OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

INF FAKULTÄT FÜR INFORMATIK

Path interpolation: Trajectories and velocity profiles

T. Baier, A. Belov, S. Hagemann

25.11.2013

Agenda

1. Definition "Trajectory"
2. Velocity profiles
3. Classification of trajectories
4. Point–to–point–Control
5. Linear Continuous–path–Control
6. Problems and Potentials

Definition of “Trajectory”

Continuous series of points with a position and an Orientation

→ Geometric representation

Path ≠ Trajectory

Geometric Path + information about the time

→ Time is represented as velocity, acceleration und the chronological sequence of the points

Velocity Profiles

- Mathematical functions which describe the velocity progress during the trajectory

Why do we need this?

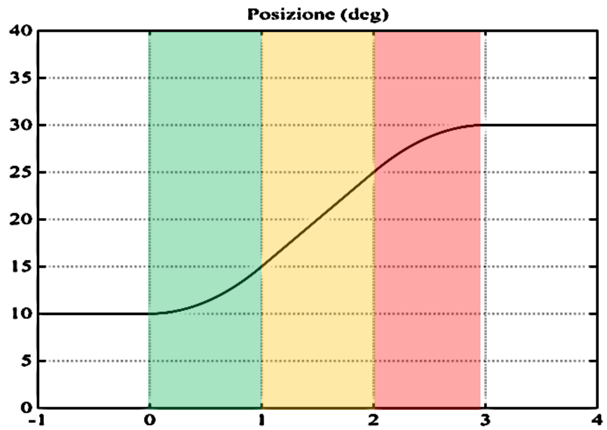
- Enables a smooth distribution of interpolated points
- Avoids high abrasion

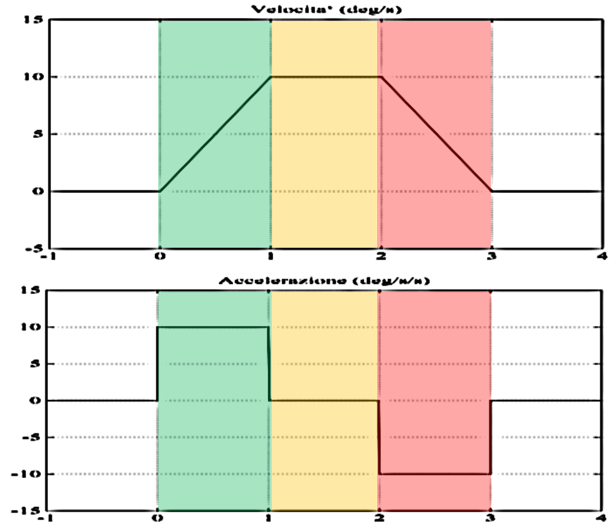
Many types of different velocity profiles described by i.e.:

- Trapezoid / Triangle
- $\sin^2$–function
- …

Velocity Profiles

Trapezoidal Profile





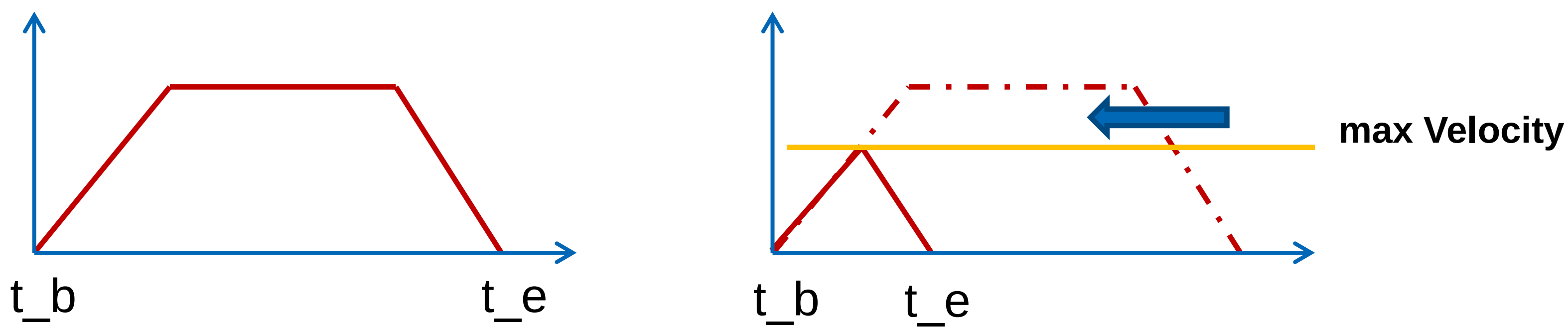
- Linear increasing and decreasing of velocity up to v_{max}
- Accelaration changes abrupt
- Three parts:
  - (1) Acceleration
  - (2) constant velocity
  - (3) Negative acceleration

Velocity Profiles

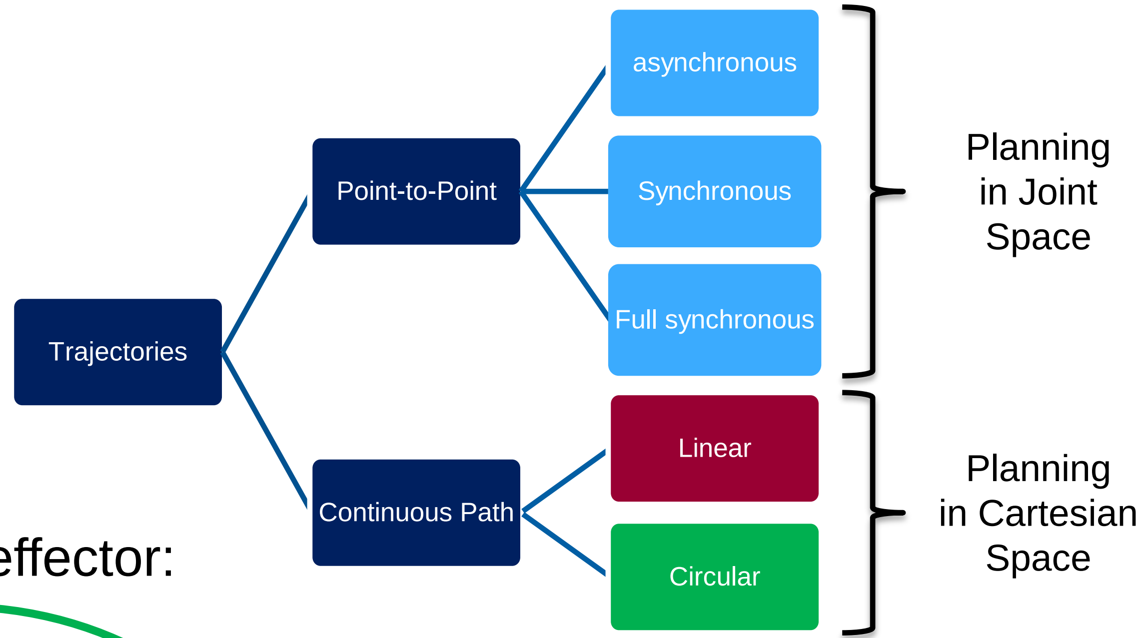
Special case of trapezoid profile: Triangle Profile

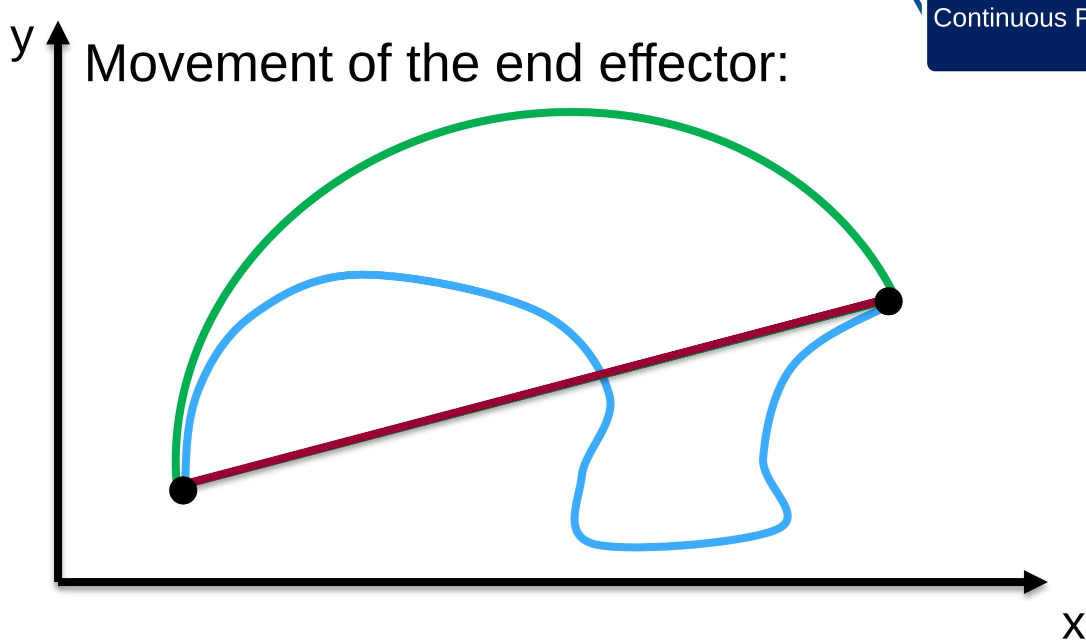
Trapezoid

Triangle



Classification of Trajectories





PTP–Control

- Point–to–Point
- Trajectory planning in joint space
- Goal: Find the shortest way for each joint between two points

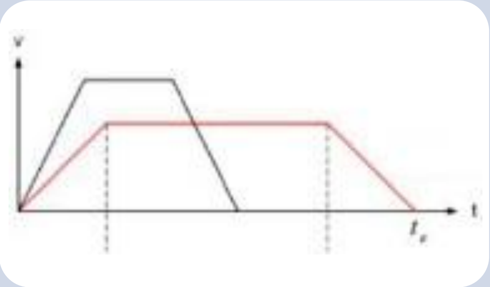
- Process:
  - Choose the synchronization type
  - Compute interpolation points in joint space with a the help of a velocity profile for each axis

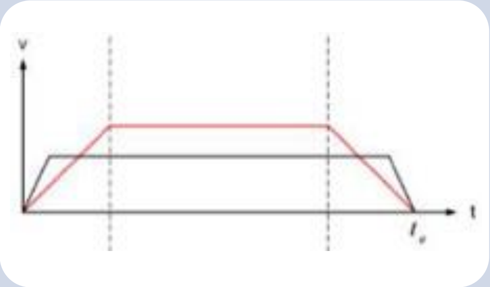
Be careful!

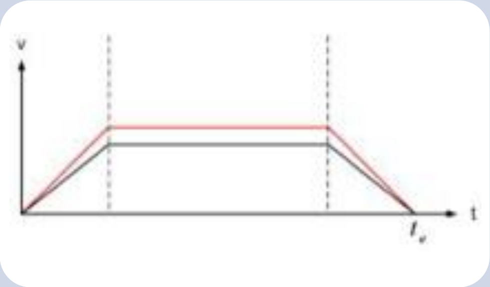
- The end effector reaches the final point but its path, velocity and acceleration are not predictable

PTP–Control

Synchronization Types







Asynchronous

- Moving of the single joints with highest velocity and acceleration
- movements of joints will not end at the same time

Synchronous

- Movement time of all axes will be adjusted to the slowest axis (leading axis)
- finish their movements at the same time

Full synchronous

- additionally adjusting the acceleration time to the leading axis

PTP-Control // Asynchronous movement

Given:

N – amount of degrees of freedom

p1[1..N] – start point of the trajectory [rad]

p2[1..N] – end point of the trajectory [rad]

vMax[1..N] – max velocities for the joints [rad/s]

aMax[1..N] – max accelerations for the joints [rad/s^2]

frequency – calculations per second

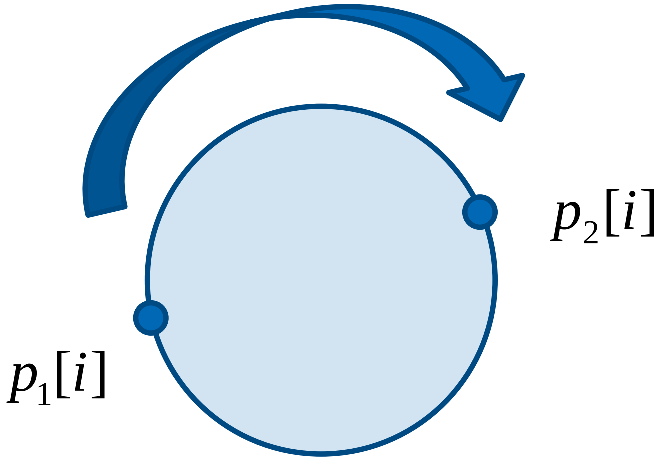
Find:

trajectoryAngle[1..n_step] – angles of joints for each point in time

So that the robot changes its joint configuration according to the velocity profile.

PTP-Control // Asynchronous movement

Step 1
For each Joint: what is the angle distance and change direction?



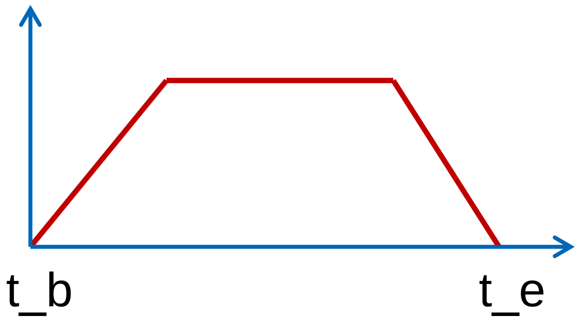
$$changes[i] = p_2[i] - p_1[i]$$

$$if \quad changes[i] \geq 0 \quad sign[i] = 1 \quad else \quad sign[i] = -1$$

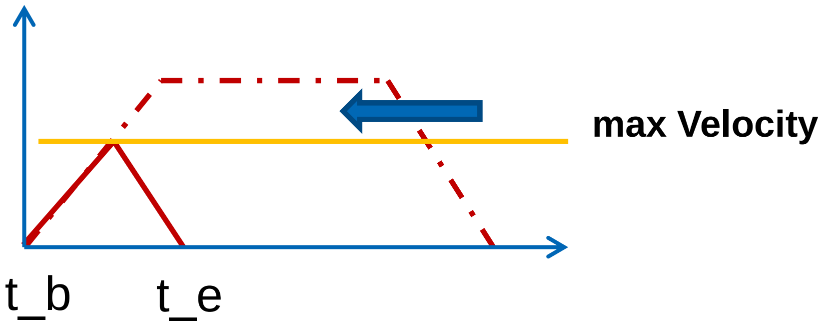
PTP–Control // Asynchronous movement

Step 2
For each Joint: which kind of Velocity Profile?

Trapezoid



Triangle

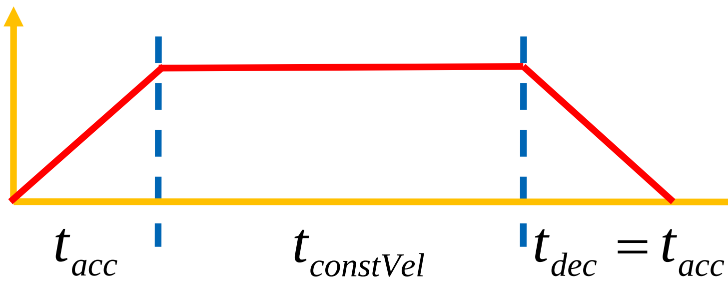


$$v[i] = \sqrt{a_{\max}[i] \cdot abs(changes[i])}$$

$$if \quad v[i] < v_{\max} \; then \qquad v_{\max}[i] = v[i]$$

PTP–Control // Asynchronous movement

Step 3
For each Joint:
Acceleration and Deceleration duration?



$$t_{acc}[i] = \frac{v_{\max}[i]}{a_{\max}[i]}$$ acceleration time

$$dist_{acc}[i] = \frac{a_{\max}[i] \cdot t_{acc}^{\;2}}{2}$$ acceleration distance

$$dist_{constVel}[i] = abs(changes[i] - 2 \cdot dist_{acc}[i])$$ monotone motion distance

$$t_{constVel}[i] = \frac{dist_{constVel}[i]}{v_{\max}[i]}$$ monotone motion time

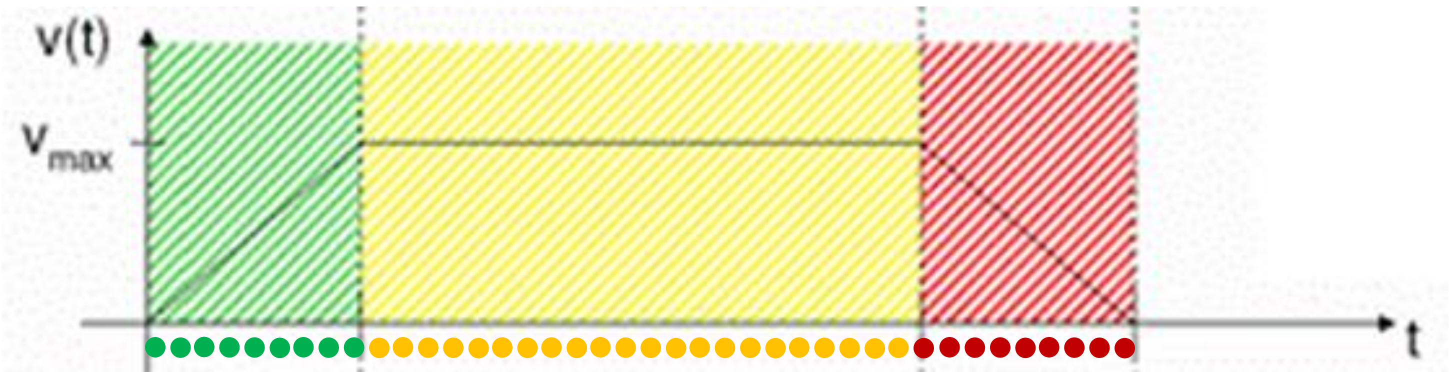
$$t_{total}[i] = 2 \cdot t_{acc}[i] + t_{constVel}[i]$$ total time for a joint

$$T = \max(t_{total}[i])$$ total motion time

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG INF FAKULTÄT FÜR INFORMATIK

PTP–Control // Asynchronous movement

Step 4
For each Joint: What is count of waypoints?



$$n\_steps = frequency \cdot T$$

amount of descret steps

PTP–Control // Asynchronous movement

Step 5 For each Joint: Compute all way points!

$cur\_T = 0$

$for \quad i \quad in \quad [0, n\_steps]$ //go through all way points

$\quad cur\_T = cur\_T + time\_step$

$\quad for \quad j \quad in \quad [1, N]$ //find current velocity for each joint

$\quad\quad if \quad (cur\_T \le t_{acc}[j])$ //acceleration phase

$\quad\quad\quad cur\_v[j] = cur\_v[j] + a_{\max}[j] \cdot time\_step$

$\quad\quad else \quad if \quad (cur\_T > t_{acc}[j] + t_{constVel}[j] \quad \& \quad cur\_T \le T)$ //deceleration phase

$\quad\quad\quad cur\_v[j] = \max(cur\_v[j] - a_{\max}[j] \cdot time\_step, 0)$

$\quad\quad dAngle = sign[j] \cdot cur\_v[j] \cdot time\_step$ //how does each angle change

$\quad\quad cur\_p[j] = cur\_p[j] + dAngle$

$\quad trajectAngle[i] = cur\_p$ //save the found angles to the trajectory vector

PTP–Control // Asynchronous movement

Warning!

You need to handle the calculations at the last point in time, because a discretization error can occur!

Just control the physical values and break the cycle
if they are not plausible at the end point of time. For example:

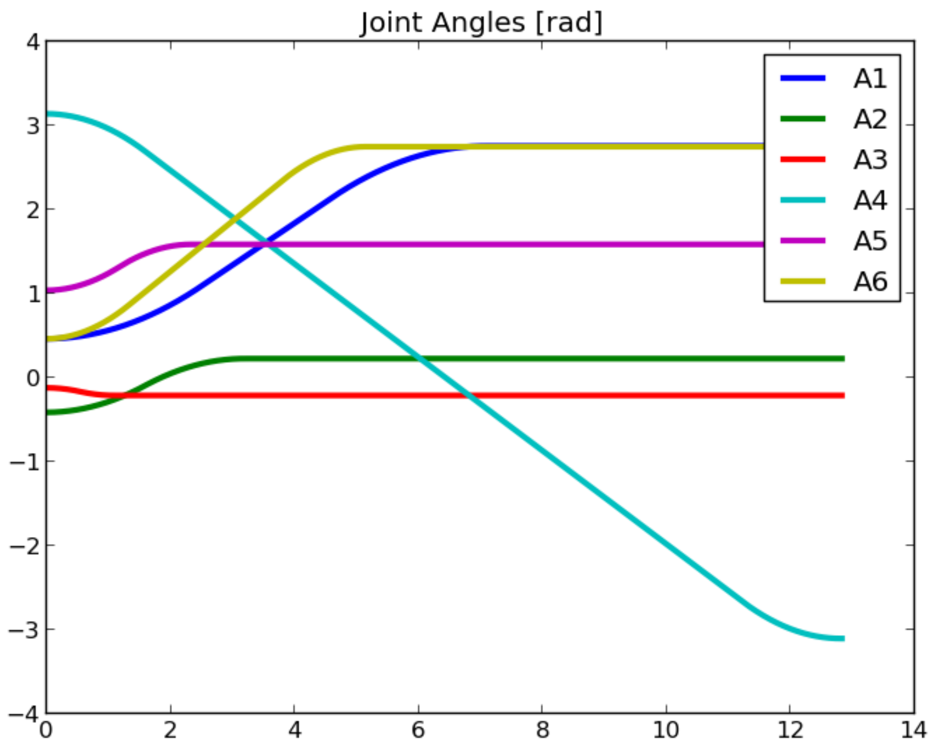
$$\mathit{if}\ cur\_v < 0 \quad \mathit{then}$$

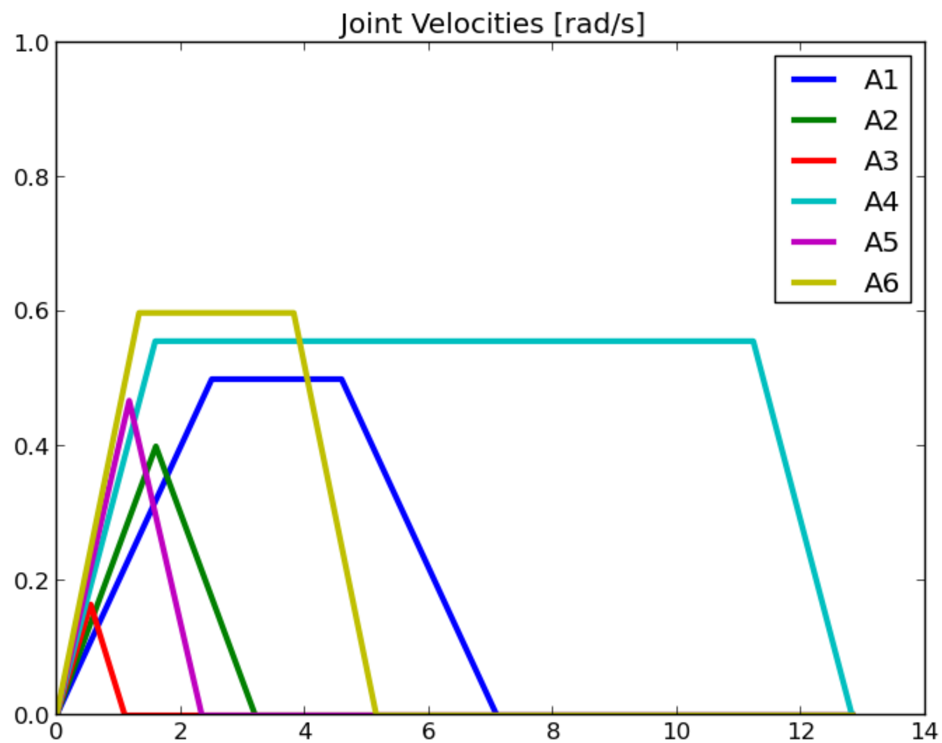
$$cur\_v = 0$$

$$\mathit{break}$$

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG
INF FAKULTÄT FÜR INFORMATIK

PTP-Control // Asynchronous movement





PTP–Control // Synchronous movement

Given:

N – amount of degrees of freedom

p1[1..N] – start point of the trajectory [rad]

p2[1..N] – end point of the trajectory [rad]

vMax[1..N] – max velocities for the joints [rad/s]

aMax[1..N] – max accelerations for the joints [rad/s^2]

frequency – calculations per second

Find:

trajectoryAngle[1..n_step] – angles of joints for each point in time

So that all joints change their angles synchronously and finish at the same point in time.

PTP–Control // Synchronous movement

Step 1

Define motion times for joints

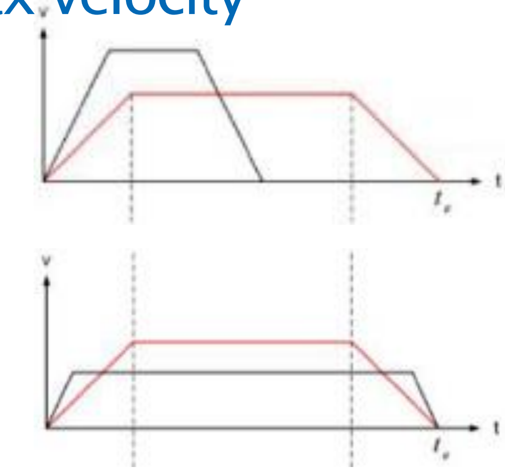
(the same way as in the asynchronous method)

Define the leading joint!

PTP–Control // Synchronous movement

Step 2

Express the longest motion time through the max velocity



$$T = 2 \cdot t_{acc} + t_{constVel}$$

$$t_{acc}[i] = \frac{v_{\max}[i]}{a_{\max}[i]} \qquad t_{constVel}[i] = \frac{dist_{constVel}[i]}{v_{\max}[i]}$$

$$dist_{acc}[i] = \frac{1}{2} \cdot \frac{v_{\max}[i]^2}{a_{\max}}$$

$$dist_{constVel}[i] = changes[i] - 2 \cdot dist_{acc} = changes[i] - \frac{v_{\max}[i]^2}{a_{\max}}$$

$$\boxed{v_{\max}[i]^2 - T \cdot a_{\max}[i] \cdot v_{\max}[i] + a_{\max}[i] \cdot abs(changes[i]) = 0}$$

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG INF FAKULTÄT FÜR INFORMATIK

PTP–Control // Synchronous movement

Step 3
Solve the quadratic equation for velocity

$$a \cdot x^2 + b \cdot x + c = 0$$

$a = 1$ $\quad b = -T \cdot a_{\max}[i]$ $\quad c = a_{\max}[i] \cdot abs(changes[i])$

$$D = b^2 - 4 \cdot a \cdot c$$

$$x = \frac{-b \pm \sqrt{D}}{2 \cdot a}$$

From the two possible results the constraint $2 \cdot t_{acc} \leq T$ annuls the larger one!!!

$$\boxed{v_{\max}[i] = \frac{T \cdot a_{\max}[i] - \sqrt{T^2 \cdot a_{\max}[i]^2 - 4 \cdot a_{\max}[i] \cdot abs(changes[i])}}{2}}$$

PTP-Control // Synchronous movement

Step 4

For each Joint: Acceleration and Deceleration duration?

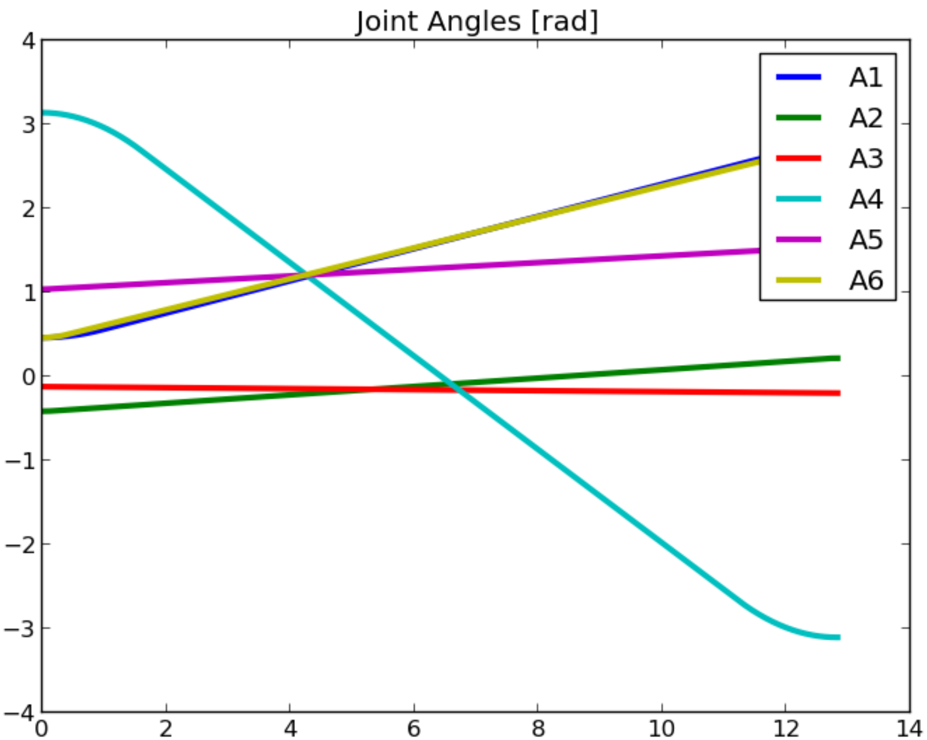
(the same way as in the asynchronous method)

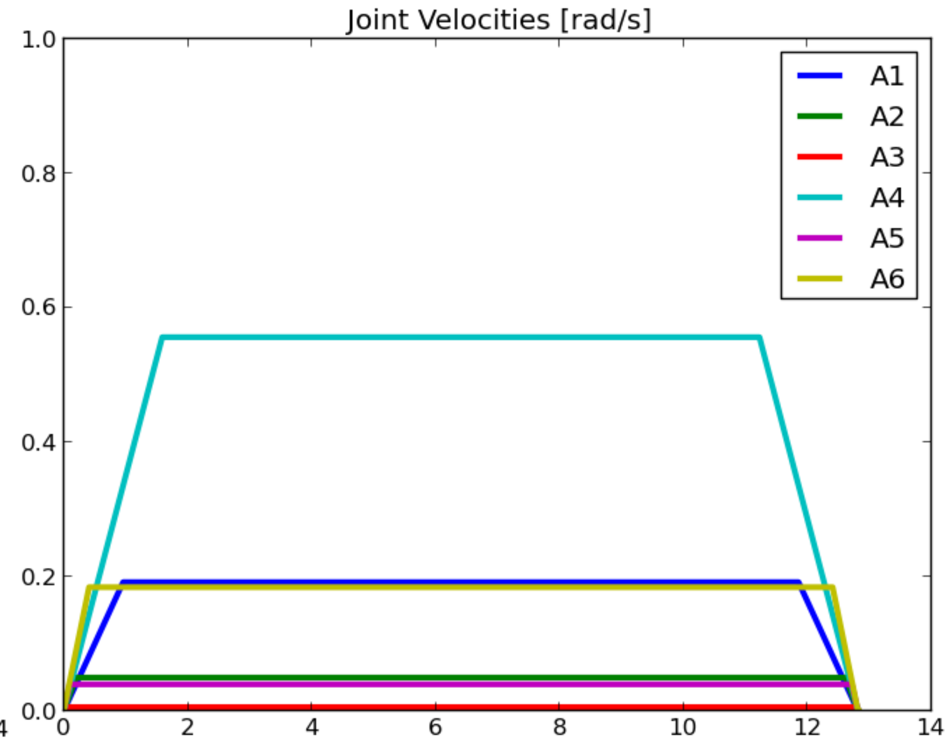
Step 5

For each Joint: Compute all way points!

(the same way as in the asynchronous method)

PTP–Control // Synchronous movement





PTP–Control // Full synchronous movement

Given:

N – amount of degrees of freedom

p1[1..N] – start point of the trajectory [rad]

p2[1..N] – end point of the trajectory [rad]

vMax[1..N] – max velocities for the joints [rad/s]

aMax[1..N] – max accelerations for the joints [rad/s^2]

frequency – calculations per second

Find:

trajectoryAngle[1..n_step] – angles of joints for each point in time

So that all joints change their angles synchronously, finish at the same point in time and have the same acceleration and deceleration phases.

PTP–Control // Full synchronous movement

Step 1 Define the leading joint (the longest acceleration phase)

$$t_{accMax} = \max(t_{acc}[j])$$

//duration of the longest acceleration phase
//use it for all joints!!!

Step 2 What are aMax* and vMax* for each joint to fulfill constraints?

$$T = 2 \cdot t_{accMax} + \frac{dist_{constVel}[j]}{a^*_{\max}[j] \cdot t_{accMax}}$$

//express the overall time through $a^*_{\max}[j]$

$$dist_{constVel}[j] = changes[j] - a^*_{\max}[j] \cdot t^2_{accMax}$$

//const velocity distance for each joint

Solve equations for aMax* and vMax*

$$a^*_{\max}[j] = \frac{changes[j]}{t_{accMax} \cdot (T - t_{accMax})}$$

$$v^*_{\max}[j] = a^*_{\max}[j] \cdot t_{accMax}$$

Check the constraints:

$$a^*_{\max}[j] \le a_{\max}[j]$$

$$v^*_{\max}[j] \le v_{\max}[j]$$

Do other steps in the same way like for synchronous movement!

PTP–Control // Full synchronous movement

Warning!

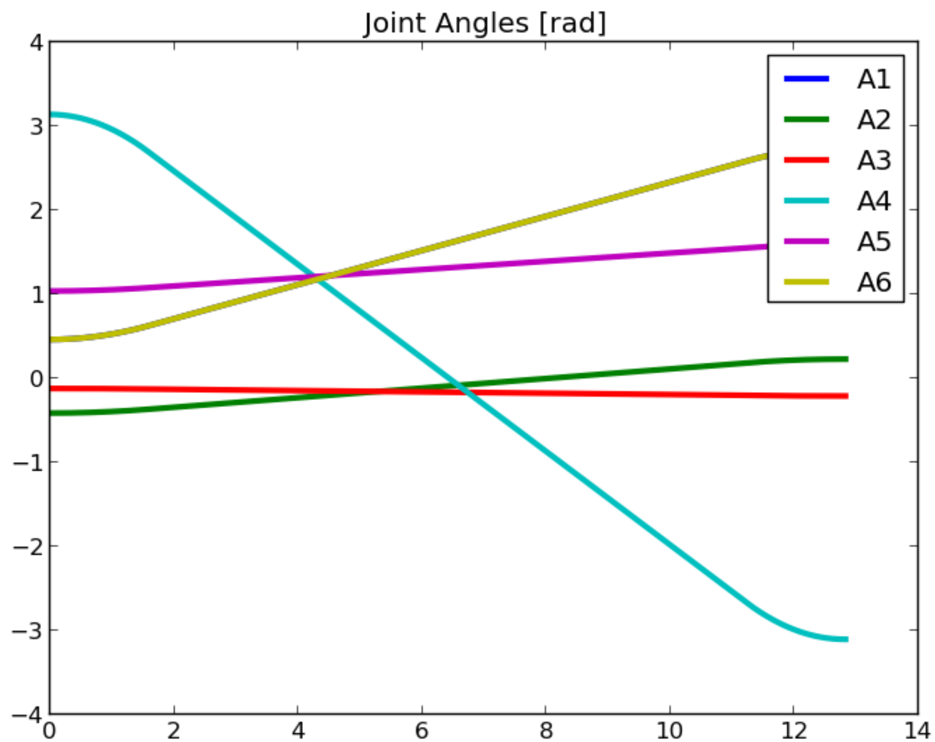
Check the constraints:

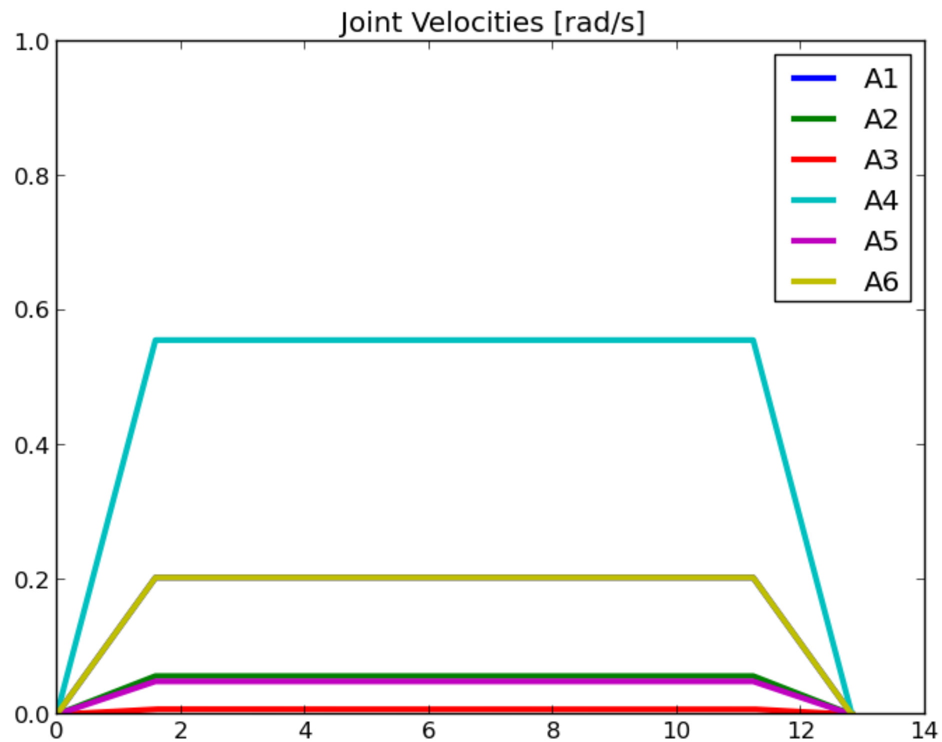
$$a^*_{\max}[j] \le a_{\max}[j]$$

$$v^*_{\max}[j] \le v_{\max}[j]$$

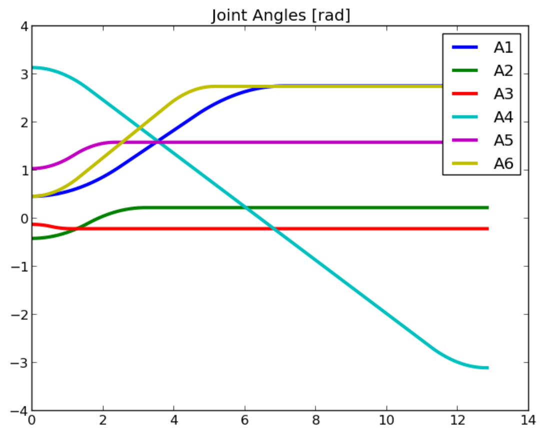
If these constraints are not fulfilled you need to expand the whole transformation time and do the calculations again and again.

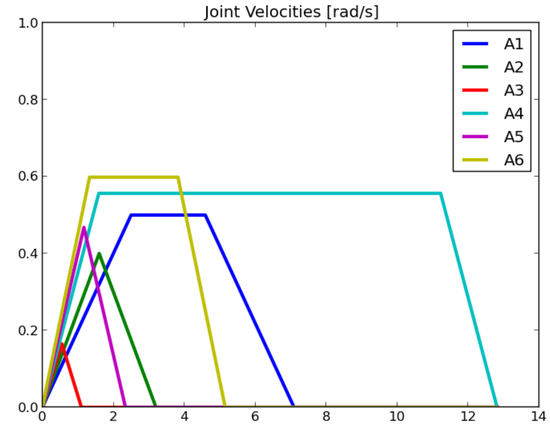
PTP–Control // Full synchronous movement



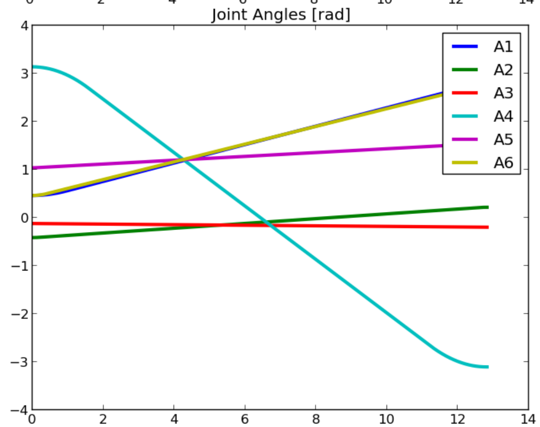


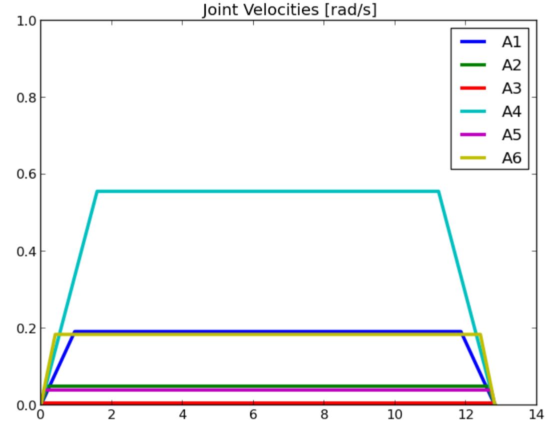
Asynchronous



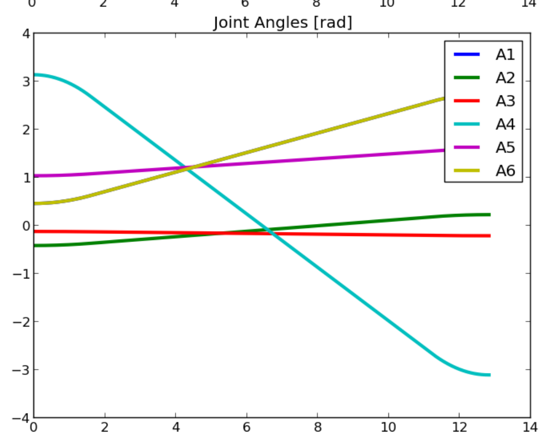


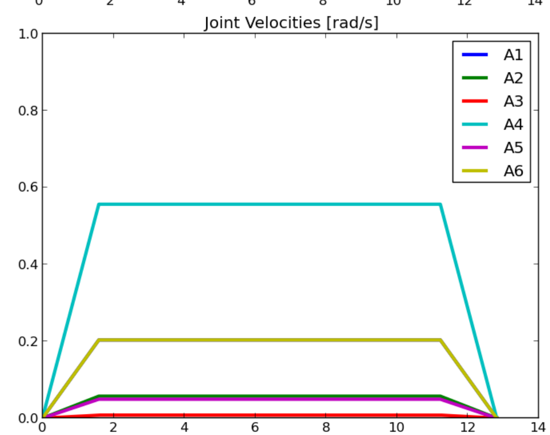
Synchronous





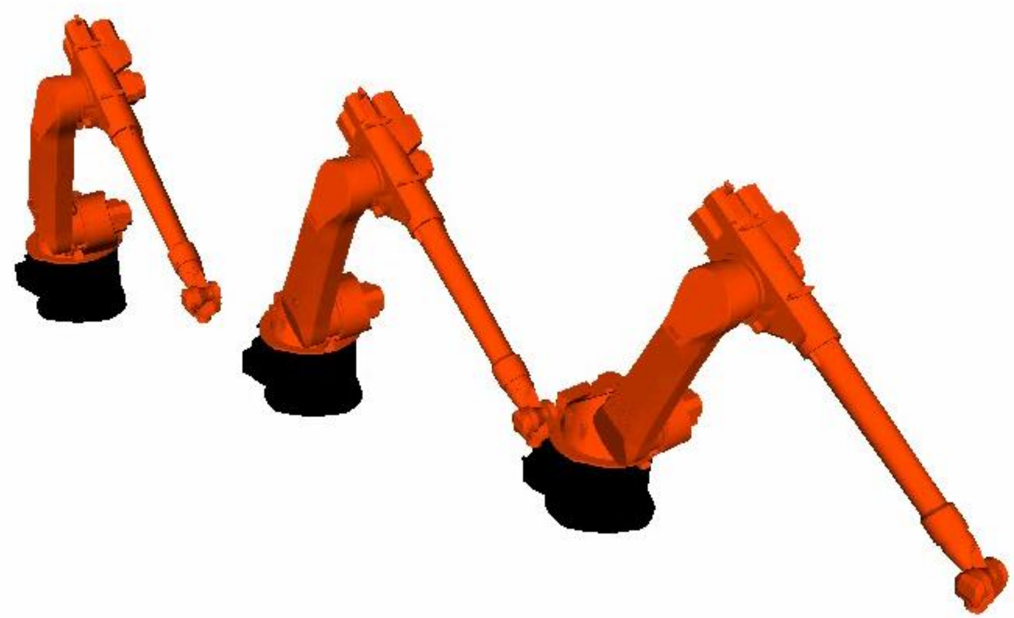
Full synchronous





OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG INF FAKULTÄT FÜR INFORMATIK

Video for all synchronization types



CP–Control

- Continuous Path
- Trajectory planning in Cartesian space
- Goal: Move the end effector along an exact path

- Process:
  - Compute interpolation points on the planned path in short time intervals in Cartesian space
  - Compute the inverse kinematics for every single interpolation point

- Usually needs more joint movements

  → longer movement time

CP-Control // Linear

Given:

- P_1 – start point of the trajectory [x,y,z]
- P_2 – end point of the trajectory [x,y,z]
- v_{max} – max velocity for the Endeffektor [m/s]
- a_{max} – max acceleration for the Endeffektor [m/s^2]
- frequency – calculations per second [1/s]

Find:

- trajectoryAngle[1..n_step] – angles of joints for each point in time

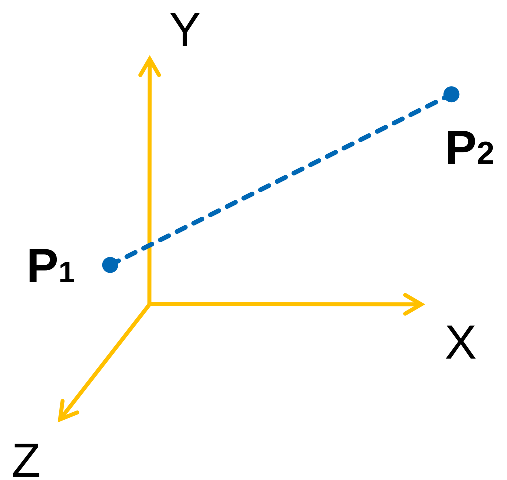
So that the robot changes its Endeffektor Position according to the velocity profile

CP–Control // Linear

Step 1

Compute length of the trajectory and type of the velocity profile

$$length = \sqrt{(p2_x - p1_x)^2 + (p2_y - p1_y)^2 + (p2_z - p1_z)^2}$$



$lengthAMax$ //length for the full acceleration

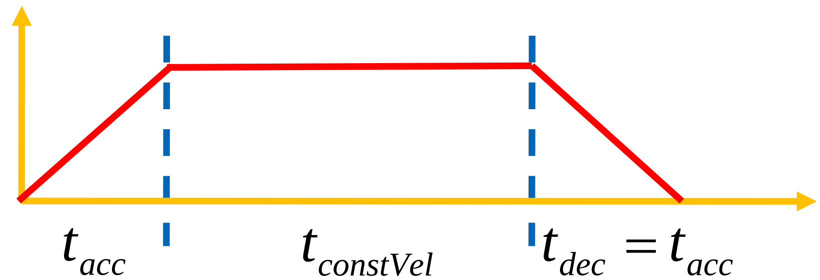
$$lengthAMax = \frac{vMax^2}{aMax}$$

$if \quad lengthAMax \geq length \quad then$ //robot can't achieve constant velocity phase

$$vMax = \sqrt{aMax \cdot length}$$ //reduce the max velocity

//get a triangle profile

CP–Control // Linear



Step 2
Find the overall time of the transformation

$$T = 2 \cdot t_{acc} + t_{constVel}$$ //overall time of the transformation

$$t_{acc} = \frac{vMax}{aMax}$$ //duration of the acceleration/deceleration phases

$$S_{acc} = \frac{aMax \cdot t_{acc}^2}{2} = \frac{vMax^2}{2 \cdot aMax}$$ //length of the acceleration/deceleration phases

$$S_{constVel} = length - 2 \cdot S_{acc}$$ //length of the constant velocity phase

$$t_{constVel} = \frac{S_{constVel}}{V_{\max}}$$ //duration of the constant velocity phase

CP–Control // Linear

Step 3 Get number of discrete calculation steps and their length

$$n\_step = T \cdot frequency$$ //number of discrete steps

$$time\_step = \frac{1}{frequency}$$ //length of each step

Step 4 Find a normal vector between start and finish points

$$\bar{n} = \begin{pmatrix} \frac{p2_x - p1_x}{length} \\ \frac{p2_y - p1_y}{length} \\ \frac{p2_z - p1_z}{length} \end{pmatrix}$$ //this vector defines the direction between p1 and p2

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG
INF FAKULTÄT FÜR INFORMATIK

CP–Control // Linear

Step 5 Find all points of the trajectory

$cur\_T = 0$

$for \quad i \quad in \quad [0, n\_steps]$ //go through all way points

$cur\_T = cur\_T + time\_step$

$if \quad (cur\_T \le t_{acc})$ //acceleration phase

$cur\_s = a_{\max} \cdot cur\_T^2 / 2$

$else \quad if \quad (cur\_T < t_{acc} + t_{constVel})$ //constant velocity phase

$cur\_s = s_{acc} + vMax \cdot dt_{constVel}$

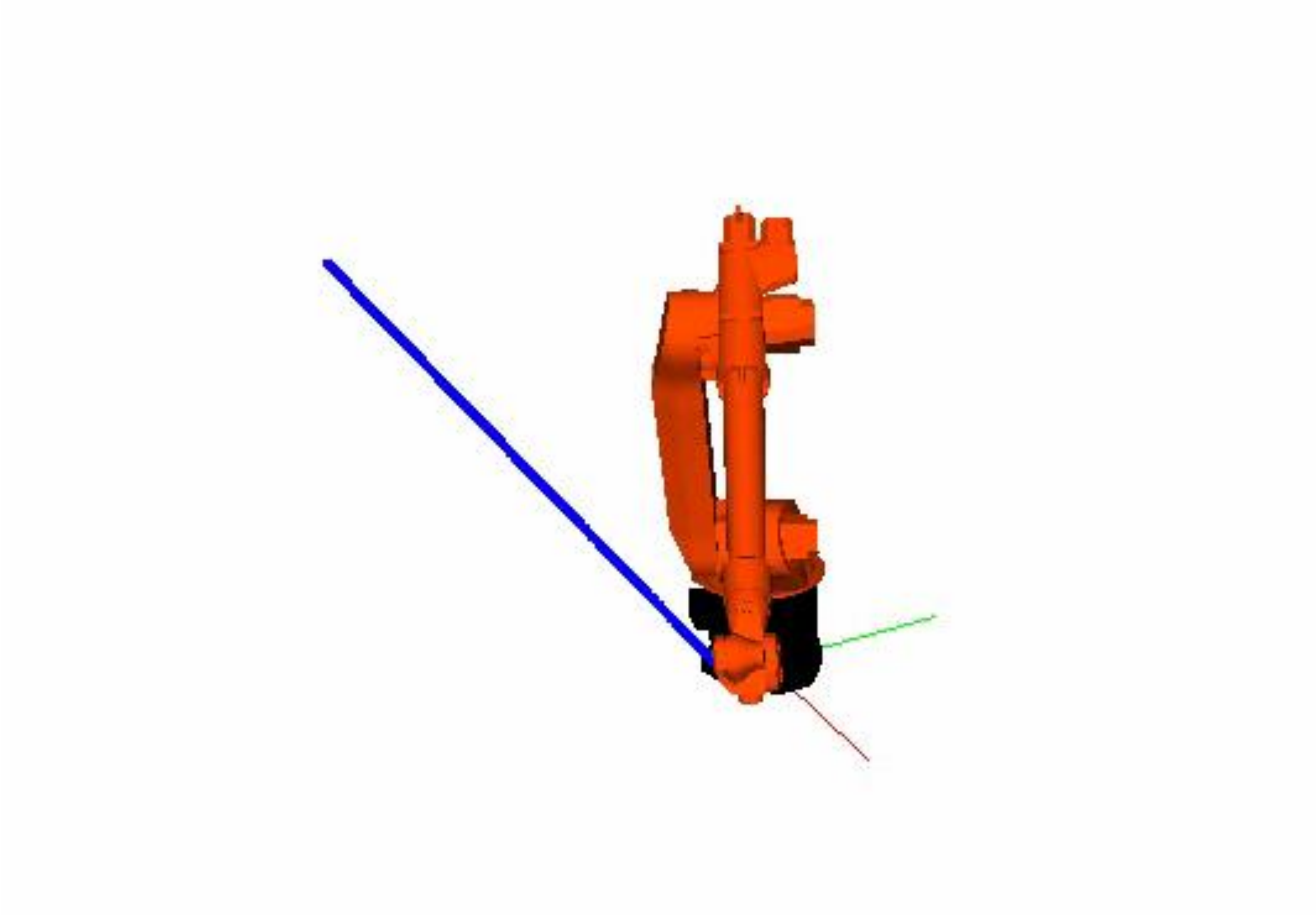
$else \quad if \quad (cur\_T < T)$ //deceleration phase

$cur\_s = s_{acc} + s_{constVel} + (vMax \cdot dt_{dec} - aMax \cdot dt_{dec}^2)/2$

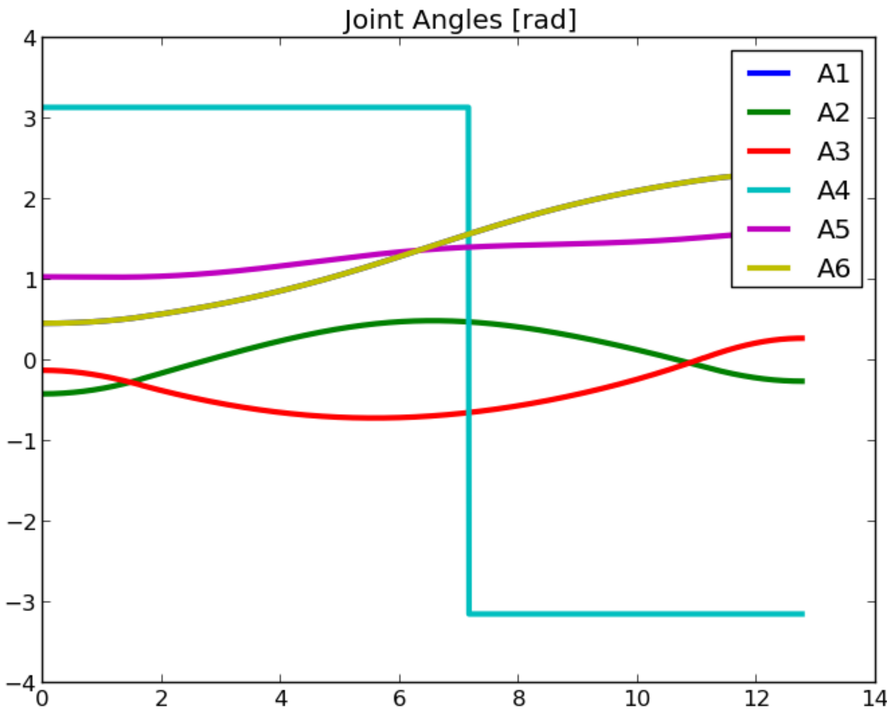
$cur\_p = p1 + \bar{n} \cdot cur\_s$ //get current point (x, y, z)

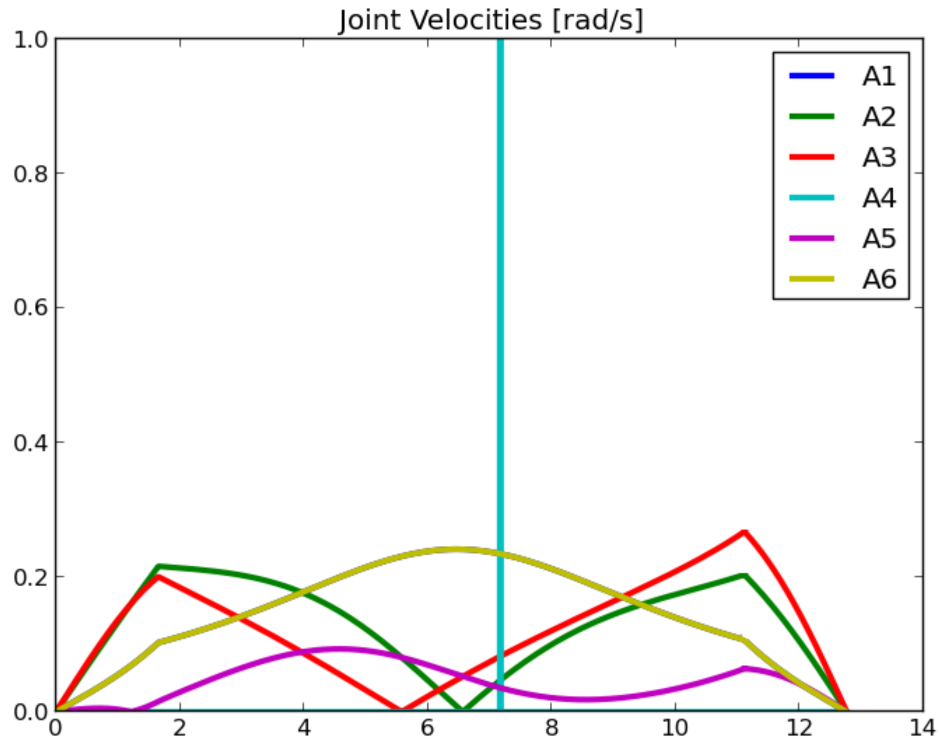
$trajectAngle[i] = getInverseKinematics(cur\_p)$ //get and save IK solution

Video of CP-Control // Linear



CP–Control // Linear





Advantages and Drawbacks

PTP-Control	CP-Control
Advantages	
• Easy for the controller • Good usage for positioning movements • Is possible to avoid singularities	• Exact movements with a tool possible • Obstacle avoidance
Drawbacks	
• No predictable path for end-effector • No exact movements with a tool or near obstacles possible	• Possibility of infinite solutions or singularities by computing the inverse Kinematics • Ambiguity

Problems and Potentials

The presented algorithms have some disadvantages:

- “Bang–Bang” velocity profiles
  - Better: smoother velocity profiles (i.e. $\sin^2$–profile)
- Linear paths
  - Better: Higher–order–Polynomials or Splines for trajectory planning
- Each planned point on the path is reached exactly
  - Better: Fly–by–points
- Behavior with obstacles is not considered
  - Better: Collision Detection

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG INF FAKULTÄT FÜR INFORMATIK

Sources

[Trajectory Planning for Automatic Machines and Robots; by Biagiotti, Melchiorri; Heidelberg, 2008]

[KUKA Arm Tutorials; by KUKA Roboter GmbH; http://www.kuka-robotics.com/en/products/software/educational_framework/arm_tutorials/ ; 24.11.2013]

[Robotics: Modelling, Planning and Control; by Siciliano, Sciavicco, Villani, Oriolo; London, 2009]

[Lecture «Robotics 1: Trajectory planning»; by De Luca; Rome, 2013]

[Lecture «Trajectory Planning for Robot Manipulators»; by Melchiorri; Bologna, 2013]

Thank you very much for your attention!