Adaptive cruise control

- Cruise control: Control system that adjust throttle to keep speed constant
- Adaptice cruise control (ACC): Adapt speed when vehicle in front of you

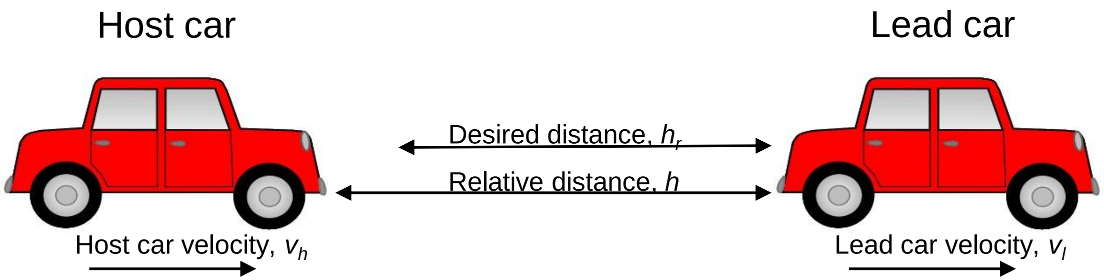
  Two modes of control:
  - Speed control when no vehicle in front
  - Distance control when vehicle in front – a radar sensor measures distance

https://www.continentalmitsubishi.com/blog/mitsubishi-adaptive-cruise-control-explained/

- Simplified case study: MPC for distance control

Inspired by
- Takahama, Akasaka: “Model Predictive Control Approach to Design Practical Adaptive Cruise Control for Traffic Jam”, 2017
- ACC example in Matlab MPC toolbox

ACC Modeling



$$\begin{aligned}
\text{desired (safe) distance/headway:} \quad & h_r = T v_h + h_o \\
\text{distance error:} \quad & \Delta h = h - h_r = h - T v_h - h_o \\
\text{relative speed:} \quad & \Delta v = v_l - v_h \\
\text{Newton's law, host car:} \quad & m \dot{v}_h = m a_f - F_r \\
\text{Throttle dynamics:} \quad & \dot{a}_f = -\frac{1}{T_f} a_f + \frac{1}{T_f} u, \quad T_f = 0.5
\end{aligned}$$

u : Input, acceleration command (engine or brake)

F_r : Resistive force (air drag, rolling resistance, road inclination, etc.)

$$\text{States:} \quad \begin{cases}
\dot{\Delta h} & = \dot{h} - \dot{h}_r = \Delta v - T \dot{v}_h = \Delta v - T \left(a_f - \frac{1}{m} F_r \right) \\
\dot{\Delta v} & = \dot{v}_l - \dot{v}_h = \dot{v}_l - a_f + \frac{1}{m} F_r \\
\dot{a}_f & = -\frac{1}{T_f} a_f + \frac{1}{T_f} u
\end{cases}$$

Note that “throttle dynamics” is most inaccurate part of this model. In reality acceleration and braking has quite different dynamics.

ACC state-space model

A linear state-space model on the form

$$\dot{x} = \tilde{A}x + \tilde{B}u + \tilde{E}d$$

can be written down as

$$\begin{pmatrix} \dot{\Delta h} \\ \dot{\Delta v} \\ \dot{a}_f \end{pmatrix} = \begin{pmatrix} 0 & 1 & -T \\ 0 & 0 & -1 \\ 0 & 0 & -\frac{1}{T_f} \end{pmatrix} \begin{pmatrix} \Delta h \\ \Delta v \\ a_f \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \frac{1}{T_f} \end{pmatrix} u + \begin{pmatrix} \frac{T}{m} & 0 \\ \frac{1}{m} & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} F_r \\ \dot{v}_l \end{pmatrix}$$

A discrete-time model to be used for MPC is written

$$x_{t+1} = Ax_t + Bu_t + Ed_t$$

where the matrices are found by using the Matlab-command c2d, using Euler discretization, or similar.

ACC control performance specifications

- Control distance to vehicle in front

$$\min \ (\Delta h)^2$$

- While not using too much input (not accelerating excessively)

$$\min \ u^2$$

- Limit maximum accelerations

$$-3 \text{ m/s}^2 \leq \dot{v}_h \leq 2 \text{ m/s}^2$$

- Limit speed

$$v_h \leq v_{\text{set}}$$

However, v_h is not a state, while $\Delta v = v_l - v_h$ is. We reformulate the constraint in terms of Δv using $v_h = v_l - \Delta v$:

$$\Delta v \geq v_l - v_{\text{set}}$$

The acceleration, $\dot{v}_h$, can be expressed in terms of state and disturbance from force balance:

$$\dot{v}_h = a_f - \frac{1}{m} F_r$$

MPC formulation

$$x = \begin{pmatrix} \Delta h \\ \Delta v \\ a_f \end{pmatrix}$$

- Assume *d = 0*

$$\begin{aligned} \min \sum_{t=0}^{N-1} & \frac{1}{2} x_{t+1}^\top Q x_{t+1} + \frac{1}{2} u_t^\top R u_t \\ \text{s.t.} \quad & x_{t+1} = A x_t + B u_t, \quad t = 0, \dots, N-1 \\ & x^{\text{low}} \le x_t \le x^{\text{high}}, \quad t = 1, \dots, N \\ & u^{\text{low}} \le u_t \le u^{\text{high}}, \quad t = 0, \dots, N-1 \end{aligned}$$

- Weights:

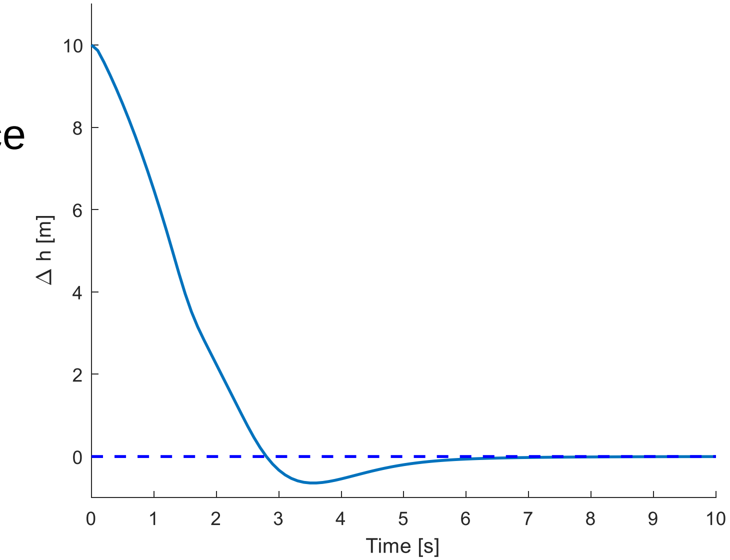
$$Q = \begin{pmatrix} q_{\Delta h} & 0 & 0 \\ 0 & q_{\Delta v} & 0 \\ 0 & 0 & q_{a_f} \end{pmatrix}, \quad R = r$$

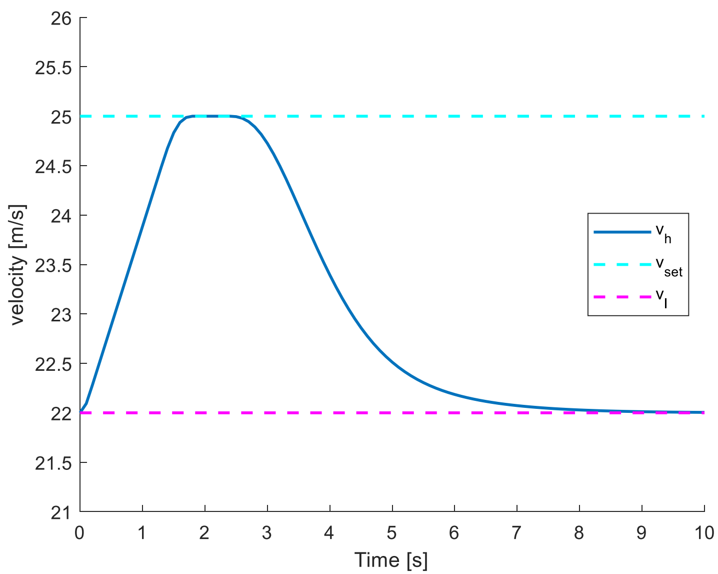
- Constraints:

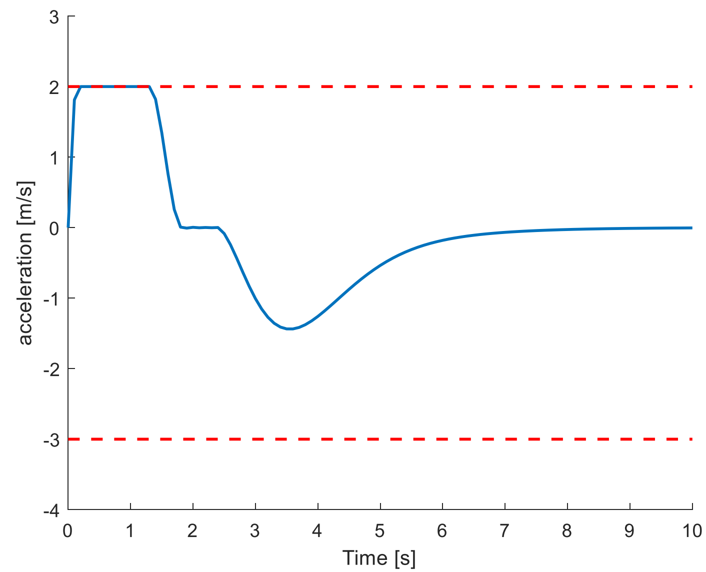
$$x^{\text{low}} = \begin{pmatrix} -\infty \\ v_l - v_{\text{set}} \\ -3 + \frac{1}{m} F_r \end{pmatrix}, \quad x^{\text{high}} = \begin{pmatrix} \infty \\ \infty \\ 2 + \frac{1}{m} F_r \end{pmatrix}$$

Scenario 1: No disturbances

- No disturbances
  - lead car velocity constant, no resistant force
- Start 10m behind desired headway, same velocity as leader car

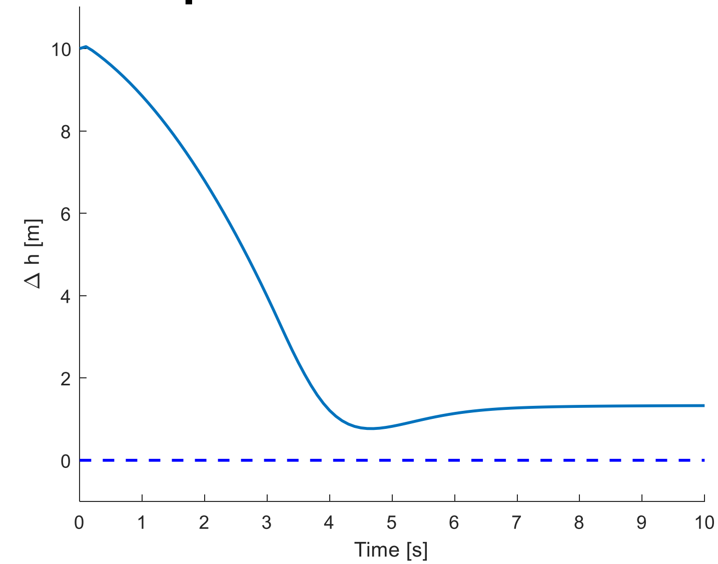


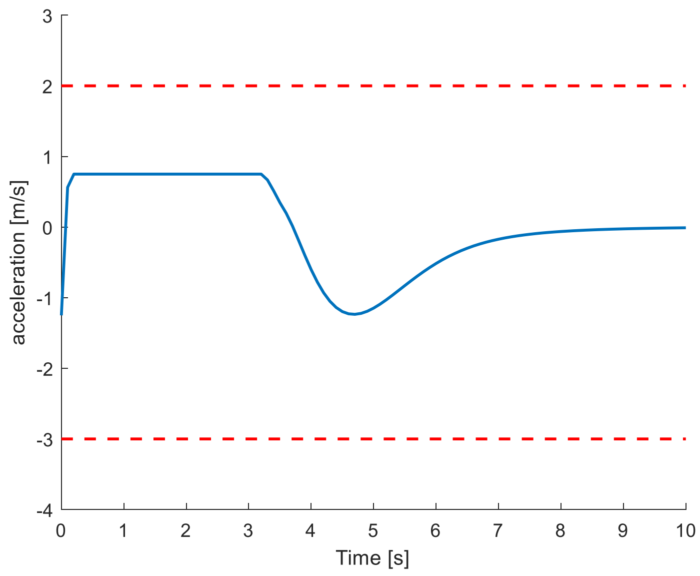


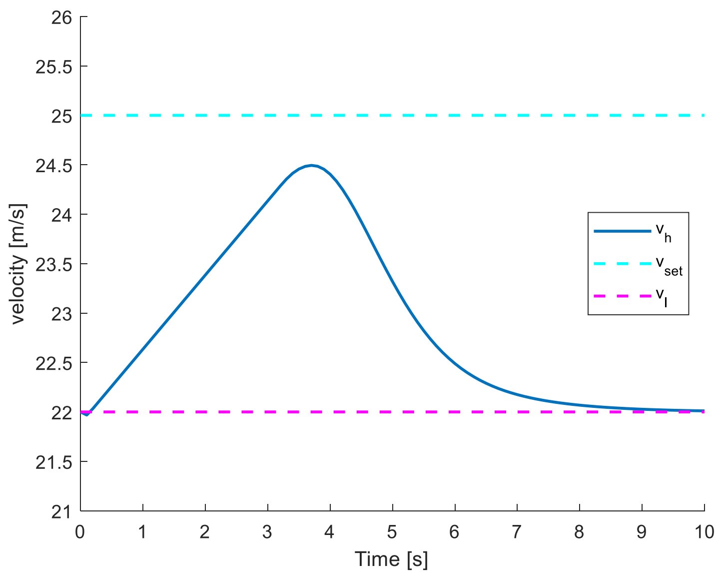


Scenario 2: Disturbance, no compensation

- $F_r = 1000 \text{ N}, \quad \dot{v}_l = 0 \text{ m/s}^2$







Offset free control

- Make disturbance observer:

The disturbance observer for

$$
\begin{aligned}
x_{t+1} &= Ax_t + Bu_t + Ed_t \\
y_t &= Cx_t + Fd_t
\end{aligned}
$$

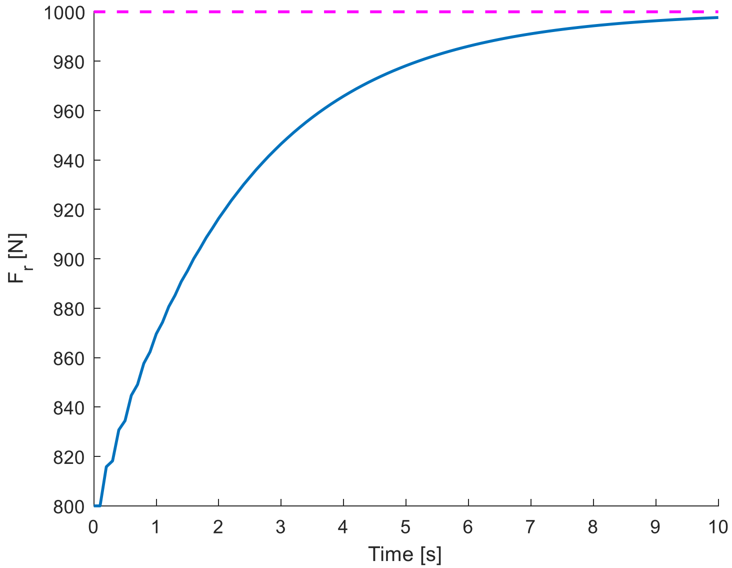
is

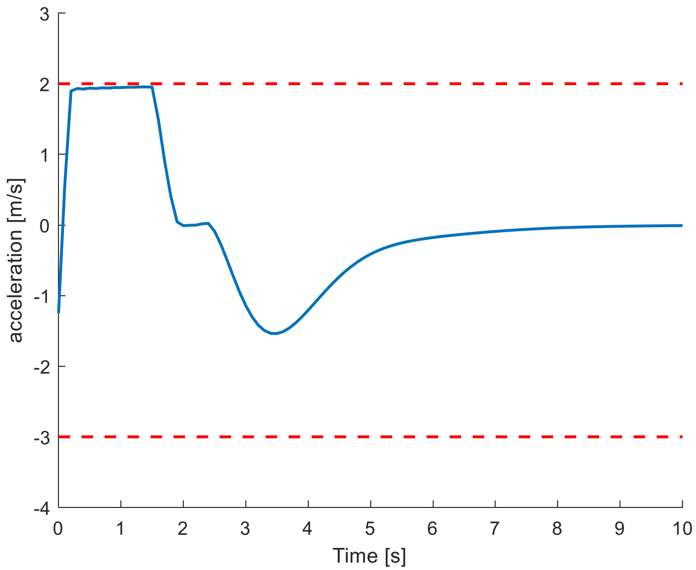
$$
\begin{pmatrix} \hat{x}_{t+1} \\ \hat{d}_{t+1} \end{pmatrix} = \begin{pmatrix} A & E \\ 0 & I \end{pmatrix} \begin{pmatrix} \hat{x}_t \\ \hat{d}_t \end{pmatrix} + \begin{pmatrix} B \\ 0 \end{pmatrix} u_t + K_F \left(y_t - \begin{pmatrix} C & F \end{pmatrix} \begin{pmatrix} \hat{x}_t \\ \hat{d}_t \end{pmatrix} \right)
$$

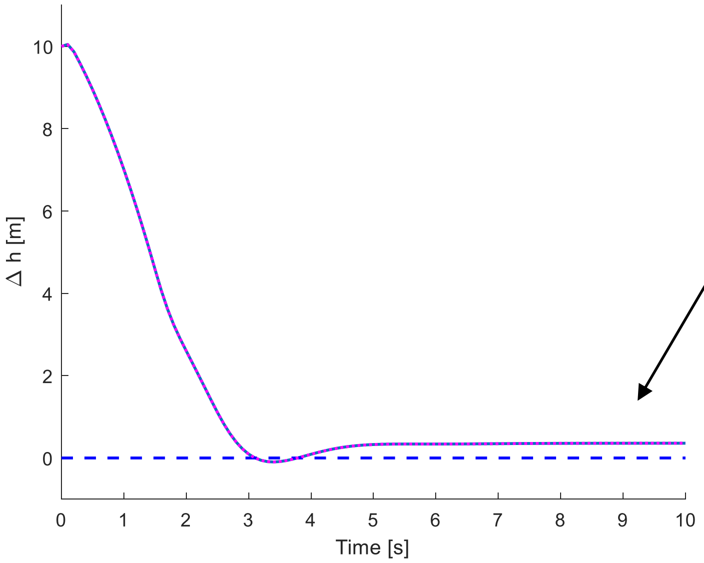
Note: The MPC must now use the augmented model as prediction model

- Do target calculation
  - Not implemented yet

Scenario 3: Disturbance, Offset-free MPC







Still not completely offset-free, since I haven't added target calculation