre. Los parámetros *aux_x, aux_y* y *aux_z* solo son usados cuando la interpolación es *circular* y determinan el punto intermedio necesario para formar el arco de circunferencia.
- ***set_frame_position(int i)***: esta función permite guardar el punto en el espacio donde está el efector final de robot como un nuevo marco coordenado *i*-ésimo. El valor 0 del marco coordenado está reservado para el inercial del robot por lo que no puede

ser modificado, es decir $i > 0$. Por *default* el marco coordenado tiene la misma orientación que el marco inercial y solo pueden guardarse hasta 15 marcos coordenados.

- ***set_frame_orientation(int i, vector3 Dir_x, vector3 Dir_y, vector3 Dir_z)***: esta función permite cambiar las direcciones de los ejes x,y,z del marco i-ésimo. Se deben seleccionar correctamente los valores de estos vectores de tal forma que se forme un sistema derecho, de lo contrario la función mandará una advertencia y no se actualizará la orientación.
- ***get_joint_angle(int joint)***: esta función permite obtener la posición que tiene la junta especificada en *joint* medida en grados.
- ***get_position(int frame)***: esta función permite obtener el valor x,y,z de la ubicación actual del efector final respecto al marco *frame*.
- ***get_orientation(int frame)***: esta función permite obtener la matriz de orientación del efector final respecto al marco *frame* usando la convención RPY o ángulos de Euler según esté especificado en *config()*.
- ***get_temperature()***: permite obtener el valor en °C del sensor de temperatura.
- ***get_humidity()***: permite obtener el valor de humedad del sensor de humedad.
- ***get_current(int joint)***: permite obtener el valor de corriente que está siendo suministrada a la junta *joint*.
- ***get_active_joints()***: permite obtener el número de grados de libertad conectados en la configuración del robot actual.
- ***get_PID_param(int joint)***: permite obtener las ganancias del controlador usado para la junta *joint*.
- ***move_X(float x, int frame, float speed)***: permite desplazar el efecto final *x* milímetros en la dirección X del marco coordenado especificado en *frame*.
- ***move_Y(float y, int frame, float speed)***: permite desplazar el efecto final *y* milímetros en la dirección Y del marco coordenado *frame*.
- ***move_Z(float z, int frame, float speed)***: permite desplazar el efecto final *z* milímetros en la dirección Z del marco coordenado *frame*.
- ***config()***: esta función tiene las siguientes opciones para configurar dentro de su argumento:
 - *Euler/RPY*: al seleccionar uno u otro, la orientación del efector final usará la convención de ángulos de Euler o de roll,pitch y yaw.

- *joint/linnear/circular*: el seleccionar uno de estos permite hacer que las interpolaciones a los puntos sean: libres (*joint*), lineales (*linnear*) o mediante arcos de circunferencia (*circular*). Cuando se use la interpolación circular, se debe especificar un punto intermedio adicional en la función *set_position()*.
- *Tool0/1/2/.../n*: especifica el efector final que se va a usar. 0 es que no usará ninguno.
- *Security system off/on*: permite activar o desactivar el paro de emergencia que ocasiona el sistema de prevención de colisiones, por *default* siempre está activado.
- ***home()***: esta función se debe ejecutar antes de apagar el robot, permite regresarlo a su posición de home y guarda todos los datos necesarios en memoria para poder asegurar su correcta la siguiente vez que se vaya a hacer uso del mismo.

Dependiendo del efector final son las funciones que se pueden realizar con él. Por ejemplo, para la Tool 1 (*gripper*) tenemos:

- ***set_tool1(float aperture)***: establece la apertura en milímetros del *gripper*.
- ***get_tool1_apperture()***: permite obtener el valor de apertura que tiene el *gripper* en milímetros.

Funciones especiales que involucran memoria:

- ***clear_frames()***: borra toda la información de los marcos coordenados que tenía el sistema dejando solamente el inicial.
- ***clear_points()***: borra todos los puntos guardados que tenía el sistema.
- ***load_points(int i)***: carga al programa el punto *i*-ésimo guardado, si *i=0* se cargan a memoria todos los puntos guardados.
- ***load_sensor_data(string sens)***: esta función permite obtener los valores que tuvo el sensor o proceso especificado en *sens* a lo largo del último movimiento establecido. Las opciones de sensor son las siguientes:
 - ***current_i***: *i* puede ir desde 1-6 y representa el nivel de corriente suministrado al motor de la junta *i*-ésima.
 - ***joint_i***: *i* puede ir desde 1-6 y representa el ángulo de la junta *i*-ésima.
 - ***errorp_i***: *i* puede ir desde 1-6 y representa el error de posición que tuvo la junta *i*-ésima.
 - ***time***: representa el instante de tiempo en el cual se muestrearon los parámetros antes mencionados.

Funciones Modo Manual

Solo podemos acceder a estas funciones mediante botones en la HMI, que es lo que se usa para operar en el modo manual.

- ***select_frame()***: al presionar el botón podemos seleccionar respecto a que marco de referencia de los guardados en memoria vamos a realizar las demás acciones.
- ***save_point()***: al presionar el botón, se guarda el punto en el espacio en el que se encuentra el efector final respecto al marco activo. El sistema es capaz de guardar hasta 1000 puntos.
- ***set_speed()***: permite seleccionar el porcentaje de velocidad al cual se realizarán los movimientos.
- ***move_x()***: son dos botones, al presionar uno de ellos el efector final se mueve a lo largo del X del marco seleccionado en la dirección positiva o negativa según corresponda.
- ***move_y()***: son dos botones, al presionar uno de ellos el efector final se mueve a lo largo del Y del marco seleccionado en la dirección positiva o negativa según corresponda.
- ***move_z()***: son dos botones, al presionar uno de ellos el efector final se mueve a lo largo del Z del marco seleccionado en la dirección positiva o negativa según corresponda.
- ***move_joint(int i)***: son 2 botones, al presionar uno, se incrementa o disminuye, según corresponda, el ángulo de la junta *i*-ésima de acuerdo a la velocidad especificada en *set_speed()*.

Funciones Modo diagnóstico

Detallado en el UML

Funciones Modo de emergencia

El modo de emergencia se activa cuando el sistema de prevención de colisiones detecta un obstáculo o bien, cuando se presiona el botón de paro de emergencia. Una vez en este estado solo son posibles las siguientes acciones y deben efectuarse desde la HMI con botones:

- Acceso a los botones de movimiento con baja velocidad para alejarse del obstáculo y poder reiniciar el sistema
- ***save_state()***: permite guardar todos los estados del sistema para generar un reporte de error. Es decir, guarda la posición y orientación del efector final que tienen al momento de que se disparará el paro de emergencia, así como las variables de junta, número de juntas activas, modo de operación, y parámetros de configuración del sistema.

- ***reset()***: reinicia el sistema como si se acabara de prender