ROBOT KINEMATICS, MODELLING AND SIMULATION

ROBOTICS 2023

OLE MADSEN
OM@M-TECH.AAU.DK

AALBORG UNIVERSITY
DENMARK

1

Lecture plan

No	Responsible	Contents
1	Ole Madsen	Introduction to: • the course; • robotics and robot terminology.
2	Ole Madsen	Spatial descriptions and transformation matrices
3	Ole Madsen	Orientation
4 (FIB)	Ole Madsen	Practical exercise with the on-line programming (1.5 timer/gruppe).
5	Ole Madsen	Forward Kinematics I
6	Ole Madsen	Forward Kinematics II (go though 6 DOF robot) – exercise, you go though your robot
7	Ole Madsen	Inverse kinematics I
8	Ole Madsen	Inverse kinematics II (go through 6DOF robot) – you start on your robot
9	Ole Madsen	Trajectory generation and control (joint)
10	Ole Madsen	Trajectory generation and control (cartesian)
11	Ole Madsen	Jacobian/Exam preparation

2

TRAJECTORY GENERATION II

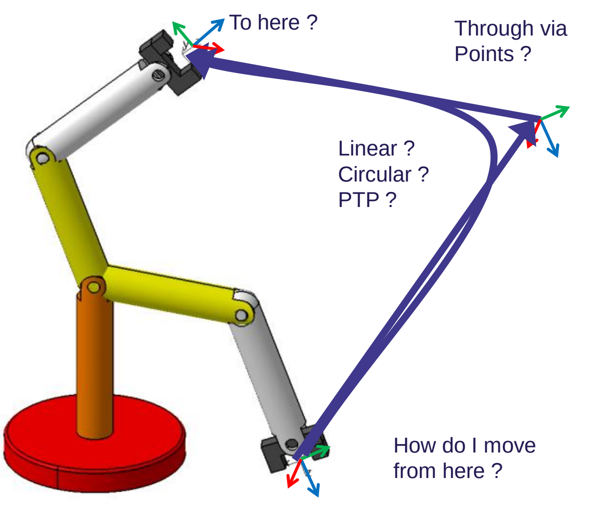
ROBOTICS 2023

OLE MADSEN
OM@M-TECH.AAU.DK

AALBORG UNIVERSITY
DENMARK

3

Trajectory generation



4

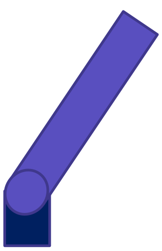
Requirements to trajectory generation

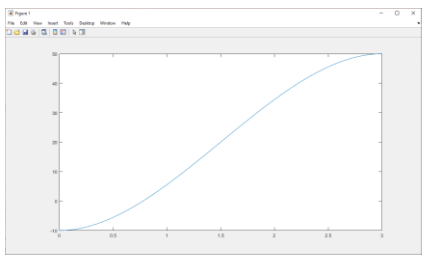
- The motion of the manipulator must be smooth:
  - Continuous path
  - Continuous velocity profile
  - Sometimes continuous acceleration
- All joints reach their target location at the same time

5

Moving one joint axes from one location to another (no via points)

- Robot with one axis
- Desired movement:
  - Start in θ_0
  - End in θ_f
  - Time for movement: t_f
  - Velocities in start and end are zero





We are looking for a function:

$\Theta(t)$

which fullfills the requirements

We will examine two options:
- Cubic polyminials
- Parabolic blend

6

Cubic polynominal –no viapoints

We know:

$$\theta(t) = a_0 + a_1 \cdot t + a_2 t^2 + a_3 t^3$$

$$\theta(\dot{t}) = a_1 + 2a_2 \cdot t + 3a_3 t^2$$

We have the following constraints:

$$\theta(0) = \theta_0$$
$$\theta\left(t_f\right) = \theta_f$$
$$\dot{\theta}(0) = 0$$
$$\dot{\theta}(t_f) = 0$$

$$\theta(0) = a_0 = \theta_0$$

$$\dot{\theta}(0) = a_1 = 0$$

$$\theta(t_f) = \theta_0 + a_2 t_f{}^2 + a_3 t_f{}^3 = \theta_f$$

$$\dot{\theta}(t_f) = 2a_2 \cdot tf + 3a_3 t_f{}^2 = 0$$

$$a_2 = \frac{3}{t_f{}^2}(\theta_f - \theta_0)$$

$$a_3 = -\frac{2}{t_f{}^3}(\theta_f - \theta_0)$$

7

Cubic polynominal –no viapoints

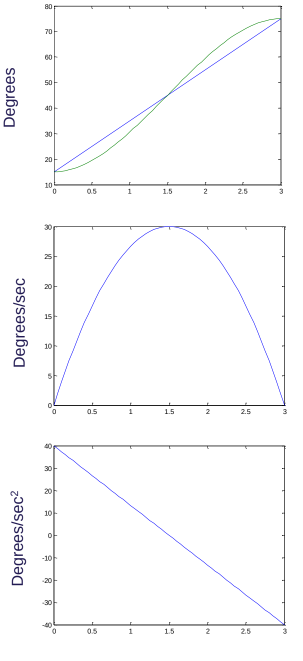
Find the a’s realising the following movement:

- Start in $\theta_0 = 15^\circ$
- End in $\theta_f = 75^\circ$
- Time for movement: $t_f = 3$ sec
- Velocities in start and finish are zero

What is the position at time 1.76 sec ?

Solution

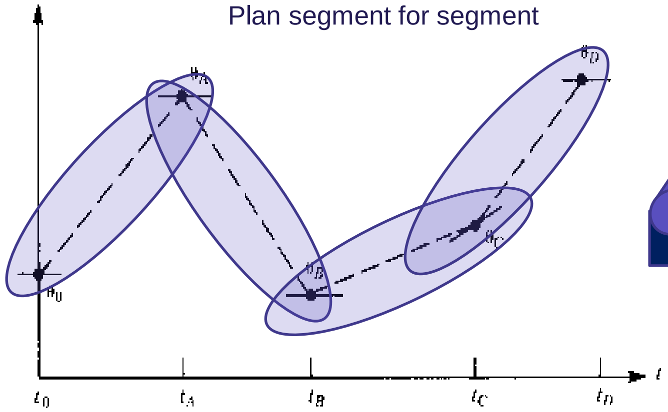
$$\theta(t) = 15 + 20t^2 - 4.44t^3$$
$$\theta(\dot{t}) = 40 \cdot t - 13.33t^2$$
$$\theta(\ddot{t}) = 40 - 26.64 \cdot t$$

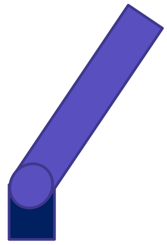


8

Cubic polynominal with via points

Plan segment for segment





9

Cubic polynominal –with viapoints

We know:

$$\theta(t) = a_0 + a_1 \cdot t + a_2 t^2 + a_3 t^3$$

$$\dot{\theta}(t) = a_1 + 2a_2 \cdot t + 3a_3 t^2$$

We have the following constraints:

$$\theta(0) = \theta_0$$

$$\theta(t_f) = \theta_f$$

$$\dot{\theta}(0) = \dot{\theta}_0$$

$$\dot{\theta}(t_f) = \dot{\theta}_f$$

$$\theta(0) = a_0 = \theta_0$$

$$\dot{\theta}(0) = a_1 = \dot{\theta}_0$$

$$\theta(t_f) = \theta_0 + \dot{\theta}_0 t_f + a_2 t_f^{\,2} + a_3 t_f^{\,3} = \theta_f$$

$$\dot{\theta}(t_f) = \dot{\theta}_0 + 2a_2 \cdot tf + 3a_3 t_f^{\,2} = \dot{\theta}_f$$

$$a_2 = \frac{3}{t_f^2}(\theta_f - \theta_0) - \frac{2}{t_f}\dot{\theta}_0 - \frac{1}{t_f}\dot{\theta}_f$$

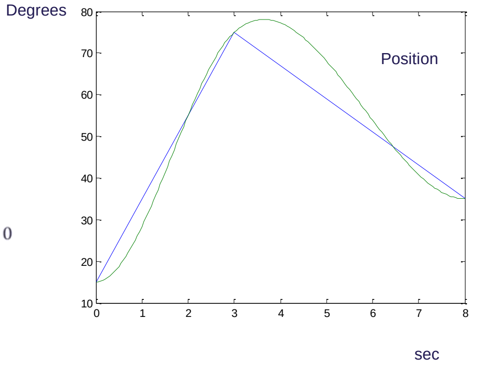
$$a_3 = -\frac{2}{t_f^{\,3}}(\theta_f - \theta_0) + \frac{1}{t_f^2}(\dot{\theta}_f + \dot{\theta}_0)$$

10

Cubic Polynomials- with via points - Example

Given:

- t_{f12} = 3 sec
- t_{f23} = 5 sec
- $\Theta(0)$ = 15 degrees
- $\Theta(t_{f12})$ = 75 degrees
- $\Theta(t_{f23})$ = 35 degrees
- Start and stop from rest
  - $\dot{\theta}(0) = 0; \dot{\theta}(t_{f12} + t_{f23}) = 0$
- Speed in via point:
  - $\dot{\theta}(t_{f12}) = 10 degrees/sec$

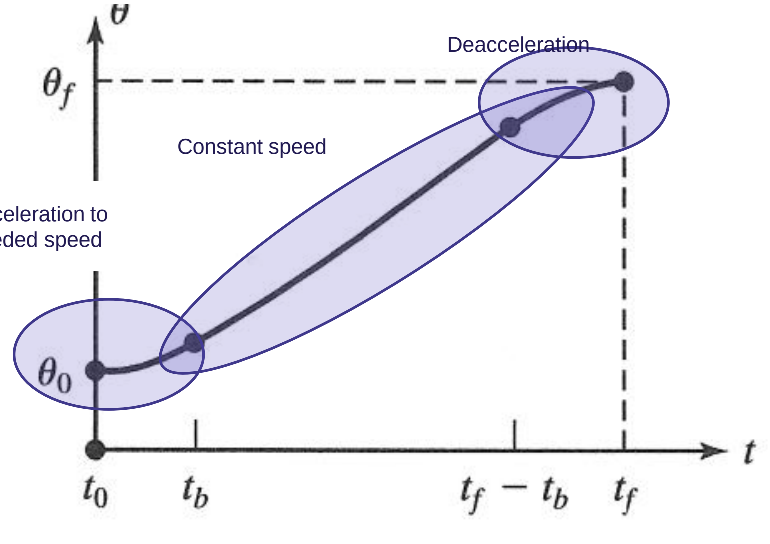


$$\theta 12(t) = 15.00 + 16.66t^2 - 3.33t^3$$

$$\theta 23(t) = 75.00 + 10.00(t - t_{f12}) - 8.80(t - t_{f12})^2 - 1.04(t - t_{f12})^3$$

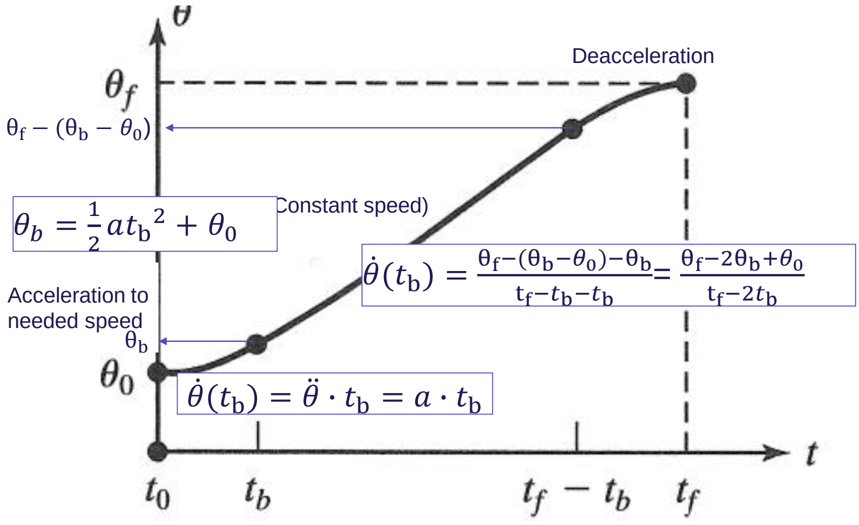
11

Liniar function with parabolic blend



12

Liniar function with parabolic blend



13

Liniar function with parabolic blend

$$\dot{\theta}(t_b) = a \cdot t_b$$

$$\dot{\theta}(t_b) = \frac{\theta_f - 2\theta_b + \theta_0}{t_f - 2t_b}$$

$$\theta_b = \frac{1}{2} a {t_b}^2 + \theta_0$$

$$\frac{\theta_f - 2(\frac{1}{2} a {t_b}^2 + \theta_0) + \theta_0}{t_f - 2t_b} = a \cdot t_b$$

$$\theta_f - a{t_b}^2 - \theta_0 = a \cdot t_b (t_f - 2t_b)$$

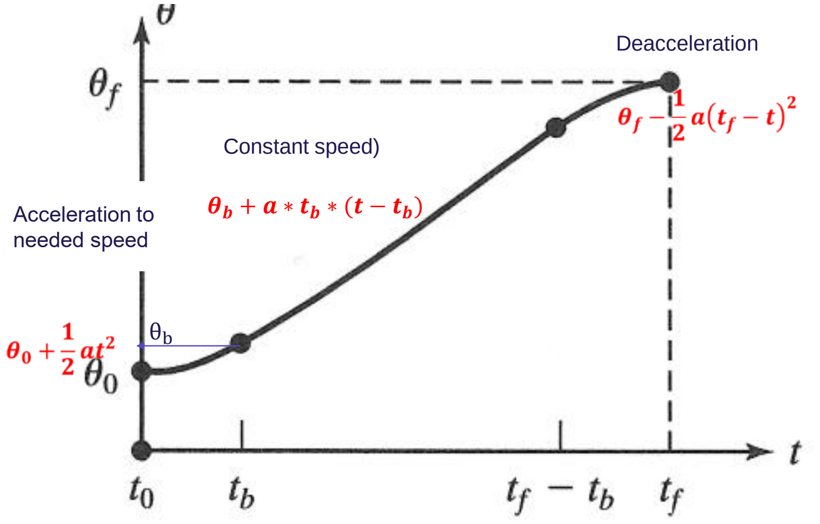
$$a{t_b}^2 - a t_f t_b + \theta_f - \theta_0 = 0$$

$$t_b = \frac{t_f}{2} - \frac{\sqrt{a^2 t_f^2 - 4a(\theta_f - \theta_0)}}{2a}$$

$$a >= \frac{4(\theta_f - \theta_0)}{t_f^2}$$

14

Liniar function with parabolic blend



15

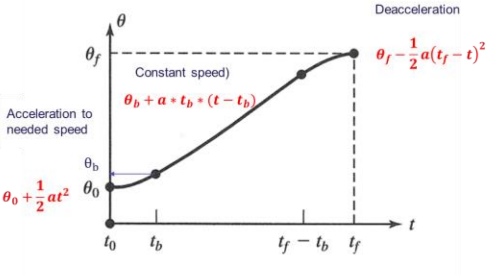
Liniar function with parabolic blend

Assume the following is given:

- Θ_0, θ_f, t_f and a

Compute t_b and θ_b :

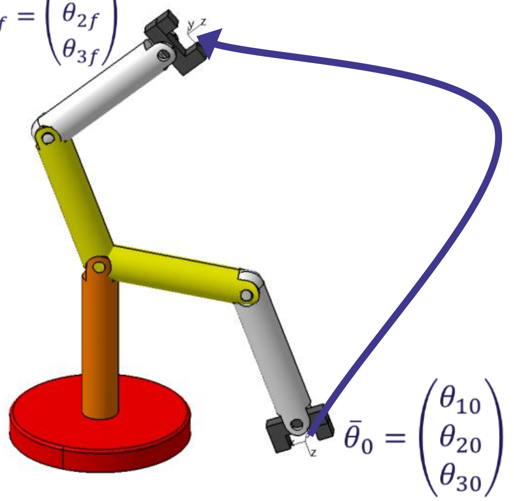
$$t_b = \frac{t_f}{2} - \frac{\sqrt{a^2 t_f^2 - 4a(\theta_f - \theta_0)}}{2a} \qquad \theta_b = \frac{1}{2} a t_b^2 + \theta_0$$



16

PTP-movement (MoveJ, J, Joint movement)

$$\bar{\theta}_f = \begin{pmatrix} \theta_{1f} \\ \theta_{2f} \\ \theta_{3f} \end{pmatrix}$$

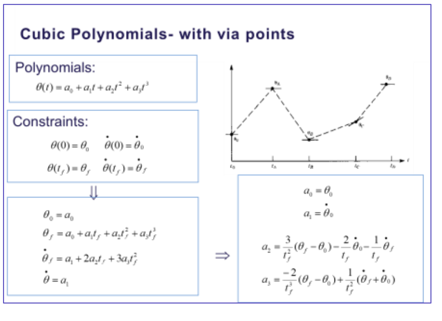


$$\bar{\theta}_0 = \begin{pmatrix} \theta_{10} \\ \theta_{20} \\ \theta_{30} \end{pmatrix}$$

17

PTP-movement (MoveJ, J, Joint movement)

1. Plan the movement e.g: if we use cubic polynominals: find the a's.



2. Interpolate find the location for a given t ∈ $[0;t_f]$:

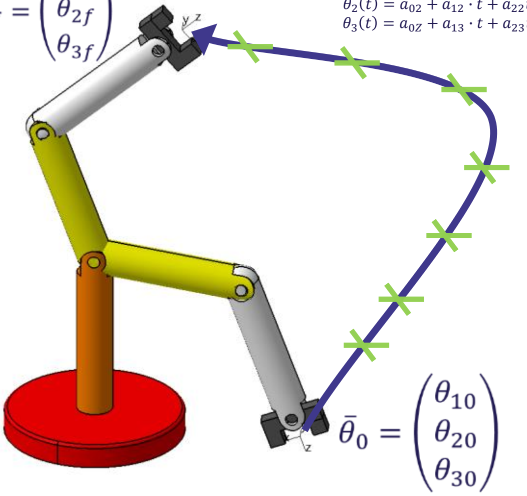
$$\theta_1(t) = a_{01} + a_{11} \cdot t + a_{21}t^2 + a_{31}t^3$$
$$\theta_2(t) = a_{02} + a_{12} \cdot t + a_{22}t^2 + a_{32}t^3$$
$$\theta_3(t) = a_{0Z} + a_{13} \cdot t + a_{23}t^2 + a_{33}t^3$$

18

PTP-movement (MoveJ, J, Joint movement)

$$\bar{\theta}_f = \begin{pmatrix} \theta_{1f} \\ \theta_{2f} \\ \theta_{3f} \end{pmatrix}$$

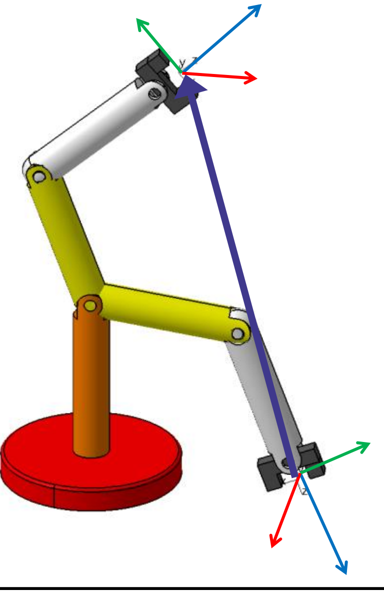
$$\theta_1(t) = a_{01} + a_{11} \cdot t + a_{21}t^2 + a_{31}t^3$$
$$\theta_2(t) = a_{02} + a_{12} \cdot t + a_{22}t^2 + a_{32}t^3$$
$$\theta_3(t) = a_{0Z} + a_{13} \cdot t + a_{23}t^2 + a_{33}t^3$$



$$\bar{\theta}_0 = \begin{pmatrix} \theta_{10} \\ \theta_{20} \\ \theta_{30} \end{pmatrix}$$

19

Cartesian trajectory generation (e.g MoveL)



Move linear from a start location to an end location.

20

Cartesian trajectory generation (e.g MoveL)

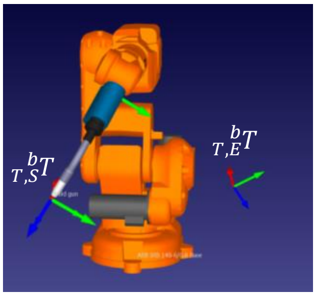
Given:

$^{B}_{TS}T$ = Start location of tool relatively to the robot base

$^{B}_{TE}T$ = End location of tool relatively to the robot base

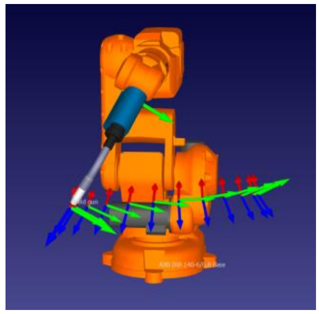
t_f = The time for the movement

(if we know the desired average velocity for the movement t_f can be found from the distance between the origins of the start and end locations)



Output:

- A trajectory of tool locations moving it from the start to the end location under the given constrains



21

Agenda:

1. Trajectory planning in Cartesian space:
   - Method 1: Interpolation in XYZ (keeping the orientation constant)
   - Method 2: Interpolation using RPY
   - Method 3: Interpolation using equivalent angle axis.

2. Problems with cartesian planning

3. Putting it all together in a robot controller

22

Method 1: Interpolate in XYZ (works only if orientation is unchanged)

1. Find start (Xs,Ys,Zs) and end (Xe,Ye,Ze) positions from ${}_{TS}^{B}T$ and ${}_{TE}^{B}T$

$$ {}_{B}^{A}T = \begin{bmatrix} {}^{A}i_{B,X} & {}^{A}j_{B,X} & {}^{A}k_{B,X} & {}^{A}P_{BORG,X} \\ {}^{A}i_{B,Y} & {}^{A}j_{B,Y} & {}^{A}k_{B,Y} & {}^{A}P_{BORG,Y} \\ {}^{A}i_{B,Z} & {}^{A}j_{B,Z} & {}^{A}k_{B,Z} & {}^{A}P_{BORG,Z} \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Position

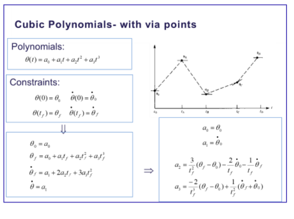
2. If the desired average velocity (Vac) is know compute t_f as:

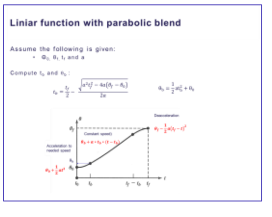
$$ t_f = \frac{\sqrt{(Xe - Xs)^2 + (Ye - Ys)^2 + (Ze - Zs)^2}}{Vac} $$

23

Method 1: Interpolate in XYZ (works only if orientation is unchanged)

3. Plan the motion using parabolic blends or cubic polynominals





4. Use the parameters to find intermediate positions.

$$ X(t) = a_{0X} + a_{1X} \cdot t + a_{2X}t^2 + a_{3X}t^3 $$
$$ Y(t) = a_{0Y} + a_{1Y} \cdot t + a_{2Y}t^2 + a_{3Y}t^3 $$
$$ Z(t) = a_{0Z} + a_{1Z} \cdot t + a_{2Z}t^2 + a_{3Z}t^3 $$

24

Method 1: Interpolate in XYZ (works only if orientation is unchanged)

5. Insert the computed values into a transformation matrix (keeping the rotation matrix from the start pose).

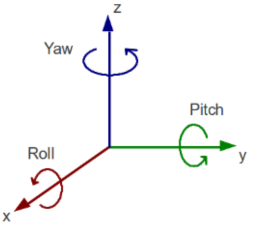
25

Agenda:

1. Trajectory planning in Cartesian space:
    - Method 1: Interpolation in XYZ (keeping the orientation constant)
    - Method 2: Interpolation using RPY
    - Method 3: Interpolation using equivalent angle axis.
2. Problems with cartesian planning
3. Putting it all together in a robot controller

26

Interpolate using RPY (or another 3 angle representation)



RPY (Roll, Pitch, Yaw):

- Start with a frame {B} coincident with a known reference frame {A}:
  - rotate {B} i_A (X) by an angle γ (Roll)
  - rotate about j_A (Y) by an angle β (Pitch)
  - rotate about k_A (Z) by an angle α (Yaw)
- Basic idea of interpolation:
  - Interpolate on XYZ as described in method 1.
  - Find the RPY-values in start and end location
  - Use cubic polynominals (or parabolic blends) to find values in-between start and end.
  - Transform the intermediate RPY to an rotation matrix

27

Method 2: Interpolate using XYZ-RPY – step 1

1. Transform start and end locations (${}^{B}_{TS}T$ and ${}^{B}_{TE}T$) to 6x1 vectors (XYZ, RPY).

Inverse Extrinsic (Fixed angel) XYZ rotation (roll, pitch, yaw)

$$ {}^{A}_{B}R_{XYZ} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{22} & r_{23} & r_{33} \end{bmatrix} $$

$$\beta = A\tan 2(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2})$$

$$\alpha = A\tan 2(\frac{r_{21}}{c\beta}, \frac{r_{11}}{c\beta})$$

$$\gamma = A\tan 2(\frac{r_{32}}{c\beta}, \frac{r_{33}}{c\beta})$$

$$\beta = 90$$
$$\alpha = 0.0$$
$$\gamma = A\tan 2(r_{12}, r_{22})$$

$$\beta = -90$$
$$\alpha = 0.0$$
$$\gamma = -A\tan 2(r_{12}, r_{22})$$

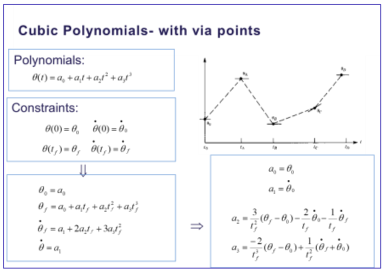
28

Method 2: Interpolate using XYZ-RPY – step 2, 3

2. If the desired average velocity is know compute t_f as:

$$t_f = \frac{\sqrt{(Xe-Xs)^2+(Ye-Ys)^2+(Ze-Zs)^2}}{Vac}$$

3. Use parabolic blends (or cubic polynominals) to represent a path that brings X1 to X2, Y1 to Y2, Z1 to Z2, roll1 to roll2, pitch1 to pitch2, yaw1 to yaw2.



$$\begin{gathered}
X(t) = a_{0X} + a_{1X}\cdot t + a_{2X}t^2 + a_{3X}t^3 \\
Y(t) = a_{0Y} + a_{1Y}\cdot t + a_{2Y}t^2 + a_{3Y}t^3 \\
Z(t) = a_{0Z} + a_{1Z}\cdot t + a_{2Z}t^2 + a_{3Z}t^3 \\
Roll(t) = a_{0R} + a_{1R}\cdot t + a_{2R}t^2 + a_{3R}t^3 \\
Pitch(t) = a_{0P} + a_{1P}\cdot t + a_{2P}t^2 + a_{3P}t^3 \\
Yaw(t) = a_{0Y} + a_{1Y}\cdot t + a_{2Y}t^2 + a_{3Y}t^3
\end{gathered}$$

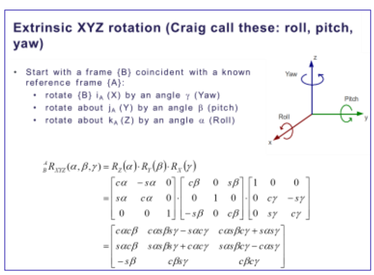
29

Method 2: Interpolate using XYZ-RPY – step 4-5

4. To obtain the location for a given time t $[0;t_f]$, compute the X, Y, Z, Roll, Pitch, Yaw

$$\begin{gathered}
X(t) = a_{0X} + a_{1X}\cdot t + a_{2X}t^2 + a_{3X}t^3 \\
Y(t) = a_{0Y} + a_{1Y}\cdot t + a_{2Y}t^2 + a_{3Y}t^3 \\
Z(t) = a_{0Z} + a_{1Z}\cdot t + a_{2Z}t^2 + a_{3Z}t^3 \\
Roll(t) = a_{0R} + a_{1R}\cdot t + a_{2R}t^2 + a_{3R}t^3 \\
Pitch(t) = a_{0P} + a_{1P}\cdot t + a_{2P}t^2 + a_{3P}t^3 \\
Yaw(t) = a_{0Y} + a_{1Y}\cdot t + a_{2Y}t^2 + a_{3Y}t^3
\end{gathered}$$

5. Transform back to a transformation matrix



30

Agenda:

1. Trajectory planning in Cartesian space:
   - Method 1: Interpolation in XYZ (keeping the orientation constant)
   - Method 2: Interpolation using RPY
   - Method 3: Interpolation using equivalent angle axis.

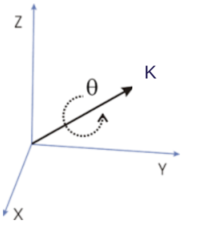
2. Problems with cartesian planning

3. Putting it all together in a robot controller

31

Interpolation using Equivalent angle-axis

Equivalent angle-axis representation:
- Start with a frame {B} coincident with a known reference frame {A}.
- Then rotate {B} about the vector K by an angle θ according to the right-hand rule.



- Basic idea of interpolation:
  - Interpolate on XYZ as described in method 1.
  - Find the angle axis representation bringing start orientation to end orientation (finding K and theta)
  - Use cubic polynominals (or parabolic blends) to intrepolate on the angle theta.
  - Transform back to rotation matrix

32

Method 3: Interpolation using equivalent angle axis.

1. Determine the transformation describing the end location seen from the start:

$$ {}^{TS}_{TE}T = \left({}^{B}_{TS}T\right)^{-1} \cdot {}^{B}_{TE}T $$

2. Convert the rotation matrix of ${}^{TS}_{TE}T$ to the angle-axis representation

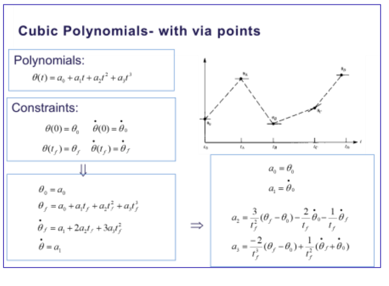
$$ \theta = A\cos\left(\frac{r_{11} + r_{22} + r_{33} - 1}{2}\right) $$

$$ \overline{K} = \frac{1}{2\sin\theta}\begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix} $$

33

Method 3: Interpolation using equivalent angle axis.

3. Computer the parameters of the cubic polynominals (or parabolic blends) for x(t), y(t), z(t) and θ(t) that describe the movements between 0 and t_f.



$$ X(t) = a_{0X} + a_{1X} \cdot t + a_{2X}t^2 + a_{3X}t^3 $$
$$ Y(t) = a_{0Y} + a_{1Y} \cdot t + a_{2Y}t^2 + a_{3Y}t^3 $$
$$ Z(t) = a_{0Z} + a_{1Z} \cdot t + a_{2Z}t^2 + a_{3Z}t^3 $$
$$ \theta(t) = a_{0R} + a_{1R} \cdot t + a_{2R}t^2 + a_{3R}t^3 $$

34

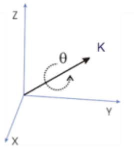
Method 3: Interpolation using equivalent angle axis.

4. For a given time t $[0;t_f]$, compute the X, Y, Z and θ and insert the result in the following equation

Equivalent angle-axis

Start with a frame {B} coincident with a known reference frame {A}.

Then rotate {B} about the vector K by an angle θ according to the right-hand rule.



$$
{}^{A}_{B}R_K(\theta) = \begin{bmatrix} k_x k_x(1-c\theta)+c\theta & k_x k_y(1-c\theta)-k_z s\theta & k_x k_z(1-c\theta)+k_y s\theta \\ k_x k_y(1-c\theta)+k_z s\theta & k_y k_y(1-c\theta)+c\theta & k_y k_z(1-c\theta)-k_x s\theta \\ k_x k_z(1-c\theta)-k_y s\theta & k_y k_z(1-c\theta)+k_x s\theta & k_z k_z(1-c\theta)+c\theta \end{bmatrix}
$$

35

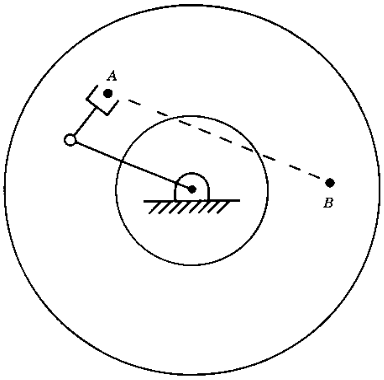
Agenda:

1. Trajectory planning in Cartesian space:
    - Method 1: Interpolation in XYZ (keeping the orientation constant)
    - Method 2: Interpolation using RPY
    - Method 3: Interpolation using equivalent angle axis.
2. Problems with cartesian planning
3. Putting it all together in a robot controller

36

Problems due to Cartesian Interpolation

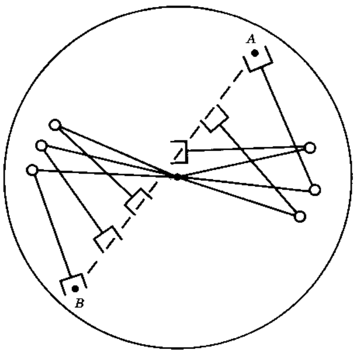
Intermediate points unreachable



37

Problems due to Cartesian Interpolation

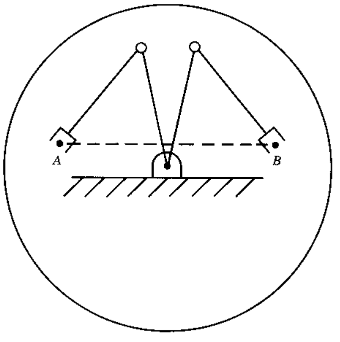
Singularities in the cartesian path



38

Problems due to Cartesian Interpolation

Path points reachable in different solutions/configurations



39

Agenda:

1. Trajectory planning in Cartesian space:
   - Method 1: Interpolation in XYZ (keeping the orientation constant)
   - Method 2: Interpolation using RPY
   - Method 3: Interpolation using equivalent angle axis.
2. Problems with cartesian planning
3. Putting it all together in a robot controller

40

Program

```
MODULE SVEJS
PROC main()
    MoveJ,*,v100,z50,tool0;
    ArcL\On,*,v200,sm1,wd1,wv1,fine,tool0;
    ArcL\Off,*,v100,sm1,wd1,wv1,fine,tool0;
    MoveL,*,v200,fine,tool0;
  ENDPROC
ENDMODULE
```

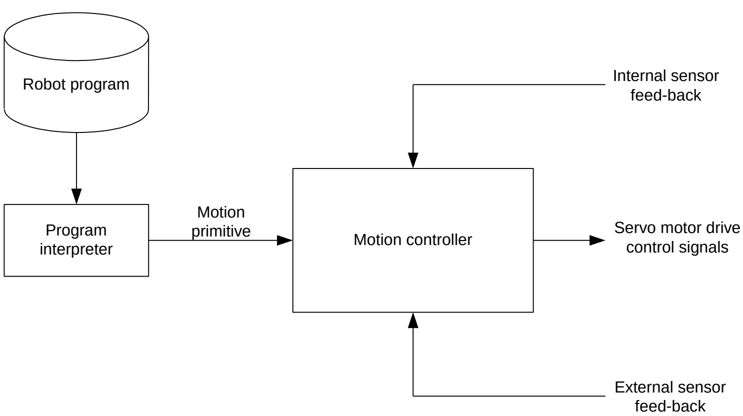
Controller

Machine

Control Data

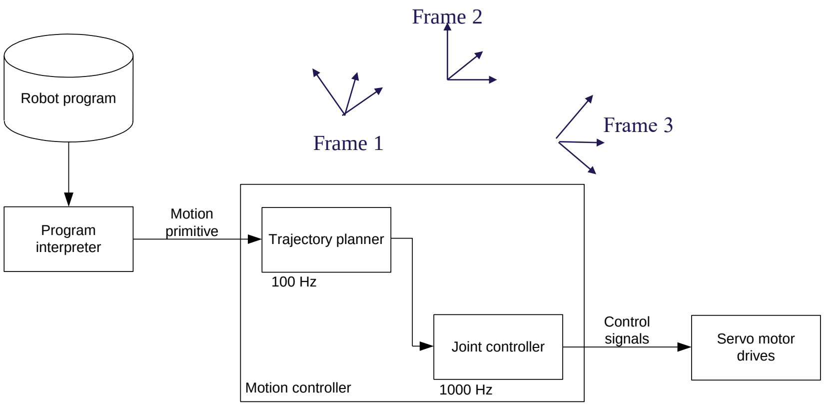
41

A Generic Robot Controller



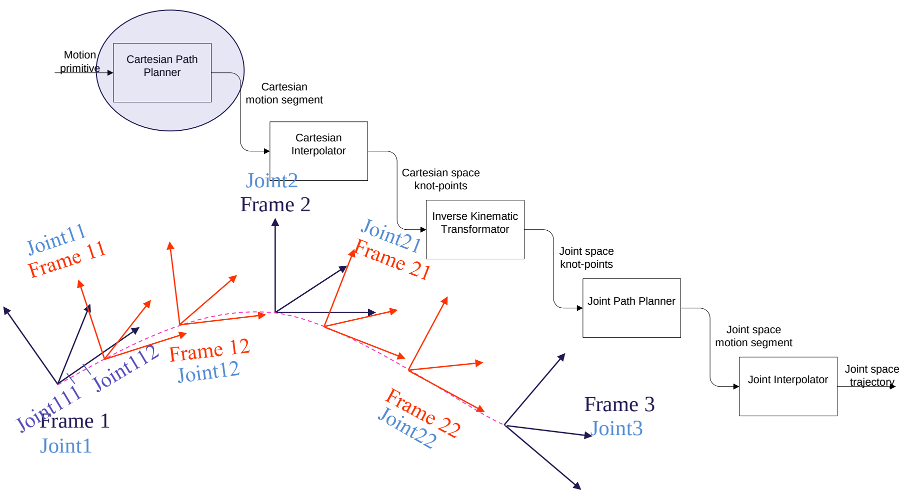
42

Motion controller



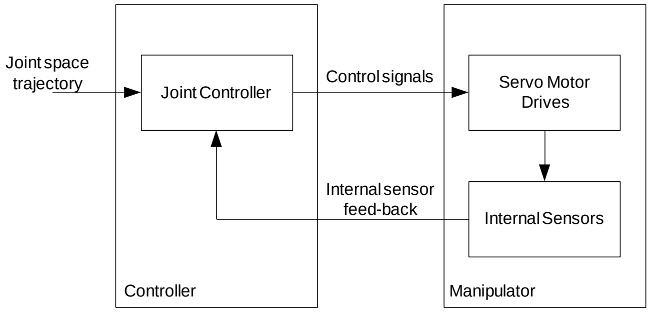
43

Trajectory planner



44

Joint controller



45

Exercises

1. We have a robot tool in a start location {S} and we want the tool to move approximately linear to the end location {E} with a velocity of 100 mm/sec.

    - The locations of the tool relatively to the robot base {B} are given by (where RPY are the roll, pitch, yaw angles):

$$
{}^b_sT = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ R_1 \\ P_1 \\ Y_1 \end{pmatrix} = \begin{pmatrix} 289.48\,\text{mm} \\ 334{,}78\,\text{mm} \\ 818{,}21\,\text{mm} \\ 55.04° \\ -20.95° \\ 139.63° \end{pmatrix} \qquad {}^b_eT = \begin{pmatrix} x_2 \\ y_2 \\ z_2 \\ R_2 \\ P_2 \\ Y_2 \end{pmatrix} = \begin{pmatrix} 561.11\,\text{mm} \\ -71.56\,\text{mm} \\ 1010.43\,\text{mm} \\ -8.28° \\ -55.20° \\ 139.95° \end{pmatrix}
$$

    - Setup the equations describing the movements of the tool (use cubic polynominals).
    - Test the result in matlab (or in RoboDK ?)

46

Exercises

2. Repeat question 2.
    - This time setup the equations describing the movements of the tool (using equivalent angle-axis).
    - Test the result in matlab (or in RoboDK ?)

Exercises (3/3)

The next movement of the manipulator in Figure 3 must be linear in Cartesian space. The manipulator can move in all three spacial directions (X, Y, and Z), and can rotate it's end-effector around Z.

Assume that the robot is configured so that its tool is located in:

$$P_{start} = \begin{pmatrix} X_{start} \\ Y_{start} \\ Z_{start} \\ Rotz_{start} \end{pmatrix} = \begin{pmatrix} 100.0\,\text{mm} \\ -50.0\,\text{mm} \\ 40.0\,\text{mm} \\ 45.0° \end{pmatrix}$$

From here we want to move the robot tool to the following Cartesian location:

$$P_{end} = \begin{pmatrix} X_{end} \\ Y_{end} \\ Z_{end} \\ Rotz_{end} \end{pmatrix} = \begin{pmatrix} 100.0\,\text{mm} \\ 140.0\,\text{mm} \\ 40.0\,\text{mm} \\ 45.0° \end{pmatrix}$$

The movement must take 8 sec.

a) Compute the transformation matrices describing the location of the start- and end-locations relative to the robot base coordinate system (i.e: ${}^{B}_{T}T_{start}$ and ${}^{B}_{T}T_{end}$).
b) Plan the linear motion moving the tool from the start location to the end location. Use parabolic functions (second order polynominals) and assume that the acceleration is $a = 20\,\text{mm/s}^2$.
c) Compute where the tool is located after 0.5 sec.

The robot manipulator in Figure 3 has four joints; one prismatic and three revolute.

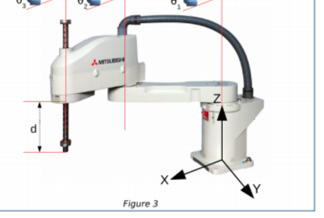


Figure 3

AALBORG UNIVERSITY
DENMARK

49