Fundamental on Robotics: Pallet assembly guide using Staubli TX60

Guillaume Lemaître - Miroslav Radojević
Heriot-Watt University, Universitat de Girona, Université de Bourgogne
g.lemaitre58@gmail.com - miroslav.radojevic@gmail.com

I. Introduction

The aim of this laboratory is to be familiar with the robot Staubli TX60. The activity asked is a palletization. We will present first the specification of the robot. Then, we will introduce the positions recording during the manipulation. Finally, we will present the scheme and the program implemented to carry the palletization.

II. Specification of the robot

Figure 1 presents all degrees of freedom of the robot. The Mitsubishi RV-M1 is composed of five degrees of freedom. All degree of freedom are rotational joints.

Referring to the figure 1, the degrees of freedom are:

- J_1: Rotational joint of $\pm 180°$.
- J_2: Rotational joint of $\pm 127.5°$.
- J_3: Rotational joint of $\pm 142.5°$.
- J_4: Rotational joint of $\pm 270°$.
- J_5: Rotational joint of $+133.5, -122.5°$.
- J_6: Rotational joint of $\pm 270°$.

III. Acquisition position

In order to perform the palletization, several positions were saved:

A. World positions

- *pospalappro*:
  - x = 309.5239, y = 0.6892, z = -157.2786
  - rot_x = -0.7818, rot_y = 178.8200, rot_z = -1.0460
- *posgrab*:
  - x = 292.1072, y = -34.9257, z = -224.1558
  - rot_x = -0.7873, rot_y = 178.8221, rot_z = -1.0488
- *outpos*:
  - x = 462.4474, y = -262.3270, z = -161.7083
  - rot_x = 4.2406, rot_y = -169.2928, rot_z = 17.4632

These positions are corresponding to:

- *pospalappro*: Approach position of the pallet.
- *posgrab*: Position to grab the object. This position is initially on the object at the top-right corner of the pallet.
- *outpos*: Position of the garbage to unfill the pallet.

B. Joint positions

- *jposini*:
  - j_1 = -4.4282, j_2 = 6.5870, j_3 = 88.7293
  - j_4 = -1.7652, j_5 = 87.5666, j_6 = -0.1095

The position *jposini* is the HOME position of the robot. The different values of the position are the different value of the different joint of the robot.

IV. Scheme and program

A. Description of the task

So to perform the manipulation, these different steps have to be executed:

- Step 1: Initialization
  - Enable and clean MCP screen and switch on the robot.
  - Define the speed (30 over 100).
  - Define the offsets between different objects.
  - Define the number of row and column of the pallet.
- Step 2: Move to the HOME position
- Step 3: Set up the speed (10 over 100)
- Step 4: Approach the pallet
- Step 5: Approach the object
- Step 6: Set up the speed (2 over 100)

- Step 7: Go and grab the object
- Step 8: Set up the speed (10 over 100)
- Step 9: Go to the previous approach position
- Step 10: Go to the approach pallet position
- Step 11: Approach the garbage
- Step 12: Go to the garbage and let the object
- Step 13: Go to the approach garbage position
- Step 14: Update the next position where the robot will grab the object and go to the step 3

B. Scheme

The previous step describe in the previous section are shown on the figure 2.

C. Program and description

```
//Routine to initialize the robot
//sets MCP screen as a user interface
userPage()
//cleans the MCP screen
cls()
//Set_robot_power()
//sets the robot power
enablePower()
//avoids robot to stop between consecutive
    movements
autoConnectMove(true)
//message on the screen (indicates the
   beginning of the application)
putln("program starts")
//Move_Robot(jpos_Inicial)
```

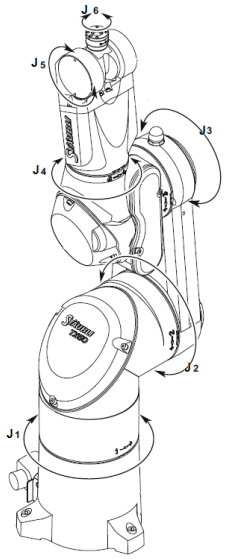

Figure 1. Presentation of degree of freedom

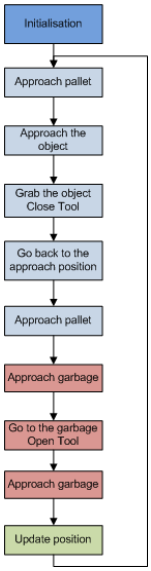

Figure 2. Scheme of the program

```
//////////////////
///Definition variable
///Speed variable
mNomSpeed.vel=30
/// number of pieces
nPieces=8
///Define offset
offsetX=30
offsetY=25
//Define the number of row and column of
    the pallet
row=2
column=4

//////////
//Start program
//joint type robot movement
movej(jposini,teina,mNomSpeed)
//waits current movement to end before
    executing any other instruction
waitEndMove()
```

In the initialization phase, we will initialize the robot. Thus, we will enable and clean the MCP screen and switch on the robot. Then, the speed of the robot is fixed at 30 over 100. The number of column and row of the pallet is defined as the offset in X and Y. Then, the robot will move to the HOME position.

```
mNomSpeed.vel=10
////////
//Approach the pallet
////////
movel(pospalappro,teina,mNomSpeed)
```

```
//waits current movement to end before
    executing any other instruction
waitEndMove()

////////
//Go to grab the object
//////

//Set up the grab position as approach
    position
posgrab.trsf.z=posgrab.trsf.z+50
movel(posgrab,teina,mNomSpeed)
//waits current movement to end before
    executing any other instruction
waitEndMove()

//Set up the speed slower
mNomSpeed.vel=2

//Go to grab the object
posgrab.trsf.z=posgrab.trsf.z-50
movel(posgrab,teina,mNomSpeed)

//Grab the object
close(teina)

//Set up the speed faster
mNomSpeed.vel=10

//Go to the approach position
posgrab.trsf.z=posgrab.trsf.z+50
movel(posgrab,teina,mNomSpeed)
//waits current movement to end before
    executing any other instruction
waitEndMove()

//Reset the grab position
posgrab.trsf.z=posgrab.trsf.z-50

//////
//Go to the approach pallet
/////

//Go to the approach position of the
    pallet
movel(pospalappro,teina,mNomSpeed)
//waits current movement to end before
    executing any other instruction
waitEndMove()
```

In order to go accurately to grab the object, the robot will approach first the object and only after go to grab it. Hence, the approach position is defined as fifty millimeters above the object. When the robot reaches this position, it will go slow to grab the object. Then, it will come back to the approach object position. Then the robot moves to the approach pallet position.

```
////////
//Go to the garbage and let the object
//////

//Approach the garbage
outpos.trsf.z=outpos.trsf.z+150
movel(outpos,teina,mNomSpeed)
//waits current movement to end before
    executing any other instruction
waitEndMove()
//Go to the garbage
outpos.trsf.z=outpos.trsf.z-150
movel(outpos,teina,mNomSpeed)
//waits current movement to end before
    executing any other instruction
waitEndMove()

//Open the tool
open(teina)
//Go to the approach position
outpos.trsf.z=outpos.trsf.z+150
movel(outpos,teina,mNomSpeed)
//waits current movement to end before
    executing any other instruction
waitEndMove()
//Reset the position of the garbage
outpos.trsf.z=outpos.trsf.z-150
```

As for the object, the robot approaches the garbage. Then go to let the object on the garbage. The robot will come back to the approach garbage position.

```
//////
//Update grabbing position
//////
//Update the next position of the next
    object on the pallet
//Go to the next column
posgrab.trsf.y=posgrab.trsf.y+offsetY
endFor
//Update the next position of the next
    object on the pallet
//Go to the next row
posgrab.trsf.x=posgrab.trsf.x+offsetX
posgrab.trsf.y=posgrab.trsf.y-(offsetY*
    column)
endFor
```

Before to restart the routine to grab an object, the next position of the pallet where the robot will grab the object is computed.

V. Conclusion

In this paper, we presented an example of palletization using the Staubli TX60. First, we presented the specifications of this robot. Then, we introduce the position

stored in order to perform the palletization. Finally, we described the program allowing the palletization.

APPENDIX A
CODE

```
begin
  //Routine to initialize the robot
  //sets MCP screen as a user interface
  userPage()
  //cleans the MCP screen
  cls()
  //Set_robot_power()
  //sets the robot power
  enablePower()
  //avoids robot to stop between consecutive movements
  autoConnectMove(true)
  //message on the screen (indicates the beginning of the application)
  putln("program starts")
  //Move_Robot(jpos_Inicial)
  //////////////////////////////////////////////////
  ///Definition variable
  ///Speed variable
  mNomSpeed.vel=30
  /// number of pieces
  nPieces=8
  ///Define offset
  offsetX=30
  offsetY=25
  //Define the number of roz and column of the pallet
  row=2
  column=4
  //////////////////////////////////////////////////
  // ---------------- Start program ------------------//
  //joint type robot movement
  movej(jposini,teina,mNomSpeed)
  //waits current movement to end before executing any other instruction
  waitEndMove()
  ///Routine to grab each object
  for i=1 to row
    for j=1 to column
      mNomSpeed.vel=10
      //////////////////////////
      //Approach the pallet
      //////////////////////////
      movel(pospalappro,teina,mNomSpeed)
      //waits current movement to end before executing any other instruction
                waitEndMove()

                //////////////////////////
      //Go to grab the object
      //////////////////////////

      //Set up the grab position as approach position
      posgrab.trsf.z=posgrab.trsf.z+50
      movel(posgrab,teina,mNomSpeed)
      //waits current movement to end before executing any other instruction
                waitEndMove()

                //Set up the speed slower
      mNomSpeed.vel=2

      //Go to grab the object
      posgrab.trsf.z=posgrab.trsf.z-50
      movel(posgrab,teina,mNomSpeed)
```

```
//Grab the object
close(teina)

//Set up the speed faster
mNomSpeed.vel=10

//Go to the approach position
posgrab.trsf.z=posgrab.trsf.z+50
movel(posgrab,teina,mNomSpeed)
//waits current movement to end before executing any other instruction
          waitEndMove()

          //Reset the grab position
posgrab.trsf.z=posgrab.trsf.z-50

//////////////////////////////////
//Go to the approach pallet
//////////////////////////////////

//Go to the approach position of the pallet
movel(pospalappro,teina,mNomSpeed)
//waits current movement to end before executing any other instruction
          waitEndMove()

          //////////////////////////////////
          //Go to the garbage and let the object
          //////////////////////////////////

//Approach the garbage
outpos.trsf.z=outpos.trsf.z+150
movel(outpos,teina,mNomSpeed)
//waits current movement to end before executing any other instruction
          waitEndMove()
          //Go to the garbage
outpos.trsf.z=outpos.trsf.z-150
movel(outpos,teina,mNomSpeed)
//waits current movement to end before executing any other instruction
          waitEndMove()

          //Open the tool
open(teina)
//Go to the approach position
outpos.trsf.z=outpos.trsf.z+150
movel(outpos,teina,mNomSpeed)
//waits current movement to end before executing any other instruction
          waitEndMove()
          //Reset the position of the garbage
outpos.trsf.z=outpos.trsf.z-150

//////////////////////////////////
//Approach the pallet
                  //////////////////////////////////
movel(pospalappro,teina,mNomSpeed)


//////////////////////////////
//Update grabbing position
//////////////////////////////
//Update the next position of the next object on the pallet
//Go to the next column
```

```
      posgrab.trsf.y=posgrab.trsf.y+offsetY
    endFor
    //Update the next position of the next object on the pallet
    //Go to the next row
    posgrab.trsf.x=posgrab.trsf.x+offsetX
    posgrab.trsf.y=posgrab.trsf.y-(offsetY*column)
  endFor
  //---------------- End program ------------------------//
end
```