CONTENTS

PART# 1

In this we show how the first part can be done using botrhe Te or through Python.

PART1 IN TRE E

In Tree approach w,e can make use of sliders for the six joints angles of rob, cohtange the x,y,z and euler angels ins ection ' Tool frame wreifetrehn cre fersapmeec't itno fiogru rsei.m1p ly by using 'Alt and drag and dropto bring the robot to the desried position with respect to the frames 4,7 and 5.

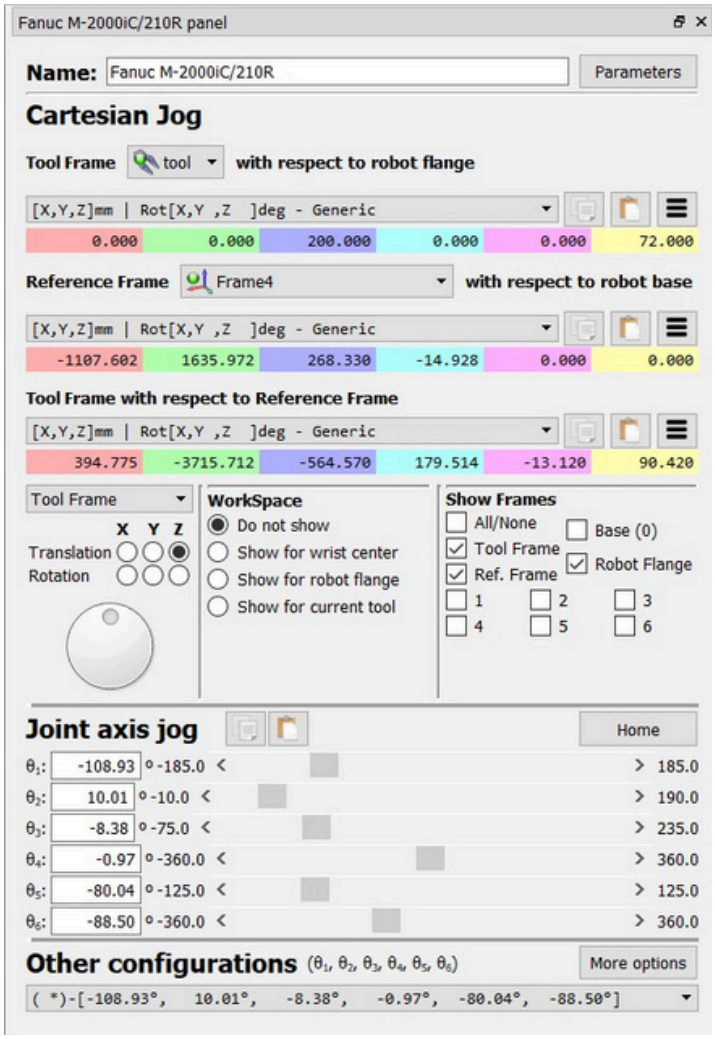


Figure1 Accessible features of robo. t

Then by clicking  **tab we cwaen add sxi targets to the frames 4,7 and 5 respectively as shown in figure.2 for approaching the carts and picking the pieces steps into the tre e.**

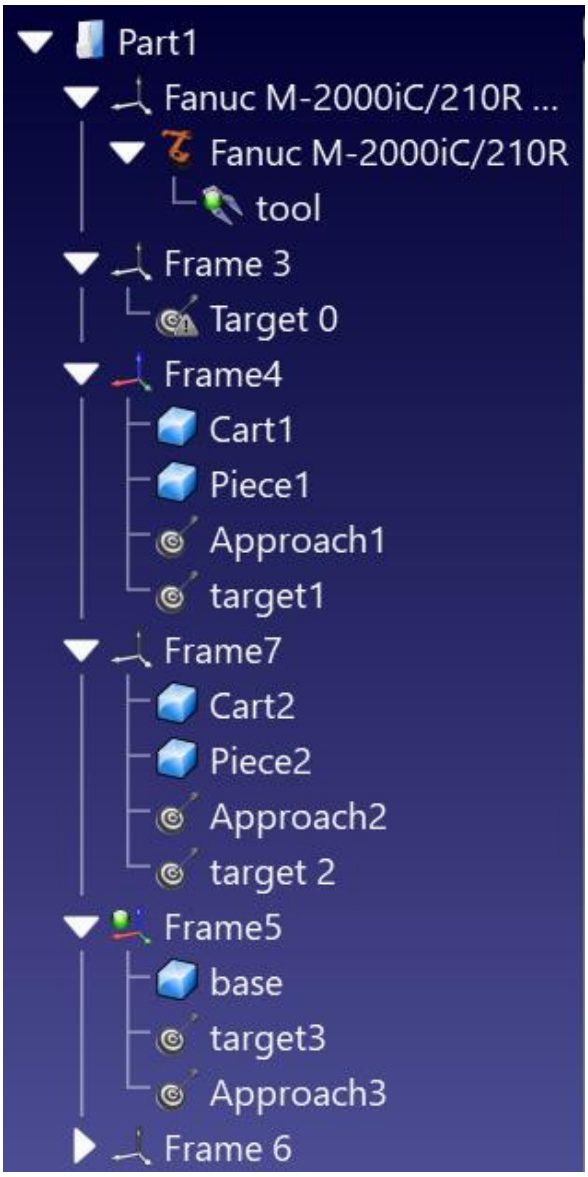
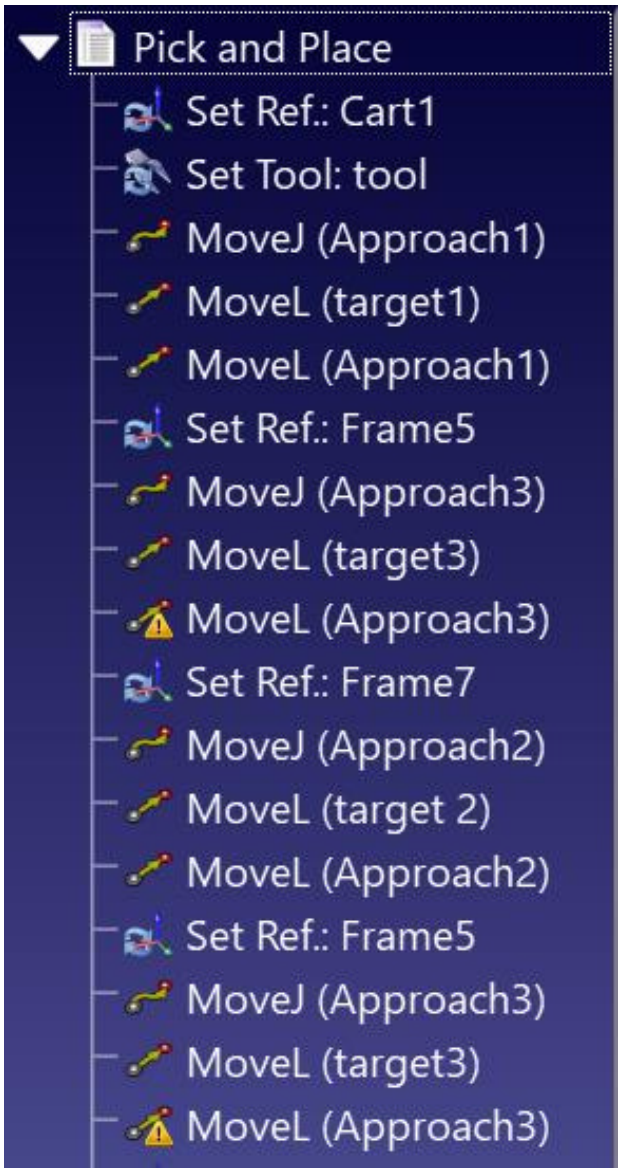


Figure2 Tree of the stationc ontents.



F i g ure3 Ende-Effector Movements.

To move the robo tbetween the carts and conveyer we can click the tab  **and for movement between 'Aphp'r aon'adc Pick'i Tanrggets we can s e l t aebc.t T o operate in each frame, we need to activate that frame in advance as shown in figure .3**

PART 1 IN PYTHO N

The other way is to make the movements, target, object and framest hrough python language. For movements inside frame 4and between frames 4 and 5t he codes are asf ollowing:

```
robot.setPoseFrame(frame4 )

robot.MoveJ(Approach1)

robot.MoveL(target1)

robot.MoveL(Approach1 )

robot.setPoseFrame(frame5 )

robot.MoveJ(Approach3 )

robot.MoveL(target3)

robot.MoveL(Approach3 )
```

To add objects, targets and frames to the tree through the python programing, a typical set of code s is presented heretogenerate frame 5, conveyer and holder frame .

```
frame5=RDK.AddFrame('frame5 ')

frame5.setPose(trans-l(812.638,1005.325,0)*rotz(45*pi/180)*roty(0*pi/180)*rotx(0*pi/180) )

convoyer = RDK.AddFile(r'\CU:sers\fredy\Desktop\carts\convoyer.stl',frame5)

holder = RDK.AddFile(r'\CU:sers\fredy\Desktop\carts\holder.stl',frame5)

convoyer.setPose(transl(596.9,1752.6,0)*rotz(90*pi/180)*roty(0*pi/180)*rotx(0*pi/180 ))

holder.setPose(trans-l(848.180,1326.987-1,076.296)*rotz(90*pi/180)*roty(0*pi/180)*rotx(0*pi/180))

Approach3 = RDK.AddTarget('Approach3',fram e5)

Approach3=Approach3.setPose(transl(576.577,814.278,1200)*r-otz(
89.591*pi/180)*roty(1.808*pi/180)*rotx(179.622*pi/180))
target3=RDK.AddTarget('target3',frame 5)

target3=target3.setPose(transl(576.577,814.217085,0)*rotz(-89.591*pi/180)*roty(1.808*pi/180)*rotx(179.622*pi/180))
```

we use‘ RDK.AddFram’e to add a new frame,‘R DK.AddTarge’ to add a new target and‘R DK.AddFil’e to bring the convoyer and holder to the station. To move the objects or targets to the deseidr positions we cana lso use‘ .setPose (transl(x,y,z)*rotz(rz*pi/180)*roty(ry*pi/180)*rotx(rx*pi/180))’.

PART2

For this part the number of ipeces available on each cart and theidr istribution are generic .A robot program was designed that prompts the user to input an excel file in .csv fmorat.

INPUT TABLE FORM APTART2

The table can have any desirendu mber of rows where: each row of the first column has the number of objects in the first column and the corresponding chart to seleinc tt he second column .

Table 1 Table for Part 2.

NUMBER OF OBJEC TS	CHART SELECTIO N
10	Chart x
5	Chart y
4	Chart x
31	Chart y
2	Chart x

PYTHONP ROGRAM PART 2

The program was written in python. The main steps are discussed here in b rief:

1. **In the first step the program agthers the robot and other items from the station.**

```
# Gather required items from the station
robot = RDK.Item('Fanuc M-2000iC/210R')
a1 = RDK.Item(' Approach1 ')
a2 = RDK.Item(' Approach2 ')
a3 = RDK.Item(' Approach3 ')
t1 = RDK.Item(' target1 ')
t2 = RDK.Item(' Target 2 ')
t3 = RDK.Item('Target')
frame1 = RDK.Item('Frame4')
frame2 = RDK.Item('Frame7')
frame3 = RDK.Item('Frame5')
```

2. **In second stepit prompts the user for an input u"s fuincgt tihoe "nm foborx an excel tabl .csv format and imports the file from directory usin"g impor. t csv"**

```
#Prompts the user to input(type the name)a CSV file in proper directory
while(1)  :
 import csv
 task = []
 chartxy = []
 prod = mbox ('Enter the name of csvfile in directory for input Table' , entry =True)
 prod = str(prod)
 with open(prod) as f:
 #e.g with open('Book1.csv') as f:
    reader = csv.reader(f)
```

3. **Then the program reads the first column of each row as the number of objects it needs to pick and from second column the corresponding c a(rt''cxha'r't i s ''chart1'' or ''chart y it needs to pick from .**

```
   reader = csv.reader(f)
   for row in reader:
    #Reads number of tasks(objects in each cart) from 1st column of Excel Table
    task = row[0]
    #reads from 2nd column the corres
    chartxy = row[1]
```

4. **Then the program starts to pick and place from the chart to the conveyor. The full list of movementsi s in python program file.**

```
    chartx = "chartx"
    charty = "charty"
    if chartxy == chartx :
     print('if loopchartx')
     for x in range(tasknum) :
      print('for loop chart x task',tasknum )
      robot.setPoseFrame(frame1)
      robot.MoveJ(a1)
      robot.MoveL(t1)
      robot.MoveL(a1)
      robot.setPoseFrame(frame3)
      robot.MoveJ(a3)
      robot.MoveL(t3)
      robot.MoveL(a3)
    elif chartxy == charty :
     print('else loop chart y')
     for y in range(tasknum) :
      print('for loop chart y task',tasknum )
      robot.setPoseFrame(frame2)
      robot.MoveJ(a2)
      robot.MoveL(t2)
      robot.MoveL(a2)
      robot.setPoseFrame(frame3)
      robot.MoveJ(a3)
      robot.MoveL(t3)
      robot.MoveL(a3)
```

5. **After finishing all its tasksf or all the rows in the table,t he program goes back to the start of the loop and asks for another table inpu itf the while loop is included or ends if the while looips removed.**

PART 3

For this part the refence frames correpsonding to the chart and conveyor are the parameters of the program. A robot program was desgined considering that one cart and one conyvoer could be in any location based on each assigend task every-time.

STATION SETUFPO R PAR T3

Considering ‘n1’ and ‘n2’ to be the number of charts and number of con number of tasks ‘n’. The n tasks can be performed using one cart for n conveyor for n2 configurations.

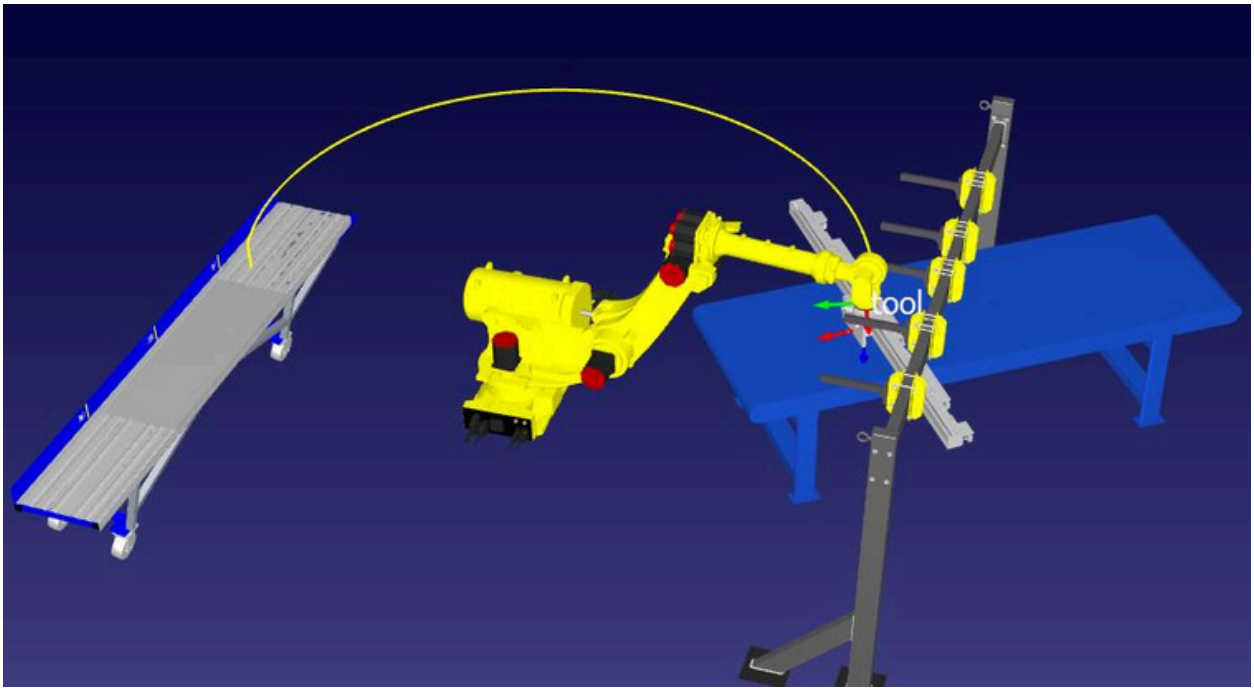


Figure4 Station setup using one cart and one convey. or

INPUT TABL FEOR PART 3

A table in .csv format was used to make a list of inputs to perform the desired ta sks.

The table can have annyu mber of desired rows where each row corresponds to the following columns in order:

- **¶ The first column has the number of objects to pick from for the corresponding configuration of chart and conveyor in that row .**
- **¶ The 2nd to 7th column has (Px1,Py1,Pz1,Rx1,Ry1,Rz1) Pose values for the Cart .**
- **¶ The 8th to 13th column has (Px2,Py2,Pz2,Rx2,Ry2,Rz2) Pose values for the Con veyor.**

Table 2 Table for Part 3.

No.Task	Px1	Py1	Pz1	Rx1	Ry1	Rz1		Px2	Py2	Pz2	Rx2	Ry2	Rz2	
5	1940	-373	788	0	0	-30		-1011	1004	0	0	0	20.24	
3	1800	-373	700	0	0	-20		-1011	1004	0	0	0	20.5	
2	1940	-373	788	0	0	-40		-1011	1204	0	0	0	20.24	

PYTHON PROGRAM PA3R T

The main steps for the program are discussed here in br ief:

1. **In the first step the program gathers robot and oth eitrems from the station.**

```
# Gather required items from the station
robot = RDK.Item('Fanuc M-2000iC/210R')
a1 = RDK.Item(' Approach1 ')

a3 = RDK.Item(' Approach3 ')
t1 = RDK.Item(' target1 ')

t3 = RDK.Item('Target')
frame1 = RDK.Item('Frame4') #cart

frame3 = RDK.Item('Frame5') #conveyor
```

2. **Then using it creates a prompt for user input for the table file in .csv format discussed be fore:**

```
#Ask For Input table of part3 table in .csv format

while(1)  :
 import csv
 task = []
 chartxy = []
 prod = mbox ('Enter the name of csvfile in directory for input Table' , entry =True)
 prod = str(prod)
 with open(prod) as f:
 #with open('Book1.csv') as f:
    reader = csv.reader(f)
```

3. **Then for Each row of the table it reads the corresponding data as discussed be fore:**

```
#Chart Data Read
XYZABC2 = XYZABC
XYZABC2[0] = float (row[1])
XYZABC2[1] = float (row[2])
XYZABC2 [2] = float (row[3])
XYZABC2 [3] = float (row[4])
XYZABC2 [4] = float (row[5])
XYZABC2 [5] = float (row[6])
```

```
#Conveyor data Read
convpose=convpose1

convpose[0] = float (row[7])
convpose[1] = float (row[8])
convpose[2] = float (row[9])
convpose[3] = float (row[10])
convpose[4] = float (row[11])
convpose[5] = float (row[12])
```

4. **Then it sets the new Cart and Conveyor reference frame Poses according to the data it read for the corresponding row.**

```
#set frame Locations
frame1.setPoseAbs(posemat)  #Cart
frame3.setPoseAbs(posemat2) #Conveyor
```

5. **Then according to the number of objects in first column of the ro wit ;performs the number of pick and place for this configuration of cart and conveyor based on the data it read from the first column of the table.T he complete list of movements is in the python program f ile.**

```
#starts to pick and place from cart to conveyor

tasknum = int(task)
print(tasknum)
for x in range(0,tasknum) :
        robot.setPoseFrame(frame1)
        robot.MoveJ(a1)
        robot.MoveL(t1)
        robot.MoveL(a1)
        robot.setPoseFrame(frame3)
        robot.MoveJ(a3)
        robot.MoveL(t3)
        robot.MoveL(a3)
```

6. **After completing these steps fo arll rows of the table the program goes back to the start of the loop and asks for another .csv table inp uif ta while loop is included or ends if its exclud.e d**

CONCLUSIO N

From the abovep rograms it can beo bserved that for just two carts in a fixed position setup, the 2nd program is better suitable for performing tasks, becaus eit doesn re'qutire too many inputs in the table.

But for more generict asks, the 3rd program is best suitable which rqeuires the posesf or the desired configuration of each cart andc onveyor, and the number of objects to pick in each row of a tab le.