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

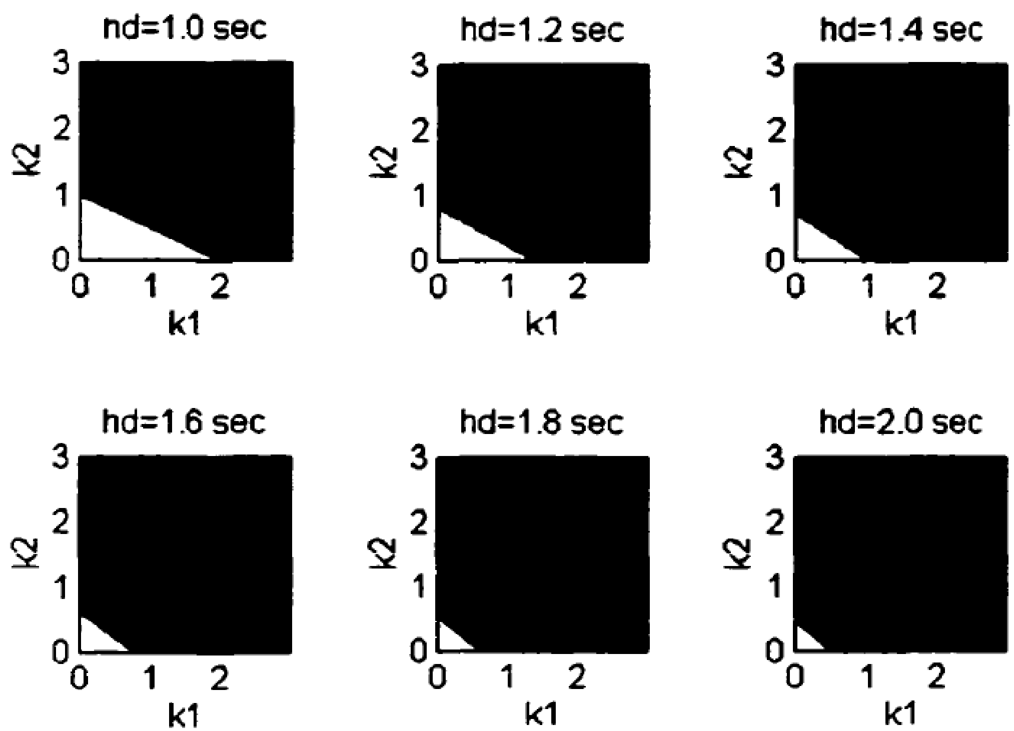
- $\frac{\epsilon_i}{\epsilon_{i-1}} = \frac{\epsilon_{vi}}{\epsilon_{vi-1}} = \frac{R_i}{R_{i-1}} = \frac{v_i}{v_{i-1}} = G$
- Substituting all the equations from the previous page
- $\frac{\epsilon_i}{\epsilon_{i-1}} = \frac{1/s(1-G_i-s \cdot T_h \cdot G_i)v_{i-1}}{1/s(1-G_{i-1}-s \cdot T_h \cdot G_{i-1})v_{i-2}} = \frac{1/s(1-G-s \cdot T_h \cdot G)Gv_{i-2}}{1/s(1-G-s \cdot T_h \cdot G)v_{i-2}} = G$
- 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 in to 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: $0 \leq C_1 \leq 1$, 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

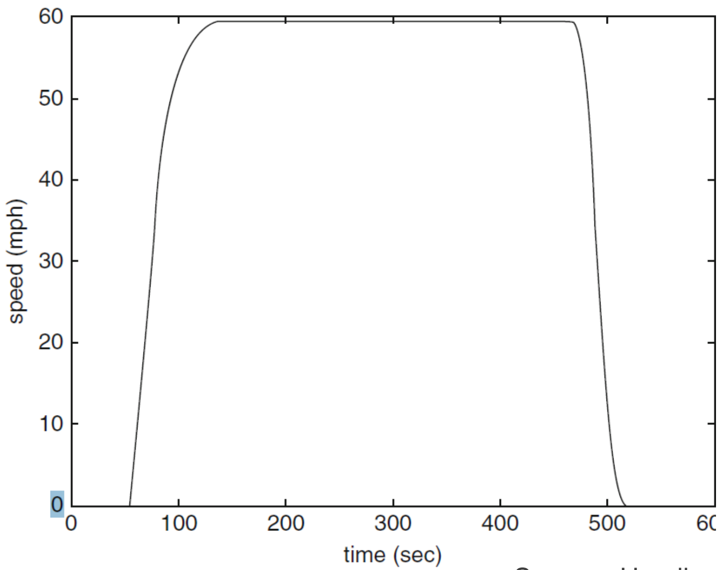
- Lower level controller
  - Throttle and brake actuator puts are determined so as to track the desired acceleration
  - Again, standard sliding surface control technique
  - If the torque is chosen as $T_{net,i} = \frac{J_e}{Rh}\ddot{x}_{i_{des}} + \left[c_a R^3 h^3 \omega_e^2 + R\left(hF_f + T_{br}\right)\right]_j$, 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 $m_{a_{des}}$

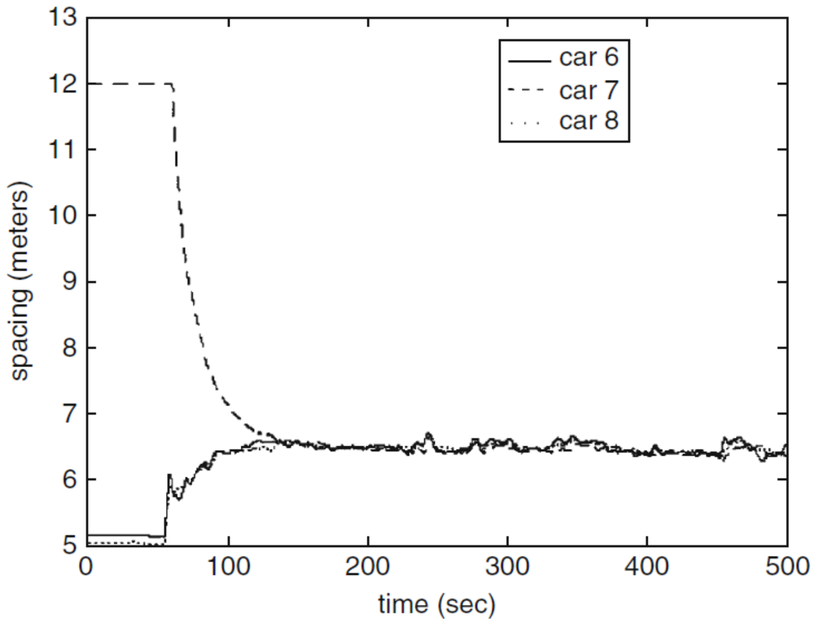
Lower Level Controller

- 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

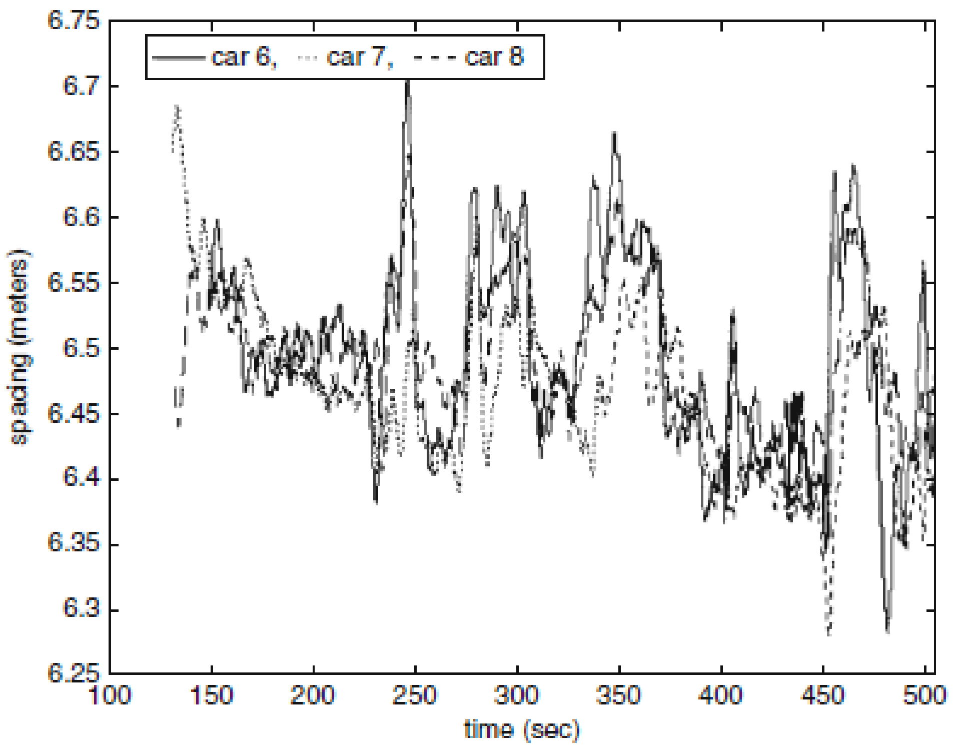




Source: Handbook of Intelligent Vehicles

Experimental Results from PATH Project

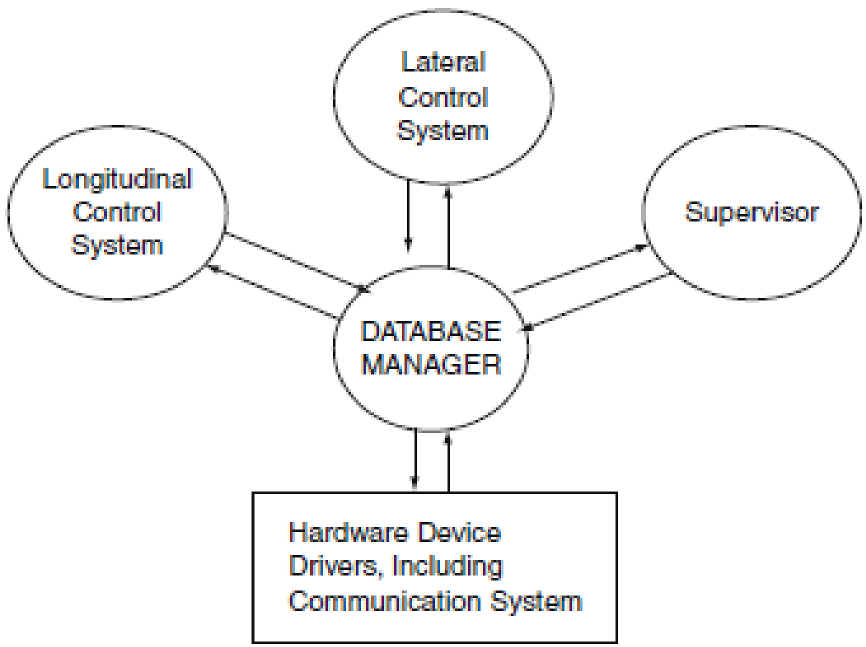
- Response to disturbance
  - Uphill, downhill



Source: Handbook of Intelligent Vehicles

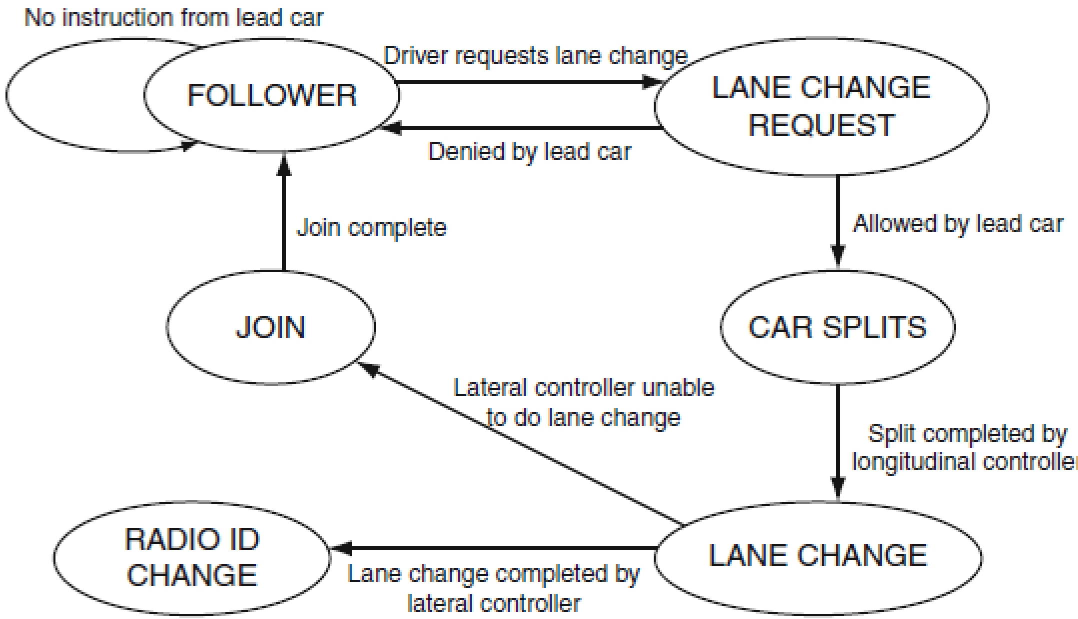
Integration with Lateral Control System

- 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



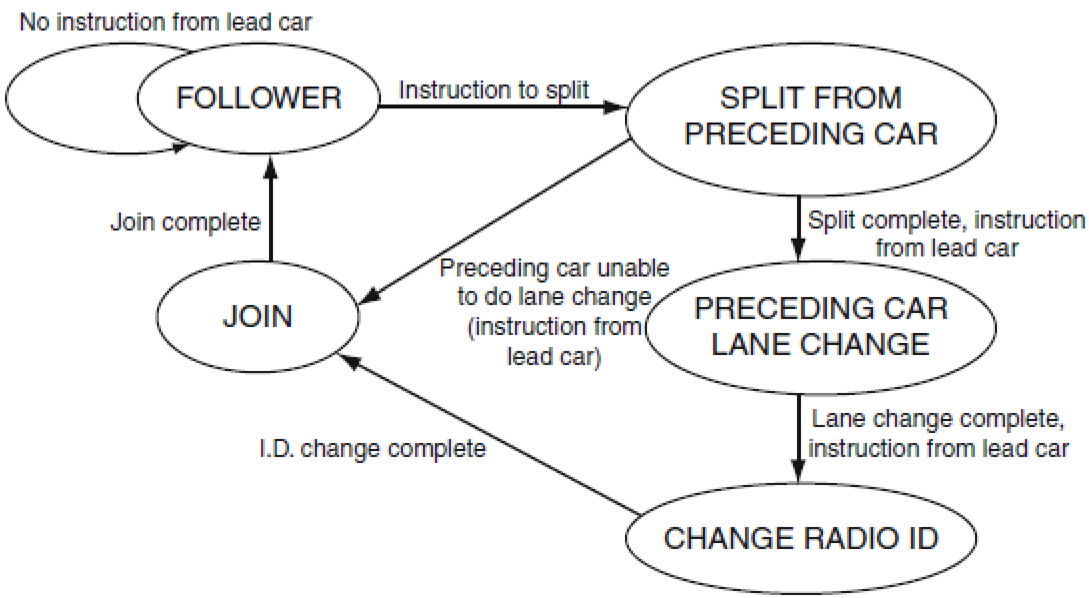
Integration with Lateral Control System

- Supervisor of the vehicle requesting to join 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

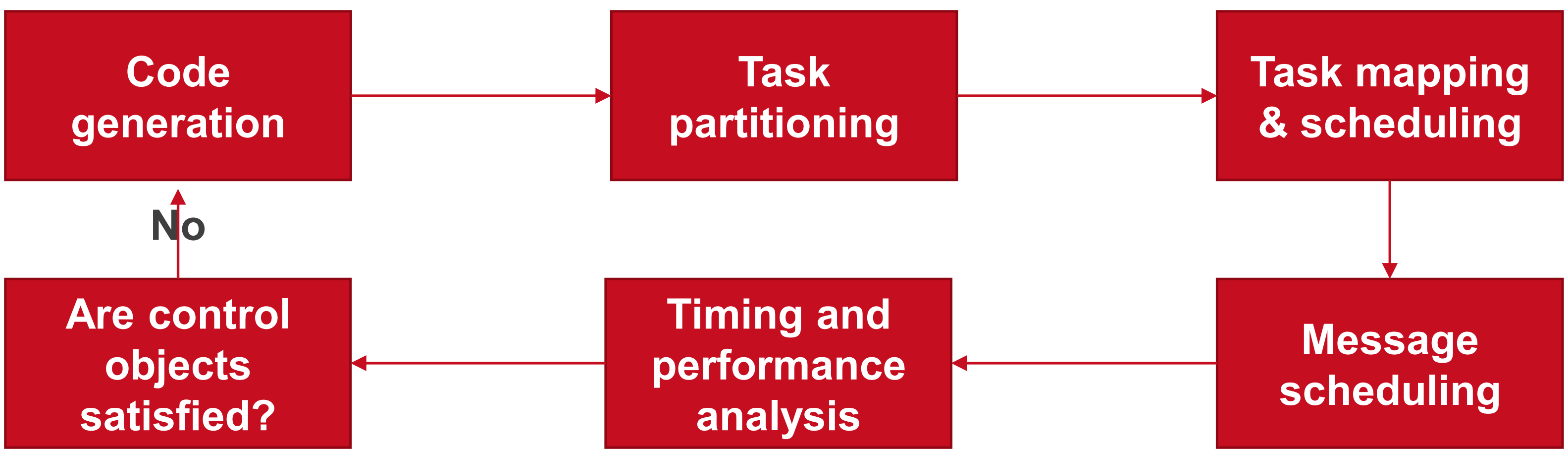
Prof. Sangyoung Park

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

Control System Design

- Controller design
  - Using equations

- Controller implementation

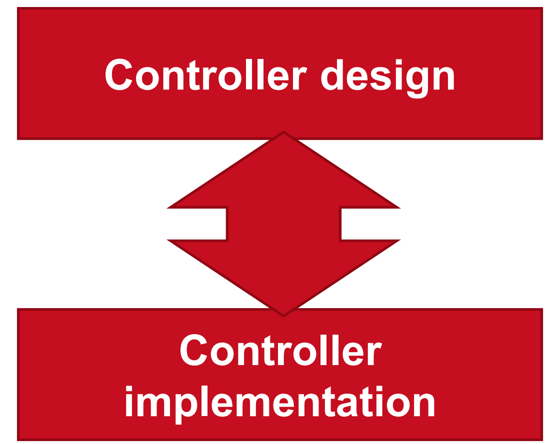


Control System Design

- 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

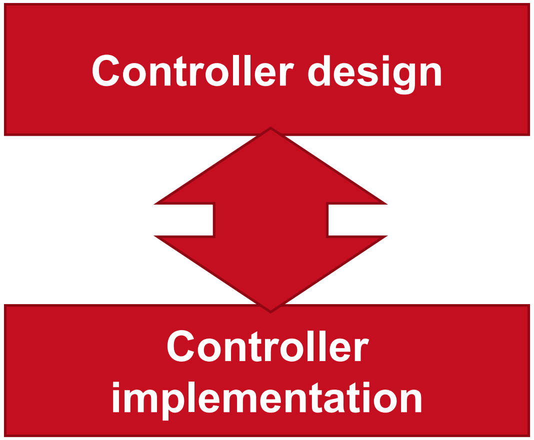
Semantic Gap

- 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
  - ...

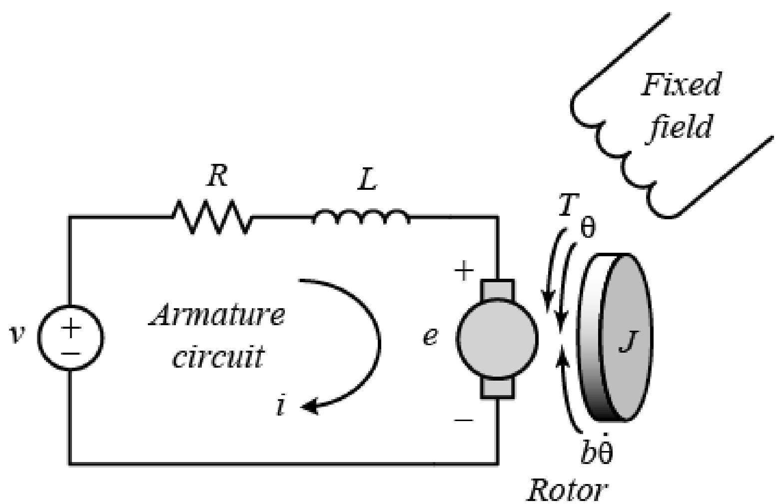


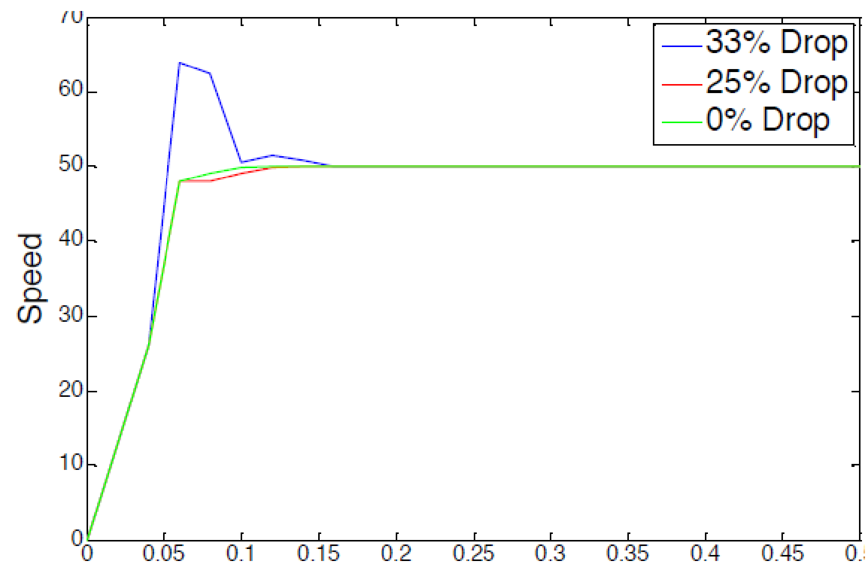
Control Task Characteristics

- 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

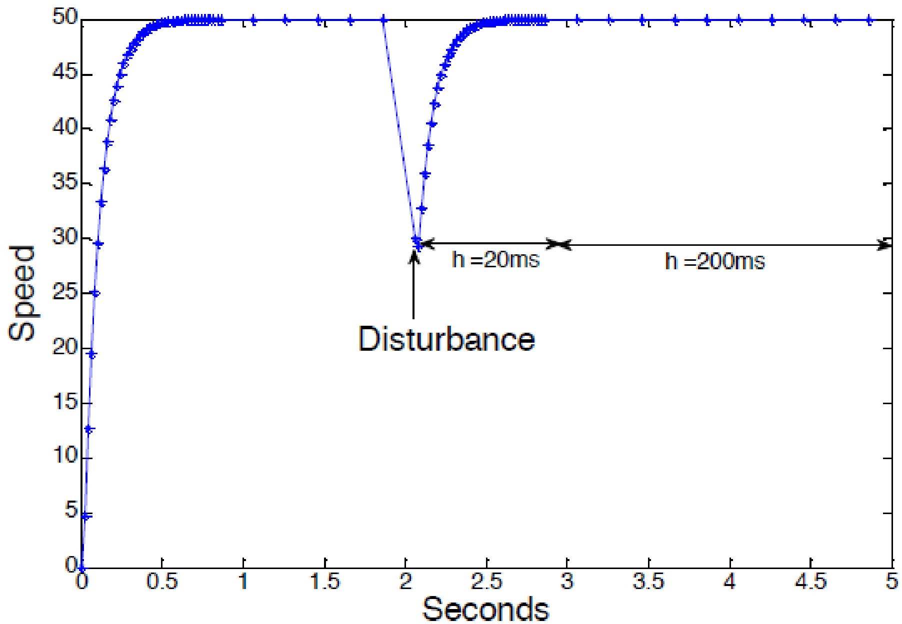




http://ctms.engin.umich.edu/CTMS/index.php?example=MotorSpeed&section=SystemModeling

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)

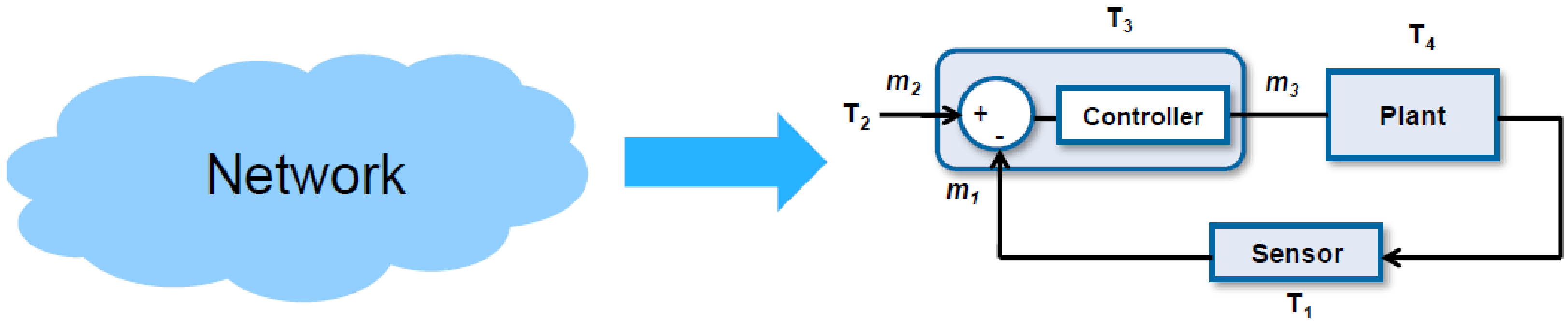


Bottomline

- 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?

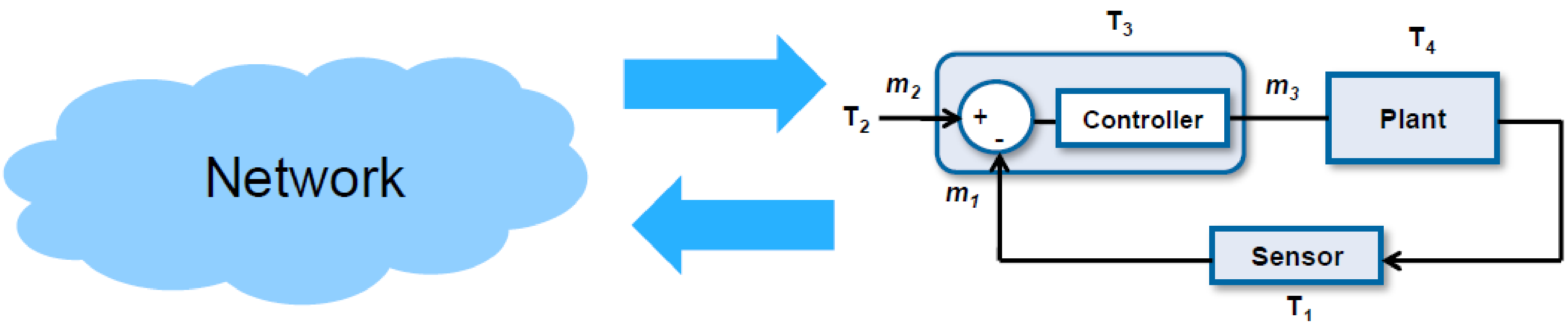
- Networked Computer Systems



- 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