TAMPERE UNIVERSITY OF TECHNOLOGY

Robot Modelling, Control and Programming

Assignment 02

Group member:

Farid Khosravi 267964

Mehdi Mahmoodpour 267958

Palash Halder 267962

ASE – 9406Autumn 2016

Contents

Jacobians: (question 1 to 4)

1) Jacobian is used to define the relation between speed in Cartesian space and speed in joint space of the manipulator. ${}^{0}\boldsymbol{J}$ is the **Jacobian** of the base frame and ${}^{n}\boldsymbol{J}$ is the **Jacobian** of the last frame (i.e. the end effector) .
2) ${}^{0}\boldsymbol{J}$ is calculated by multiplication of ${}^{n}\boldsymbol{J}$ and ${}^{0}_{n}\boldsymbol{R}$. We need ${}^{0}\boldsymbol{J}$ because when we compute linear velocity of a manipulator with respect to speed of the joints, it is easier to compute it with respect to the base frame (frame zero).
3) The sequence of robot movement is described below:
   Step1: end effector is located in the ready_pos_L
   Step2: it goes down for -50mm along Z axis and touches the surface of the metal sheet in the point p10 and starts welding.
   Step3: robot is welding from p10 to p20 and it is moving along Y axis for 160mm.
   Step4: when end effector reaches to the p20 it must have a -90° rotation about Z axis.
   Step5: the robot continues welding from p20 to p30 along X axis for 40mm.
   Step6: when end effector reaches to the p30 it must have a -90° rotation about Z axis.
   Step7: the robot continues welding from p30 to p40 along Y axis for -120mm.
   Step8: when end effector reaches to the p40 it must have a 90° rotation about Z axis.
   Step9: the robot continues welding from p40 to p50 along X axis for 80mm.
   Step10: when end effector reaches to the p50 it must have a -90° rotation about Z axis.
   Step11: the robot continues welding from p50 to p60 along Y axis for -40mm.
   Step12: when end effector reaches to the p60 it must have a -90° rotation about Z axis.
   Step13: the robot continues welding from p60 to p10 along X axis for -120mm.
   Step14: after reaching to p10, the end effector goes up to the ready_pos_L along Z axis for 50mm.
   Step15: robot moves the end effector from ready_pos_L to ready_pos_C.
   Step16: the end effector goes down for -50mm along Z axis and touches the surface of the metal sheet at the point p70 and starts welding.
   Step17: the robot continues welding from p70 to p90 in a half circular with radius 80mm.
   Step18: the robot continues welding from p90 to p100 along Y axis for -40mm.
   Step19: the robot continues welding from p100 to p120 in a half circular with radius 40mm.
   Step20: the robot continues welding from p120 to p70 along Y axis for -40mm.
   Step21: after reaching to p70, the end effector goes up to the ready_pos_L along Z axis for 50mm.
   Step22: robot moves the end effector from ready_pos_C to ready_pos_L and waits for the new sheet.
   All the movements are in **Cartesian space** except movement between ready_pos_L to ready_pos_C and vice versa which is in **Joint space** trajectory.
4) During movement in the joint space segment, no specific path is defined. The end effector moves from one point to another. The points are defined by the joint angles and the path followed is not considered. The path traversed by the end-effector is circular by changing the

joint angles only. The path traversed will be a combination of circular path traversed by changing the joint angles.

Trajectory planning: (question 5 & 6)

The Cubic Polynomial Trajectories defines the path between two points θ(t0) and θ(tf). Figure.1 depicts the angular position of the joint considered. The initial position is θ(t0) = 25° which also can be derived by putting t=0 in the equation below:

$$\theta = a0 + a1 \times t + a2 \times t^2 + a3 \times t^3$$

Then, at t = 0.65s it starts to lift off and the degree of joint increases significantly to reach the set down point when the robot is approaching to the final position. As it can be observed in the figure at the final position is $\theta(t_f) = 90°$ at t = 4 s.

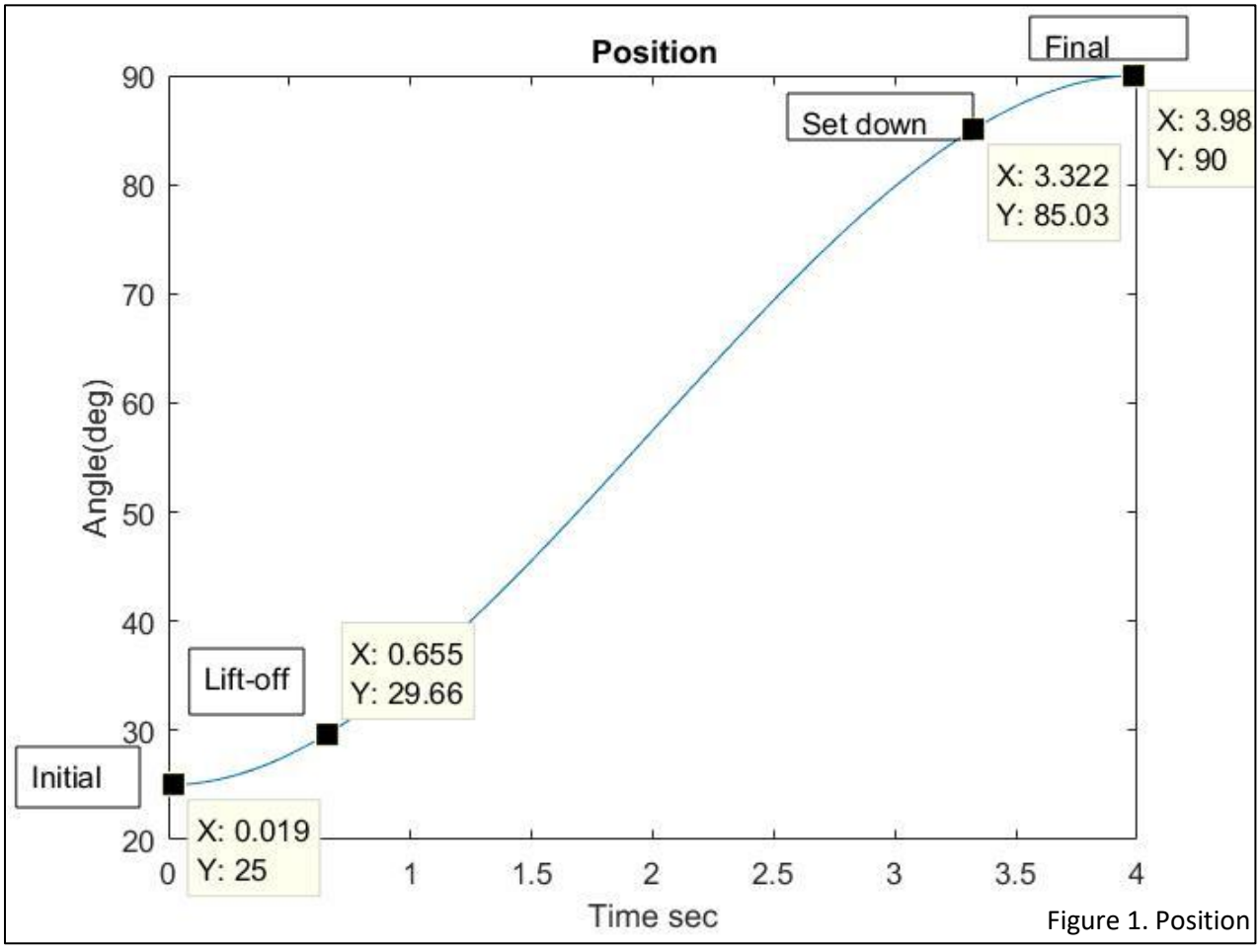


Figure 1. Position

Figure-2, shows the velocity which is derived by differentiating the equation for position with respect to time. It can be observed that it is a parabolic curve. The initial velocity for the given conditions mentioned in this assignment is equal to zero. The slope of curve is the acceleration and is positive in the beginning and it gradually decreases till the velocity reaches to its maximum at t = 2s at which, acceleration is equal to zero. Also, Figure3 can approve this conclusion that has been plotted between

acceleration and time. From t = 2s the velocity starts decreasing and reaches to the final value zero at t = 4 s. Moreover, the acceleration from t = 2 is negative which means, the robot is slowing down and the acceleration at t=4s is zero.

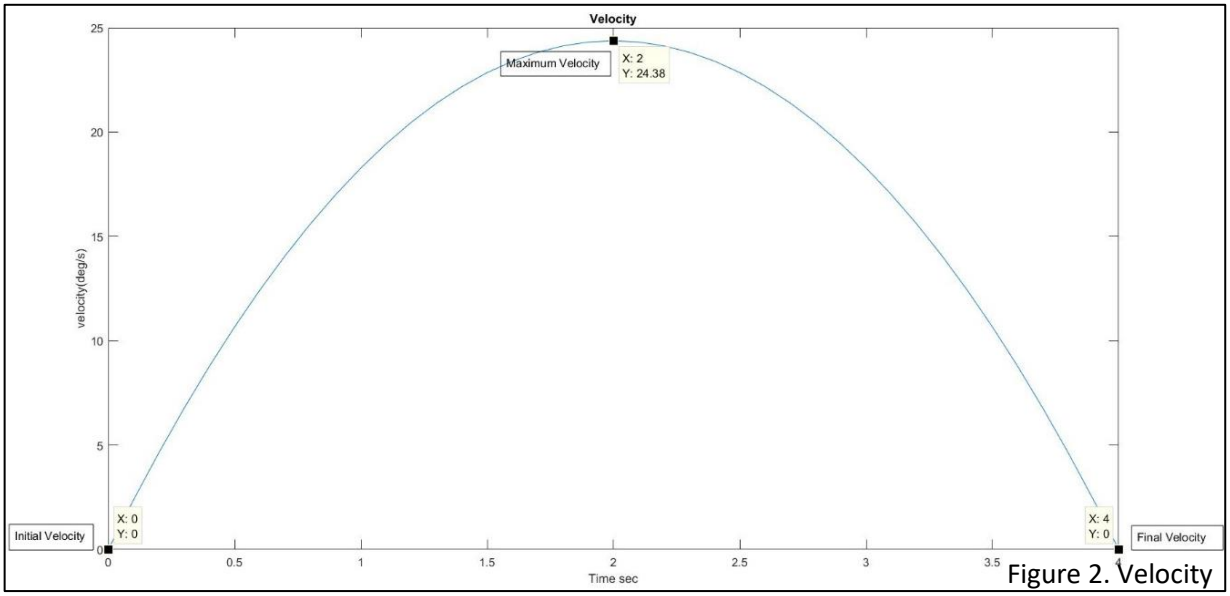


Figure 2. Velocity

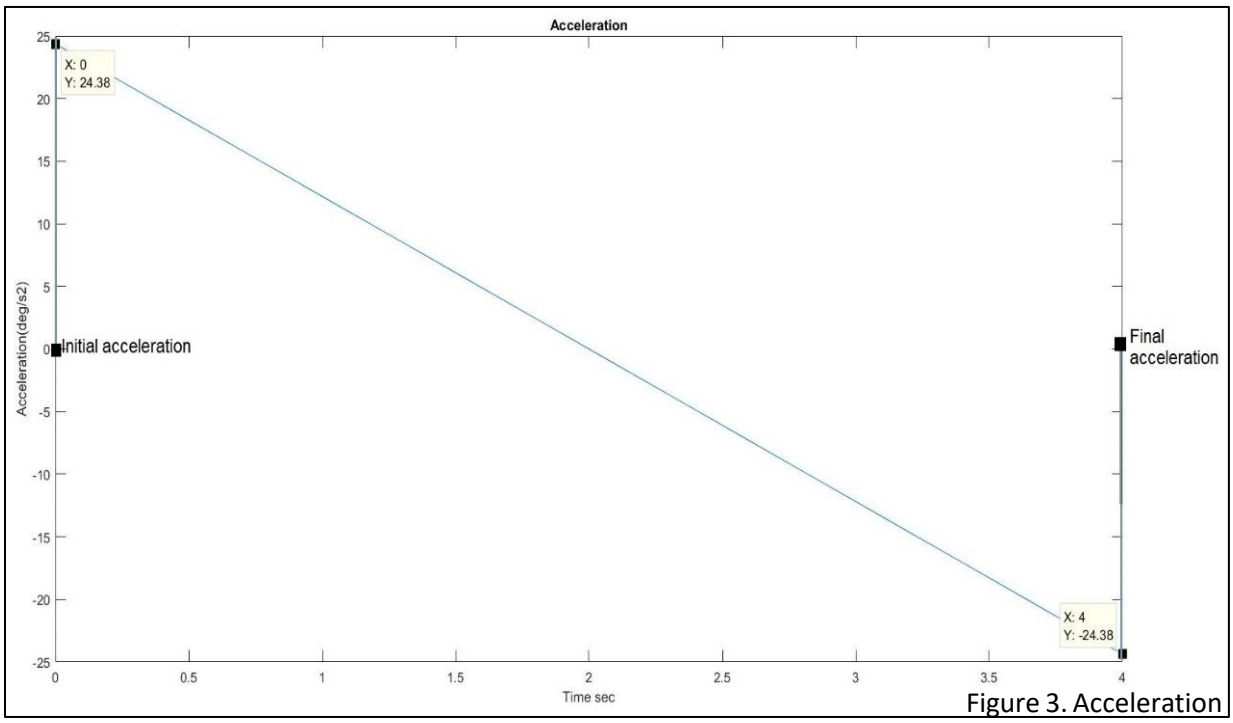


Figure 3. Acceleration

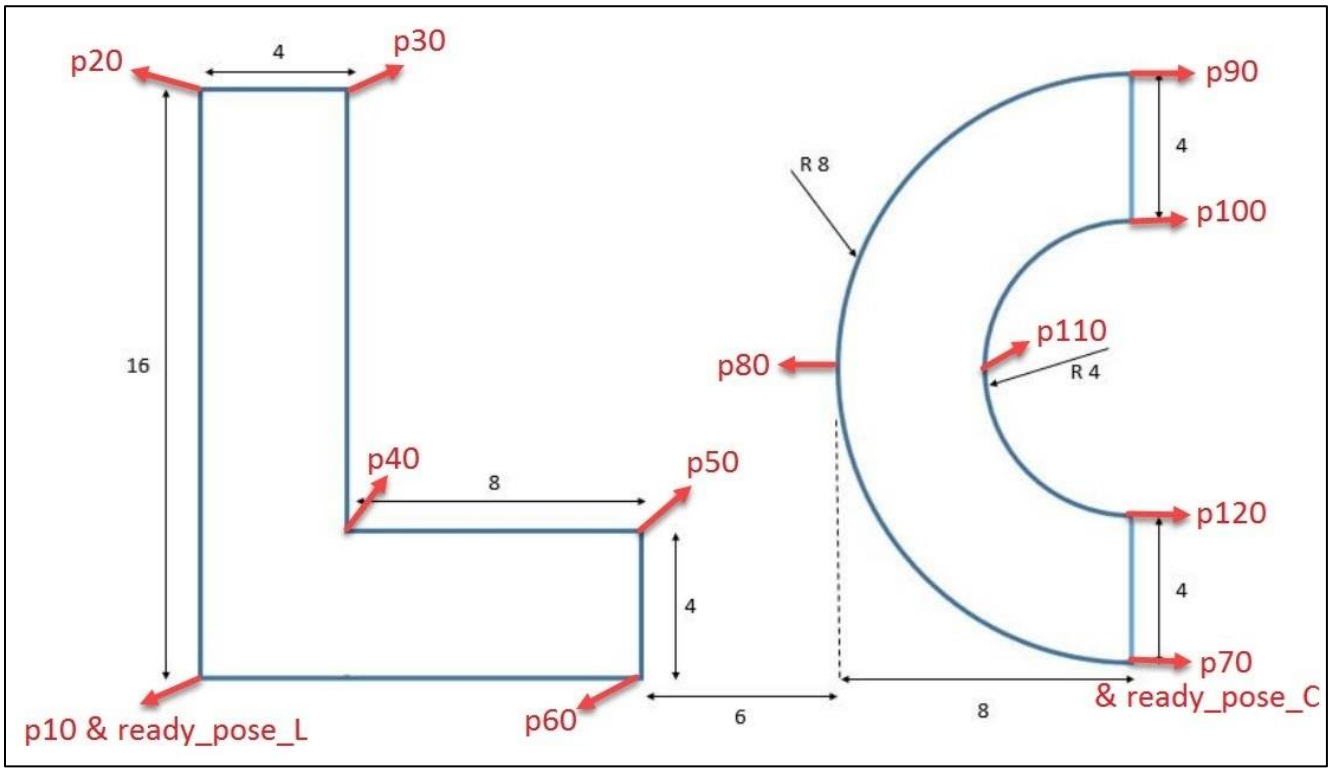
Advantages of cubic Interpolation:

- Provides a relatively “smooth” interpolation and easier computation is required.
- Usually more accurate than linear interpolation.

Disadvantage of cubic interpolation:

- Requires more calculation than linear interpolation.
- In comparison to higher degree interpolation it’s assumptions are less accurate.

ABB Robot Programming: (question 7 & 8)



Program code:

```
VAR num i;
VAR num j := 6;



PROC main()
  for i from 0 TO 10 DO      //the process will be done for 10 metal sheets but if
                                    I had used WHILE loop it will never stop
    j := j*i;                //j is used for increasing position of the tTorch for
                                    next metal sheet along Z axis
   VAR robtarget p10 := [ [ 40, 40, j ] ];
    VAR robtarget ready_pos_L := offs(p10, 0, 0, 50);   //ready point for welding
                                    L shape
   VAR robtarget p20 := offs(p10, 0, 160, 0);
   VAR robtarget p30 := offs(p10,40, 160, 0);
   VAR robtarget p40 := offs(p10, 40, 40, 0);
   VAR robtarget p50 := offs(p10, 120, 40, 0);
   VAR robtarget p60 := offs(p10, 120, 0, 0);
   VAR robtarget ready_pos_C := offs(p10, 260, 0, 50);  //ready point for welding
                                    C shape
```

```
    VAR robtarget p70 := offs(p10, 260, 0, 0);
   VAR robtarget p80 := offs(p10, 180, 80, 0);    //the point in the middle of the
                                                  outer half circle of the C shape
    VAR robtarget p90 := offs(p10, 260, 160, 0);
    VAR robtarget p100 := offs(p10, 260, 120, 0);
    VAR robtarget p110 := offs(p10, 220, 80, 0);  //the point in the middle of the
                                                  inner half circle of the C shape
    VAR robtarget p120 := offs(p10, 260, 40, 0);



       MoveJ ready_pos_L, v700, fine, tTorch                    //going to the
                                                   ready point for welding L shape
    MoveL p10, v200, fine, tTorch, \WObj:=wobj1;         //welding L shape
    MoveL p20, v400, fine, tTorch, \WObj:=wobj1;
    MoveL p30, v400, fine, tTorch, \WObj:=wobj1;
    MoveL p40, v400, fine, tTorch, \WObj:=wobj1;
    MoveL p50, v400, fine, tTorch, \WObj:=wobj1;
    MoveL p60, v400, fine, tTorch, \WObj:=wobj1;
    MoveL p10, v400, fine, tTorch, \WObj:=wobj1;
    MoveL ready_pos_L, v200, fine, tTorch, \WObj:=wobj1;
      MoveJ ready_pos_C, v700, fine, tTorch, \WObj:=wobj1;    //going to the ready
                                                  point for welding C shape
    MoveL p70, v200, fine, tTorch, \WObj:=wobj1;                //welding C shape
     MoveC p80, p90, v400, fine, tTorch, \WObj:=wobj1;         //outer half circle
                                                       of the C shape
    MoveL p100, v400, fine, tTorch, \WObj:=wobj1;
    MoveC p110, p120, v400, fine,tTorch, \WObj:=wobj1;      //Inner half circle of
                                                         the C shape
    MoveL p70, v400, fine, tTorch, \WObj:=wobj1;
    MoveL ready_pos_C, v200, fine, tTorch, \WObj:=wobj1;
    MoveJ ready_pos_L, v700, fine, tTorch, \WObj:=wobj1;    //go back to the first
                                                  position of the L shape
    i := i+1;
  END FOR
END PROC
```

9) Motion of a robot end effector can be performed in the joint space or Cartesian space, the choice between the two depends on the requirement and constraints. Joint space and Cartesian space are defined below:

Joint space- in joint space the joint angles of the joints are defined. If two points are defined then the joint space is used to calculate what kind of joint angles are needed to reach from one point to another. In joint space the path followed by the end effector is not considered, as such it is assumed that there are no contraints present.

Some aspects of joint space:

- Easy to use- find the joint angles required by inverse kinematics and plan accordingly.
- It doesnot pose any problem with singularity.
- Less calculation.
- Cannot follow straight line.

Cartesian space-this frame defines the work space. Here the required path to be followed by the end effector is defined. The defined path is divided into equally spaced points and by calculating the inverse kinematics the joint angles required are defined. The end effector reaches all the defined points and tries to follow the path.

Some aspects of cartesian space:

- Tracking a shape is possible.
- More calculation required (after the path is defined we need to calculate joint angles for each points lying on the path).
- Can pose problem due to singularity.

With the above conception, we used MoveL command for Cartesian space and MoveJ for Joint space. From point ready_pos_L to ready_pos_C and vice versa we used MoveJ because the path is not important and the important thing is going from the first point to the second point, thus it is a Joint space trajectory. And we used MoveL command where the path for movement of the end effector was defined.