

Informe de Práctica 1

Programación de Robots

David Ávila Jiménez || Pedro Antonio Aguilar Lima

Índice

| | |
|----------------------------|---|
| Resumen..... | 3 |
| Introducción | 3 |
| Objetivo | 3 |
| Restricciones | 4 |
| Software usado..... | 4 |
| Control de Scrobot | 5 |
| Colocación de piezas..... | 5 |
| Contenido del código | 6 |

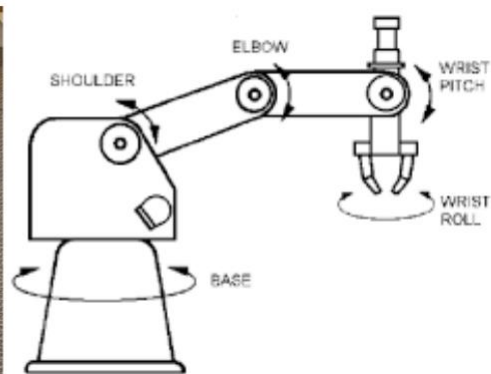
Resumen

En este informe se recoge toda la información de la práctica 1 sobre Robots Manipuladores. El objetivo de esta es aprender el uso del programa V-REP PRO además de MATLAB mediante el robot manipulador Scorbot en modo simulación, con el software, comandos y librerías proporcionadas en el Campus Virtual. Además, para comprobar que se ha entendido el manejo de cada software se debe construir una torre formada por tres piezas.

Introducción

Vamos a programar un robot manipulador, para esto descargamos la escena ("main_scene.ttt") dada en el campus virtual y que ha sido modificada para realizar la práctica, para poder escribir el código se ha creado un script en el software MATLAB ("CodigoSimulacion.m") además en la entrega se añade un archivo donde se guardan las posiciones iniciales y finales del robot ("PosicionesIniFin.mat").

El robot usado en la práctica es Scorbot, en las siguientes imágenes se puede observar, además de ver los seis grados de libertad que posee.



Objetivo

Hay que construir una pequeña torre de piezas mediante el robot Scorbot en modo simulación usando V-REP, para esto debemos usar tres piezas de la escena dando como resultado algo parecido a la siguiente imagen (dos piezas verticales con una pieza horizontal encima de estas dos).



Restricciones

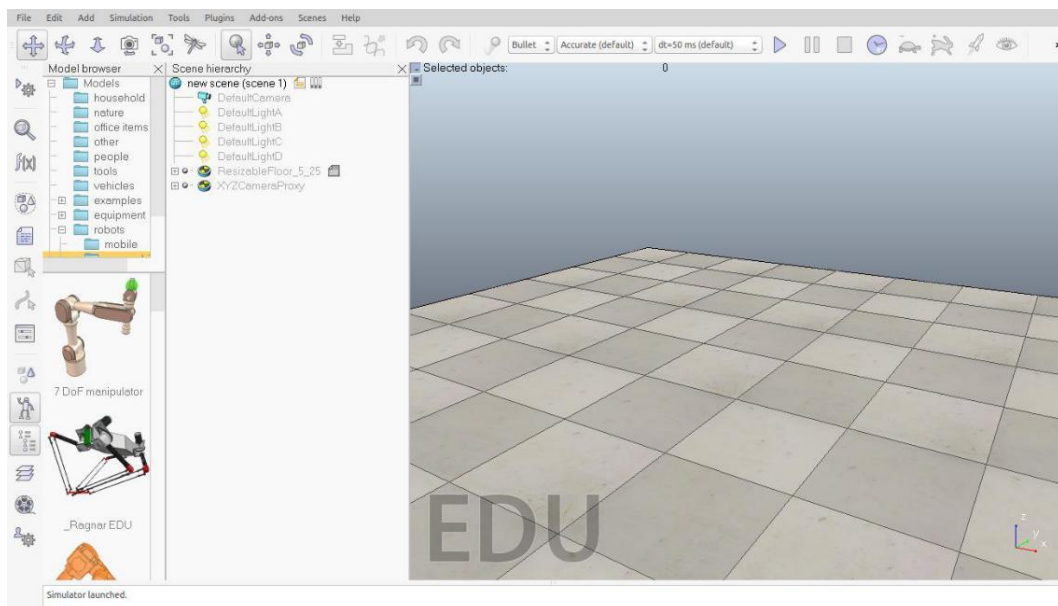
El programa hace uso únicamente de dos posiciones una inicial y otra final, ambas definidas previamente mediante la pistola simulada.

El robot en su simulación ira calculando las posiciones de las piezas nuevas mediante aproximación a las dos anteriormente definidas, de esta forma se llega de forma segura, de la misma manera vamos a calcular las posiciones para completar correctamente la torre. Se pueden emplear las rutinas `changePosXYZ`, `changePosRoll`, `changePosPitch`, con esto conseguimos cambiar la posición del robot y de su pinza.

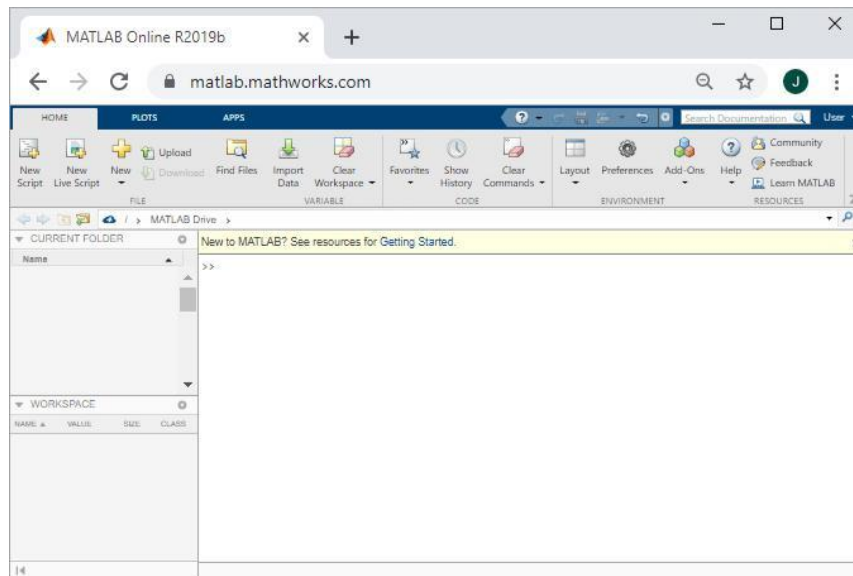
Por último, en ningún momento podemos superar el 20% de velocidad, esto debe definirse al comienzo del programa mediante la rutina `changeSpeed`. Como recomendación, antes de empezar la tarea, el robot debe comenzar desde home, para evitar de esta forma errores de aproximación a las piezas.

Software usado

VREP- PRO: Simulador robótico desarrollado por Coppelia Robotics con distribución gratuita, se trata de un software ideal para la simulación de robots manipuladores móviles en tiempo real, contiene un entorno de desarrollo integrado basada en una arquitectura de control distribuida, es decir, gracias a esto podemos controlar individualmente cada objeto además de poder controlarlos mediante diferentes formas de programación.

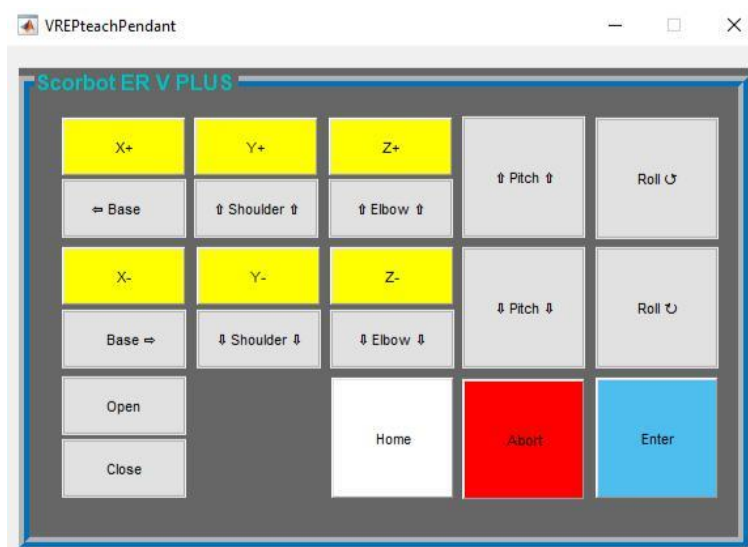


MATLAB: Abreviatura de MATrix LABoratory (laboratorio de matrices) se trata de un programa computacional que ejecuta gran variedad de operaciones y tareas matemáticas. Mediante complementos podemos realizar diferentes funciones como el conectar con el programa V-REP.



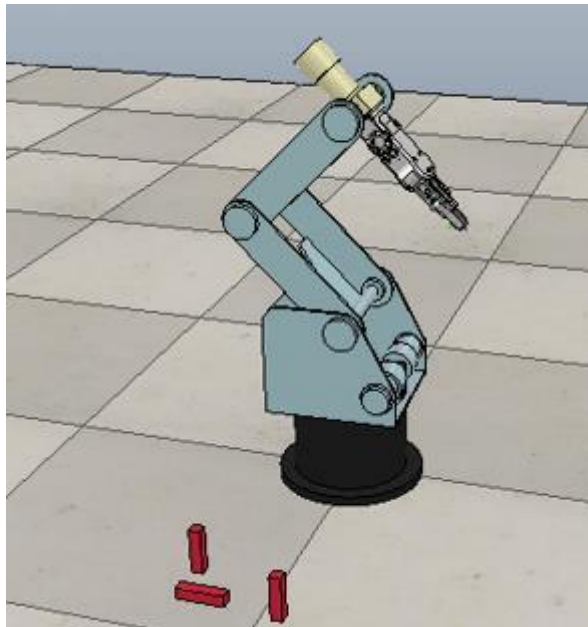
Control de Scorbot

Esto es posible mediante código, el cual se explica y se muestra más adelante, además de esto tenemos una opción gráfica, para esto, debemos ejecutar el comando `pendant` y obtendremos la siguiente pantalla para manejar el robot. En esta pantalla podremos manejar el robot a través del espacio y sus articulaciones.



Colocación de piezas

Situamos tres piezas dentro del rango de funcionamiento del robot, evitando así posibles complicaciones o rotura de este que se producen al llevar al robot a un estado inalcanzable o de singularidad. Colocaremos dos piezas tumbadas y una un poco rotada para utilizar la rutina de cambio de roll. También, una de ellas se colocará tumbada para usar de esta forma el cambio de posición del Pitch y el Roll del robot.



Contenido del código

```
%Borramos la ventana de ejecución de comandos
clear;
%Cerramos aquellos procesos abiertos anteriormente
close all;
%Llamamos al robot Scorbot desde el programa MATLAB para que esto sea
posible, antes debemos de cargar unas librerías dadas en el campus virtual
en este programa. s=Scorbot(Scorbot.MODEVREP)
%Enviamos al robot a la posición inicial para evitar complicaciones de esta
forma
s.home;
%Cargamos las posiciones si no hay nada guardado anteriormente
teach = 0;
if (~teach)
    load('PosicionesIniFin.mat');
    %La velocidad del robot debe ser igual o
    %inferior a 20
    s.changeSpeed(20);
    %Mandamos el robot a su posición inicial
    s.move(posInicial,1);
    %Aproximación a la primera pieza, esta se calcula sumando los valores de
    la pieza a la posición inicial
    s.move(s.changePosXYZ(posInicial,[posInicial.xyz(1)+296
    posInicial.xyz(2)-533 posInicial.xyz(3)-28]),1);
    %Abrimos pinza
    s.changeGripper(1);

    %Acercamos la pinza para coger mejor la pieza
    s.move(s.changePosXYZ(posInicial,[posInicial.xyz(1)+296
    posInicial.xyz(2)-533 posInicial.xyz(3)-928]),1);
    %Cerramos Pinza y movemos a posiciones anteriores
    s.changeGripper(0);
    s.move(s.changePosXYZ(posInicial,[posInicial.xyz(1)+296
    posInicial.xyz(2)-533 posInicial.xyz(3)-28]),1);
    s.move(s.changePosXYZ(posInicial,[posInicial.xyz(1)+296
    posInicial.xyz(2)-533 posInicial.xyz(3)-928]),1);
    s.move(posInicial,1);
    %Ahora llevamos la pieza uno a la posicion final
```

```

s.move(posFinal,1);
%Movemos a la posicion de la pieza y la soltamos
s.move(s.changePosXYZ(posFinal,[posFinal.xyz(1)-407 posFinal.xyz(2)+2
posFinal.xyz(3)-926]),1);
s.changeGripper(1);
s.move(posFinal,1);
s.changeGripper(0);
%Aproximacion a la segunda pieza
s.move(posInicial,1);
%Nos posicionamos encima y abrimos la pinza
s.move(s.changePosXYZ(s.changePosRoll(s.changePosPitch(posInicial,-
400),160),[posInicial.xyz(1)-2008 posInicial.xyz(2)+421
posInicial.xyz(3)-51]),1);
%Abrimos la pinza
s.changeGripper(1);
%Bajamos la pinza y cogemos la pieza cerrando la pinza
s.move(s.changePosXYZ(s.changePosRoll(s.changePosPitch(posInicial,-
400),160),[posInicial.xyz(1)-2008 posInicial.xyz(2)+421
posInicial.xyz(3)-851]),1);
s.changeGripper(0);
%Llevamos a posicion inicial y despues final
s.move(s.changePosXYZ(posInicial,[posInicial.xyz(1)+296
posInicial.xyz(2)-533 posInicial.xyz(3)-28]),1);
s.move(posInicial,1);
s.move(posFinal,1);
%Vamos a posicion de aproximacion y despues a la posicion donde lo
Soltaremos abriendo la pinza
s.move(s.changePosXYZ(posFinal,[posFinal.xyz(1)+271 posFinal.xyz(2)+2
posFinal.xyz(3)]),1);
s.move(s.changePosXYZ(posFinal,[posFinal.xyz(1)+271 posFinal.xyz(2)+2
posFinal.xyz(3)-1026]),1);
s.changeGripper(1);
%volvemos a posición original
s.move(s.changePosXYZ(posFinal,[posFinal.xyz(1)+271 posFinal.xyz(2)+2
posFinal.xyz(3)]),1);
s.changeGripper(0);
s.move(posFinal,1);
s.move(posInicial,1);
%Aproximacion a la tercera pieza
s.move(posInicial,1);
%Nos posicionamos encima y abrimos la pinza
s.move(s.changePosXYZ(s.changePosRoll(s.changePosPitch(posInicial,-
400),1040),[posInicial.xyz(1)-1093 posInicial.xyz(2)-728
posInicial.xyz(3)-816]),1);
s.changeGripper(1);
%Bajamos la pinza y cogemos la pieza
s.move(s.changePosXYZ(s.changePosRoll(s.changePosPitch(posInicial,-
400),1040),[posInicial.xyz(1)-1193 posInicial.xyz(2)-728
posInicial.xyz(3)-1116]),1); s.changeGripper(0);

%Vamos a posicion de aproximacion y despues a la posicion donde lo
%Soltaremos
s.move(s.changePosXYZ(posFinal,[posFinal.xyz(1)-75 posFinal.xyz(2)+2
posFinal.xyz(3)-426]),1);
s.changeGripper(1);
%volvemos a posicion original
s.move(posFinal,1);
s.changeGripper(0);
%Regresamos a la posición inicial
s.home;

```

```

else
    fprintf('----> Teach the robot where is the location for picking items
    and press Enter to
    finish.\n\n');
    %     Mover con pendant, y después pulsar Enter en la botonera para
    %     almacenar la posición.
    %     Tras guardar cada posición, salvar el fichero de posiciones
    %     ;Sólo las posiciones, no el objeto robot!
    posInicial = s.pendant(); % situar pieza para asegurarse de cogerla.
    Guardar x,y pieza
    posFinal = s.pendant(); % moverse en...
end

fprintf('Press any key to start picking-and-placing.\n'); pause;

```