Trajectory Planning, Setpoint Generation and Feedforward Design for High Performance Motion Systems

Paul Lambrechts

Senior Application Engineer, The MathWorks Benelux.

September 27, 2006

Overview

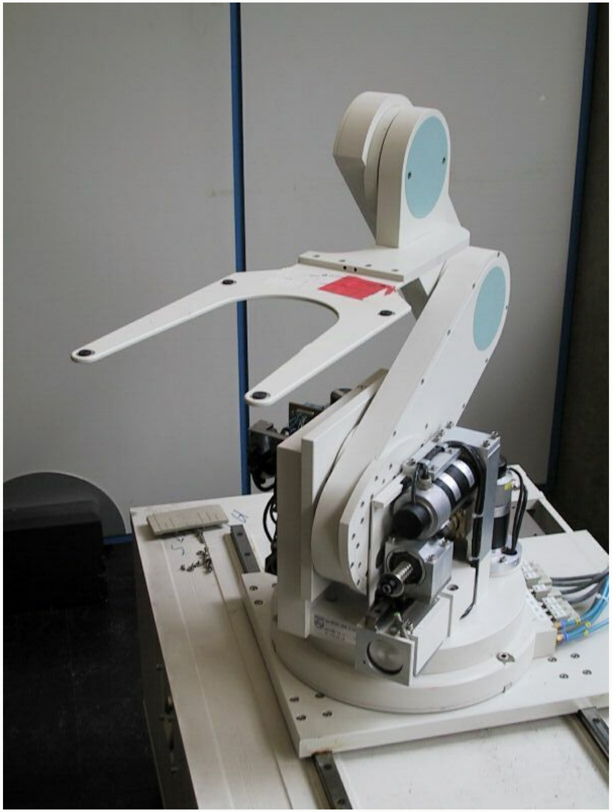
- (Industrial) motion control
  - Motion control and factory automation;
  - Current methods for feedforward control
  - Performance characteristics for trajectory planning
  - Rigid body feedforward and second order trajectory planning
- Fourth order feedforward and trajectory planning
- Implementation aspects
  - Switching times
  - Discrete time integration and synchronization
  - First order filter implementation
- Simulation and experimental results
- Conclusions

Industrial motion control

- *Robots*
- Pick-and-place units
- Wafersteppers

Motion Control Tasks:

- Safety, Communication, etc.
- System compensation
- Trajectory planning
- Feedforward control
- Feedback control



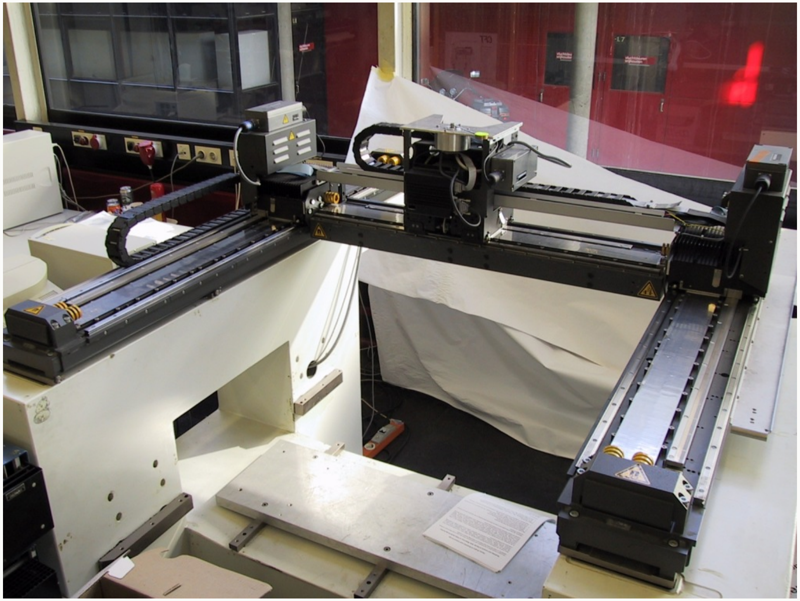
CRT Tube Handler

Industrial motion control

- Robots
- *Pick-and-place units*
- Wafersteppers

Motion Control Tasks:

- Safety, Communication, etc.
- System compensation
- Trajectory planning
- Feedforward control
- Feedback control



Component Mounter

Industrial motion control

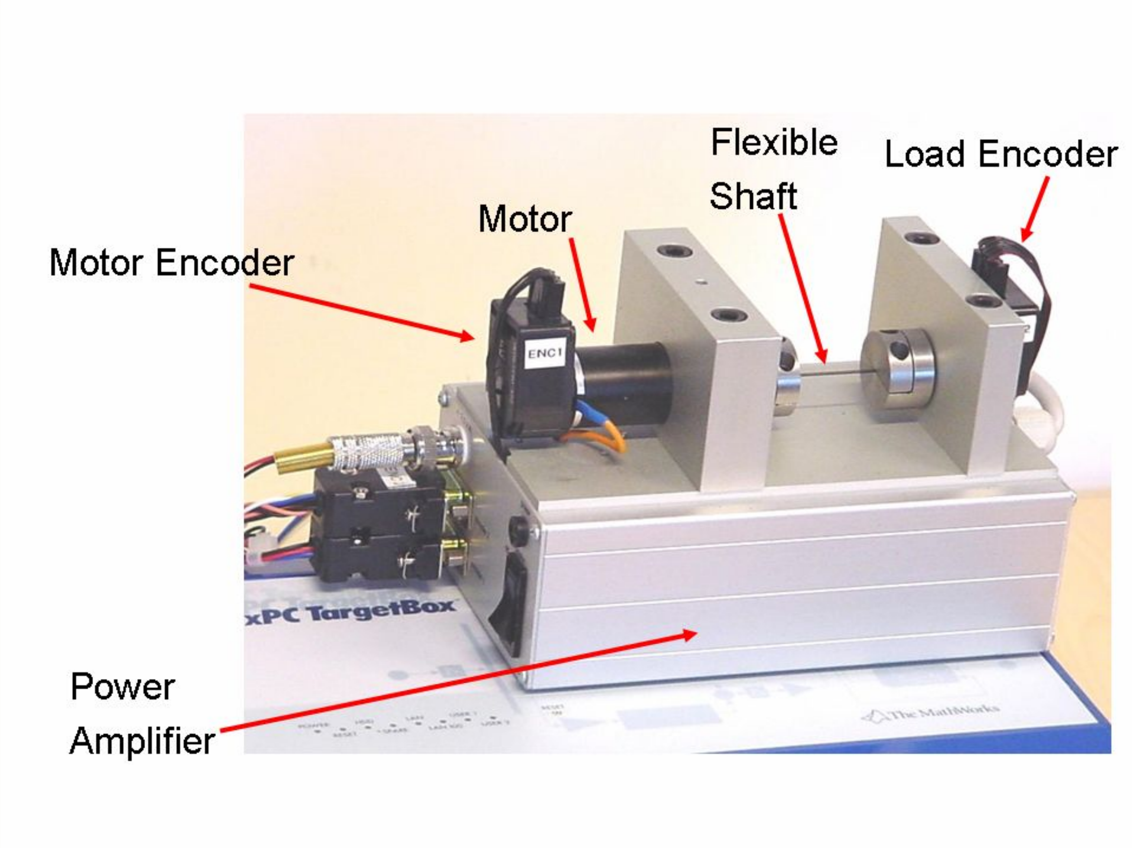
- Robots
- Pick-and-place units
- *Wafersteppers*

Motion Control Tasks:

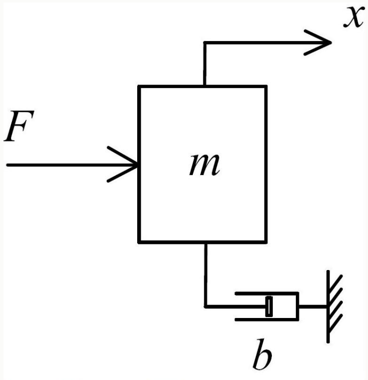
- Safety, Communication, etc.
- System compensation
- Trajectory planning
- Feedforward control
- Feedback control

300mm Waferstepper

Simple experimental setup:



Rigid-body feedforward

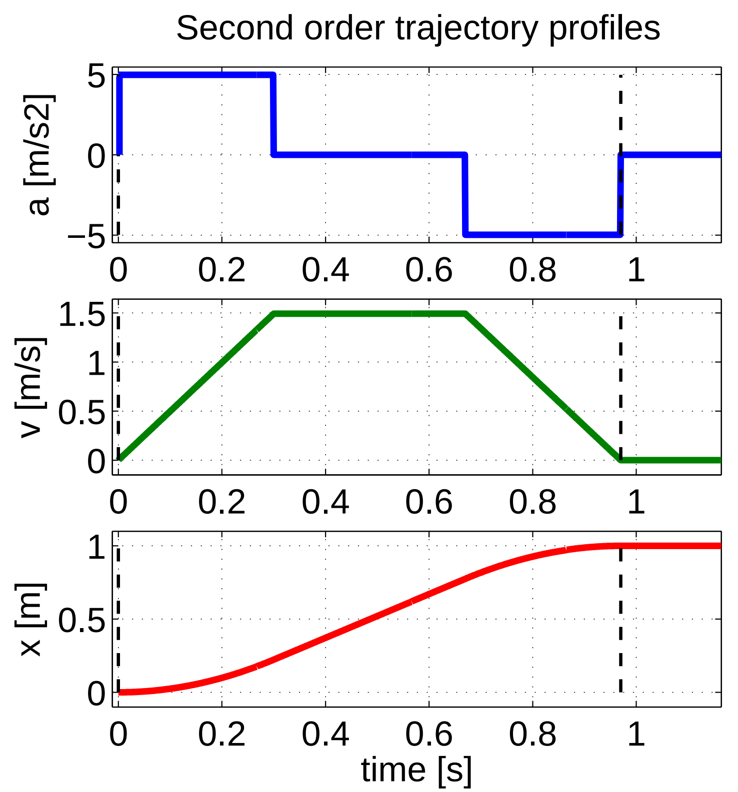


1 DOF rigid-body model

$$m\ddot{x} + b\dot{x} = F \;\Rightarrow\; F_{\mathrm{ff}} = ma + bv$$

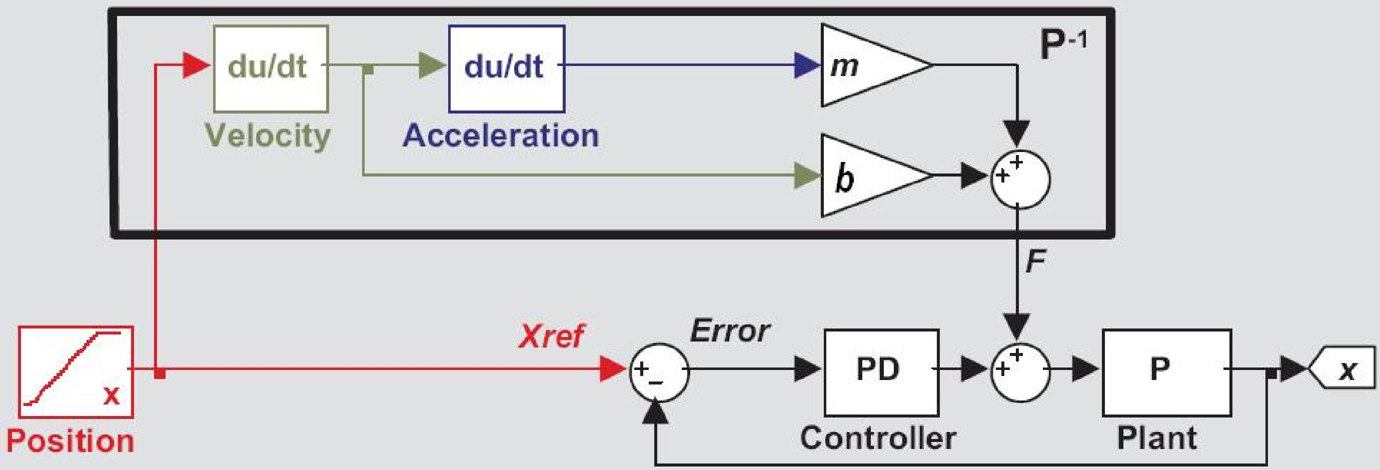
x is position, m is (equivalent) mass or inertia, b is viscous damping, F is actuator force and F_{ff} is feedforward force.

Trajectory profiles for rigid-body feedforward

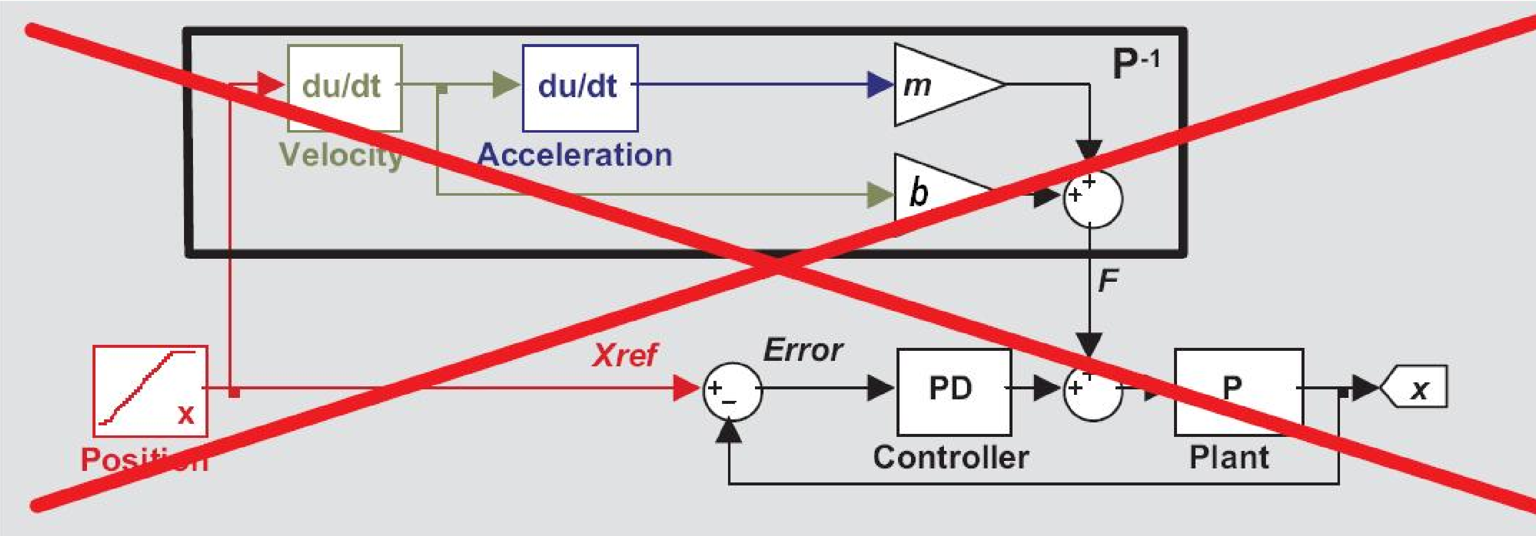


$$F_{\mathrm{ff}} = m{\color{blue}a} + b{\color{green}v}$$

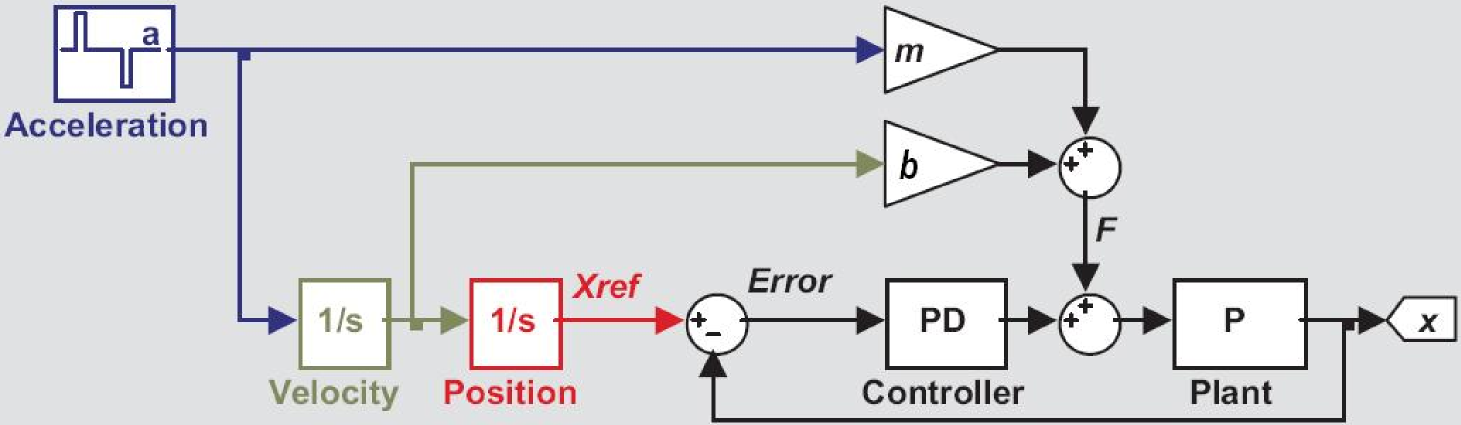
Rigid body feedforward: $F_{\text{ff}} = ma + bv$



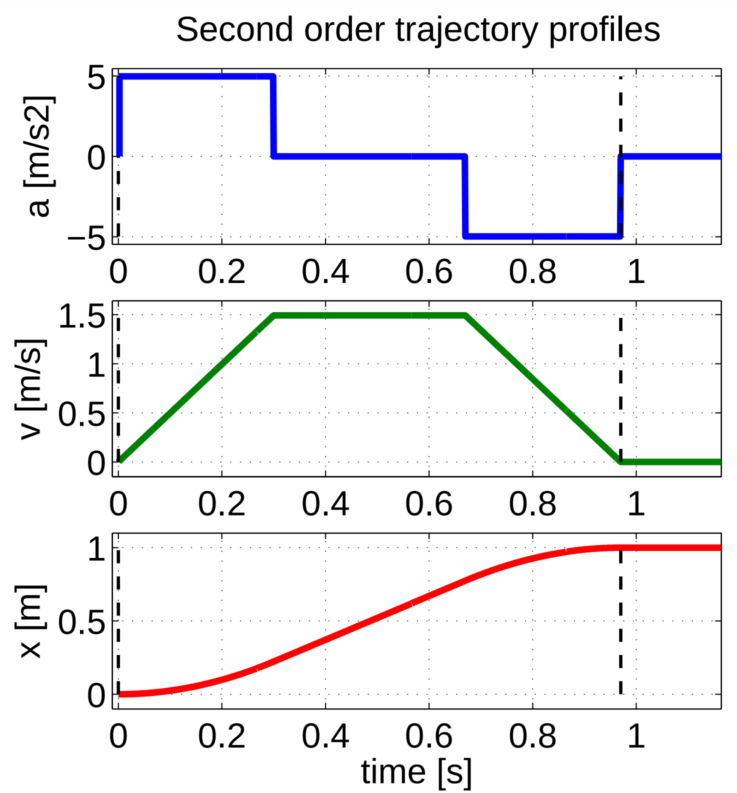
Rigid body feedforward: $F_{\text{ff}} = m{\color{blue}a} + b{\color{green}v}$



Rigid body feedforward: $F_{\mathrm{ff}} = ma + bv$



Trajectory planning performance (p2p)

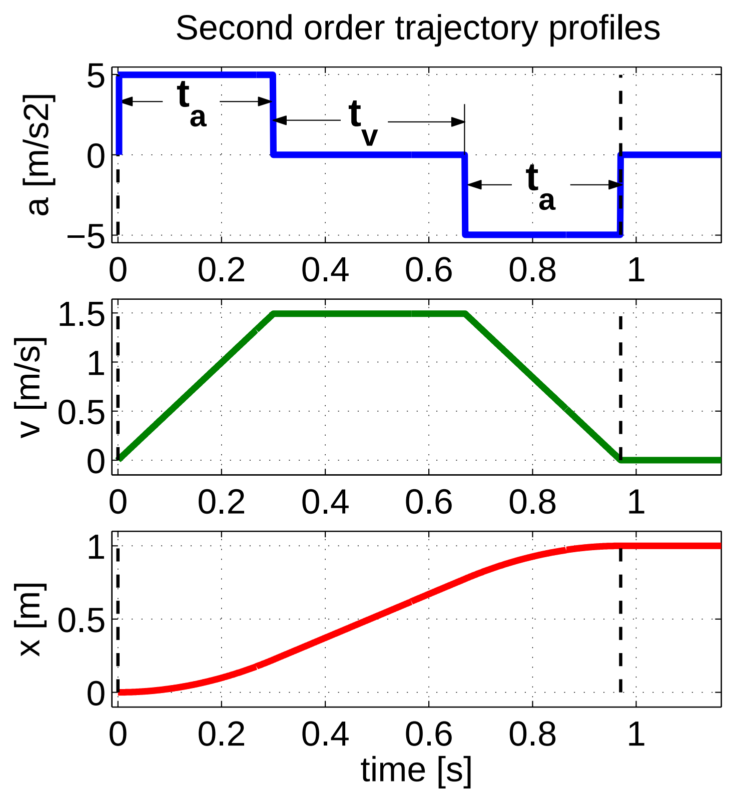


- Timing: minimal trajectory execution time
- Realizability: constrained dynamics ($\bar{a}$ and $\bar{v}$)
- Accuracy: trajectory ends at desired end position ($\bar{x}$)
- Complexity: calculation time
- Reliability: always valid solution
- Implementation: discretization, quantization

Profiles given $\bar{a}$, $\bar{v}$ and $\bar{x}$:

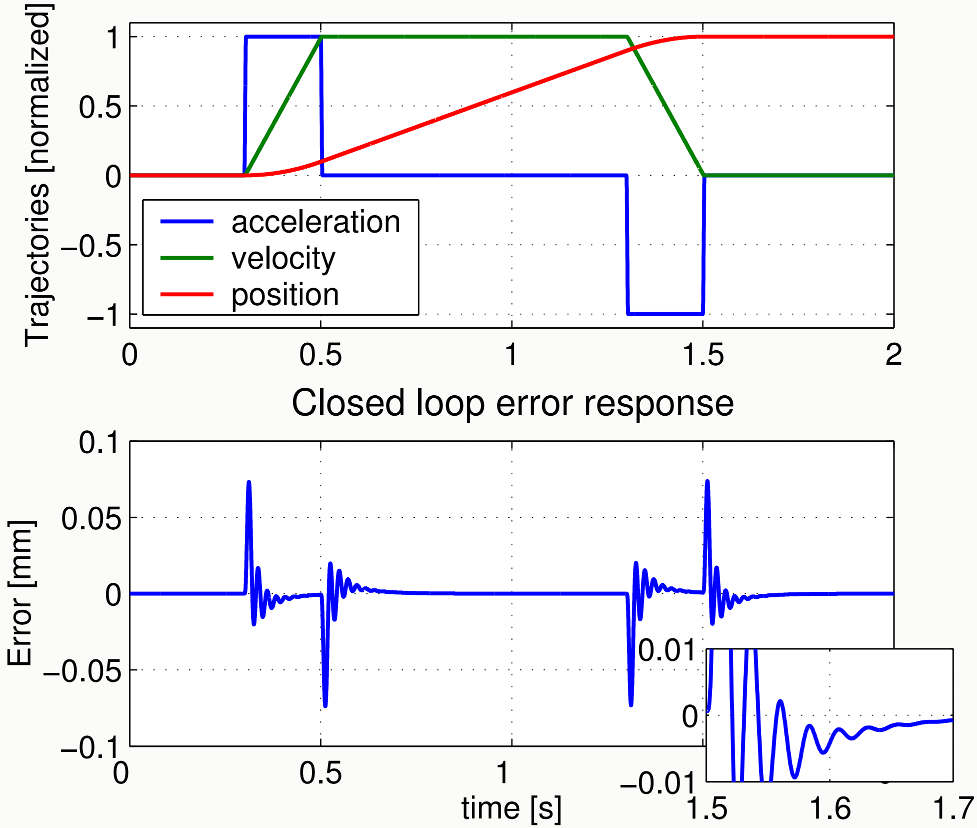
1. Forget $\bar{v}$: $\bar{x} = 2 \times \frac{1}{2}\bar{a}t^2 \Rightarrow t_{\bar{a}} = \sqrt{\frac{\bar{x}}{\bar{a}}} \Rightarrow t_{\bar{x}} = 2t_{\bar{a}}$
2. Calculate maximal velocity: $\hat{v} := \bar{a} \cdot t_{\bar{a}}$
3. $\hat{v} > \bar{v}$?; true: $t_{\bar{a}} = \frac{\bar{v}}{\bar{a}}$, false: $t_{\bar{a}} = \frac{1}{2}t_{\bar{x}}$
4. $x_{\bar{a}} := 2 \times \frac{1}{2}\bar{a}t_{\bar{a}}^2 \leq \bar{x}$
5. $t_{\bar{v}} = \frac{(\bar{x} - x_{\bar{a}})}{\bar{v}}$ $\Rightarrow \bar{a}, t_{\bar{a}}, t_{\bar{v}}$

Properties of trajectory planning algorithm



- Timing: minimal, determined by $t_{\bar{a}}$ and $t_{\bar{v}}$
- Realizablity: guaranteed by $\bar{a}$ and $\bar{v}$
- Accuracy: exact within machine accuracy
- Complexity: low
- Reliability: always valid solution
- Implementation: later

Results of rigid body feedforward:

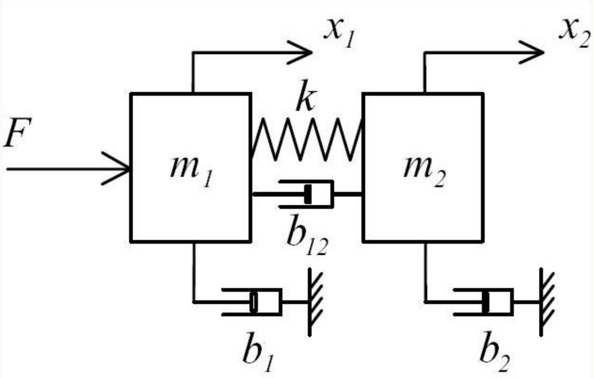


Extensions of rigid body feedforward

- Smoothing and shaping
  - third order trajectories with rigid body feedforward ?
  - filtering of second order trajectories and feedforwards ?
- (Approximate) model inversion
  - using second or third order trajectories ?
  - focus on frequency domain properties ?
  - learning techniques ?

→ back to basics !

4th order model for motion system



1 DOF 4th order model

x_1 and x_2 are actuator and load position, m_1, m_2 masses, b_1, b_2 viscous damping, k spring stiffness, b_{12} internal viscous damping, F is actuator force.

4th order feedforward

Equations of motion:

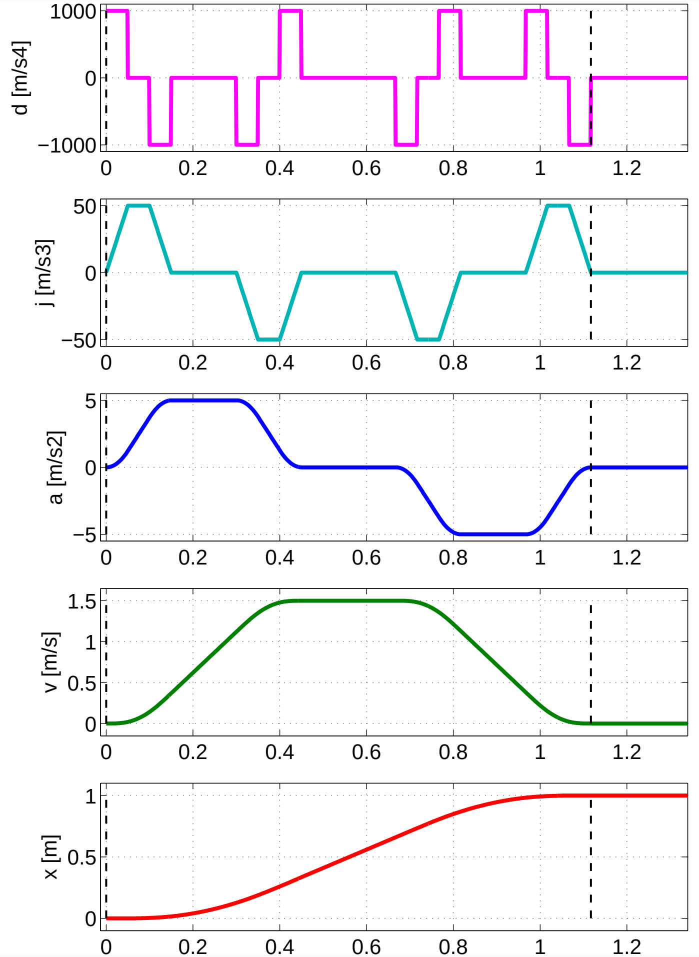
$$\begin{cases} m_1\ddot{x}_1 = -b_1\dot{x}_1 - k(x_1 - x_2) - b_{12}(\dot{x}_1 - \dot{x}_2) + F \\ m_2\ddot{x}_2 = -b_2\dot{x}_2 + k(x_1 - x_2) + b_{12}(\dot{x}_1 - \dot{x}_2) \end{cases}$$

Laplace transform and substitution:

$$F = \frac{q_1 s^4 + q_2 s^3 + q_3 s^2 + q_4 s}{b_{12}s + k} \cdot x_2 \quad \begin{cases} q_1 = m_1 m_2 \\ q_2 = (m_1 + m_2)b_{12} + m_1 b_2 + m_2 b_1 \\ q_3 = (m_1 + m_2)k + b_1 b_2 + (b_1 + b_2)b_{12} \\ q_4 = (b_1 + b_2)k \end{cases}$$

Feedforward force calculation:

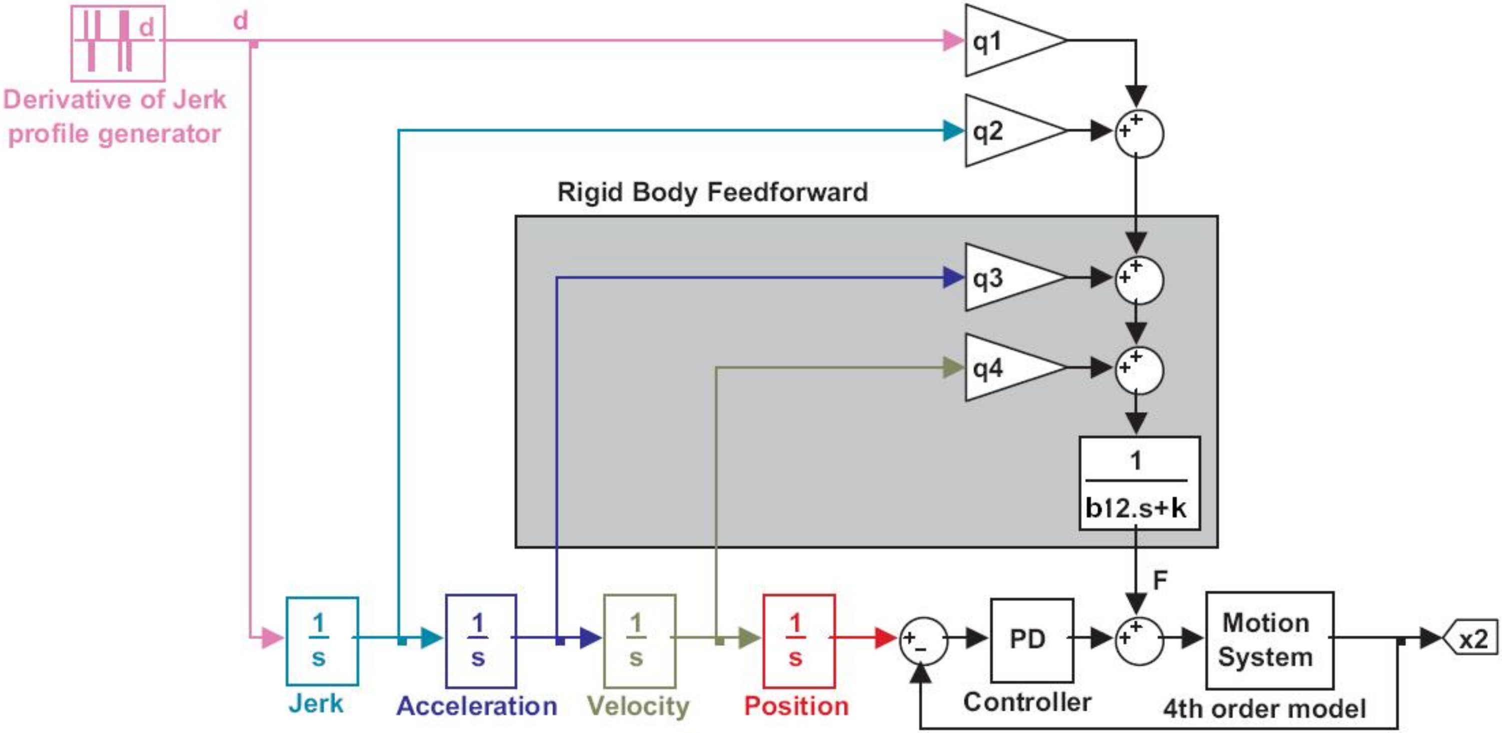
$$F_{\text{ff}} = \frac{1}{b_{12}s + k} \cdot \{q_1 d + q_2 \jmath + q_3 a + q_4 v\}$$



Trajectory profiles for 4th order feedforward

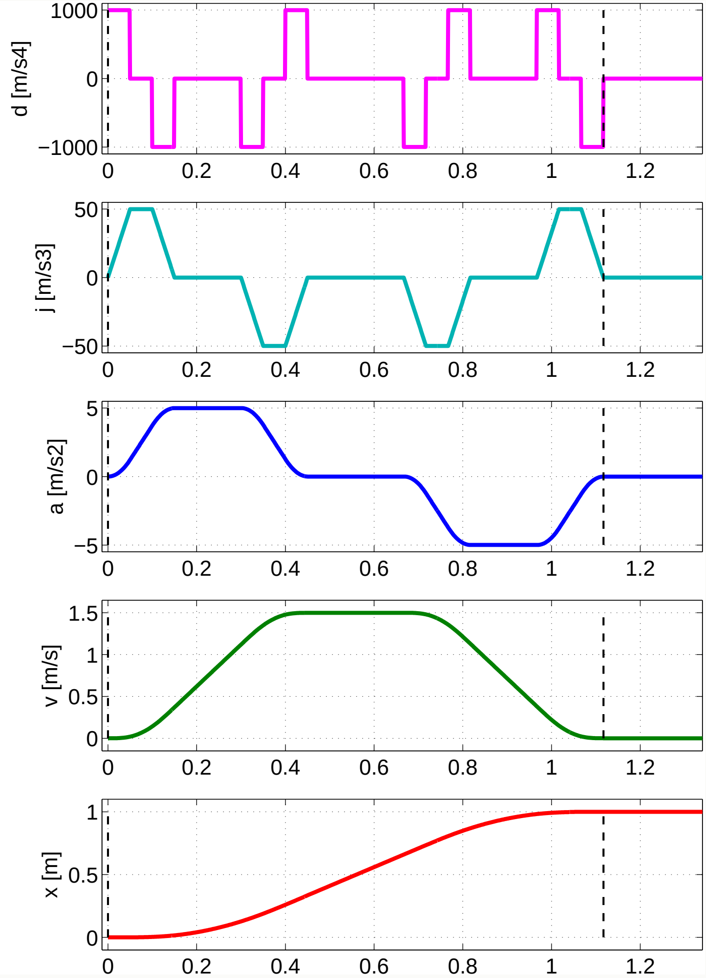
$$F_{\mathrm{ff}} = \frac{q_1 d + q_2 j + q_3 a + q_4 v}{b_{12} s + k}$$

4th order feedforward: $F_{\mathrm{ff}} = \dfrac{q_1 d + q_2 j + q_3 a + q_4 v}{b_{12} s + k}$

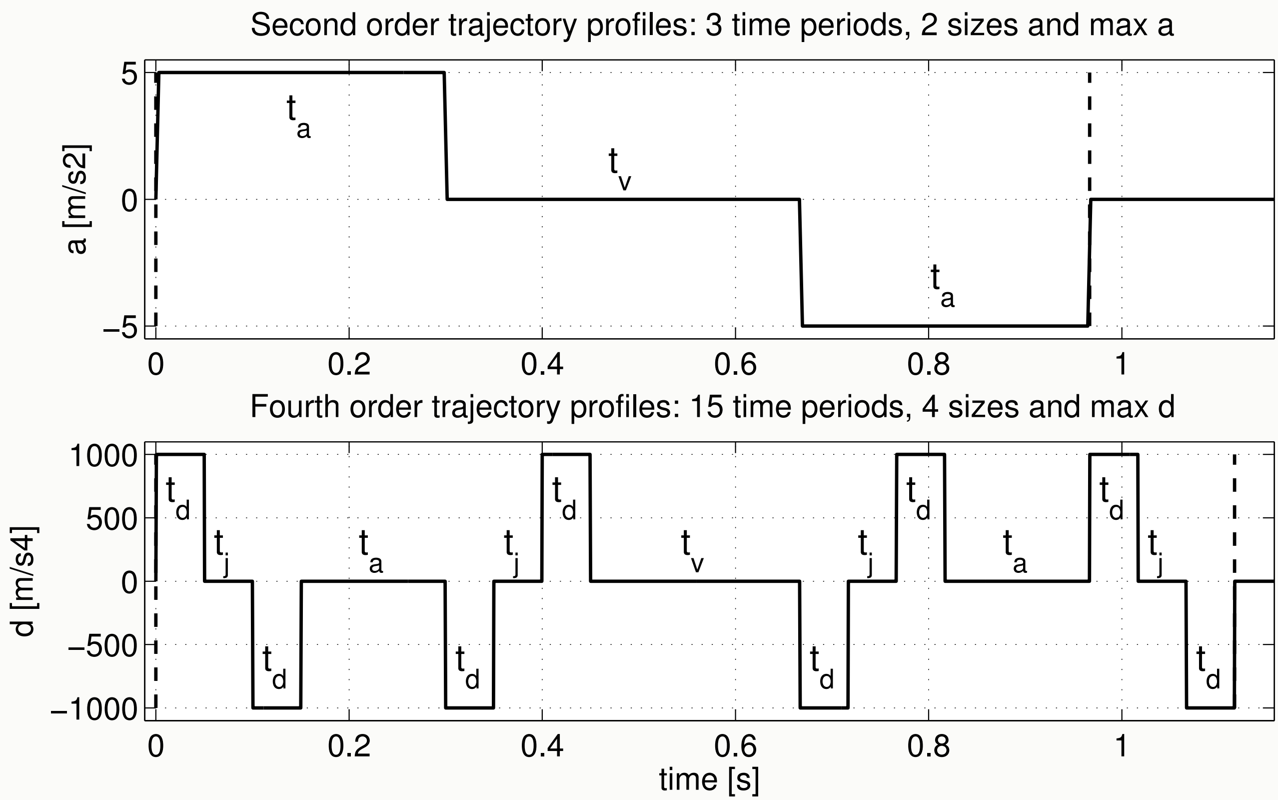


4th order trajectory planner ?

- Point-to-point move (all derivatives zero at start and end)
- Given: displacement $\bar{x}$ and bounds $\bar{d}$, $\bar{j}$, $\bar{a}$ and $\bar{v}$
- Performance criteria !?

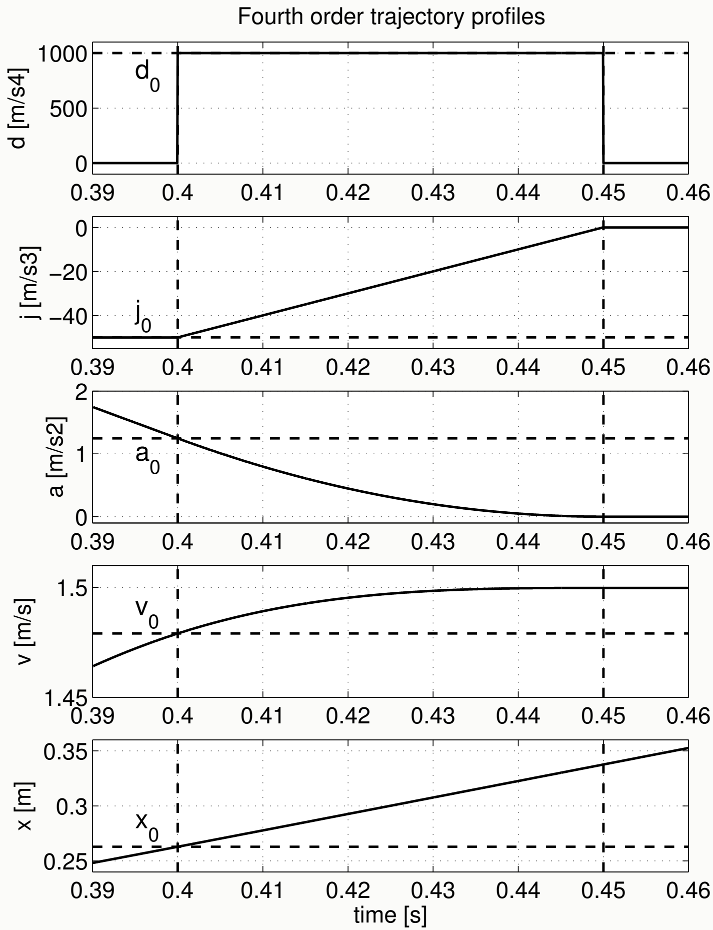


4th order trajectory specification:



4th order calculations

$$
\begin{aligned}
d(t) &= d_0 \\
\jmath(t) &= d_0 t + \jmath_0 \\
a(t) &= \tfrac{1}{2} d_0 t^2 + \jmath_0 t + a_0 \\
v(t) &= \tfrac{1}{6} d_0 t^3 + \tfrac{1}{2} \jmath_0 t^2 + a_0 t + v_0 \\
x(t) &= \tfrac{1}{24} d_0 t^4 + \tfrac{1}{6} \jmath_0 t^3 + a_0 t^2 \\
&\quad + v_0 t + x_0
\end{aligned}
$$



4th order planning calculate $t_{\bar{d}}$

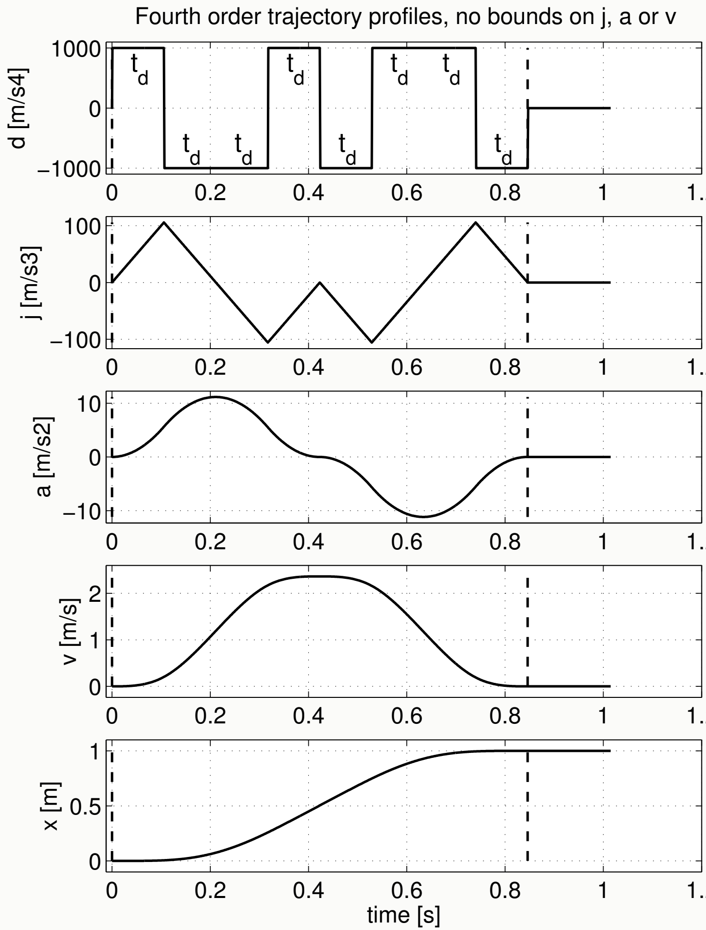
$t_{\bar{d}}$ only depends on $\bar{d}$ and $\bar{x}$

$$\Rightarrow \qquad t_{\bar{d}} = \sqrt[4]{\frac{\bar{x}}{8\bar{d}}}$$

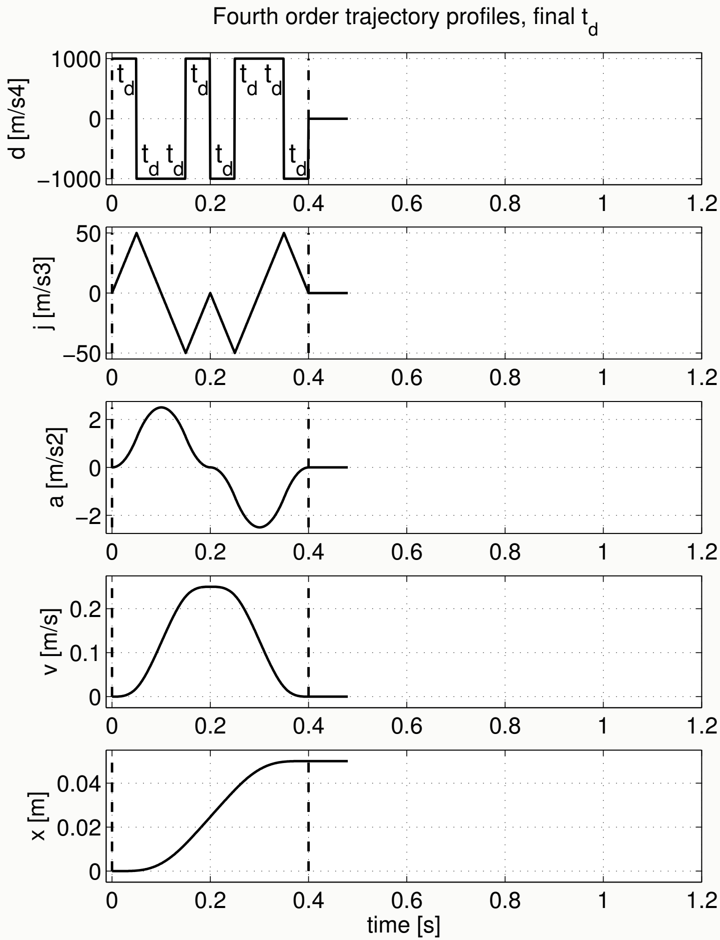
$\bar{v}$ violated: $$t_{\bar{d}} = \sqrt[3]{\frac{\bar{v}}{2\bar{d}}}$$

$\bar{a}$ violated: $$t_{\bar{d}} = \sqrt{\frac{\bar{a}}{\bar{d}}}$$

$\bar{j}$ violated: $$t_{\bar{d}} = \frac{\bar{j}}{\bar{d}}$$



4th order planning final $t_{\bar{d}}$



Note:

- No bounds violated
- Final $t_{\bar{d}}$ always $\leq$ first $t_{\bar{d}}$
- Consequently: $\bar{x}$ not reached

4th order planning
calculate $t_{\bar{\jmath}}$

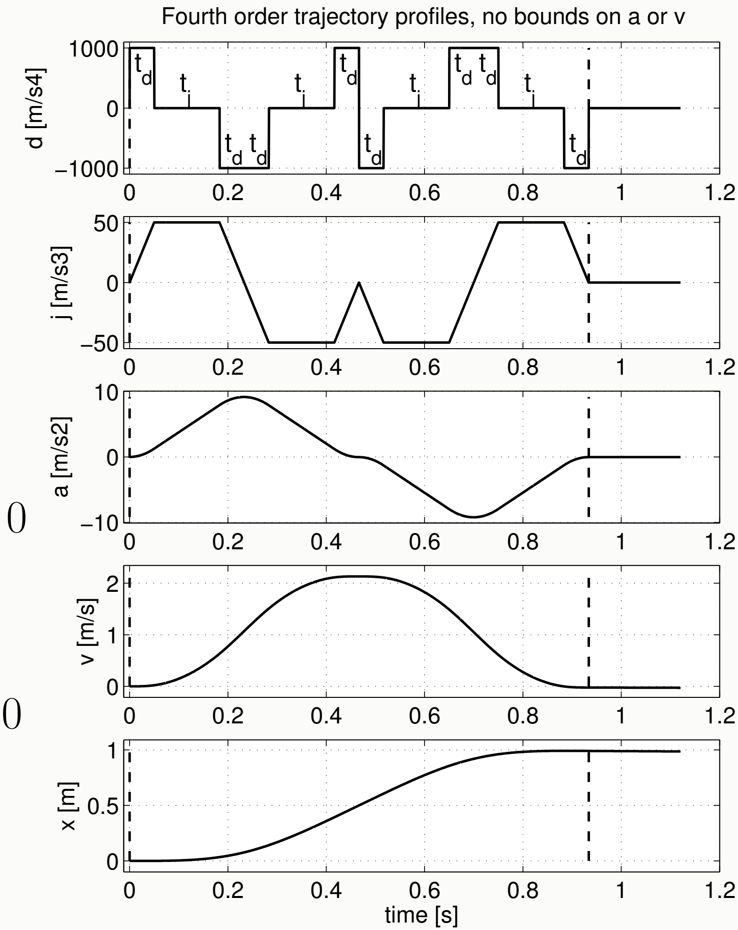
Add periods of constant jerk until $\bar{x}$ is reached again

$t_{\bar{\jmath}}$ follows from:

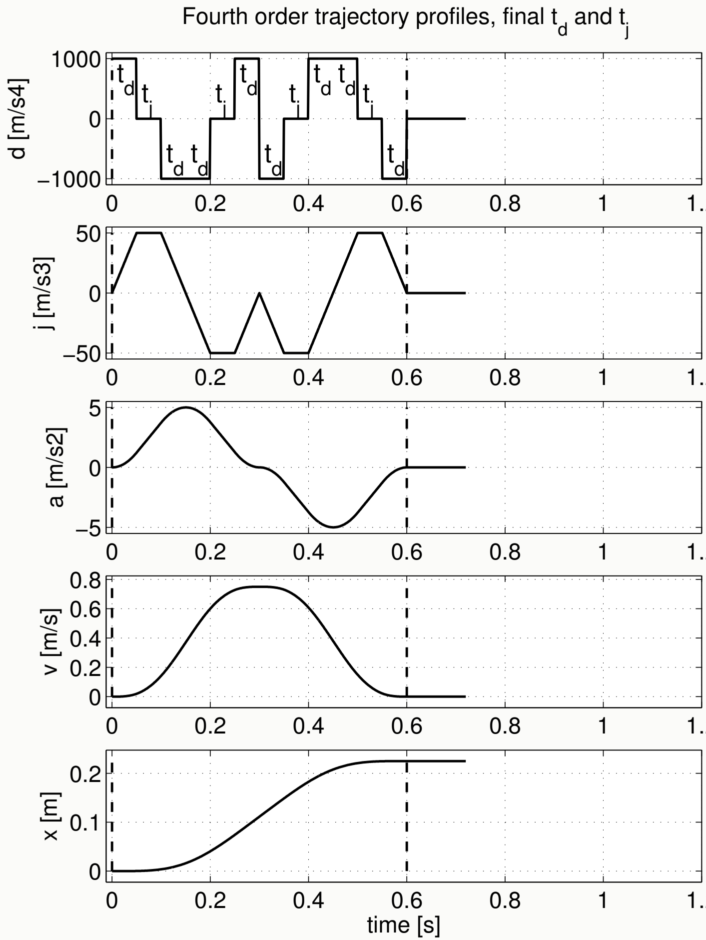
$$t_{\bar{\jmath}}^3 + (5t_{\bar{d}})t_{\bar{\jmath}}^2 + (8t_{\bar{d}}^2)t_{\bar{\jmath}} + (4t_{\bar{d}}^3 - \frac{\bar{x}}{2\bar{d}t_{\bar{d}}}) = 0$$

$\bar{v}$ violated: $\quad t_{\bar{\jmath}}^2 + 3t_{\bar{d}}t_{\bar{\jmath}} + 2t_{\bar{d}}^2 - \frac{\bar{v}}{\bar{d}t_{\bar{d}}} = 0$

$\bar{a}$ violated: $\quad t_{\bar{\jmath}} = \frac{\bar{a}}{\bar{\jmath}} - t_{\bar{d}}$



4th order planning final $t_{\bar{d}}$ and $t_{\bar{j}}$



Note:

- No bounds violated
- Final $t_{\bar{j}}$ always $\leq$ first $t_{\bar{j}}$
- Consequently: $\bar{x}$ not reached

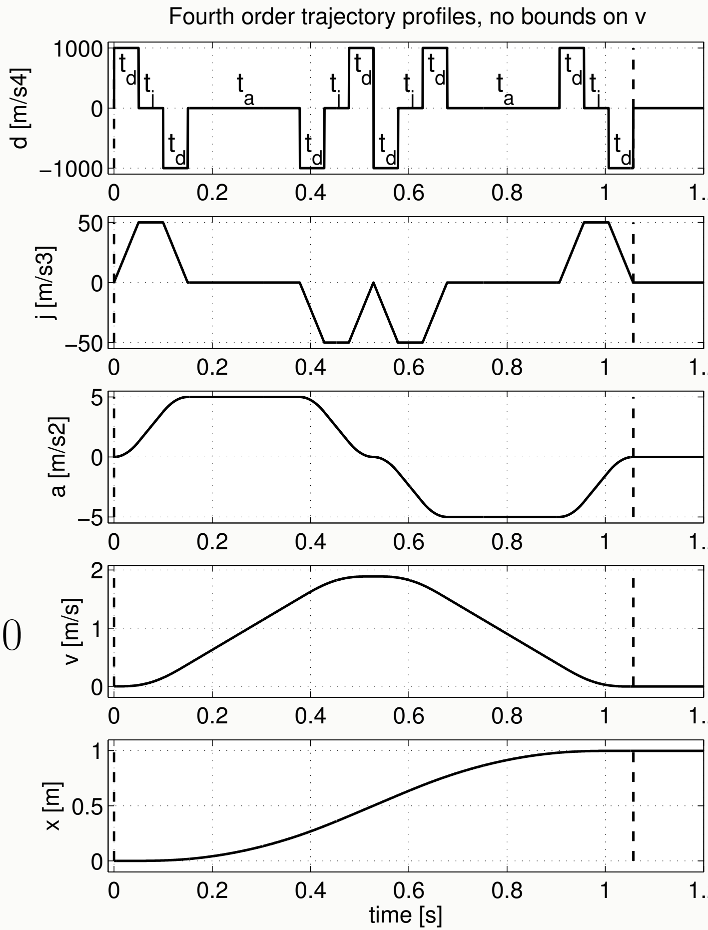
4th order planning calculate $t_{\bar{a}}$

Add periods of constant acceleration until $\bar{x}$ reached

$t_{\bar{a}}$ follows from:

$$\{t_{\bar{d}}^2 + t_{\bar{d}} t_{\bar{j}}\}\, t_{\bar{a}}^2 +$$
$$\{6t_{\bar{d}}^3 + 9t_{\bar{d}}^2 t_{\bar{j}} + 3t_{\bar{d}} t_{\bar{j}}^2\}\, t_{\bar{a}} +$$
$$\{8t_{\bar{d}}^4 + 16t_{\bar{d}}^3 t_{\bar{j}} + 10t_{\bar{d}}^2 t_{\bar{j}}^2 + 2t_{\bar{d}} t_{\bar{j}}^3 - \tfrac{\bar{x}}{\bar{d}}\} = 0$$

$\bar{v}$ violated: $t_{\bar{a}} = \frac{\bar{v} - 2\bar{d}t_{\bar{d}}^3 - 3\bar{d}t_{\bar{d}}^2 t_{\bar{j}} - \bar{d}t_{\bar{d}} t_{\bar{j}}^2}{\bar{d}t_{\bar{d}}^2 + \bar{d}t_{\bar{d}} t_{\bar{j}}}$

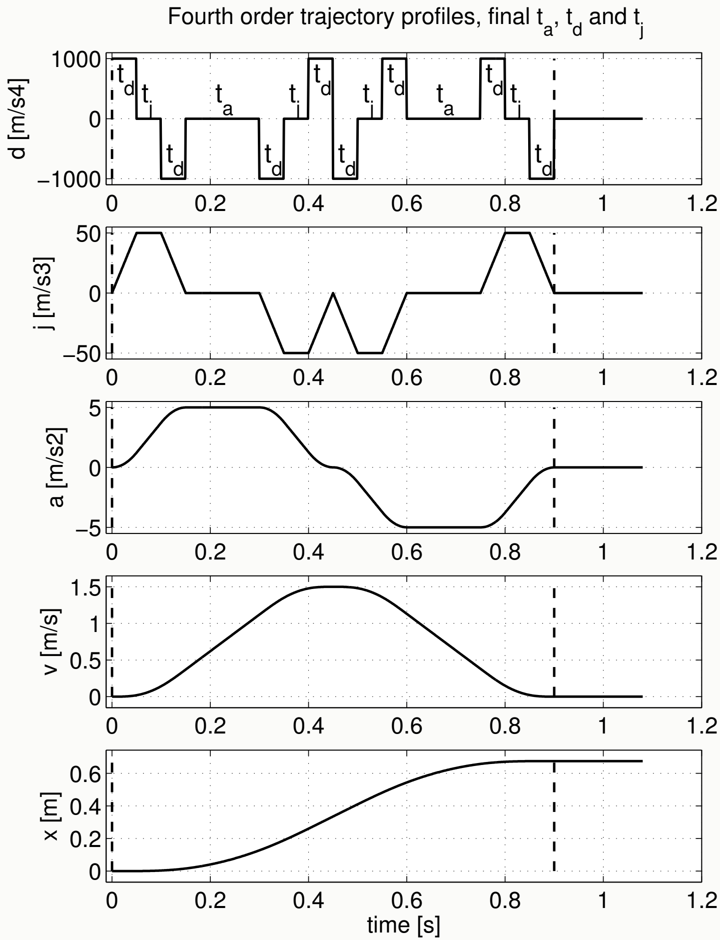


4th order planning final $t_{\bar{d}}$, $t_{\bar{j}}$ and $t_{\bar{a}}$

Note:

- No bounds violated
- Final $t_{\bar{a}}$ always $\leq$ first $t_{\bar{a}}$
- Consequently: $\bar{x}$ not reached

For final step:
determine obtained position $x_{\bar{a}}$

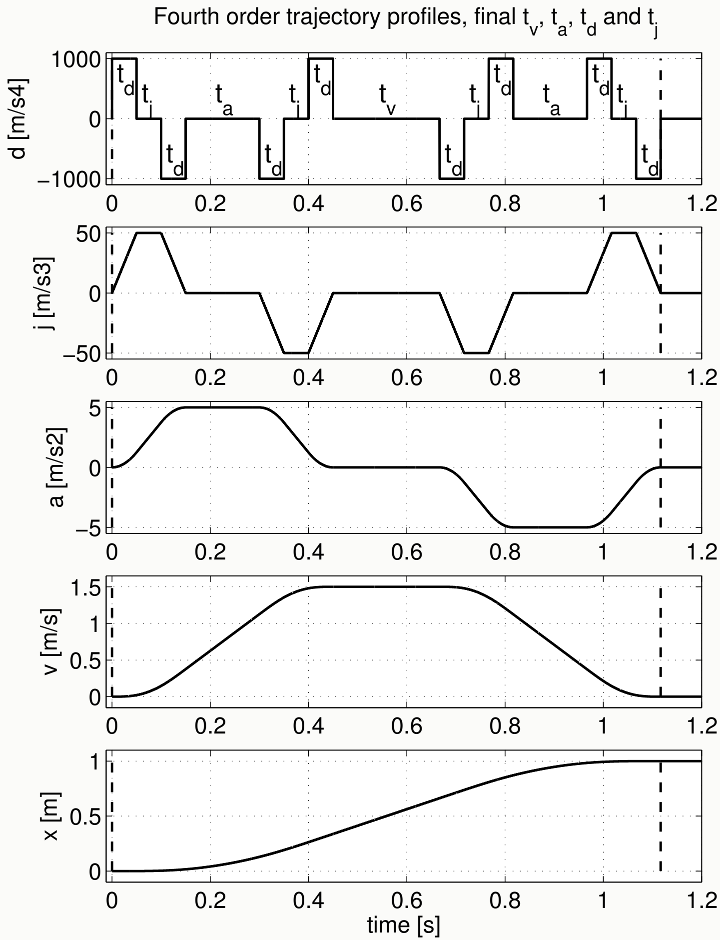


4th order planning calculate $t_{\bar{v}}$

Final step: add period of constant velocity until $\bar{x}$ reached

$$t_{\bar{v}} = \frac{\bar{x} - x_{\bar{a}}}{\bar{v}}$$

Finished: trajectory completely determined by 5 parameters !



(Further) Implementation aspects

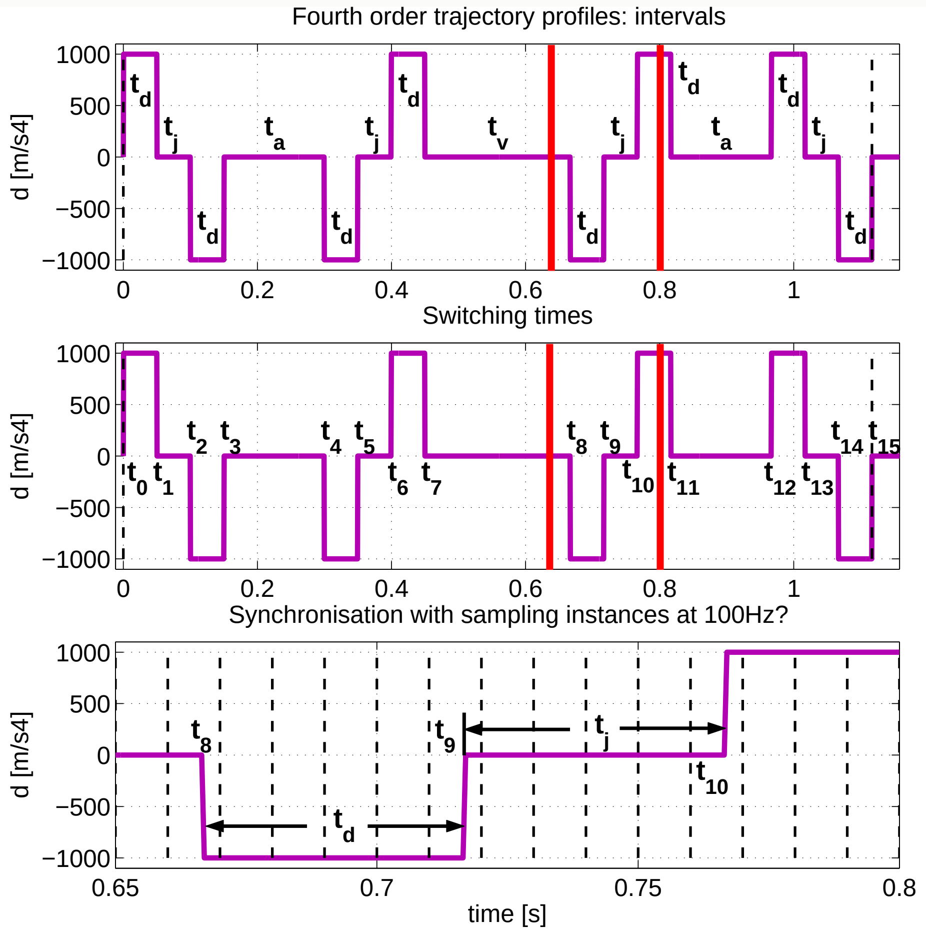
- Switching times
- Discrete time integration and synchronization
- First order filter implementation

Implementation

Switching times:

Round off intervals up to multiple of sampling time T_s.

Correct by reducing $\bar{d}$ to appropriate value.



Switching times synchronization

Make sure that each interval is a multiple of the sampling time T_s

Example:

$$t_{\bar{d}} = \sqrt[4]{\frac{\bar{x}}{8\bar{d}}}$$

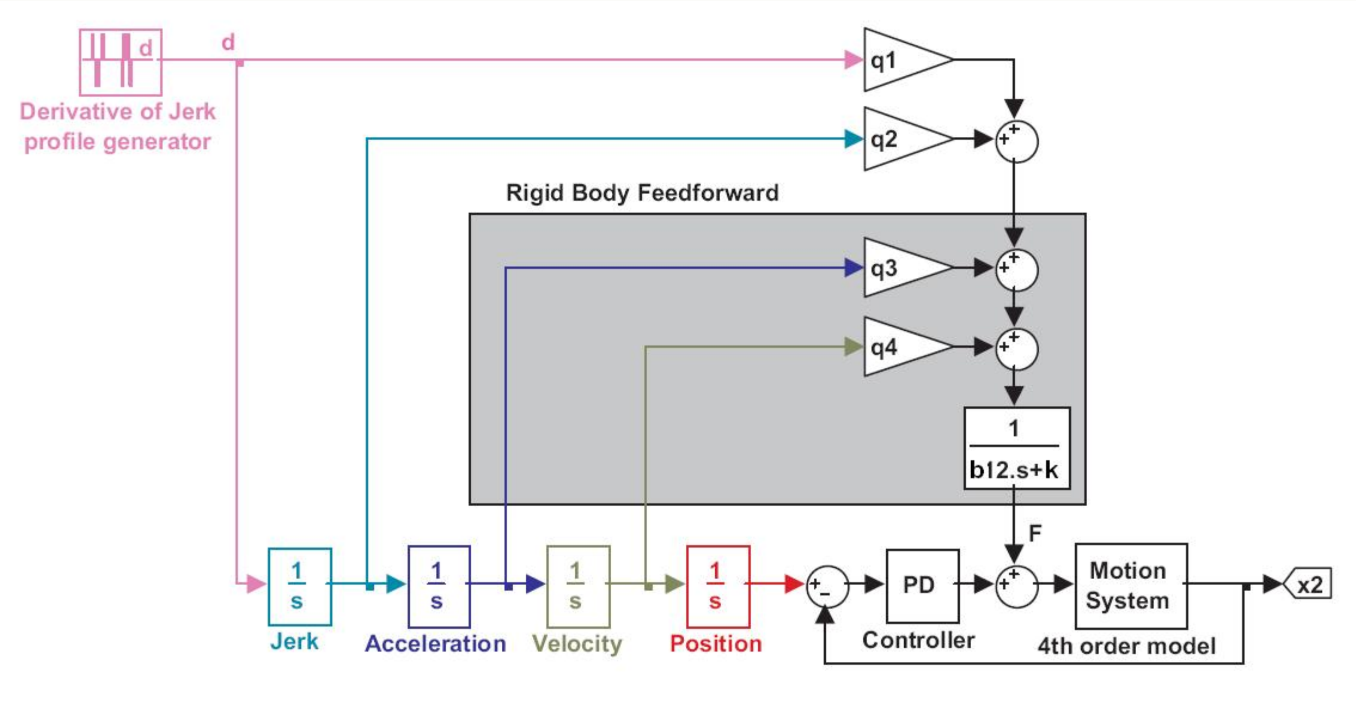
$$t'_{\bar{d}} = \text{ ceil}\left(\frac{t_{\bar{d}}}{T_s}\right) \times T_s$$

Correct $\bar{d}$

$$\bar{d}' = \frac{\bar{x}}{8{t'_{\bar{d}}}^4}$$

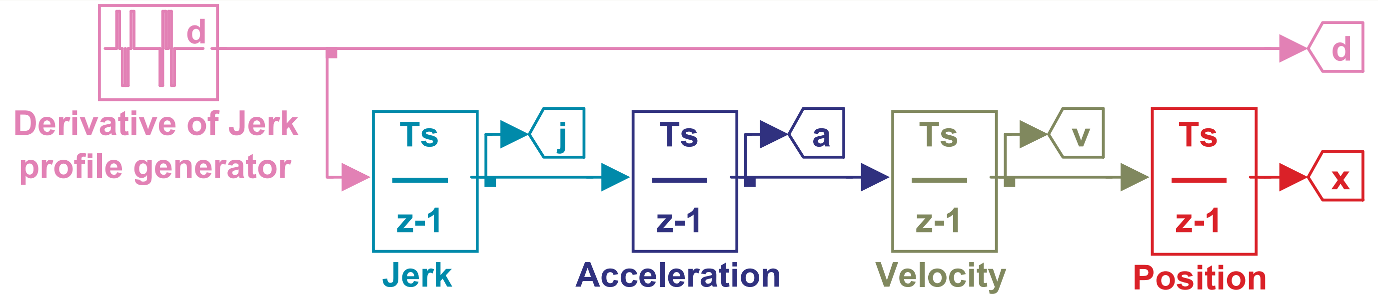
Note that with $t'_{\bar{d}} \geq t_{\bar{d}}$ we must have $\bar{d}' \leq \bar{d}$

4th order feedforward in discrete time ?



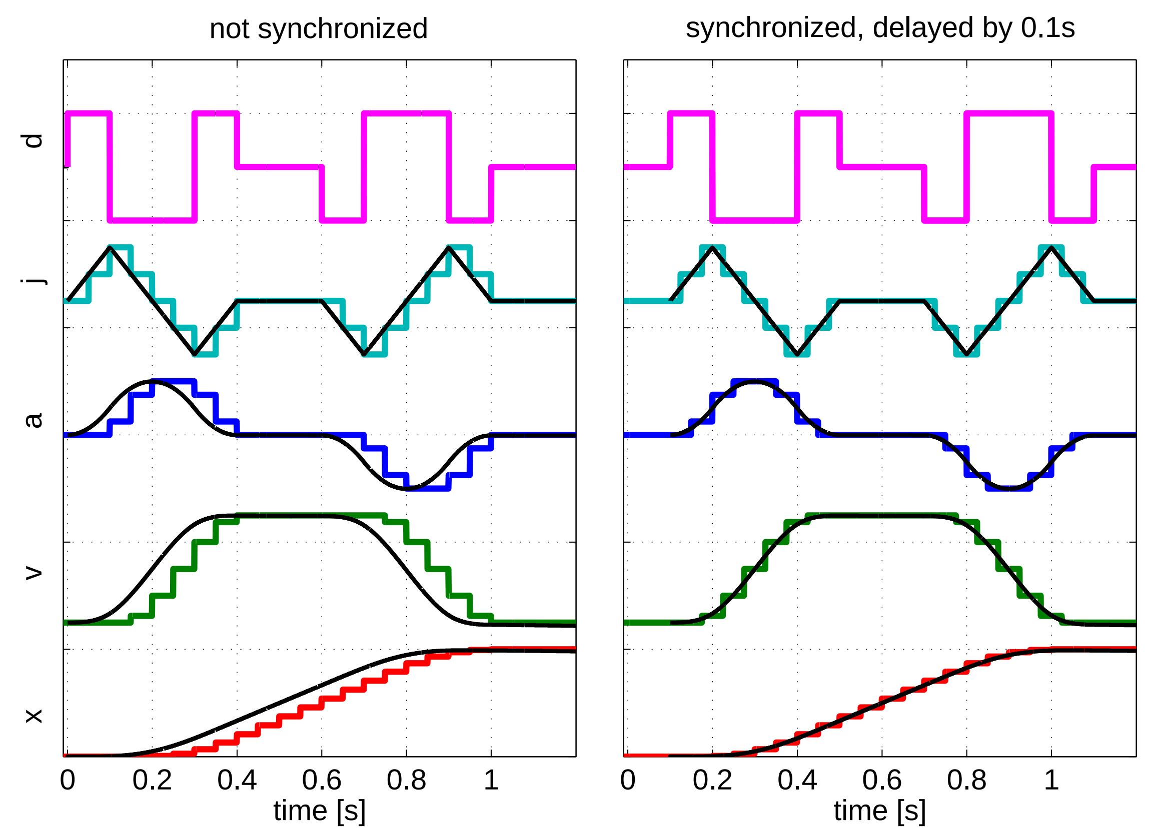
Implementation

Discrete time integration:



Synchronization of profiles is required !

Synchronization of profiles



Delay:

d with $2T_s$,

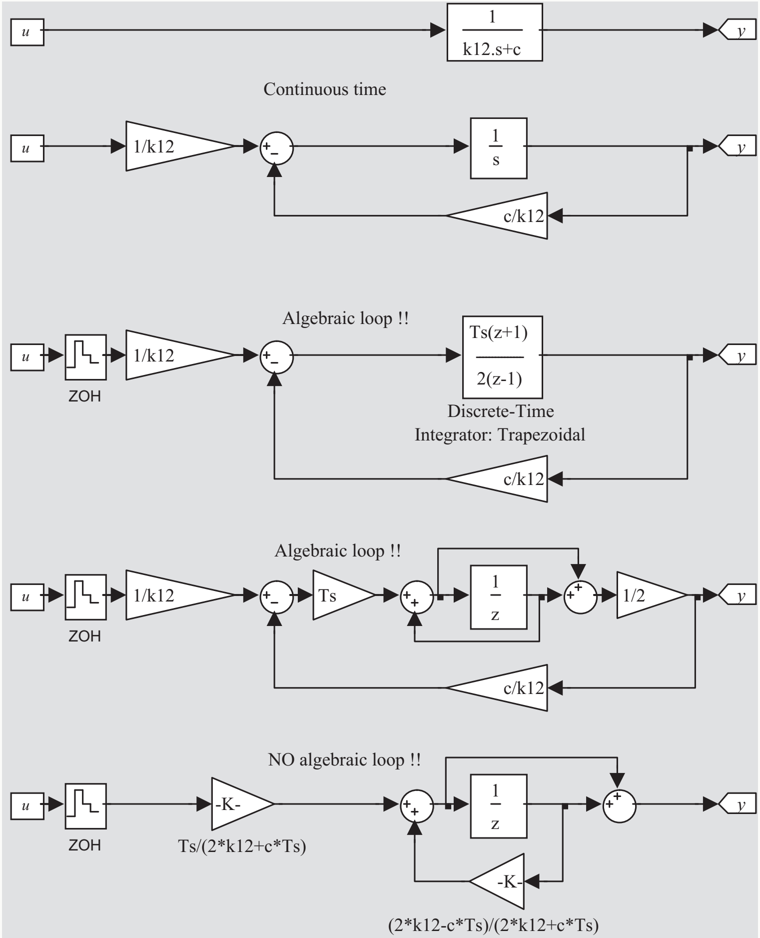
j with $1\frac{1}{2}T_s$,

a with T_s,

v with $\frac{1}{2}T_s$.

$\rightarrow \frac{1}{2}T_s$?

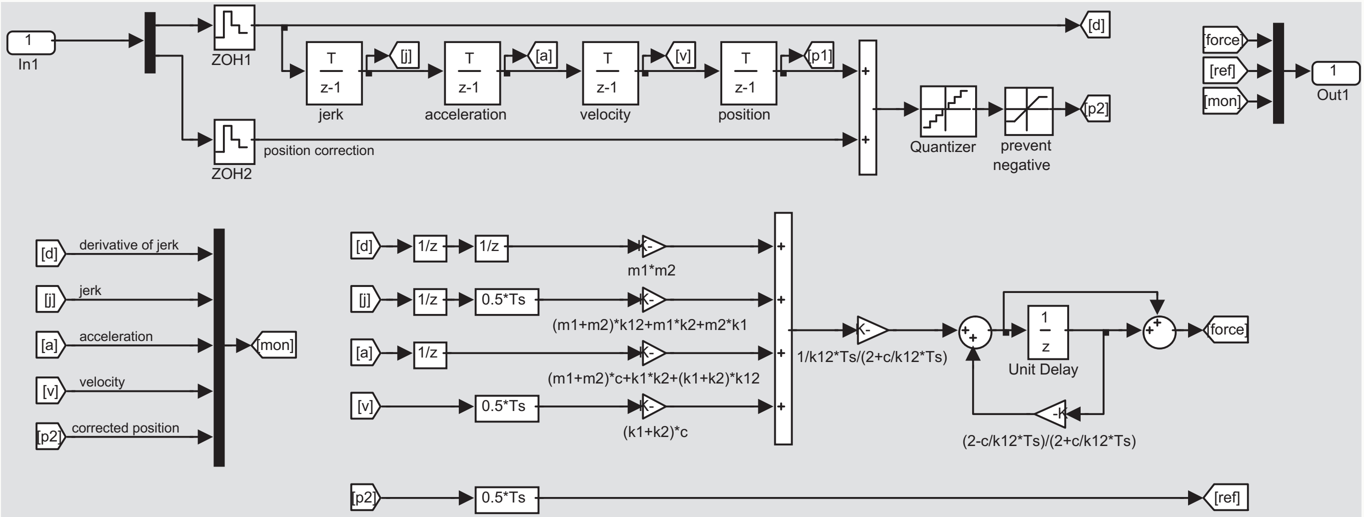
First order filter implementation



Transfer function:

$$y = \frac{\frac{T_s}{2k_{12}+cT_s}(z+1)}{z - \frac{2k_{12}-cT_s}{2k_{12}+cT_s}}u$$

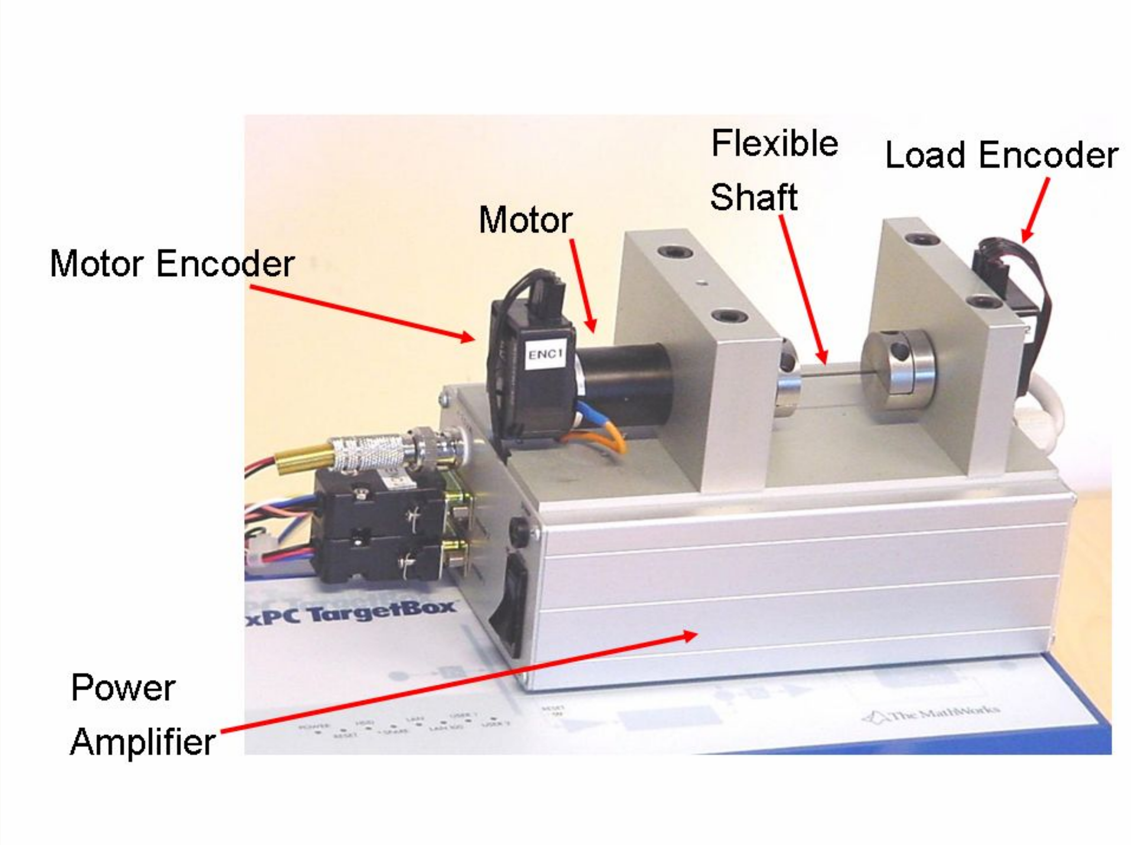
Digital 4th order feedforward



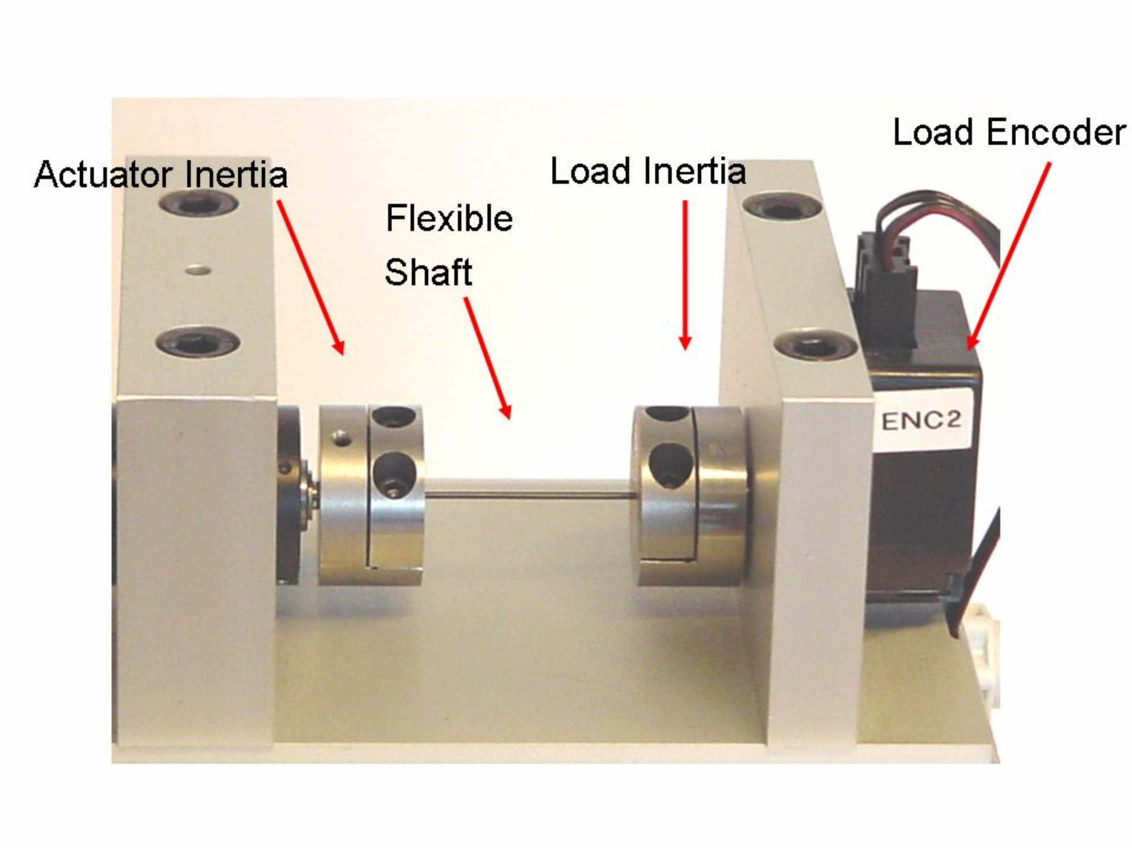
Bound selection

- Velocity:
  - back EMF smaller than power supply voltage
  - motor or gearbox specification (temperature)
- Acceleration:
  - maximum power supply or motor current
  - mechanical restrictions
- Jerk:
  - power amplifier rise time
  - mechanical restrictions
- Derivative of jerk upper bound: $\dfrac{\bar{J}}{T_s}$

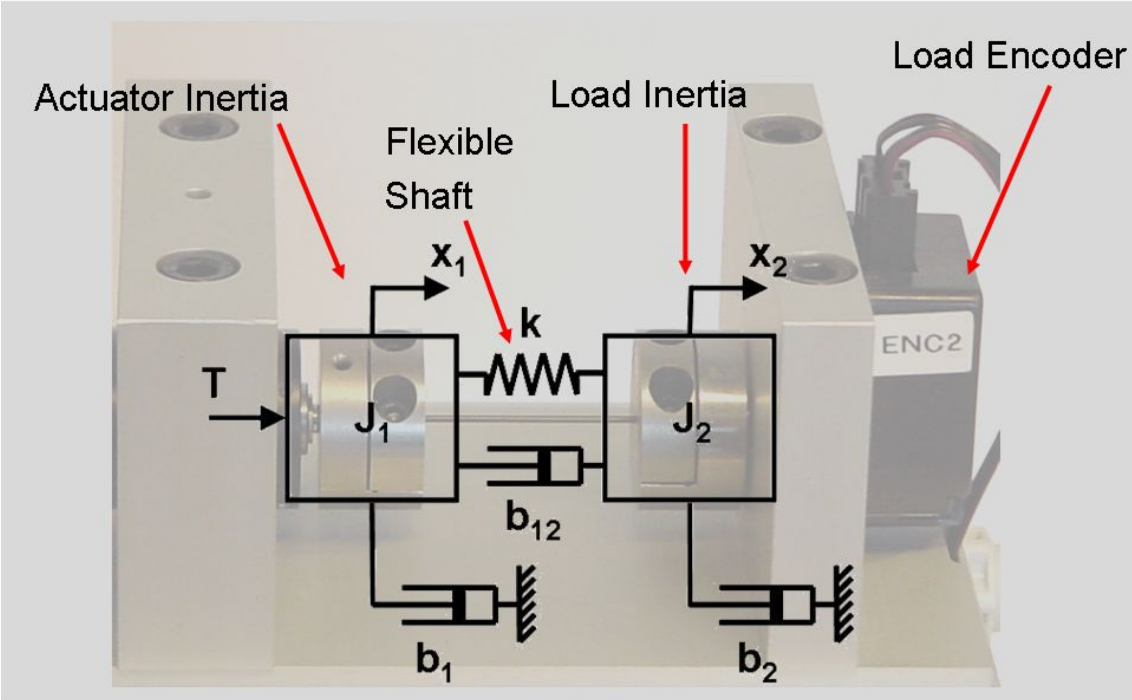
Experimental setup:



Experimental setup:

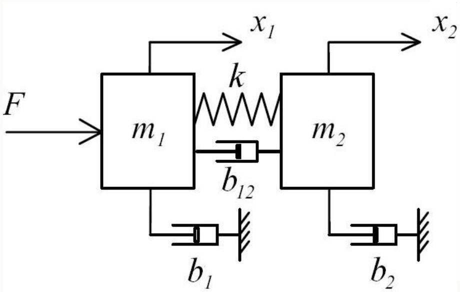


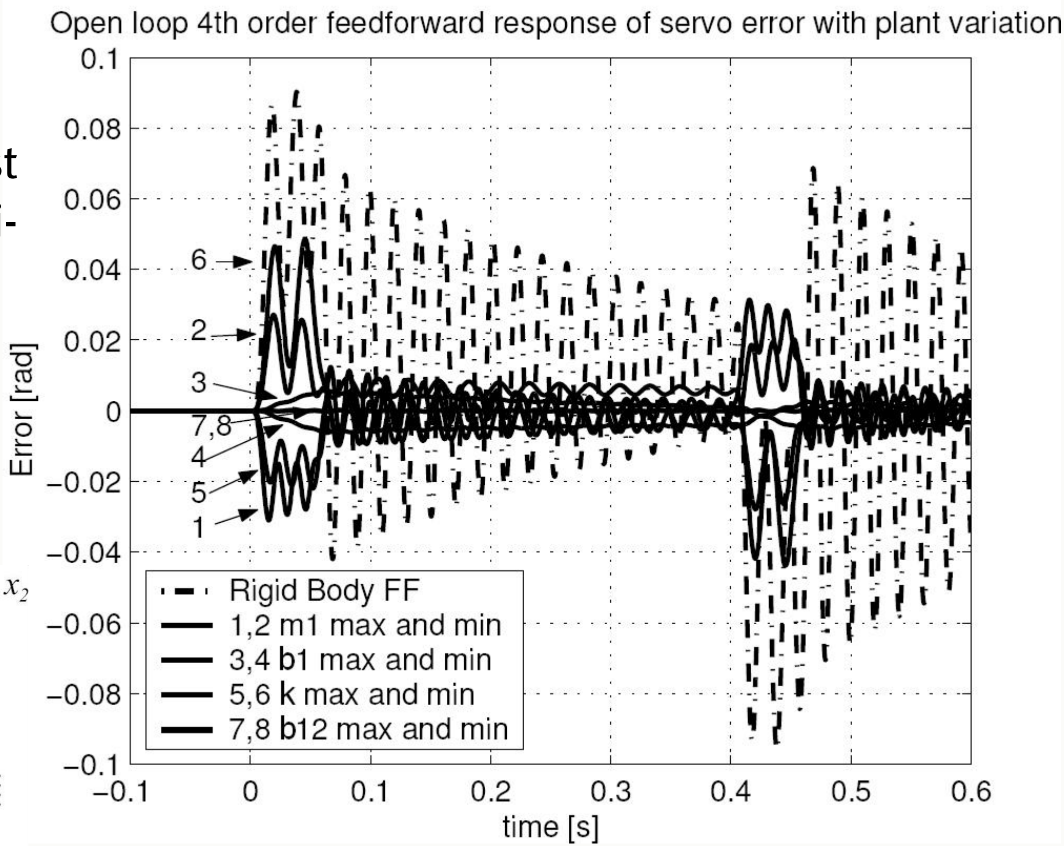
Experimental setup:



Simulation results

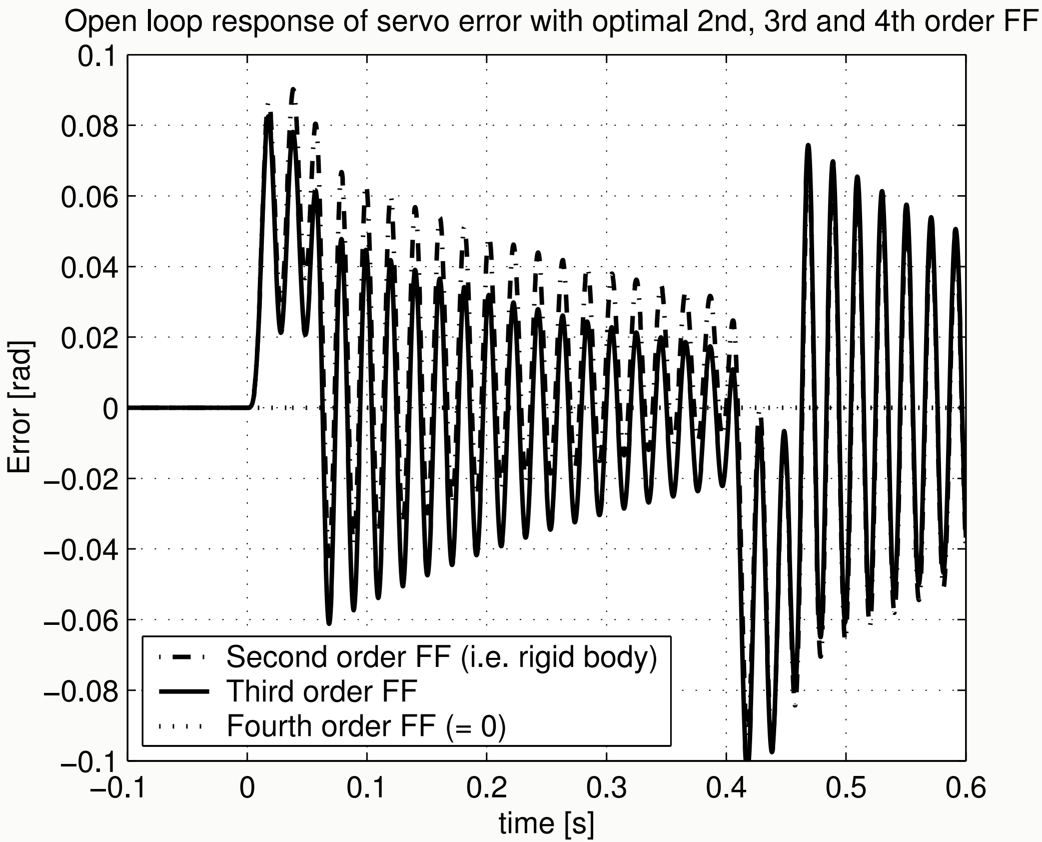
Robustness against variations in additional parameters





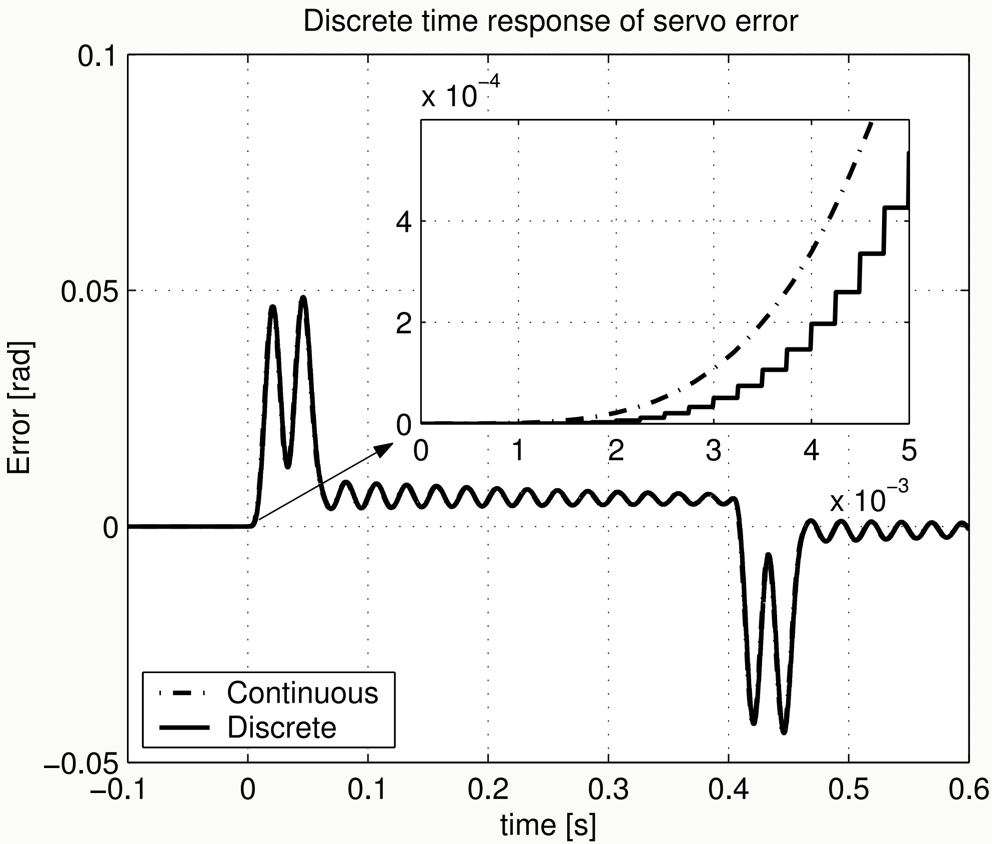
Simulation results

Effect of order of feedforward

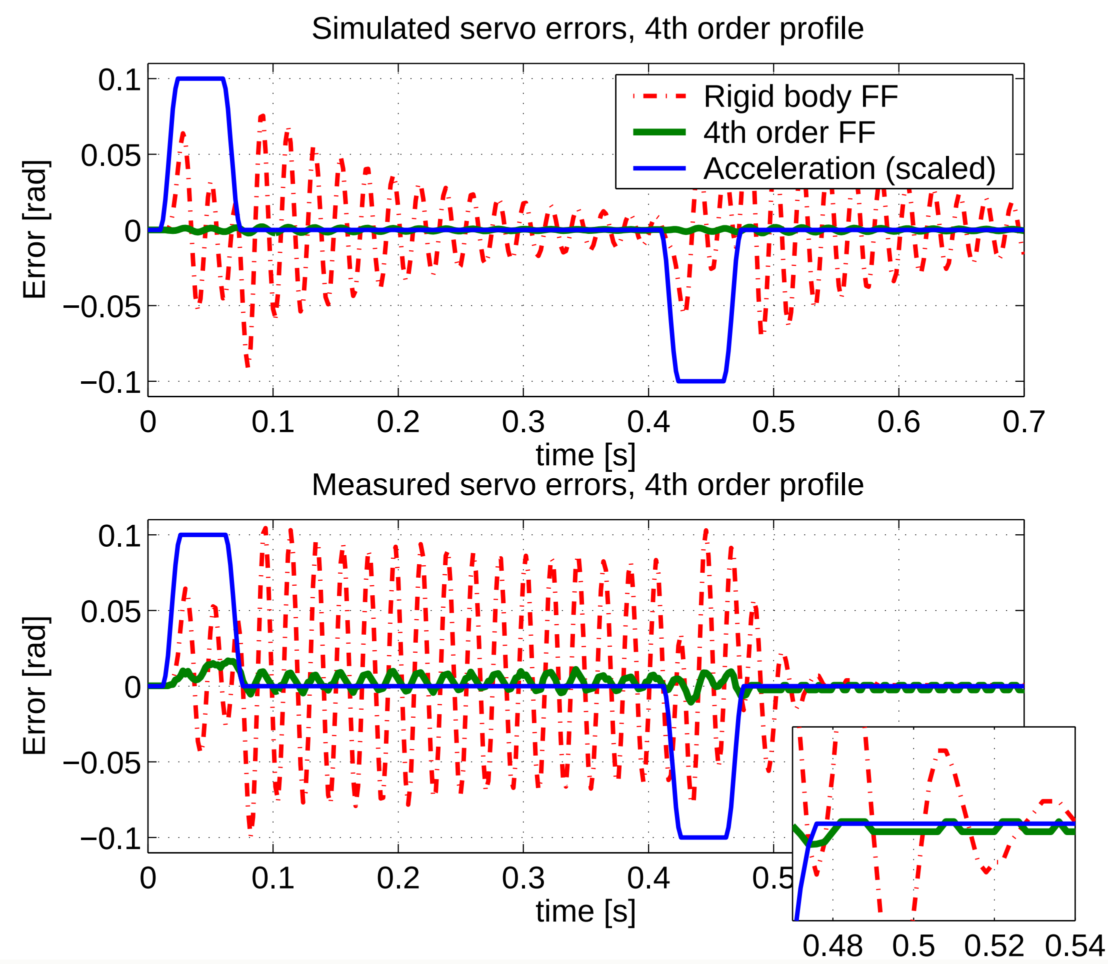


Simulation results

Discrete time vs. continuous time



Measured results



Conclusions

- Superior performance of 4th order vs. rigid-body feedforward
- Algorithm no problem for state-of-the-art motion controllers
- Especially feedforward of djerk effective (for electro-mechanical motion systems)
- Complete derivation (also for third order) available
- Simulink toolbox ‘motion’ available
- Experimental verification using MATLAB, Simulink and Real-Time Workshop