MAE 263B
Dynamics and Robotics

Lab 4

Exercise: Palletizing Functions

Instructions to Palletizing Functions of the Denso Robot

1 Task

Make a program to get familiar with the Palletizing Functions offered by the Denso robot.

2 What you will have in the workcell

- A Denso robot
- 24 aluminum blocks (1'' x 1'' x 1'')
- A fixture

3 Introduction to Palletizing

3.1 Background

Palletizing is a function for the orderly stacking of workpieces by only teaching several representative poses instead of every single pose in the program. The assumption when palletizing is that the workpieces (the objects being stacked) are all coming from a single common location – possibly the end of the conveyor system. Depalletizing is exactly the opposite operation; the pallet is disassembled with the workpieces all being deposited in a single location – possibly the beginning of a conveyor system. Since the two operations are so similar, this document will henceforth simply refer to the function as 'palletizing' and leave it to the reader to supply the required additional context.

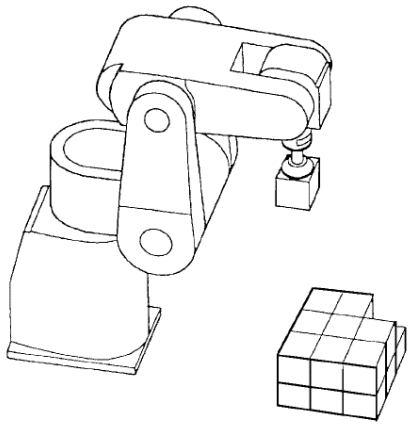


Figure 1: Palletizing Workpieces

A palletizing function consists of the following two patterns:

- Stacking Patterns: Determines the method of stacking workpieces into *rows*, *columns* and *layers*.
- Path Pattern: Determines the path along which the end effector moves when approaching or departing from the stack of workpieces.

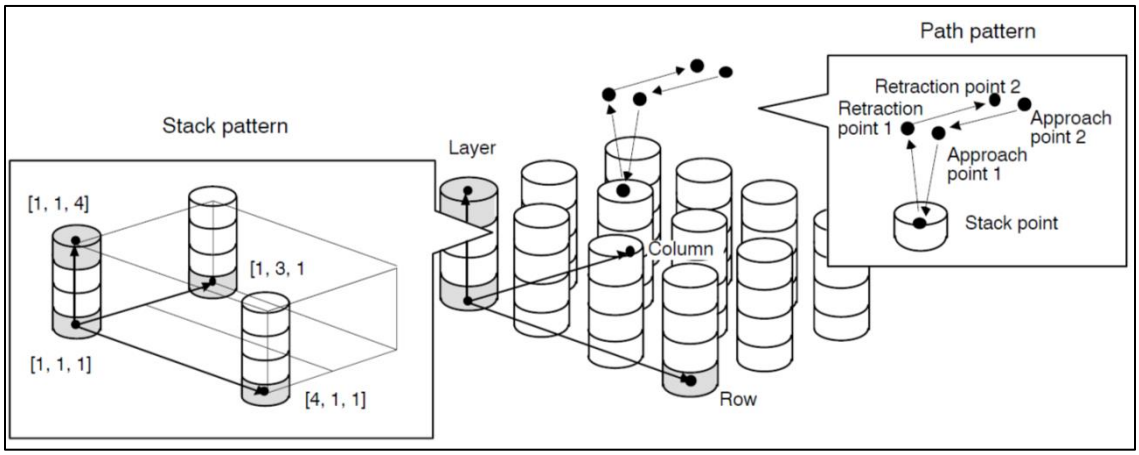
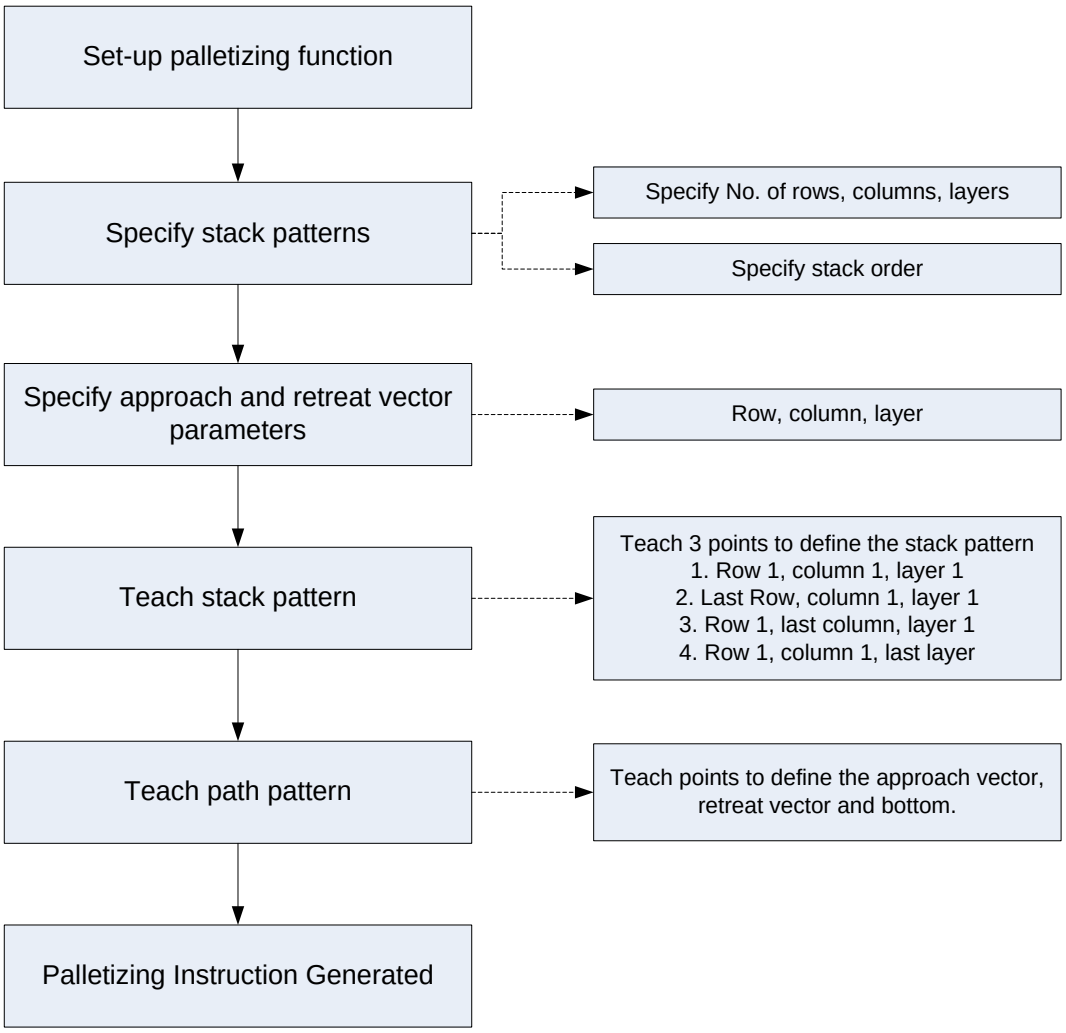


Figure 2 Stack Pattern and Path Pattern

3.2 Procedure



4 Instructions

Use the teach pendant to write an online program that completes the designated palletizing function.

4.1 Get the Robot Ready

You should follow all the instructions you have learned in labs 1 – 3, especially the safety requirements.

Register a New Program

(1) Step 1: Press [F1 Program] on the basic screen

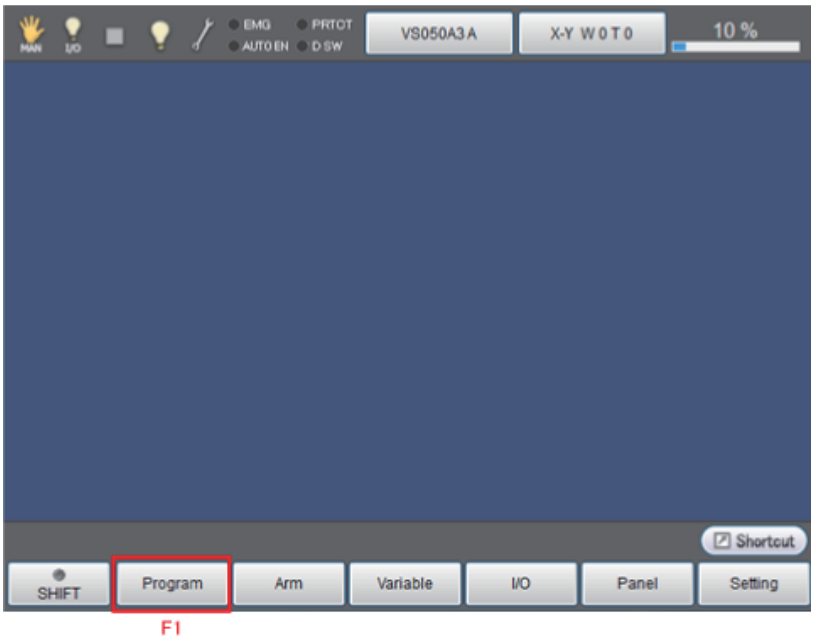


Figure 3: Basic Screen

(2) Step 2: Press [F1 New]

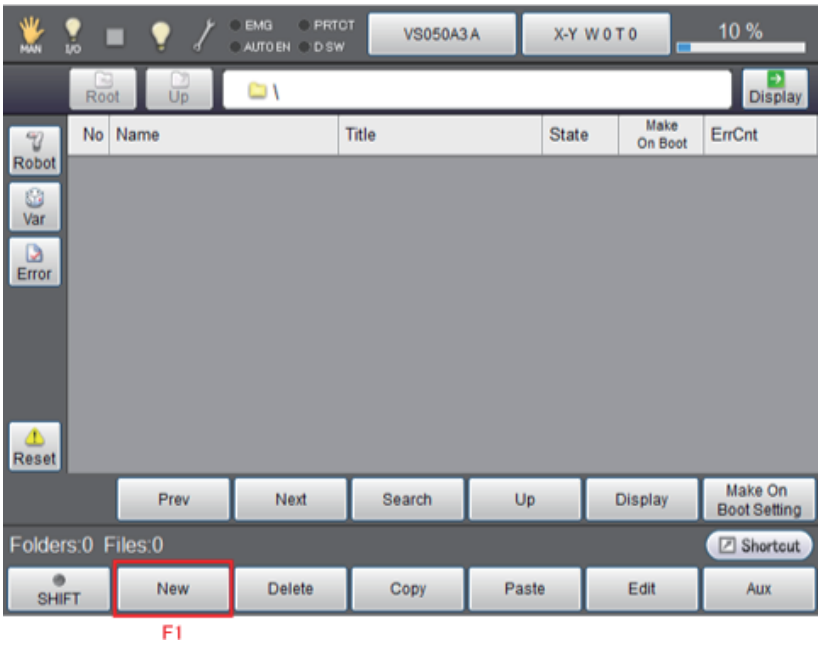


Figure 4: Step 2 of enter a new program name

(3) Step 3: Enter a program name and select [Simple] then press [OK]

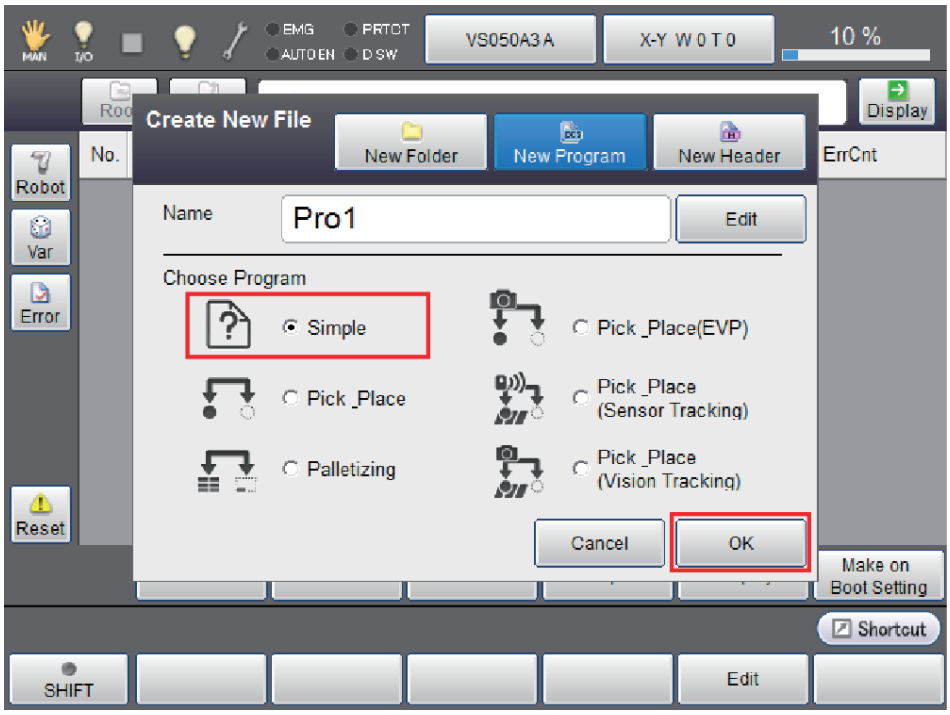


Figure 5: Step 3 of enter a new program name

(4) Step 4: This ends the preparation for program editing

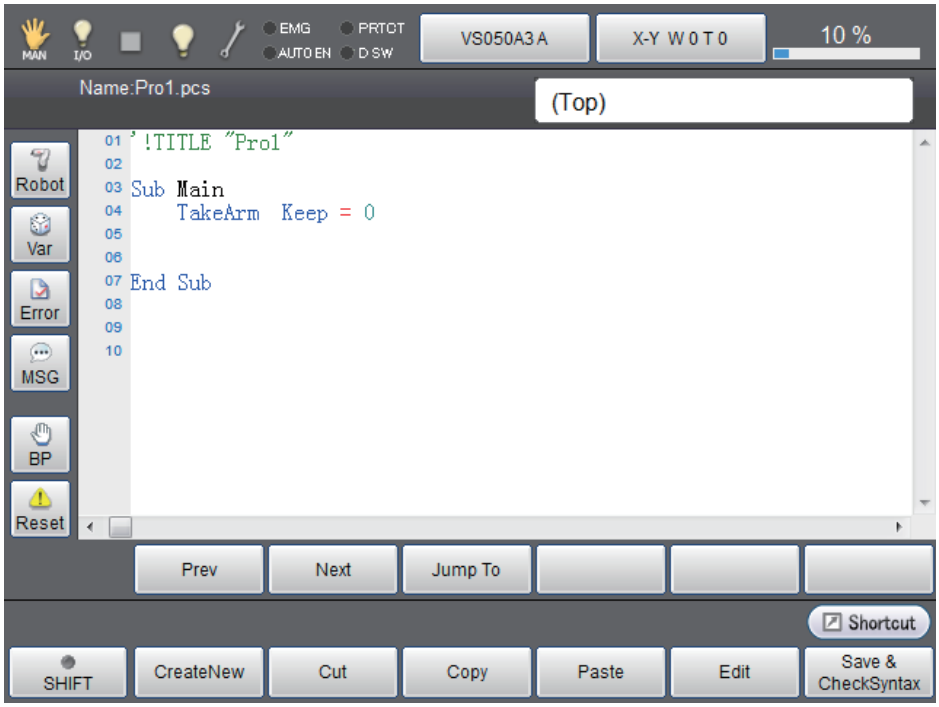


Figure 6: Step 4 of enter a new program name

For this lab exercise, create a program called GRP#L4#, replace the first # with your group number and the second # with A, B, … as you possibly create multiple versions of the program.

4.2 Palletizing Instructions

Commands	Functions
Pallet.CalcPos	Specify the pallet conditions, stack count and the target position number to retrieve the desired position.

Pallet.CalcPos

Function

Specify the pallet conditions, stack count and the target position number to retrieve the desired position.

Syntax

```
Pallet.CalcPos(P1_P3 divisions, P1_P2 divisions, Stack height,

pallet 4-corner position P1, pallet 4-corner position P2, pallet 4-corner
position P3,

pallet 4-corner position P4, pallet target position number[, Stack count][,
Robot Number])
```

Guaranteed Entry

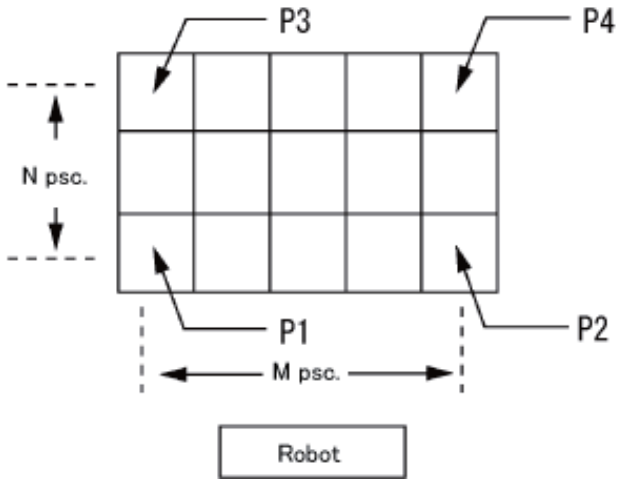


Figure 7: Guaranteed entry

- P1_P3 divisions*:* Specify the number of divisions on the P1-P3 direction (N) by integer type data.
- P1_P2 divisions: Specify the number divisions on the P1-P2 direction (M) by integer type data.
- Stack height: Specify the height of one stack by single precision real number type data. If the stack piles up, enter a positive value. If the number of stack decrease, enter a negative value. If the number of stack remains unchanged, enter 0.
- Pallet 4-corner position P1: Specify the first position in the four corners by position type data.
- Pallet 4-corner position P2: Specify the second position in the four corners by position type data.
- Pallet 4-corner position P3: Specify the third position in the four corners by position type data.
- Pallet 4-corner position P4: Specify the fourth position in the four corners by position type data.
- Pallet target position number: Specify the target position number that is counted from the first position of the stack by integer type data.
- Stack count: Specify the stack count from the first stack by integer type data.

Description

Specify the pallet conditions, stack count and the target position number to retrieve the desired position. Pallet conditions to specify are: P1-P3 division (N), P1-P2 division (M), stack height, and pallet four corners (P1 to P4).

Pallet position numbers are assigned from P1 to P2 and after P2, subsequent numbers are assigned from P1 plus 1 to P3 horizontally as indicated in the figure.

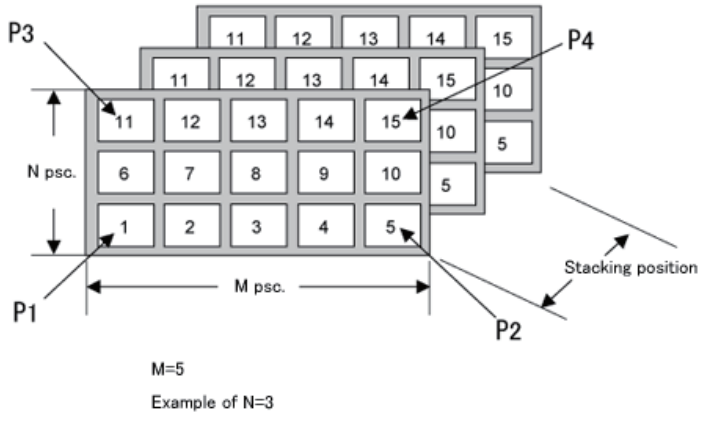


Figure 8: Pallet condition example

Example:

```
'!TITLE "Acquiring Coordinate Position of Pallet Target Position Number"
' Acquire coordinate position designated by pallet target position number and
display it on

the message output window
Sub Sample_PalletCalcPos

  Dim aaa As Position
  Dim bbb As Position
  Dim ccc As Position
  Dim ddd As Position
  Dim eee As Position
  Dim fff As Position
  Dim ggg As Position

  ' Assign position indicating pallet four corner position P1 to aaa
  aaa = P( 600, -100, 50, -180, 0, 180, 5 )

  ' Assign position indicating pallet four corner position P2 to bbb
  bbb = P( 600, 100, 50, -180, 0, 180, 5 )

  ' Assign position indicating pallet four corner position P3 to ccc
  ccc = P( 400, -100, 50, -180, 0, 180, 5 )

  ' Assign position indicating pallet four corner position P4 to ddd
  ddd = P( 400, 100, 50, -180, 0, 180, 5 )

  ' Assign coordinate position of pallet target position number 1 to eee
  eee = Pallet.CalcPos( 3, 5, 20, aaa, bbb, ccc, ddd, 1, 1 )

  ' Display coordinate position of pallet target position number 1 on the
message output window
  PrintDbg eee

  ' Assign coordinate position of pallet target position number 8 to fff
  fff = Pallet.CalcPos( 3, 5, 20, aaa, bbb, ccc, ddd, 8, 1 )
```

```
  ' Display coordinate position of pallet target position number 8 on the
message output window
  PrintDbg fff

  ' Assign coordinate position of pallet target position number 15 to ggg
  ggg = Pallet.CalcPos( 3, 5, 20, aaa, bbb, ccc, ddd, 15, 1 )

  ' Display coordinate position of pallet target position number 15 on the
message output window
  PrintDbg ggg

End Sub
```

4.3 Conditional Branch Instruction

Just as with any other programing modern language, you can perform logic computations with this programming language too. As just noted, each time you click on, the program executes once – but we need the program to continue to execute until all of the blocks have been palletized. In order to fully automate this lab exercise, we therefore need to use the conditional branch instructions, and label instruction.

In the above scenario we need to robot to keep doing the palletizing until all blocks have been handled. We therefore need the program to repeatedly execute a certain number of times. We could simply calculate the number of times needed and hardwire that value in, but then if we change the pallet dimensions we would need to recalculate that number and potentially update the counter as well. A better solution is for the program to figure out on its own how many times it needs to repeat.

Each time the program runs, it updates the location of the block to be handled. This information is recorded in the palletizing register that you specified. We need to do the following to realize the control loop.

(1) GO TO

Function

To jump to a label.

Syntax

```
GoTo label name
```

Description

The system jumps to a label designated in a *label name.*

Example

```
'!TITLE "Unconditional Execution of Program Bifurcation"
' Display results in each label after magnitude determination of aaa and bbb
Sub Sample_GoTo

  Dim aaa As Integer
  Dim bbb As Integer

  aaa = 1
  bbb = 10

  ' Magnitude determination of aaa and bbb
  If aaa > bbb Then

    ' If aaa > bbb, jump to LABEL1
    Goto LABEL1

  ElseIf aaa < bbb Then

    ' If aaa < bbb, jump to LABEL2
    Goto LABEL2

  Else

    ' If aaa = bbb, jump to LABEL3
    Goto LABEL3

  End If

  Exit Sub

LABEL1:

  ' Display the determination result (aaa > bbb) on the message output window
  PrintDbg ( aaa & " > " & bbb )

  Exit Sub
```

```
LABEL2:

  ' Display the determination result (aaa < bbb) on the message output window
  PrintDbg ( aaa & " < " & bbb )

  Exit Sub

LABEL3:

  ' Display the determination result (aaa = bbb) on the message output window
  PrintDbg ( aaa & " = " & bbb )

  Exit Sub

End Sub
```

5 Task

Write a program to palletize blocks from a single pick-up location and form them into a pallet with 4 rows, 3 columns and 2 layers as shown in Figure 14. You may choose the pick-up location to be where ever you wish, but the location shown in figure below works well (be careful to not have the robot reaching too far to reach the pick-up or drop-off location). Alternatively, write a program to de-palletize the previously described pallet with the blocks going to a single drop-off location.

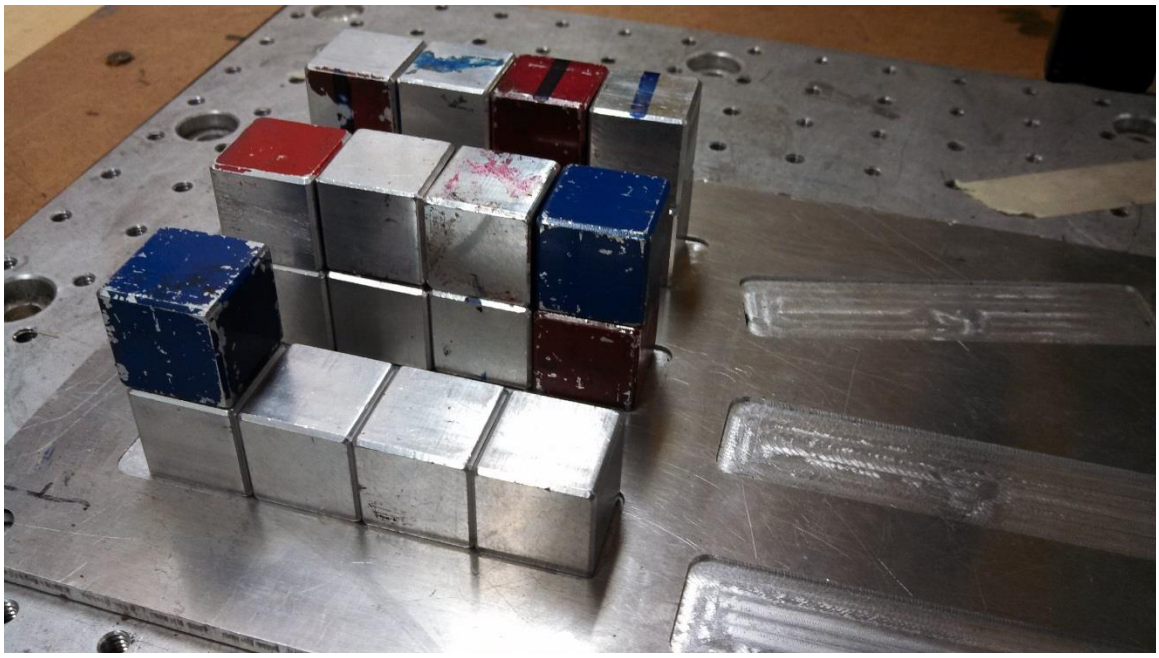


Figure 9; Palletizing / De-palletizing Task

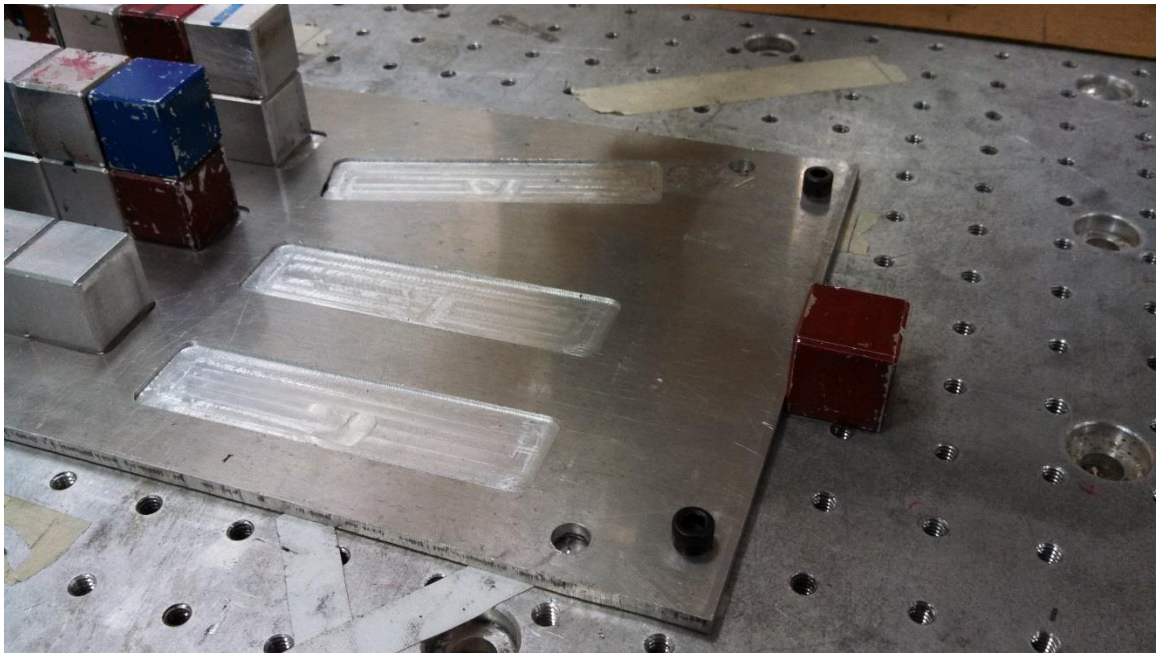


Figure 10 Suggested Pick-up or Drop-off location

5.1 Hints for the task

The easiest way to get things oriented is to start the robot out at a set of known joint angles. I like to start with $J_1 = 0$, $J_4 = 0$, and $J_6 = 0$ and then adjust joints J_2, J_3, and J_5 so that the gripper is pointing straight down at the table. In other words, the gripper's +Z is pointing the world coordinate system –Z direction.

Then switch to the tool coordinate system and start jogging the gripper to where you actually want it. With the gripper oriented straight down, a +Z motion (in the tool frame) will bring the gripper closer to the table while –Z will move the gripper away from the table. Displacements in the tool's X and Y directions will allow you to move around while hovering over the table at a constant Z. Use these motions to jog into the needed poses for the necessary motions.

For the palletizing instruction it's important to have the J_6 axis near zero when generating the various poses that are all part of the palletizing operation. If the gripper is wound around to the neighborhood of +/- 180° when you start generating your palletizing instructions you may get complaints about joint 6 axis limits later on at run time. [Whether those complaints are justified or not is another matter. Regardless, the system may generate them and keep your program from running properly.]

Follow the instructions in this document carefully and you should be successful in getting the palletizing to work. Good luck!