Lecture 5: Vehicular Control

Vehicular Control

Prof. Sangyoung Park

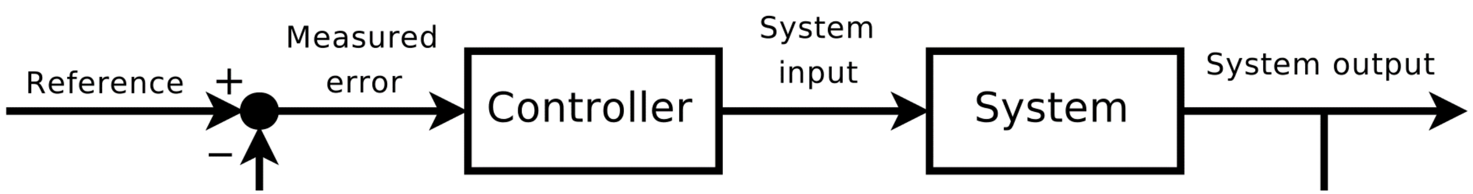
Module ”Vehicle-2-X: Communication and Control“

Contents

- Basics of Control Theory
- Vehicular Control

Control Theory

- Open-loop control vs. Closed-loop control
- Open-loop control: Control action of the controller is independent of the „process output" (no feedback)
- Closed-loop control: Control action of the controller is dependend on feedback from the process in the form of the value of the process output

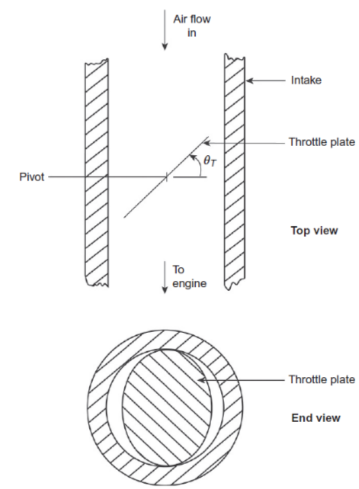


Source: Control theory, Wikipedia

Open-Loop vs Closed-Loop

- Vehicle cruise control example
  - Open-loop control
    - Lock the throttle position (recall the figure below?): controls the air intake
    - Vehicle will travel slower when climbing uphill
    - Cannot compensate for changes in circumstances
  - Closed-loop control
    - Data from sensor monitoring the vehicle speed
    - Controller continuously compared the sensor output with the desired speed
    - The „error“ determines the throttle position

Throttle valve

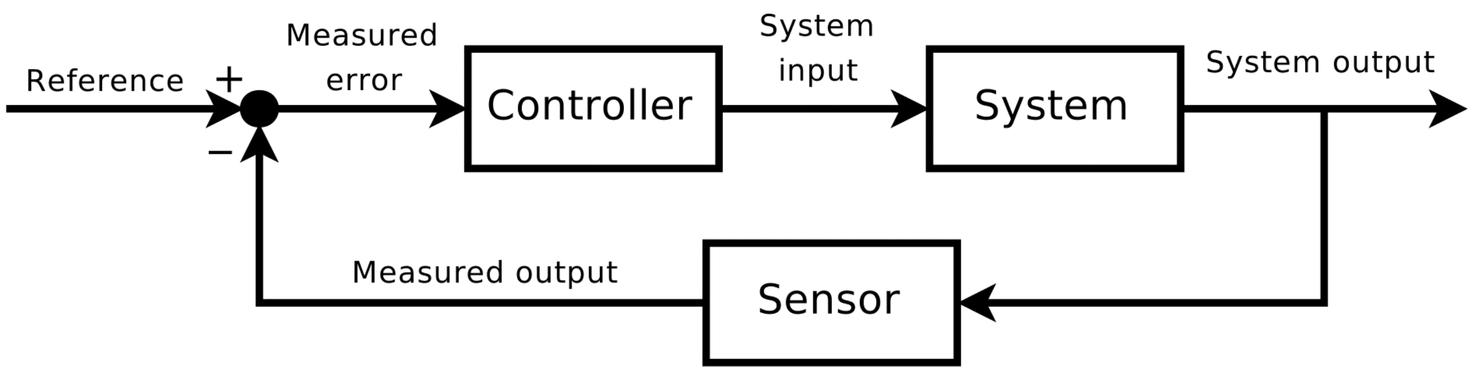


Source: W. Ribbens, “Understanding automotive electronics”

Definitions

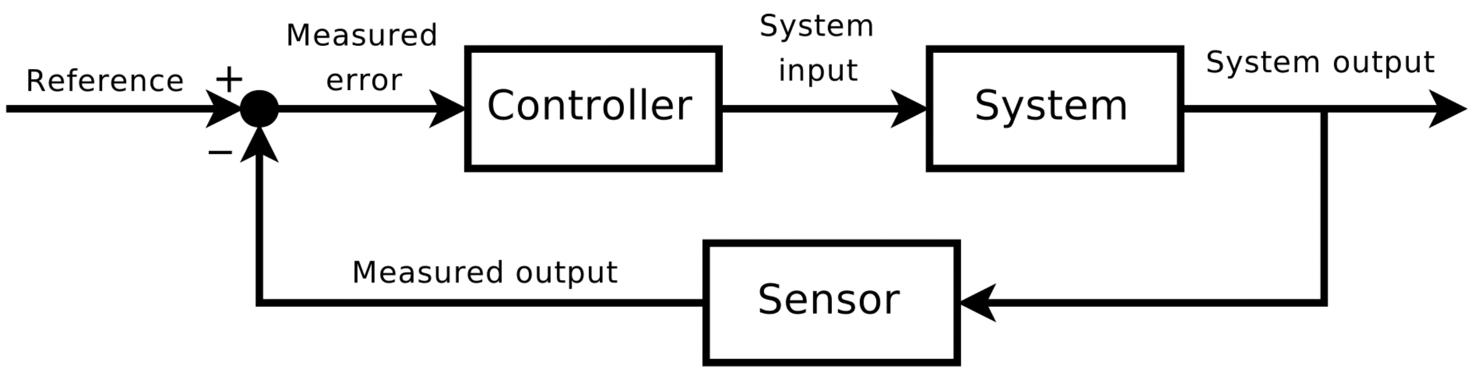
- Sensor: Component for measurement of a variable (signal)
- Plant (or system): Is the part to be controlled
- Controller: Provides the satisfactory characteristics for the total system

- Two types of control systems
  - Regulator: Maintains a physical variable at some constant value in presence of perturbances
  - Servomechanism: A physical variable is required to follow or track some time-varying function



Block Diagram and Models

- Control system is often described using block diagrams
- Block diagrams contain *models*, a mathematical description of input-output relation of components combined with block diagram

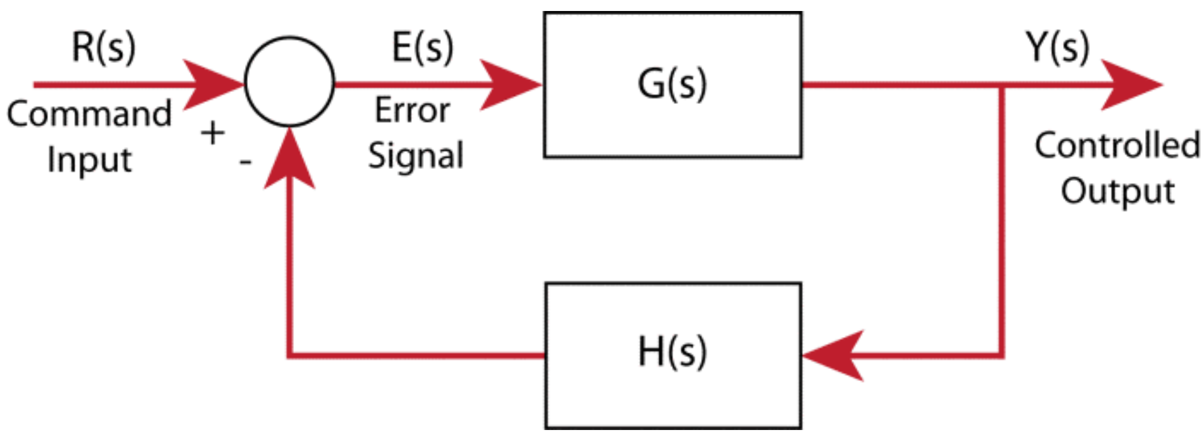


Transfer Functions

- A transfer function of a ***linear*** system is defined as the ratio of the Laplace transform of the output and the laplace transform of the input

$$Y(s) = G(s)E(s)$$
$$E(s) = R(s) - H(s)Y(s)$$
$$Y(s) = G(s)[R(s) - H(s)Y(s)]$$
$$\frac{Y(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)}$$

- System gain = forward gain / (1 + loop gain)



Source: http://bodetechnics.com/control-engineering-tutorials/transfer-function-block-diagram-manipulation/

Laplace Transform

- Laplace transform of $f(t)$ denoted by $F(s)$ or $L\{f(t)\}$, is an integral transform given by the Laplace integral:

$$L\{f(t)\} = F(s) = \int_0^\infty e^{-st} f(t)dt$$

- Provided that this integral exists
- Transformation to the frequency domain is one-to-one
- $f(t) = 1,\ F(s) = \frac{1}{s}$

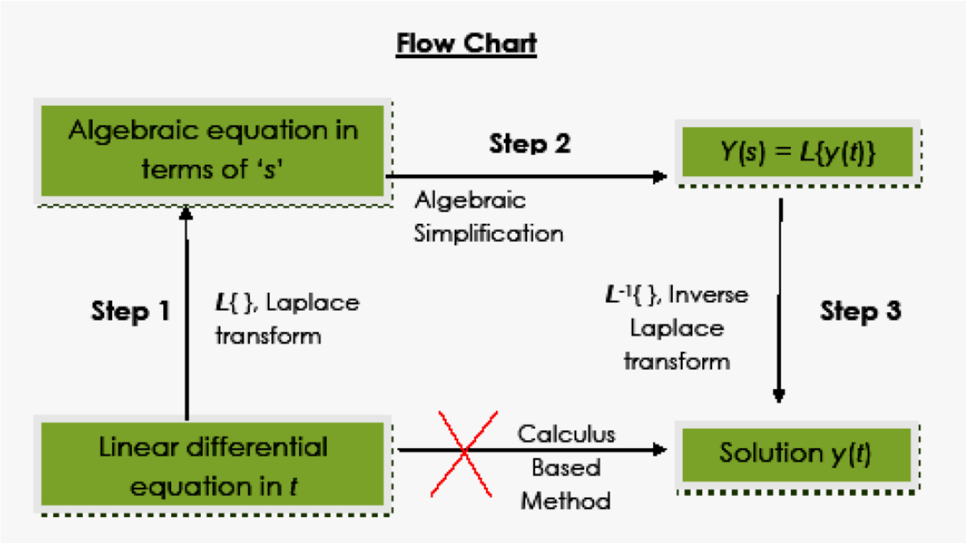
$$L\{f(t)\} = \int_0^\infty e^{-st} dt = -\frac{1}{s} e^{-st} \Big|_0^\infty = \frac{1}{s}$$

- $f(t) = t,\ F(s) = \frac{1}{s^2}$
- $f(t) = e^{at},\ F(s) = \frac{1}{s-a}$

Laplace Transform

- Some useful properties
  - $L\{f'(t)\} = sL\{f(t)\} - f(0)$
  - $L\{f''(t)\} = s^2L\{f(t)\} - sf(0) - f'(0)$
  - $L\{f'''(t)\} = s^3L\{f(t)\} - s^2f(0) - sf'(0) - f''(0)$
- Useful for solving linear differential equations

$$y'' - 6y' - 5y = 0, y(0) = 1, y'(0) = -3$$



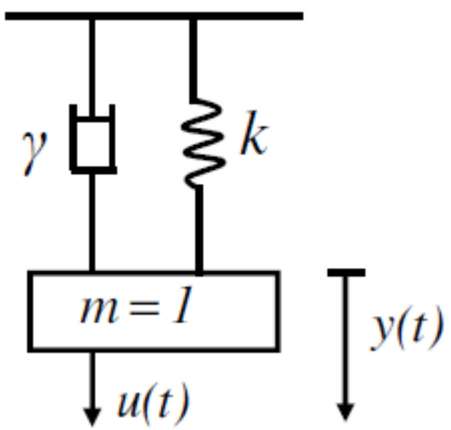
Source: Z. S. Tseng, „The Laplace Transform“, 2008

Modeling of Dynamic System

- Dynamic system modeling example
- k: spring constant, γ: damping constant, $u(t)$: force

$$\ddot{y} = -ky(t) - \gamma\dot{y}(t) + u(t)$$
$$\ddot{y}(t) + \gamma\dot{y}(t) + ky(t) = u(t)$$
$$y(0) = y_0, \dot{y}(0) = \dot{y}_0$$

- This is an linear ordinary differential equation
  - Linear: no y^2
  - Ordinary: one independent variable
    (as opposed to partial differential equations)

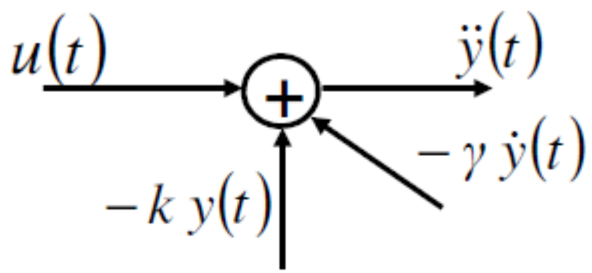


Modeling of Dynamic System

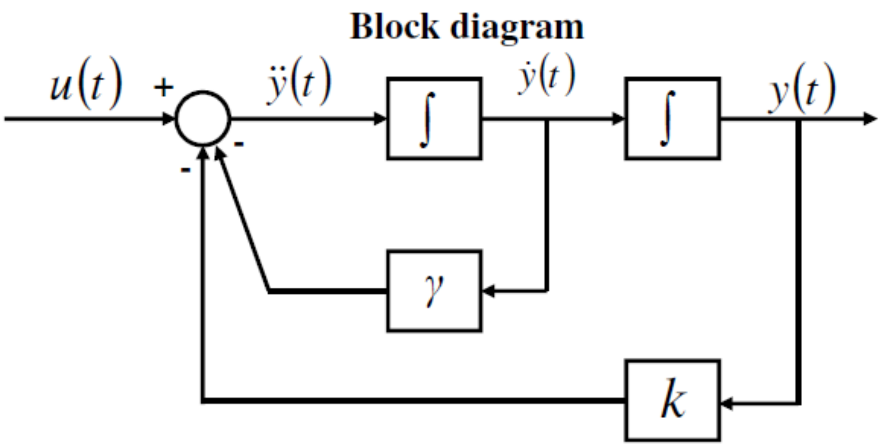
- Express the highest order term

$$\ddot{y}(t) = -ky(t) - \gamma\dot{y}(t) + u(t)$$

- Put adder in front



- Synthesize all other terms using integrators



State Space Equation

- Any system which can be presented by LODE can be represented in ***state space form*** (matrix differential equation)
- Example

$$\ddot{y} = -ky(t) - \gamma\dot{y}(t) + u(t)$$

- Step 1: Deduce set off first order differential equation in variables
  $x_j(t)$: states of system
  $x_1(t)$: Position $y(t)$
  $x_2(t)$: Velocity $\dot{y}(t)$

$$\dot{x_1}(t) = \dot{y}(t) = x_2(t)$$
$$\dot{x_2}(t) = \ddot{y}(t) = -kx_1(t) - \gamma x_2(t) + u(t)$$

- ***One*** linear ordinary differential equation (LODE) of order ***two*** is transformed into ***two*** LODE of order of ***one***

State Space Equation

- Step 2: Put everything together in a matrix differential equation

$$\begin{bmatrix} \dot{x_1}(t) \\ \dot{x_2}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k & -\gamma \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

- State equation

$$\dot{x}(t) = Ax(t) + Bu(t)$$

- Measurement equation: related observed value to the state vector

$$y(t) = [1\ 0] \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

$$y(t) = Cx(t) + Du(t)$$

- System state
  - **System state** x of a system at any time t_0 is the “amount of information that together with all inputs for $t \geq t_0$, uniquely determines the behavior of the system for all $t \geq t_0$

State Space Equation

- Linear time-invariant (LTI) system is described by standard form of the state space equation

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

- In most cases D=0

Variable	Dimension	Name
$x(t)$	$n \times 1$	State vector
A	$n \times n$	System matrix
B	$n \times r$	Input matrix
$u(t)$	$r \times 1$	Input vector
$y(t)$	$p \times 1$	Output vector
C	$p \times n$	Output matrix
D	$p \times r$	Matrix representing direct coupling with input and output

State Space Equation

- Okay why bother with state space equations?
- Computers love state space equations
- Modern control uses state space equation
- Notations are not unique

$$y^{(n)}(t) + a_{n-1}y^{(n-1)}(t) + \cdots + a_1\,\dot{y}(t) + a_0 y(t) = b_m u^{(m)}(t) + \cdots + b_1\dot{u}(t) + b_0 u(t)$$

- Control-canonical form

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix}, \mathrm{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, C = [b_0 \;\; b_1 \;\; b_2], D = b_3$$

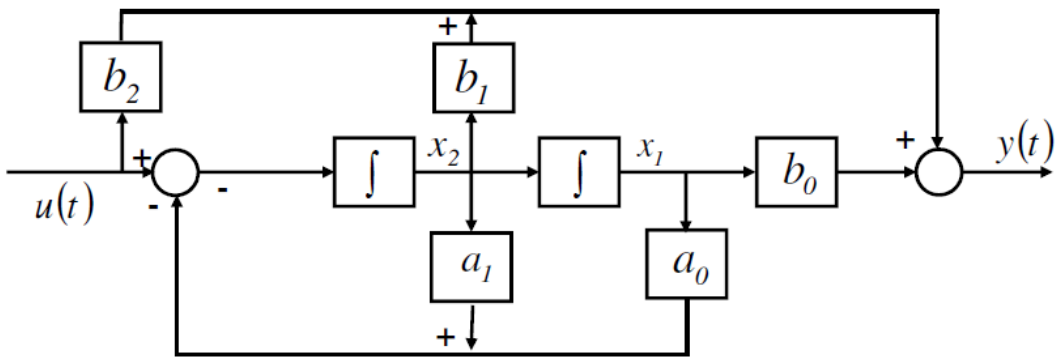
- Observer-canonical form

$$A = \begin{bmatrix} 0 & 0 & -a_0 \\ 1 & 0 & -a_1 \\ 0 & 1 & -a_2 \end{bmatrix}, \mathrm{B} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix}, C = [0 \;\; 0 \;\; 1], D = b_3$$

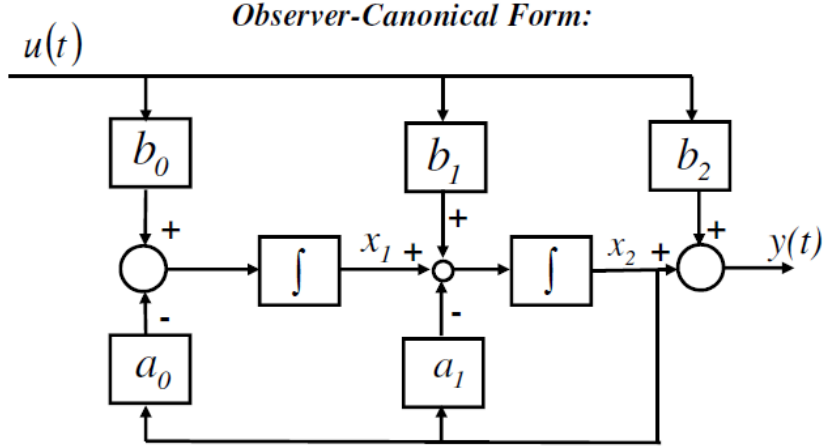
State Space Equation

Block diagrams:

Control-canonical Form:



Observer-Canonical Form:



State Space Equation

- Example

$$\frac{d^2}{dt^2}y(t) + 4\frac{d}{dt}y(t) + 3y(t) = 2u(t)$$

- State space equation
  - Let $x_1(t) = y(t)$ and $x_2(t) = \dot{y}(t)$

$$\dot{x}_1(t) = \dot{y}(t) = x_2(t)$$
$$\dot{x_2}(t) + 4x_2(t) + 3x_1(t) = 2u(t)$$
$$\dot{x_1}(t) = -3x_1(t) - 4x_2(t) + 2u(t)$$

- Write equations in matrix form
  - $x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -3 & -4 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 2 \end{bmatrix} u(t)$$
$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t)$$

Stability

- If we know transfer function $G(s)$, what can we say about the system stability?

- A linear time invariant system is called BIBO stable (bounded-input-bounded-output).
- For all bounded inputs $|u(t)| \leq M_1$ (for all t), exists a boundary for the output signal M_2, so that $|y(t)| \leq M_2$ for all t, with M_1, M_2, positive real numbers

Stability

- Example: $Y(s) = G(s)U(s)$, interator $G(s) = \frac{1}{s}$

$$u(t) = \delta(t), U(s) = 1$$

$$|y(t)| = |L^{-1}[Y(s)]| = \left|L^{-1}\left[\frac{1}{s}\right]\right| = 1$$

- What happens when the input is $u(t) = 1$?

$$u(t) = 1\,, U(s) = \frac{1}{s}$$

$$|y(t)| = |L^{-1}[Y(s)]| = \left|L^{-1}\left[\frac{1}{s^2}\right]\right| = t$$

- (unbounded)
- BIBO stability should be proven for ALL inputs

Stability

- $Y(s) = G(s)U(s)$
- By means of convolution theorem we get
- $|y(t)| = |\int_0^t g(\tau)u(t-\tau)d\tau| \leq \int_0^t |g(\tau)||u(t-\tau)|d\tau \leq M_1 \int_0^t |g(\tau)d\tau \leq M_2$
- Therefore,
- If the impulse response, $\int_0^\infty |g(\tau)d\tau < \infty$, is bounded, then the system is BIBO-stable
- But what about transfer function?

Transfer Function of a State Space Model

- State space model

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

$$sX(s) - x(0) = AX(s) + BU(s)$$
$$X(s) = (sI - A)^{-1}x(0) + (sI - A)^{-1}BU(s)$$
$$= \phi(s)x(0) + \phi(s)BU(s)$$

$$\mathrm{Y(s)} = CX(s) + DU(s)$$
$$= C[(sI - A)^{-1}]x(0) + [C(sI - A)^{-1}B + D]U(s)$$
$$= C\phi(s)x(0) + C\phi(s)BU(s) + DU(s)$$

- Transfer function

$$G(s) = \frac{Y(s)}{U(s)} = C\phi(s)B + D$$

Stability

- Can stability be determined if we know the transfer function of a system?
- State space model

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

- The transfer function is given by

$$H(s) = C(sI - A)^{-1}B + D = C\frac{Adj(sI - A)}{\det(sI - A)} + D$$

  - $Adj(A)$ is the adjugate matrix of A
  - When, $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, $Adj(A) = \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$
- The poles of H(s) are uncancelled eingenvalues of A, assuming D=0
  - Let λ_i be the i^{th} eigenvalue of A, if $\lambda_i \leq 0, \forall i$, then system is stable

Vehicle Longitudinal Control: Sensor Inputs

- Four types of information are usually considered for the longitudinal control
  - Speed and acceleration of the host vehicle
  - Distance to the preceding vehicle
  - Speed and acceleration of the preceding vehicle
  - Acceleration and speed of the first vehicle (i.e., lead vehicle)
- Speed and acceleration of the host vehicle can be measured by speed sensors and accelerometers onboard the vehicle
- Distance to the preceding vehicle can be measured by ranging sensors, e.g., radar, LIDAR, ultrasonic sensors
  - Radar has been used most commonly
  - LIDAR is affected by weather (snow and fog)

Vehicle Longitudinal Control: Sensor Inputs

- Speed and acceleration of the preceding vehicle and lead vehicle
  - Speed and acceleration of the preceding vehicle can be derived from the host vehicle
    - However, requires differentiation of the radar sensor, which can be noisy
  - Communication
    - Transmit the speed and acceleration to the succeeding vehicle
    - Reliability of communication?

Longitudinal Vehicle Model

- Assumptions
  - Time delays associated with power generation in the engine are negligible
  - Torque converter in the vehicle is locked
  - No torsion in the drive axle
  - Slip between the tires and the road is zero
- Then, vehicle speed V_x is directly related to the engine speed ω_e

$$\dot{x} = v_x = Rh\omega_e$$

where R and h are gear ratio and tire radius

Longitudinal Vehicle Model

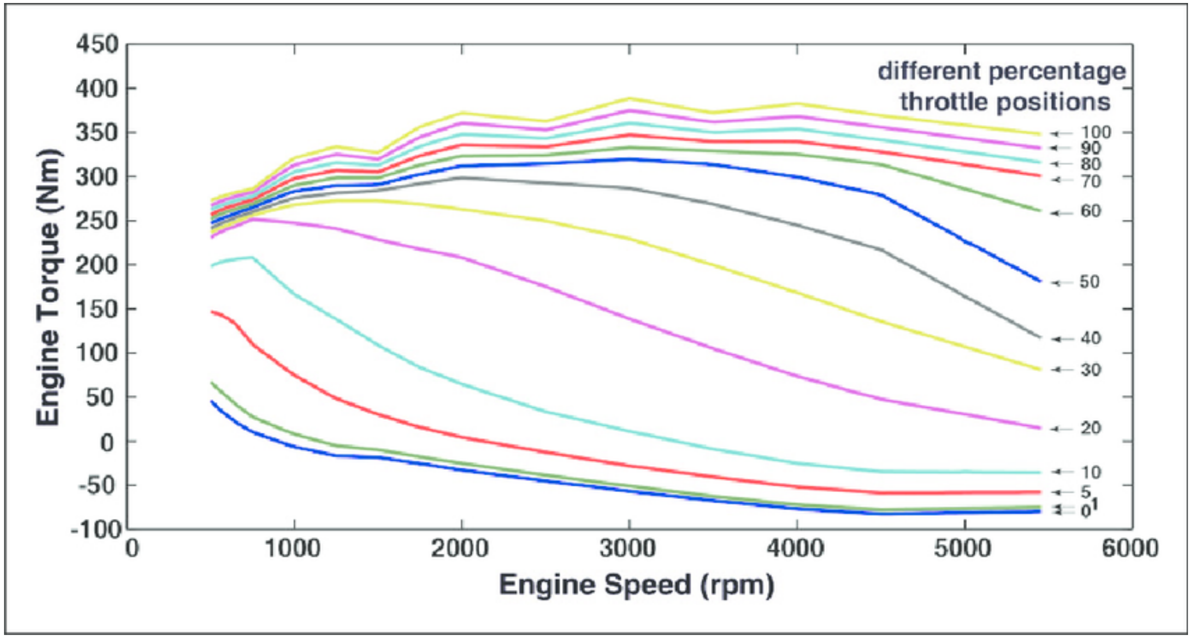
- The simplified vehicle dynamics model takes ***three*** variables as states
  - Mass of air in the intake manifold: m_a
  - Engine speed: ω_e
  - Brake torque: T_{br}
- The dynamics relating engine speed to the pseudo-inputs, net combustion torque T_{net} and brake torque T_{br} can then be modeled by

$$\dot{\omega} = \frac{T_{net} - c_a R^2 h^2 \omega_e^2 - R(hF_f + T_{br})}{J_e}$$

where c_a is the aerodynamic drag coefficient, F_f is the rolling resistance of the tires, and $J_e = I_e + (mh^2 + I_\omega)R^2$ is the effective inertia reflected on the engine side

Longitudinal Vehicle Model

- $T_{net}(\omega_e, m_a)$ is a nonlinear function obtained from steady-state engine maps available from the vehicle manufacturer



Source: S. M. M. Jaafari, „A comparison on optimal torque vectoring strategies in overall performance enhancement of a passenger car“,

Longitudinal Vehicle Model

- Dynamics relating m_a, the air mass flow in engine manifold, to the throttle angle can be modeled as

$$\dot{m}_a = \dot{m}_{ai} - \dot{m}_{ao}$$

  where m_{ai} and m_{ao} are the flow rate into the intake manifold and out from the manifold

- m_{ao} is a nonlinear function of ω_e and P_m, pressure of the air in engine manifold (from engine manufacturer)

- m_{ai} is

$$\dot{m}_{ai} = MAX \cdot TC(\alpha) PRI(m_a)$$

  where MAX is a constant dependend on the size of the throttle body, $TC(\alpha)$ is a nonlinear invertible function of the throttle angle, and PRI is the pressure influence function that describes the choked flow relationship which occurs through the throttle valve

Longitudinal Vehicle Model

- How do we measure m_a? Ideal gas law
  - $P_m V_m = m_a R_g T$
- Where R_g is a variable that depends on the vehicle transmission gear ratio, and T is the temperature
- Pressure can be measured to calculate m_a

Longitudinal Vehicle Model

- Brake model is linear and modeled by a first-order lag

$$\tau_{br}\dot{T}_{br} + T_{br} = T_{br,cmd} = K_{br}P_{br}$$

where τ_{br} is the brake system time constant, K_{br} is the total proportionality between the brake pressure P_{br} and the brake torque at the wheels

Longitudinal Vehicle Model

- Simpler models
- Assumption
  - Vehicle reacts to the acceleration input without any delay (no inertia)
- Control input: acceleration
- Control output: inter-vehicle distance

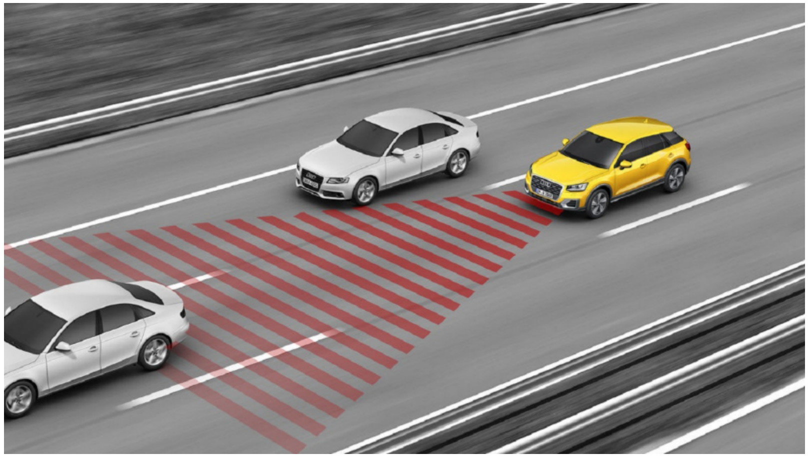
$$\dot{v}(t) = u_i(t)$$
$$\dot{X}_i(t) = A_i X_i(t) + B_i U_i(t)$$

- What would be A and B in such a case?

- $X_i(t) = \begin{bmatrix} x_i(t) \\ v_i(t) \end{bmatrix}, A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$
- $B_i = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, U_i(t) = u_i(t)$
- But, in more recent works, more sophisticated models are used

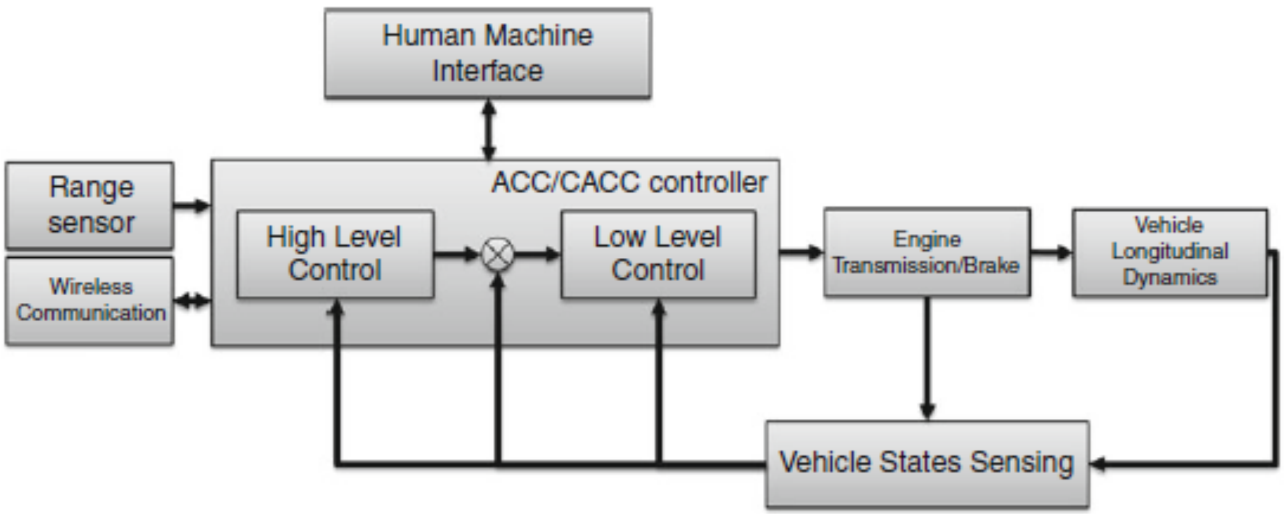
Platooning vs Adaptive Cruise Control

- Platooning
  - Maintains constant distance
  - Requires communication among vehicles
  - Minimum traffic shockwave
- Adaptive cruise control (ACC)
  - Maintains minimum distance and time headway
  - Usually achievable with sensor inputs only
  - Already in commercial vehicles



Longitudinal Control System Architecture

- Usually consists of inner loop and outer loop controllers
- Outer loop controller (upper or high level) synthesizes a desired speed or acceleration
- Inner loop controller (lower level) will generate corresponding throttle or brake commands



Longitudinal Control System Design

- Upper level controller
  - Determines the desired acceleration for each vehicle so as to
    - Objective (1): Maintain constant small spacing between the vehicles
    - Objective (2) Ensure string stability of the platoon
  - Plant model of the upper level controller is

$$\ddot{x}_i = u_i$$

    where the subscript i denotes the i-th vehicle in the platoon
  - However, due to the finite bandwidth associated with the lower level controller, each vehicle is actually expected to track the desired acceleration imperfectly

Upper Level Controller

- Performance specification of the upper level controller is therefore to meet objectives (1) and (2) in the presence of first-order lag in the lower level controller: time lag from the desired value ($\ddot{x}_{i_{des}}$)

$$\ddot{x}_i = \frac{1}{\tau s + 1} \ddot{x}_{i_{des}} = \frac{1}{\tau s + 1} u_i$$

  - This assumption is made to simplify the design of the high-level control
  - (What's first-order lag?)

$$\tau \frac{dy}{dx} + y = x$$

- The spacing error for the i-th vehicle is defined as $\epsilon_i = x_i - x_{i-1} + L$, where ϵ_i is the longitudinal spacing error of the i-th vehicle, L being the desired spacing

Spacing Policies

- Speed independent spacing policy(constant spacing)
  - $x_{rd} = d_0$
  - Usually achievable only with v2v communication
- Speed-dependent spacing policy
  - Semi-autonomous: can be implemented with sensor measurements only
  - $x_{rd} = d_0 + \dot{x} t_{hw}$
  - Time headway is used
  - Commonly used in commercial ACC systems
  - d_0 is the minimum safe distance
  - t_{hw} is the time gap
  - Similar to driver's daily experience

Speed-Dependent Spacing Policy

- Different speed-dependent spacing policies have been proposed in literature
  - Time headway policy performs poor against traffic flow fluctuation (Remember the video from the introduction?)
  - Nonlinear spacing policy for the stability of the traffic flow has been proposed
    - Junmin W., et al., "Should adaptive cruise-control systems be designed to maintain a constant time gap between vehicles?", IEEE Trans. Veh. Technol. 2004

$$x_{rd} = \frac{1}{\rho_m\left(1 - \frac{\dot{x}_i}{v_f}\right)}$$

where ρ_m is the traffic density parameter

Upper Level Controller

- The objectives (1) and (2) can be mathematically stated as

$$\epsilon_{i-1} \to 0 \Rightarrow \epsilon_i \to 0$$
$$||H(s)||_{inf} \leq 1$$

where $\widehat{H}(s)$ is the transfer function relating the spacing errors of consecutive vehicles in the platoon: $\widehat{H}(s) = \frac{\epsilon_i(s)}{\epsilon_{i-1}(s)}$

  - Notation $||H(s)||_{inf}$ denotes the largest value $H(s)$ could have according to changing value of i (vehicles)

Upper Level Controller: String Stability

- The string stability of the platoon
  - Practically, it means the gap regulation error will not be amplified from the lead vehicle to the last vehicle in the platoon
  - $||H(s)||_{inf} \leq 1$, refers to a property in which spacing errors are guaranteed to diminsh as they propagate toward the tail of the platoon
  - Example: Any errors between the second and third cars do not amplify into an extremely large spacing error between seventh and eight cars
    - (D. Swaroop, et al., „String Stability of Interconnected Systems," IEEE Transactions on Automatic Control, 1996)
  - Will be robust against internal vehicle dynamics as well as the imperfections in the lower-loop
- Traffic network point of view on string stability
  - Less shockwaves
- Driver's point of view
  - Smooth ride and safety benefit

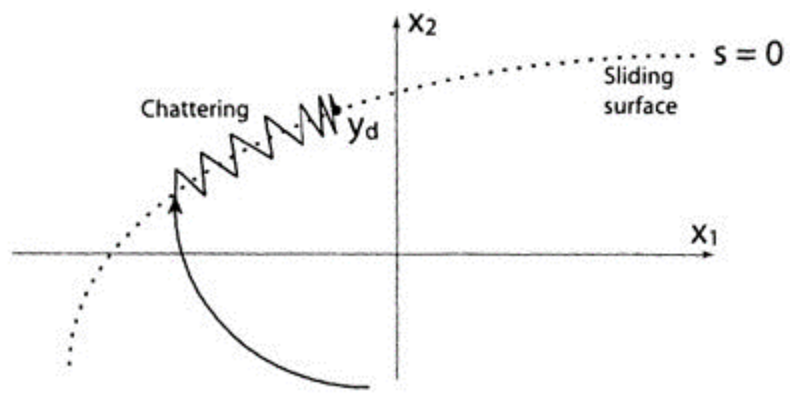
Upper Level Controller Design

- Linear Controller Design
  - Desired acceleration

$$a_{d,i} = -\frac{1}{h\left(\dot{x}_{r,i} + \lambda e_i\right)}$$

Upper Level Controller Design

- Example: Sliding mode control
  - Nonlinear control method that alters the dynamics of a nonlinear system by application of a discontinuous control signal that forces the system to „slide“ along a cross-section of the system‘s normal behavior
  - Step 1: select a sliding surface (which is stable)
  - Step 2: design a control law which will attract the system status to the sliding surface and remain there



Source: https://www.globalspec.com/reference/21394/160210/chapter-5-4-2-sliding-mode-control

Upper Level Controller Design 1

- Simple sliding mode control for ACC
- Define the sliding surface
  - $S_1 = e_i = R_i - T_h v_i$, where e_i is the range error, R_i is the the range of i-th vehicle, T_h is the time headway
- Design a control law
  - $\dot{S}_1 = \dot{e}_i = \lambda S_1$, where λ is the convergence rate to the sliding surface
- Actual input is acceleration, so if we re-arrange by substituting S
  - $a_i = \frac{\lambda}{T_h} e_i + \frac{1}{T_h} \dot{R}_i$
- What does it mean?
  - Smaller time gap means aggressive control (larger acceleration)
  - And higher risk of losing string stability $T_h \geq 2\tau$

String Stability Analysis

- Consider a group of vehicles that form a string in dense traffic
- $d_i = \frac{1}{s} v_i$
- $v_i = G_i(s) \cdot v_{i-1}$
- $G_i(s)$ is the speed transfer function of i-th vehicle
- $\epsilon_i = d_{i-1} - d_i - L$ (range error)
- $\epsilon_{vi} = v_{i-1} - v_i$ (range rate error)
- Let $L_i = T_h \cdot v_i$
- Propagation transfer function becomes,
- $\bar{G}_{i,k} = \frac{\epsilon_{i+k}}{\epsilon_i} = G_i \cdot G_{i+1} \cdot G_{i+2} \cdots G_{i+k-1} \cdot \frac{1 - G_{i+k} - s \cdot T_h \cdot G_{i+k}}{1 - G_i - s \cdot T_h \cdot G_i}$

Remark

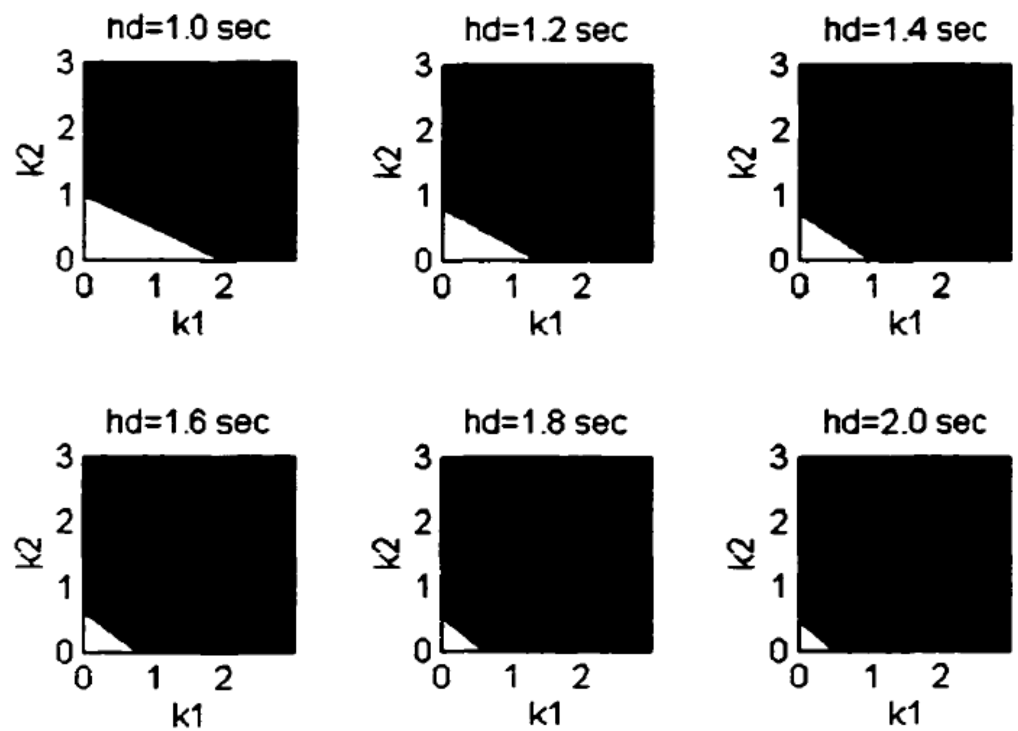
- $\frac{\epsilon_i}{\epsilon_{i-1}} = \frac{\epsilon_{vi}}{\epsilon_{vi-1}} = \frac{R_i}{R_{i-1}} = \frac{v_i}{v_{i-1}} = G$
- Substituting all the equations from the previous page
- $\frac{\epsilon_i}{\epsilon_{i-1}} = \frac{1/s(1-G_i-s{\cdot}T_h{\cdot}G_i)v_{i-1}}{1/s(1-G_{i-1}-s{\cdot}T_h{\cdot}G_{i-1})v_{i-2}} = \frac{1/s(1-G-s{\cdot}T_h{\cdot}G)Gv_{i-2}}{1/s(1-G-s{\cdot}T_h{\cdot}G)v_{i-2}} = G$
- By similar derivation process
- $\frac{\epsilon_{vi}}{\epsilon_{vi-1}} = G$ and $\frac{R_i}{R_{i-1}} = G$

String Stability Analysis

- If the ideal vehicle model is assumed
- $\dot{x}_i = A_i x_i + B_i u_i$
- $x_i = \begin{bmatrix} d_i \\ v_i \end{bmatrix}, A_i = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B_i = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$
- Let's study P-control and constant time-headway controller
- $u_i = k_1 \cdot (d_{i-1} - d_i - T_h v_i) + k_2(v_{i-1} - v_i)$
- Substituting the control law into state space equation and $R_i = d_{i-1} - d_i$ gives
- $\ddot{R}_i + (k_2 + k_1 T_h) \cdot \dot{R}_i + k_1 R_i = k_2 \dot{R}_{i-1} + k_1 \cdot R_{i-1}$
- Range propagation function is defined as
- $\left|\frac{R_i(s)}{R_{i-1}(s)}\right| = \left|\frac{k_2 s + k_1}{s^2 + (k_2 + k_1 T_h)s + k_1}\right|$
- The above function is 1 if $\omega = 0$

String Stability Analysis

- *Range propagation function*
- $\left|\frac{R_i(s)}{R_{i-1}(s)}\right| = \left|\frac{k_2 s + k_1}{s^2 + (k_2 + k_1 T_h)s + k_1}\right|$
- The above function is 1 if $\omega = 0$
- <1 for $\forall \omega > 0, k_2 = \frac{2 - k_1 T_h^2}{2T_h}$
- The controller is string stable only in the gray area



Upper Level Controller Design 2

- Sliding surface method of controller design

$$S_i = \dot{\epsilon}_i + \frac{\omega_n}{\xi + \sqrt{\xi^2 - 1}} \frac{1}{1 - C_1} \epsilon_i + \frac{C_1}{1 - C_1} (v_i - v_l)$$

where

$\dot{S}_i = -\lambda S_i$, with $\lambda = \omega_n(\xi + \sqrt{\xi^2 - 1})$

- The desired acceleration of the vehicle is then given by

$$\ddot{x}_{i,des} = (1 - C_1)\ddot{x}_{i,des} + C_1 \ddot{x}_l - 2\left(2\xi - C_1\left(\xi + \sqrt{\xi^2 - 1}\right)\right)$$

$$\omega_n \dot{\epsilon}_i - \left(\xi + \sqrt{\xi^2 - 1}\right) \omega_n C_1 (v_i - v_l) - \omega_n^2 \epsilon_i$$

- The control gains to be tuned are C_1,ξ, ω_n
  - C_1: $0 \le C_1 \le 1$, can be viewed as weighting on the lead vehicle‘s speed and acceleration
  - ξ: can be viewed as the damping ratio, critical damping if 1
  - ω_n: bandwidth of the controller

Upper Controller Design 2

- $\dot{S}_i = -\lambda S_i$, with $\lambda = \omega_n(\xi + \sqrt{\xi^2 - 1})$, ensures the system converges to the sliding surface
- Prior research shows that the system is „string stable“
  - D. Swaroop, et al., „String Stability of Interconnected Systems,“ IEEE Transactions on Automatic Control, 1996
- Robusness of the controller
  - To lags induced by the lower-level controller can also be guaranteed
- Setting $C_1 = 0$, we have the following classical second-order system

$$\ddot{x}_{i,des} = \ddot{x}_{i-1} - 2\xi\omega_n\dot{\epsilon}_i - \omega^2\epsilon_i$$

More Sophisticated Upper-Level Control?

- Control with information of “r” preceding vehicles
- Mini-platoon control strategy
  - Information from the lead vehicle increases the robustness
  - Why don’t we divide a platoon into multiple mini-platoons and have more lead vehicle information?
- Model predictive control
  - Various objectives possible
    - Minimizing gap regulating error
    - Preserving string stability
    - Driver comfort
    - Minimizing fuel consumption

Lower Level Controller

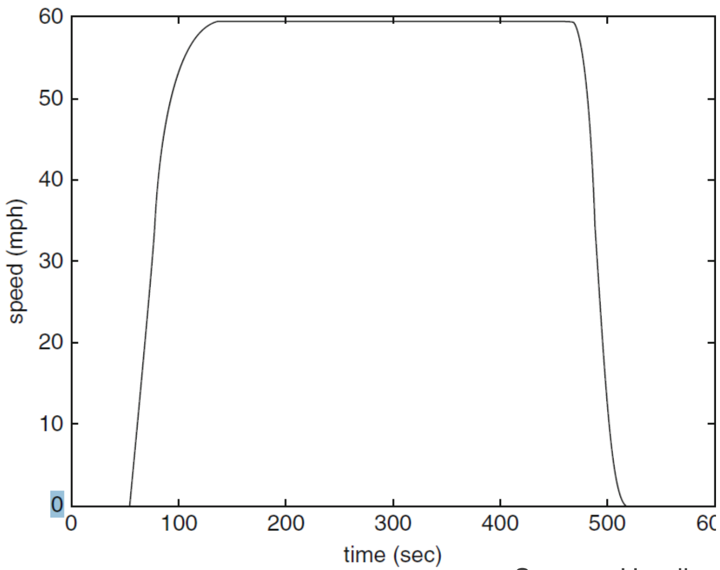
- Lower level controller
  - Throttle and brake actuator puts are determined so as to track the desired acceleration
  - Again, standard sliding surface control technique
  - If the torque is chosen as $T_{net,i} = \frac{J_e}{Rh}\ddot{x}_{i_{des}} + \left[c_a R^3 h^3 \omega_e^2 + R\left(hF_f + T_{br}\right)\right]_j$, then the acceleration of the vehicle equals the desired acceleration defined by the upper level controller $\ddot{x}_i = \ddot{x}_{i_{des}}$
  - The map $T_{net}(\omega_e, m_a)$ is inverted to obtain the desired air mass flow in engine $m_{a_{des}}$
  - A single surface controller is then used to calculate the throttle angle α to make m_a track $m_{a_{des}}$

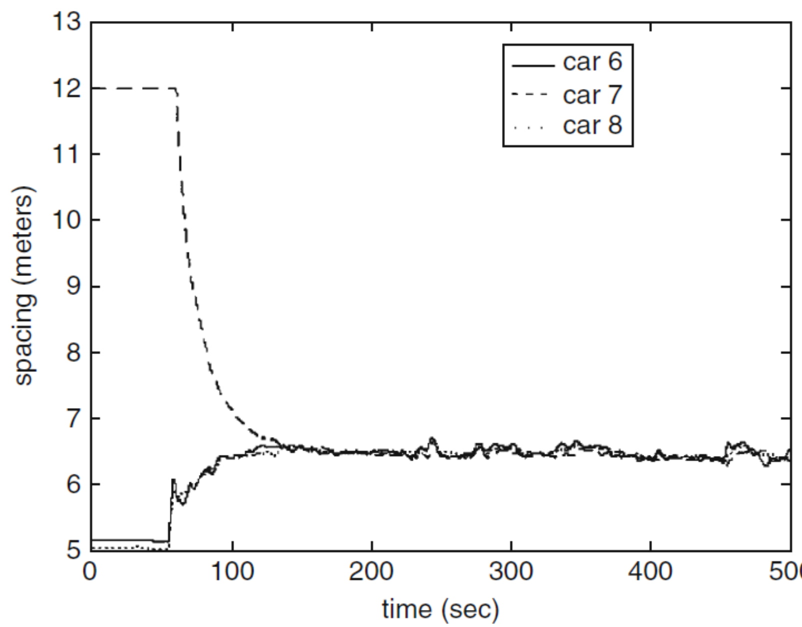
Lower Level Controller

- Define the surface as $s_2 = m_a - m_{a_{des}}$
- Setting $\dot{s}_2 = -\eta_2 s_2$,
$$MAX \cdot TC(\alpha)PRI(m_a) = \dot{m}_{ao} - \dot{m}_{a_{des}} - \eta_2 s_2$$
- Since $TC(\alpha)$ is invertible, he desired throttle angle can be calculated
- If the desired torque is negative, brake actuators is used to provide he desired torque

Experimental Results from PATH Project

- Lead vehicle velocity profile
- Convergence of inter-vehicle distance

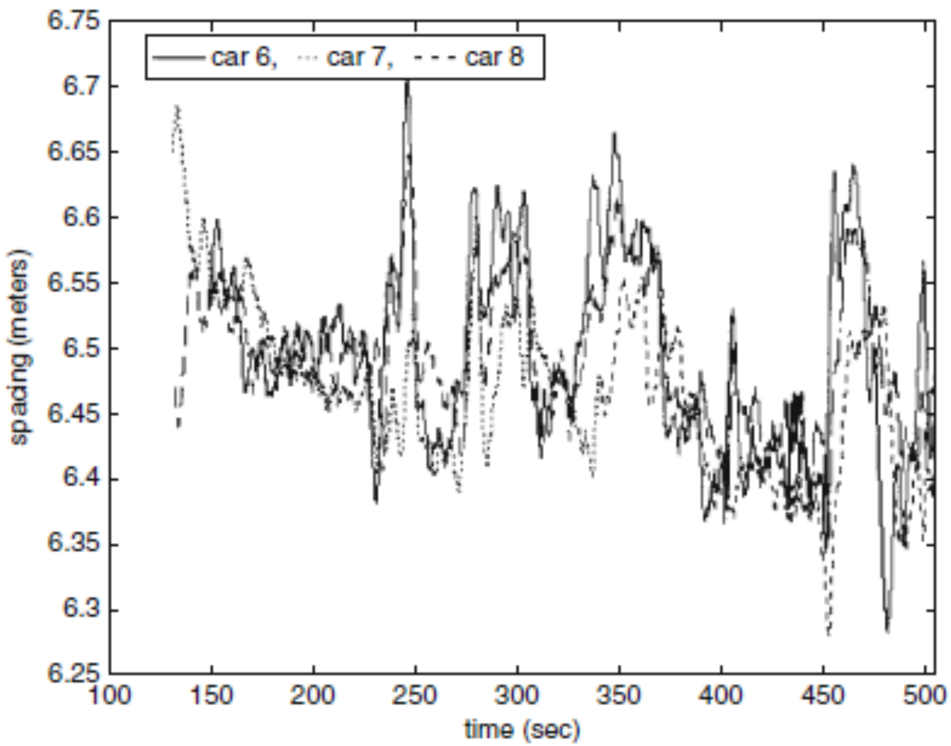




Source: Handbook of Intelligent Vehicles

Experimental Results from PATH Project

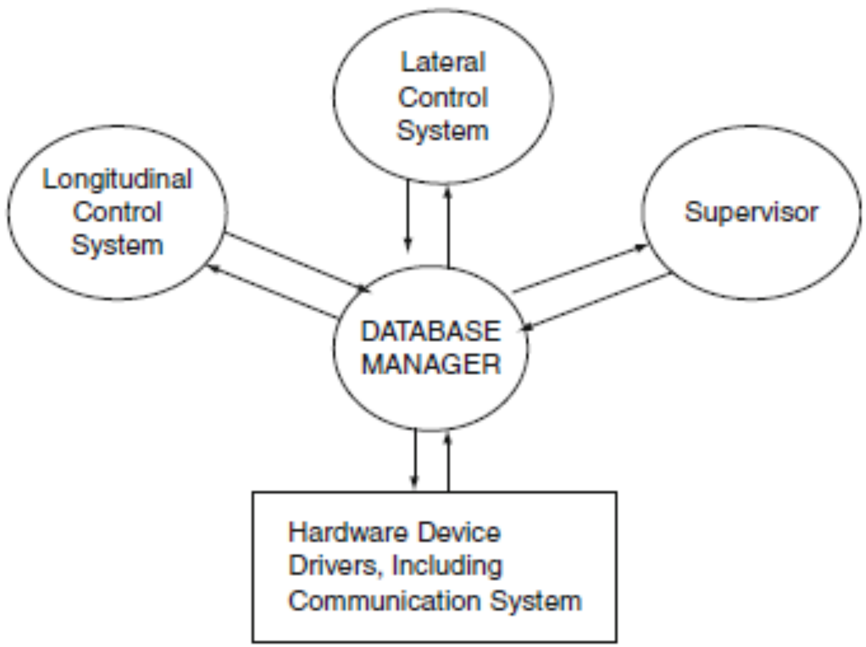
- Response to disturbance
  - Uphill, downhill



Source: Handbook of Intelligent Vehicles

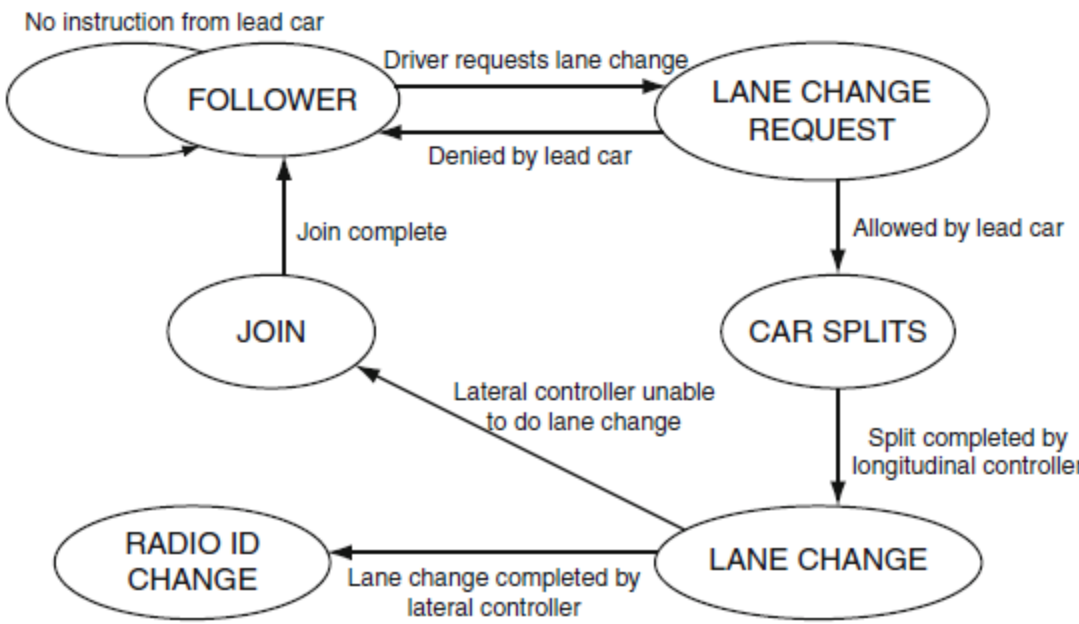
Integration with Lateral Control System

- Lane control and longitudinal control can be performed mostly independently of each other
- Coordination needed when joining or exiting a platoon
- Supervisor coordinates longitudinal and lateral control



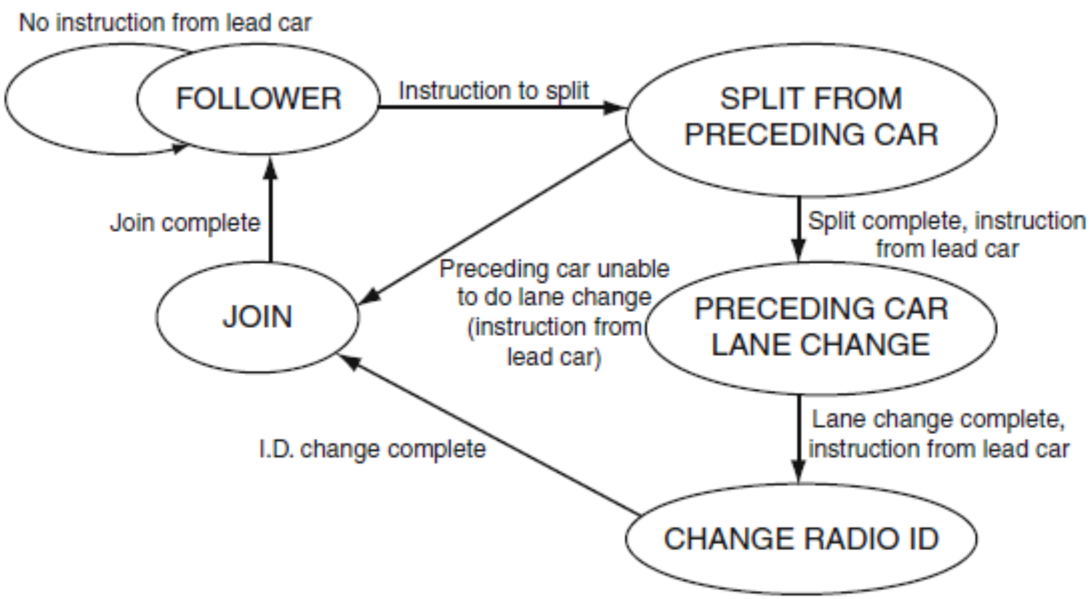
Integration with Lateral Control System

- Supervisor of the vehicle requesting to exit from a platoon



Integration with Lateral Control System

- Supervisor of the follower vehicle, which splits from the preceding car



References

- „Tutorial on Control Theory“, Stefan Simrock, ITER, 2011
- J. Zhou, et al., „Range policy of adaptive cruise control vehicles for improved flow stability and string stability,“ IEEE Transactions on Intelligent Transportation Systems, 2005
- L. Xiao, et al., “Practical String Stability of Platoon of Adaptive Cruise Control Vehicles”, IEEE Transactions on Intelligent Transportation Systems, 2011
- C.Y. Liang, “Traffic-Friendly Adaptive Cruise Control Design”, Dissertation, U. Mich. 2000

Lecture 6: Practical Issues in Digital Control

Basic Platooning Implementation

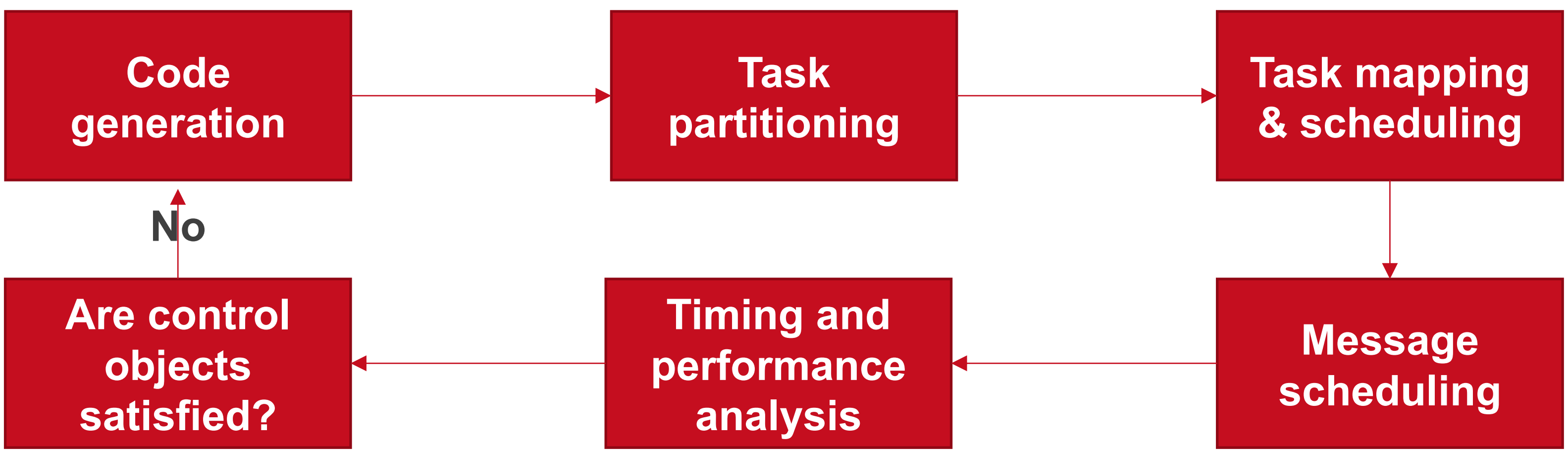
Prof. Sangyoung Park

Module ”Vehicle-2-X: Communication and Control“

Control System Design

- Controller design
  - Using equations

- Controller implementation

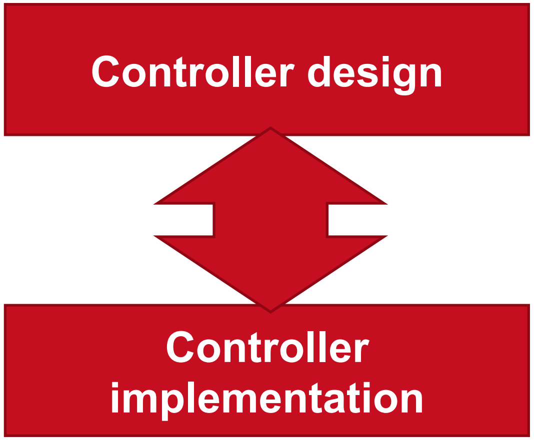


Control System Design

- Assumptions in controller design (control theorist)
  - Infinite numerical accuracy
  - Computing control law takes negligible time
  - No delay from sensor to controller
  - No delay from controller to actuator
  - No jitter
- Controller implementation (Embedded systems engineer)
  - Fix-precision arithmetic
  - Tasks have non-negligible execution times
  - Often large message delays
  - Time- and event-triggered communication

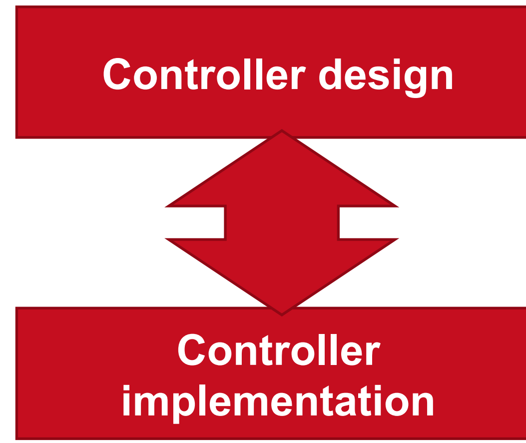
Semantic Gap

- There is a gap between model and implementation
- Control theorist:
  - “These are implementation details. Not my problem!”
- Embedded systems engineer:
  - “Model-level assumptions are not satisfied by implementation”
- Research questions
  - How do we quantify this gap?
  - How should we close this gap?
- Solution: Controller/architecture co-design



Implementation-Aware Controller Design

- Performnace metrics have been different for computer science domain and control algorithms
- Control algorithms are evaluated by
  - Stability
  - Settling time
  - Peak overshot
  - ...
- Computer programs are evaluated by
  - Computation time
  - Communication bandwidth
  - Memory footprint
  - Enegy consumption
  - ...

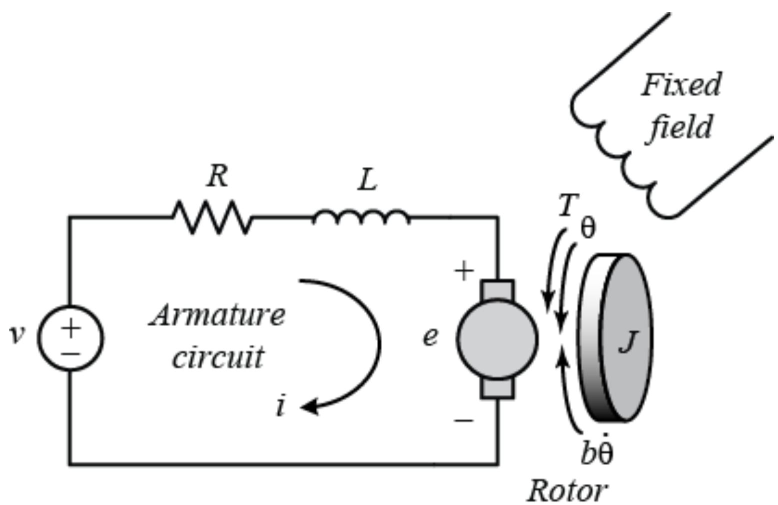


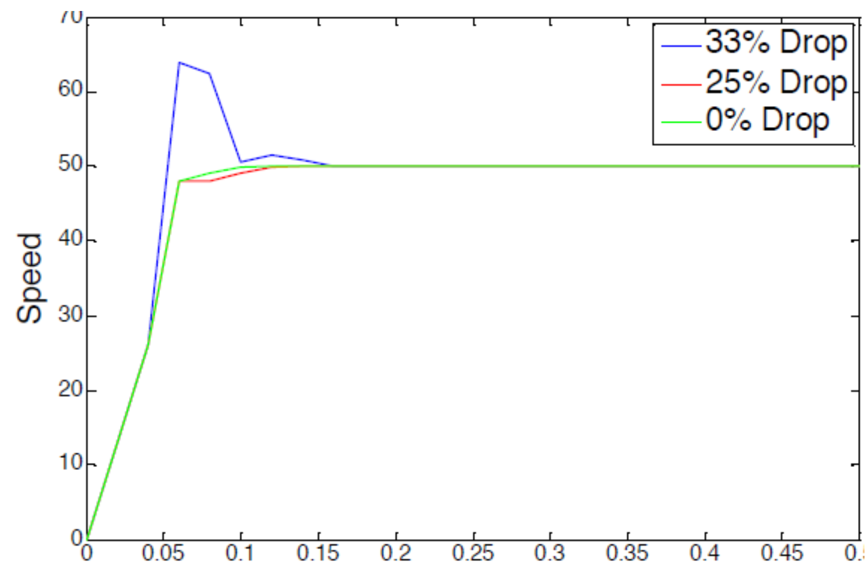
Control Task Characteristics

- The deadlines are ***not hard*** for control-related messages
- What does it mean deadline are ***hard*** or ***soft***?
  - Hard deadline: something catastrophic happens when a control task is not finished withint the given deadline
    - Aircraft crashes, battery explodes, etc
  - Soft deadline: there is degradation in performance, but a deadline miss to a certain degree is tolerable
    - Video streaming frame rate drop, etc

Control Task Characteristics

- The deadlines are ***not hard*** for control-related messages
- DC motor $\frac{d}{dt}\begin{bmatrix}\dot{\theta}\\ i\end{bmatrix} = \begin{bmatrix}-\frac{b}{J} & -\frac{K}{J}\\ -\frac{K}{L} & -\frac{R}{L}\end{bmatrix}\begin{bmatrix}\dot{\theta}\\ i\end{bmatrix} + \begin{bmatrix}0\\ \frac{1}{L}\end{bmatrix}V \rightarrow \dot{x}(t) = Ax(t) + Bu(t)$
- Objective: $\dot{\theta} = 50$
- As samples drop (ar

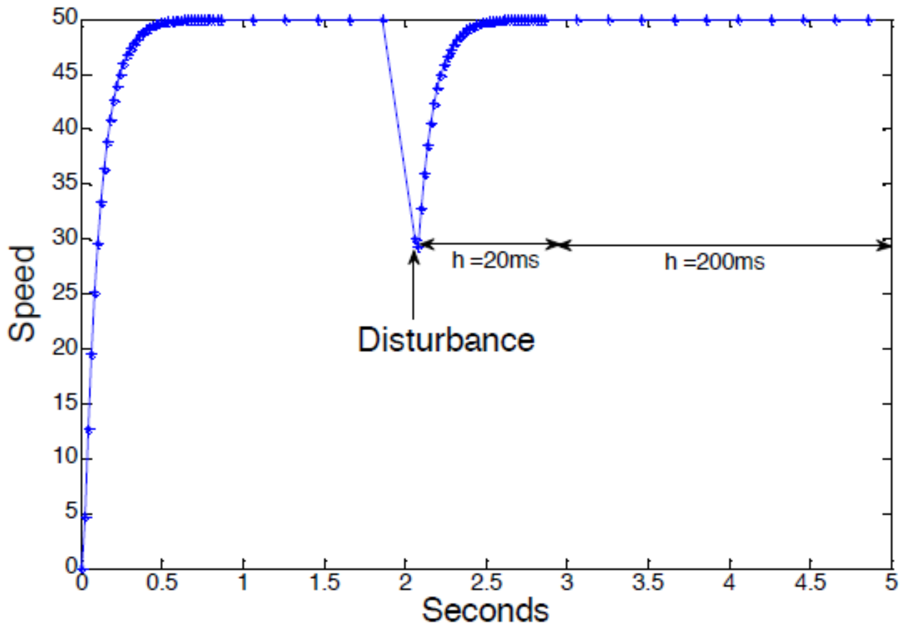




http://ctms.engin.umich.edu/CTMS/index.php?example=MotorSpeed&section=SystemModeling

Control Task Characteristics

- Sensitivity of control performance depends on the ***state*** of the controlled plant
- The computation requirement at the steady-state is less, i.e., sampling frequency can be reduced (e.g., event-triggered sampling)
- The communication requirements are less at steady-state, (e.g., ower priority can be assigned to the feedback signals)

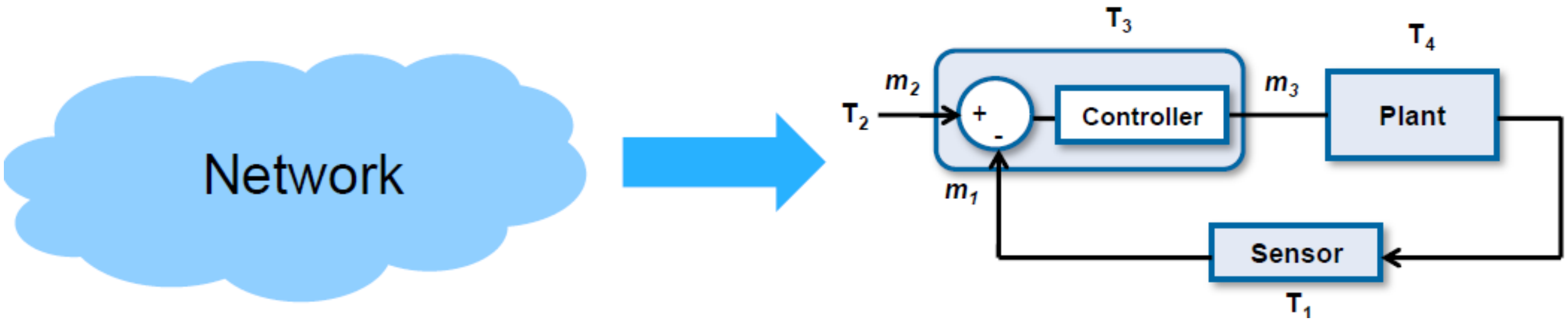


Bottomline

- Traditional Emedded control system design
  - Meeting deadilnes is of paramount importance
- Co-design
  - Deadline takes the back seat
  - Design space become bigger
  - Resuling design is robust, cost-effective, ..
- Design objectives shift from low level metrics like deadlines to metric governing system dynamics (like stability)

What about NCS?

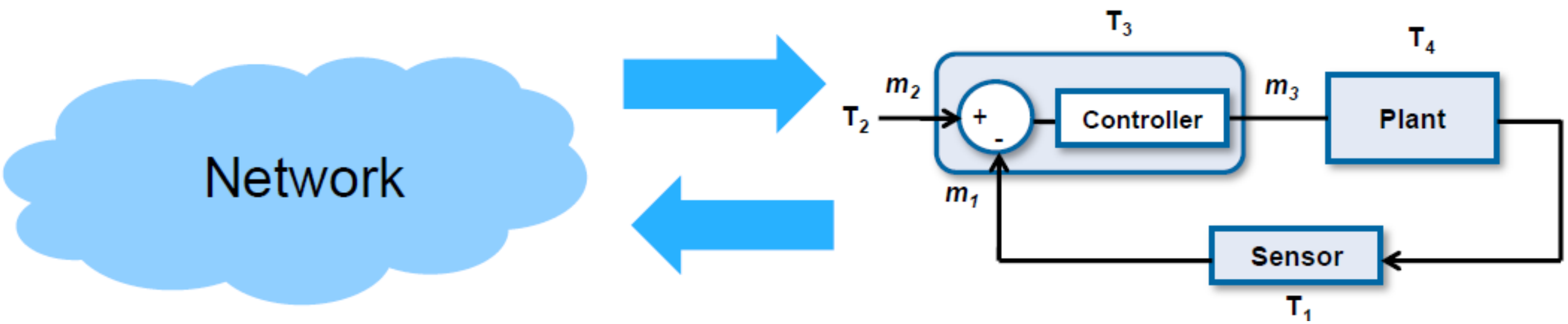
- Networked Computer Systems



- Take network characteristics into account when desining the control laws
  - Packet drops, delays, jitter, ...

What about NCS?

- Arbitrated networked control systems



- ANCS – we can design the network
  - By taking into account control performance constraints
- Problem: How to design the network?
- Given a network, how to design the controller?
  - NCS problem
- Co-design problem: How to design the network and the controller together?

References

- Samarjit Chakraborty, „Embedded Control Systems“, TU Munich