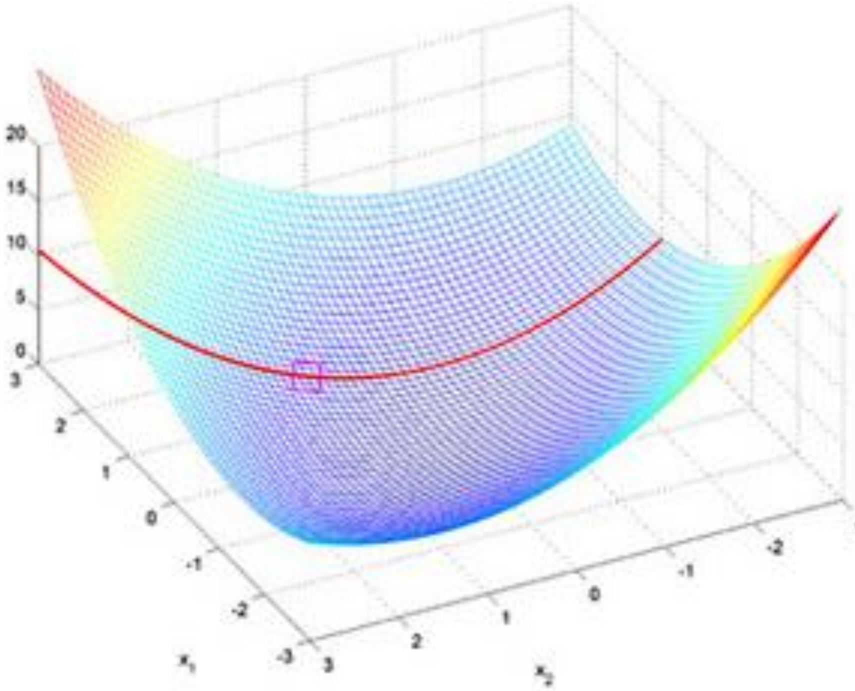
MEGR 3090/7090/8090: Advanced Optimal Control

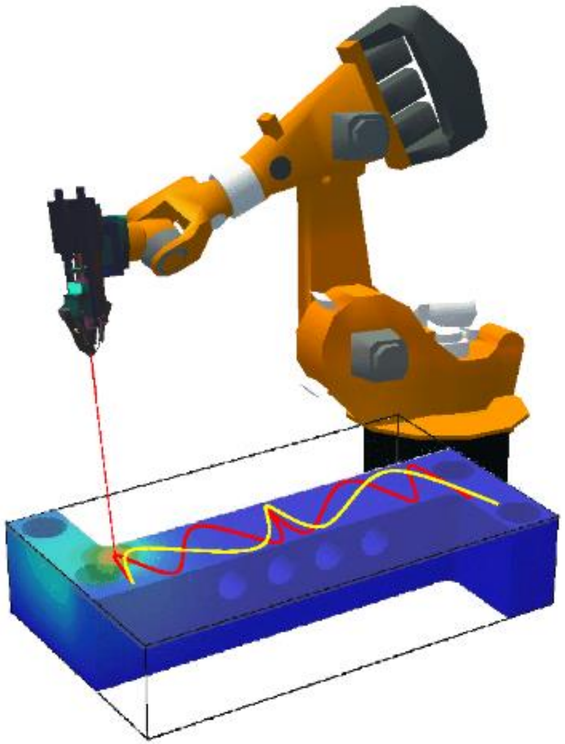
UNC CHARLOTTE



$$\boxed{V_n(\mathbf{x}_n) = \min_{\{\mathbf{u}_n, \mathbf{u}_{n+1}, \cdots, \mathbf{u}_{N-1}\}} \left[\frac{1}{2}\sum_{k=n}^{N-1}\left(\mathbf{x}_k^T \mathbf{Q}_k \mathbf{x}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k\right) + \frac{1}{2}\mathbf{x}_N^T \mathbf{Q}_N \mathbf{x}_N \right]}$$

$$\begin{aligned} V_n(\mathbf{x}_n) &= \min_{\{\mathbf{u}_n, \mathbf{u}_{n+1}, \cdots, \mathbf{u}_{N-1}\}} \left[\frac{1}{2}\sum_{k=n}^{N-1}\left(\mathbf{x}_k^T \mathbf{Q}_k \mathbf{x}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k\right) + \frac{1}{2}\mathbf{x}_N^T \mathbf{Q}_N \mathbf{x}_N \right] \\ &= \min_{\mathbf{u}_n} \left[\frac{1}{2}\left(\mathbf{x}_n^T \mathbf{Q}_n \mathbf{x}_n + \mathbf{u}_n^T \mathbf{R} \mathbf{u}_n\right) + \underbrace{\min_{\{\mathbf{u}_{n+1}, \cdots, \mathbf{u}_{N-1}\}} \left[\frac{1}{2}\sum_{k=n+1}^{N-1}\left(\mathbf{x}_k^T \mathbf{Q}_k \mathbf{x}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k\right) + \frac{1}{2}\mathbf{x}_N^T \mathbf{Q}_N \mathbf{x}_N \right]}_{V_{n+1}(\mathbf{x}_{n+1})} \right] \\ &= \min_{\mathbf{u}_n} \left[\frac{1}{2}\left(\mathbf{x}_n^T \mathbf{Q}_n \mathbf{x}_n + \mathbf{u}_n^T \mathbf{R} \mathbf{u}_n\right) + V_{n+1}(\mathbf{x}_{n+1}) \right] \end{aligned}$$

$$\boxed{V_n(\mathbf{x}_n) = \min_{\mathbf{u}_n} \left[\frac{1}{2}\left(\mathbf{x}_n^T \mathbf{Q}_n \mathbf{x}_n + \mathbf{u}_n^T \mathbf{R} \mathbf{u}_n\right) + V_{n+1}(\mathbf{x}_{n+1}) \right]}$$



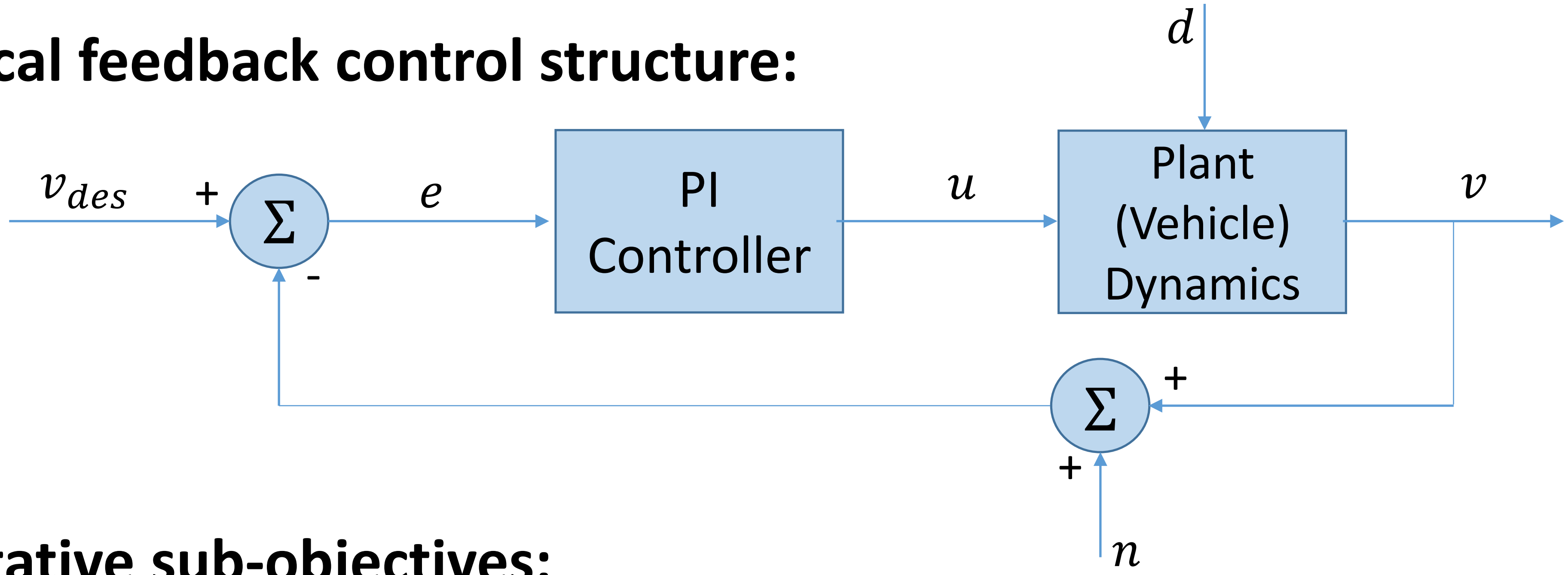
Lecture 1

August 22, 2017

Cruise control: A classic control system designed with classical control techniques

Basic objective: Drive vehicle speed (v) to user-specified target (v_{des})

Classical feedback control structure:



Qualitative sub-objectives:

- *Track* setpoints, v_{des}
- *Reject* disturbances, d
- *Reject* noise, n

Cruise Control: Classical Steady-State Objectives

Steady-state requirements (assumes closed-loop stability):

- $\frac{v(0)}{v_{des}(0)} = 1$: Steady-state setpoint tracking (constant setpoints)

- $\frac{v(0)}{d(0)} = 0$: Steady-state disturbance rejection (constant disturbances)

Cruise Control: Classical Time Domain Objectives

All time constants must be faster than τ_{des}:

- Given that the poles of the system transfer functions are given by $p_i, i = 1 \ldots n$, we require $Re(p_i) < -\frac{1}{\tau_{des}}, i = 1 \ldots n$

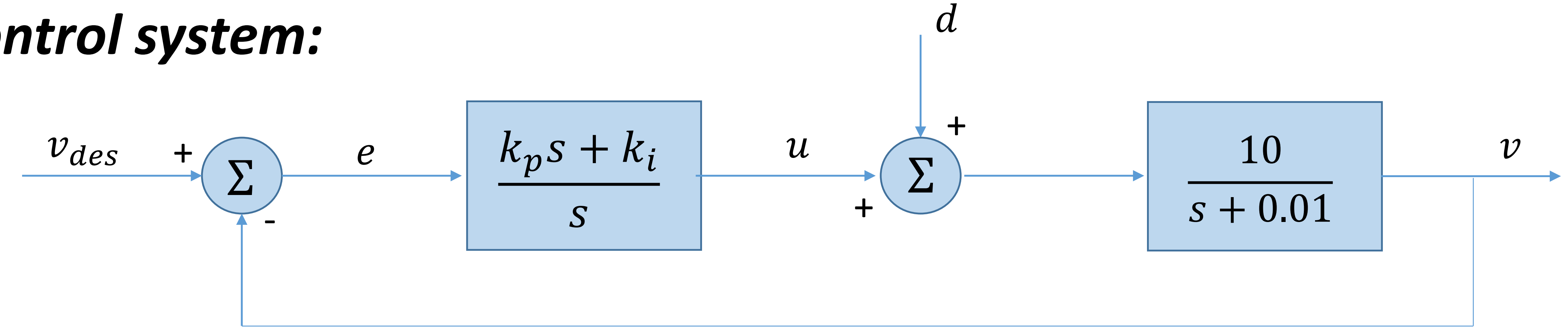
Closed-loop system should be overdamped:

- Given that the poles of the system transfer functions are given by $p_i, i = 1 \ldots n$, we require $Im(p_i) = 0, i = 1 \ldots n$

Other (example) time domain specifications: Rise time, settling time, overshoot

Designing a Cruise Control System to Achieve Time Domain Objectives - Example

Control system:



Objectives:

- Steady-state tracking of constant v_{des} and rejection of constant d
- Closed-loop time constants of 10 seconds or faster
- Overdamped closed-loop system

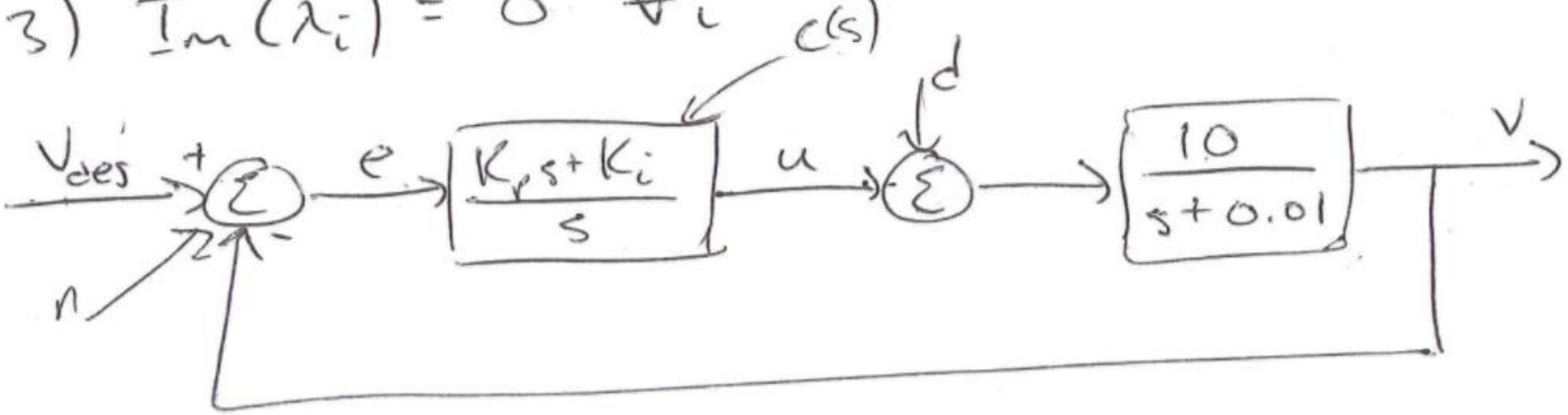
Designing a Cruise Control System to Achieve Time Domain Objectives - Example

Objectives:

1) $\dfrac{v(0)}{v_{des}(0)} = 1 \quad , \quad \dfrac{v(0)}{d(0)} = 0$

2) $Re(\lambda_i) \le -0.1 \quad \forall i$

3) $Im(\lambda_i) = 0 \quad \forall i$



Designing a Cruise Control System to Achieve Time Domain Objectives - Example

$$\frac{V(s)}{V_{des}(s)} = \frac{\frac{K_p s + K_i}{s}\,\frac{10}{s+0.01}}{1 + \frac{K_p s + K_i}{s}\cdot\frac{10}{s+0.01}}$$

$$= \frac{10(K_p s + K_i)}{s(s+0.01) + (K_p s + K_i)\cdot 10}$$

$$\Rightarrow \frac{V(0)}{V_{des}(0)} = 1 \text{ as long as}$$

K_i, K_p are stabilizing

$$\frac{V(s)}{d(s)} = \frac{\frac{10}{s+0.01}}{1 + \frac{10}{s+0.01}\cdot\frac{K_p s + K_i}{s}}$$

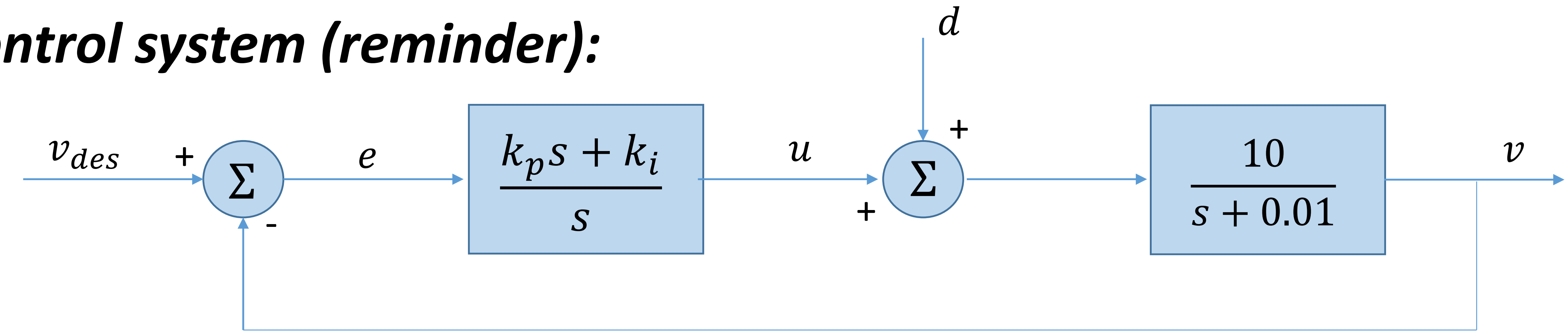
$$= \frac{10s}{s(s+0.01) + 10(K_p s + K_i)}$$

$$\Rightarrow \frac{V(0)}{d(0)} = 0 \text{ as long as}$$

K_i, K_p are stabilizing

Designing a Cruise Control System to Achieve Time Domain Objectives – Candidate Solutions

Control system (reminder):



Candidate solutions:

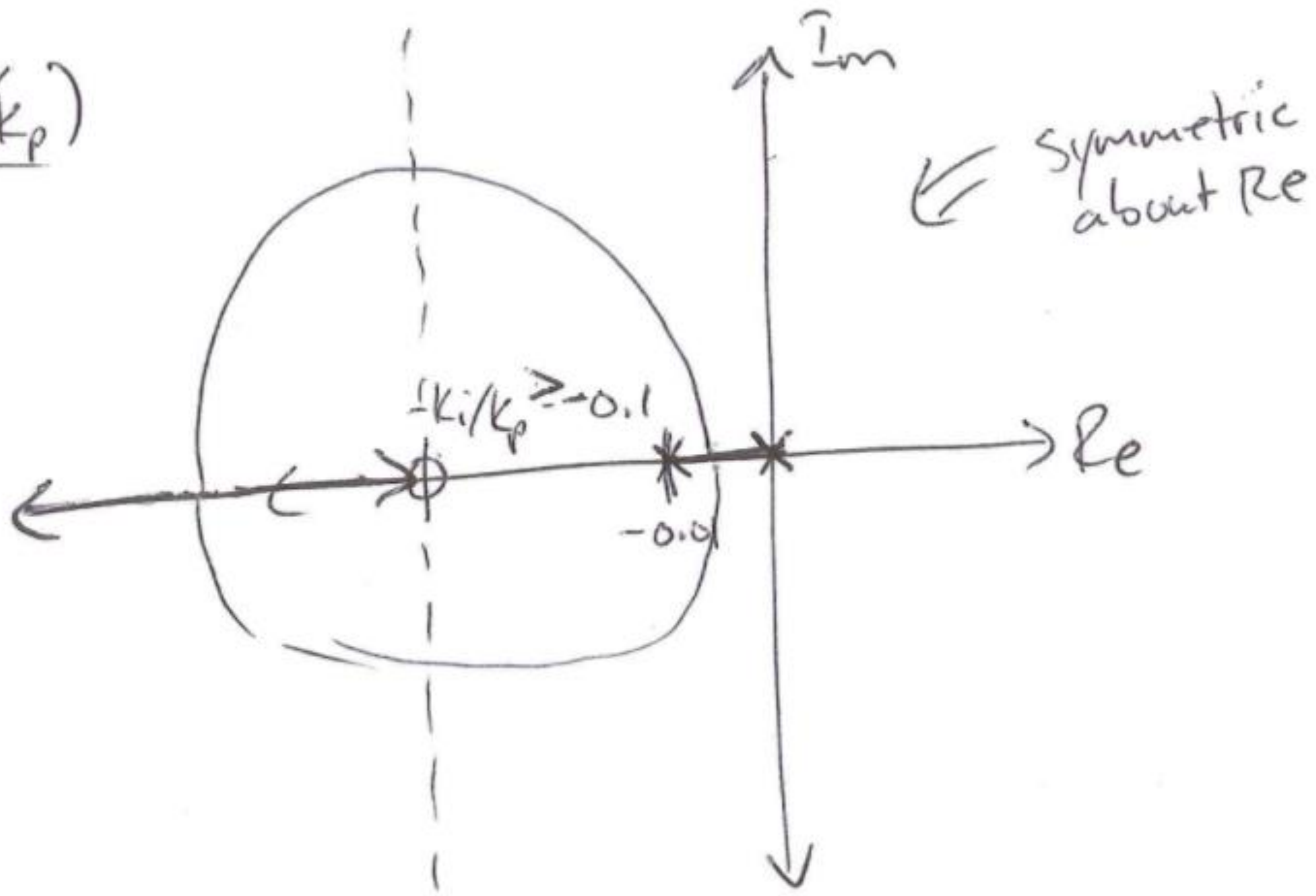
- Solution 1: $k_p = 0.08$, $k_i = 0.016$
- Solution 2: $k_p = 0.2$, $k_i = 0.04$

Questions:

- Which (if either, or maybe both) solutions achieve the prescribed objectives?
- Which solution is ***better***?

Designing a Cruise Control System to Achieve Time Domain Objectives – Candidate Solutions

$$C(s) = \frac{K_p\left(s + K_i/K_p\right)}{s}$$

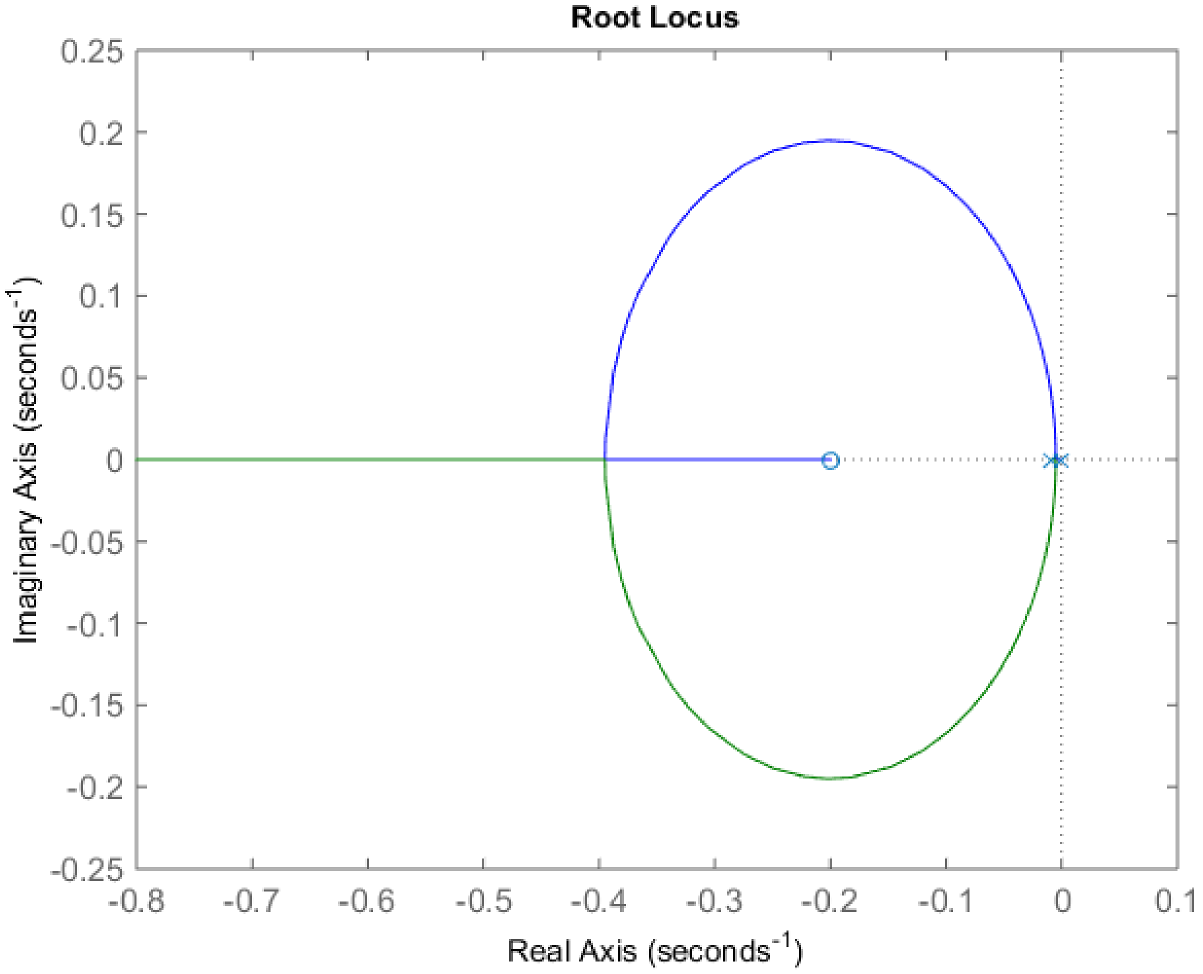


$\boxed{K_i/K_p \geq 0.1}$

Note: Both of the candidate controllers satisfy this requirement.

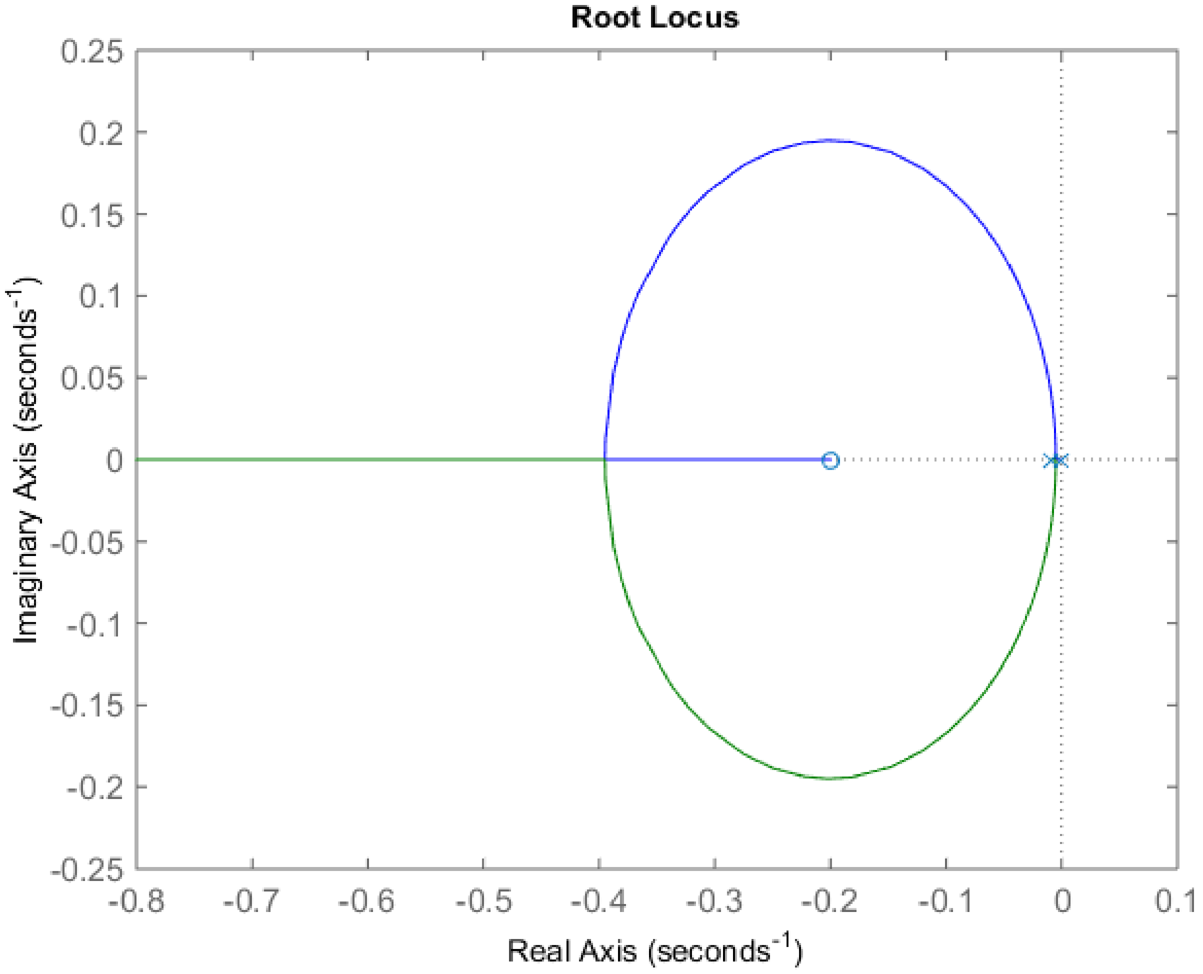
Designing a Cruise Control System to Achieve Time Domain Objectives – Assessment

Root locus for both candidate solutions (note that open loop zero and pole locations are the same in both cases, with $k_i/k_p = 0.2$***)***



Designing a Cruise Control System to Achieve Time Domain Objectives – Assessment

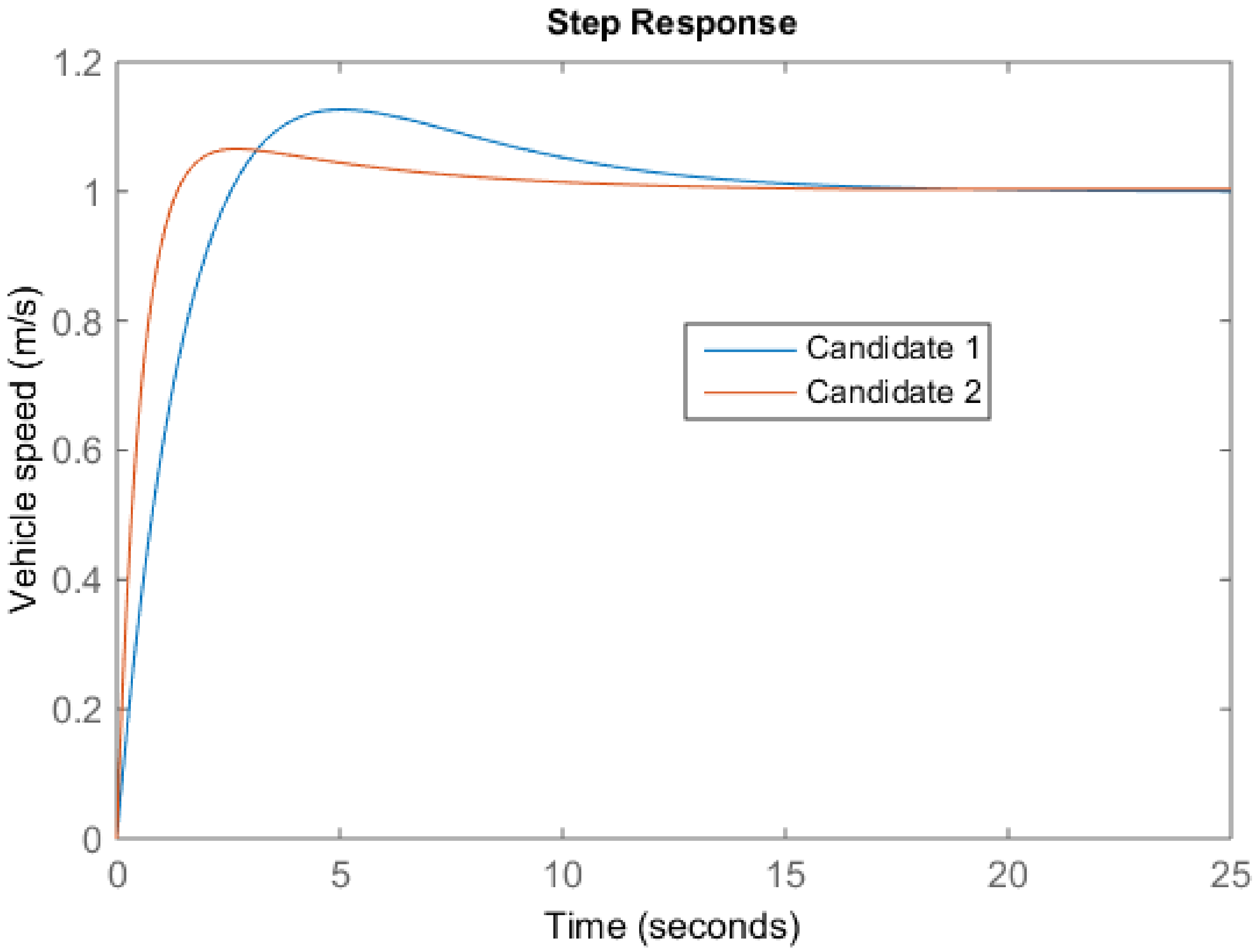
Root locus for both candidate solutions (note that open loop zero and pole locations are the same in both cases, with $^{k_i}/_{k_p} = 0.2$***)***



Conclusion: Both solutions achieve the required objectives…neither one is better than the other

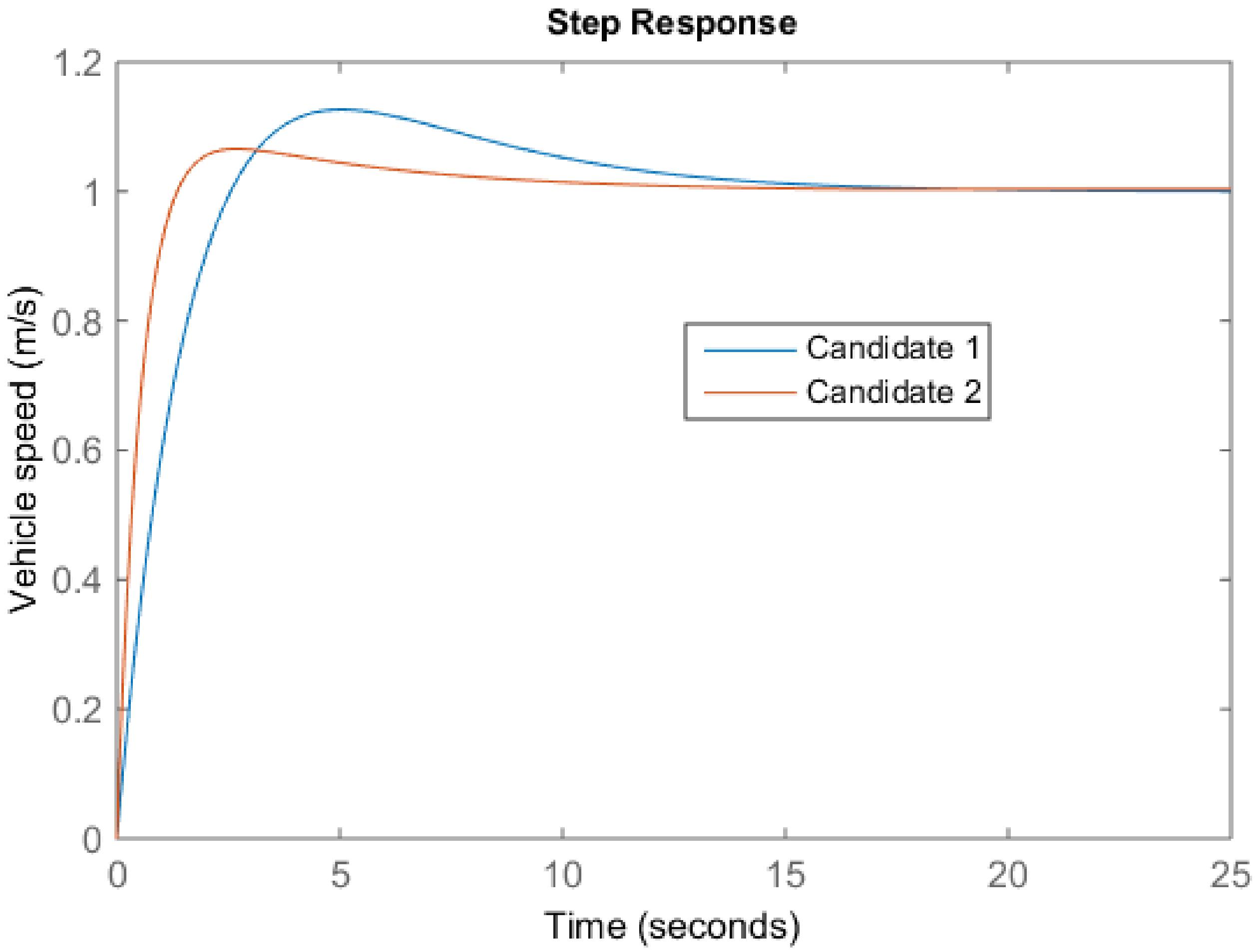
Designing a Cruise Control System to Achieve Time Domain Objectives – Assessment

Step response for both candidate solutions:



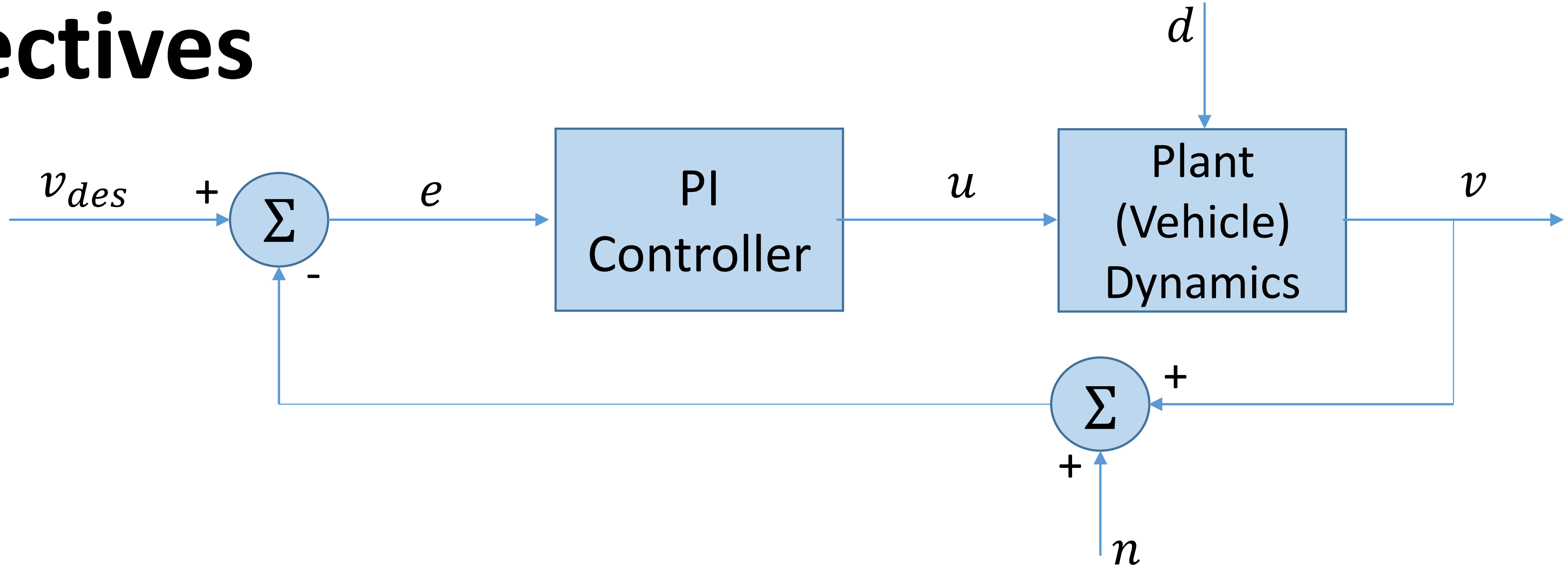
Designing a Cruise Control System to Achieve Time Domain Objectives – Assessment

Step response for both candidate solutions:



Conclusion (again): Both solutions achieve the required objectives…neither one is better than the other

Cruise Control: Classical Frequency Domain Objectives



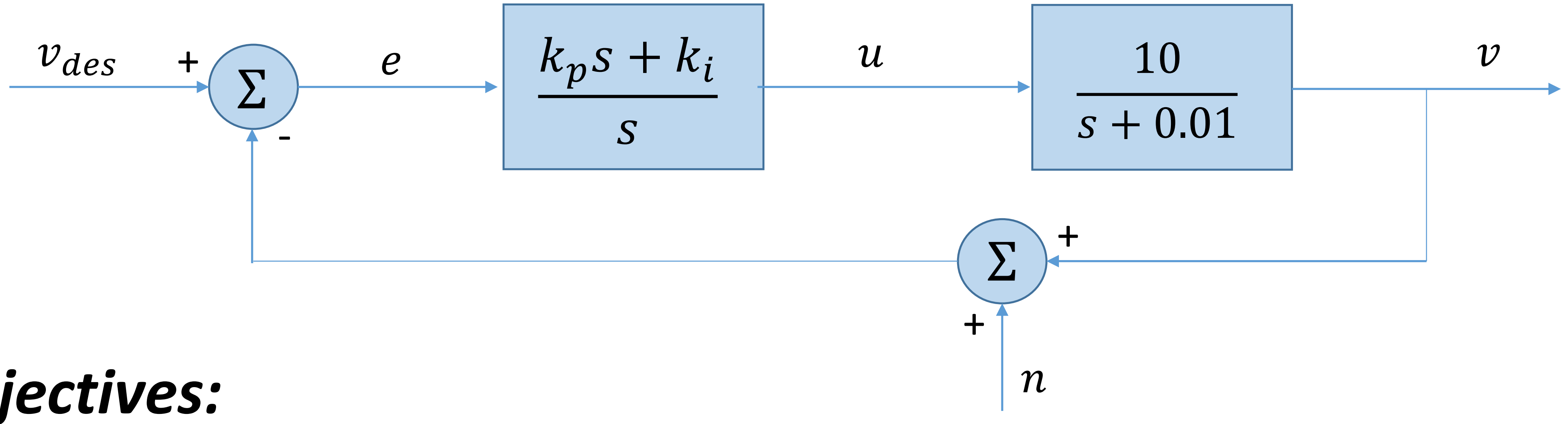
Main idea: Qualitative performance requirements (setpoint tracking, disturbance rejection, and noise rejection) are translated into transfer function magnitude requirements at critical frequencies:

$$\left|\frac{e(i\omega_{sp})}{v_{des}(i\omega_{sp})}\right| \leq K_{sp} \qquad \left|\frac{y(i\omega_d)}{d\ (i\omega_d)}\right| \leq K_d \qquad \left|\frac{y(i\omega_n)}{d\ (i\omega_n)}\right| \leq K_n$$

ω_{sp}, ω_d, ω_n = critical frequencies associated with the setpoint, disturbance, and noise, respectively

Designing a Cruise Control System to Achieve Frequency Domain Objectives - Example

Control system:



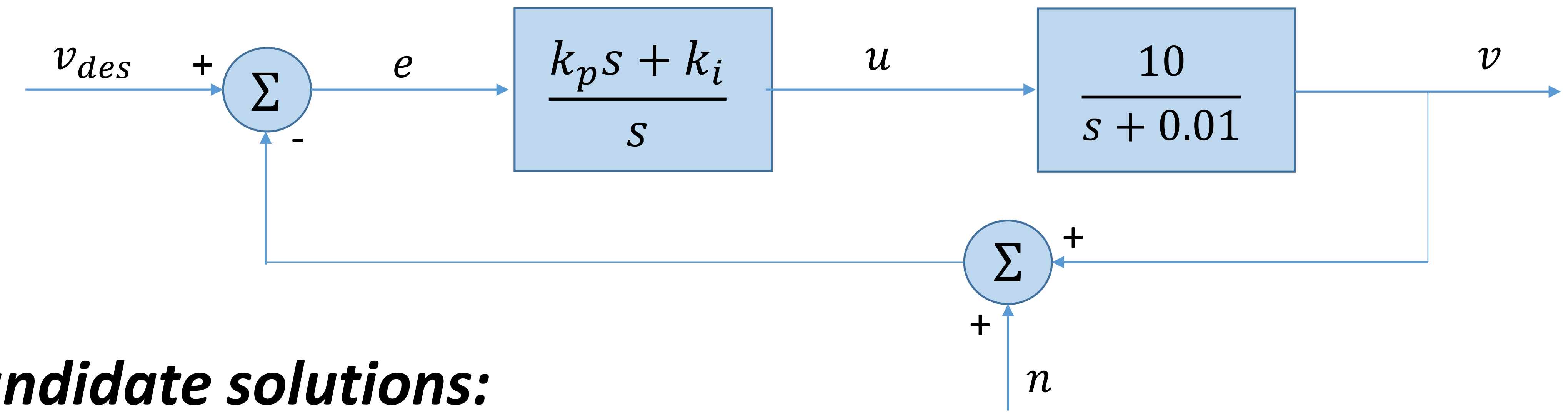
Objectives:

- Setpoint tracking: $\left|\frac{e(i\omega_{sp})}{v_{des}(i\omega_{sp})}\right| \leq 0.01, \omega_{sp} = 0$ rad/s
- Noise rejection: $\left|\frac{y(i\omega_n)}{n(i\omega_n)}\right| \leq 0.1, \omega_n = 100$ rad/s

Designing a Cruise Control System to Achieve Frequency Domain Objectives – Candidate Solutions

UNC CHARLOTTE

Control system (reminder):



Candidate solutions:

- Solution 1: $k_p = 0.08$, $k_i = 0.016$
- Solution 2: $k_p = 0.2$, $k_i = 0.04$

Questions:

- Which (if either, or maybe both) solutions achieve the prescribed objectives?
- Which solution is ***better***?

Designing a Cruise Control System to Achieve Frequency Domain Objectives – Candidate Solutions

$$\frac{v(s)}{n(s)} = \frac{-(10K_p s + 10K_i)}{s^2 + (0.01 + 10K_p)s + 10K_i}$$

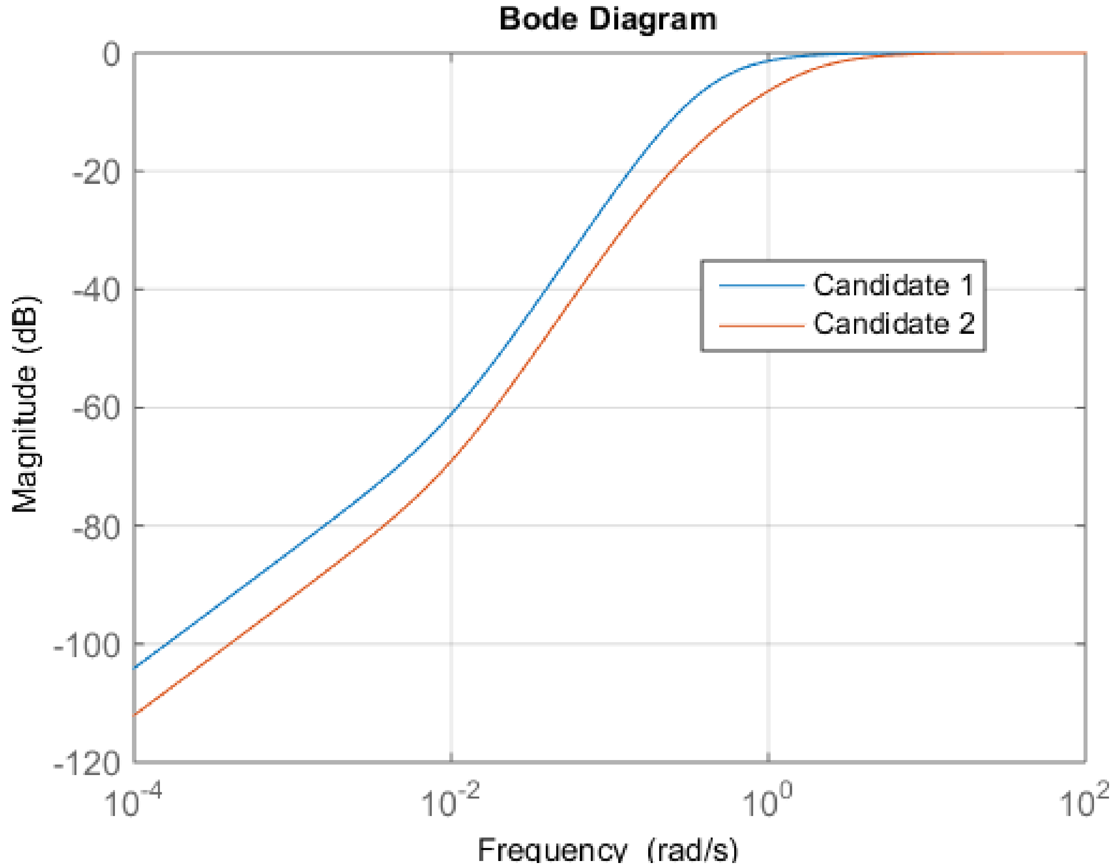
$$\frac{e(s)}{V_{des}(s)} = \frac{1}{1 + \frac{10}{s+0.01} \cdot \frac{K_p s + K_i}{s}} = \frac{s(s+0.01)}{s^2 + (0.01 + 10K_p)s + K_i \cdot 10}$$

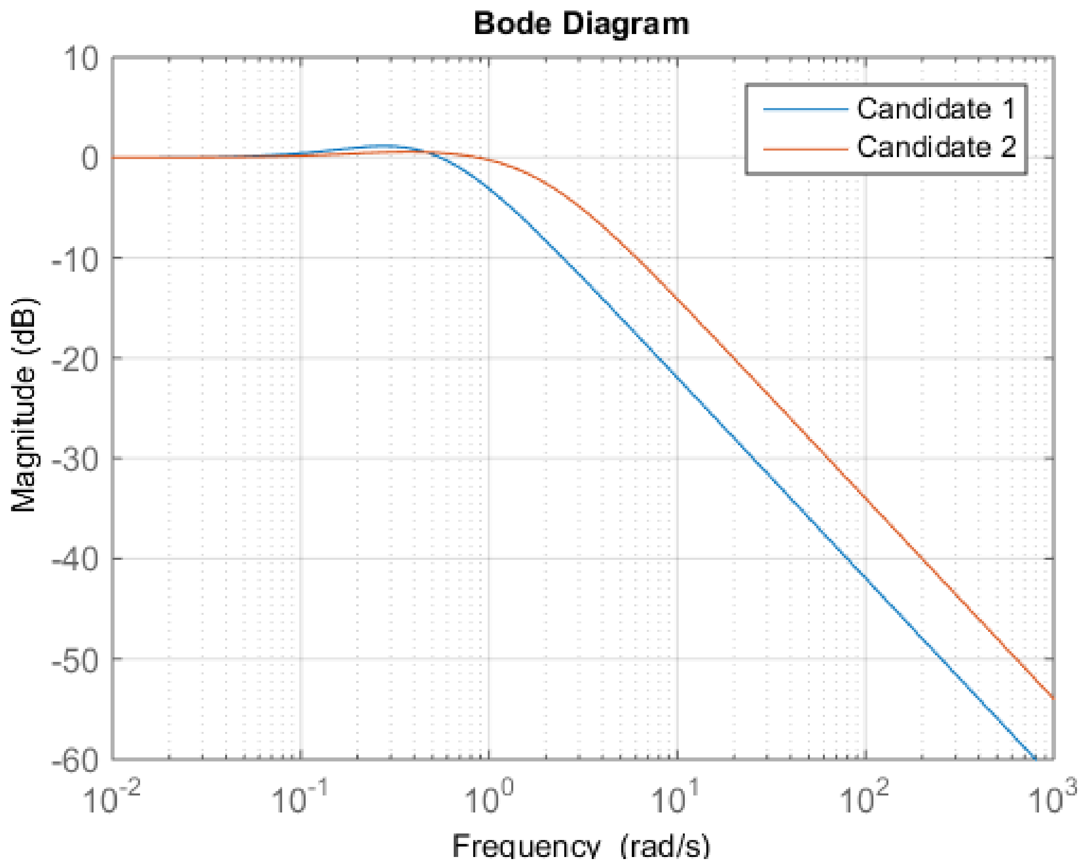
Notes from class:

- Frequency responses ($\frac{e(i\omega_{sp})}{v_{des}(i\omega_{sp})}$ and $\frac{y(i\omega_n)}{n(i\omega_n)}$) are obtained by replacing "s" with $i\omega_{sp}$
- When evaluating frequency-domain performance from an external input to an external output with a controller already in place (as we are here) we want to look at **closed-loop** Bode plots
- If, on the other hand we were trying to determine how much we could increase a control gain (implying that the controller was not already finalized) and still retain closed-loop stability, we would want to look at the **open-loop** Bode plot to evaluate **stability margins** (this is not what we're doing in this example)

Designing a Cruise Control System to Achieve Frequency Domain Objectives – Assessment

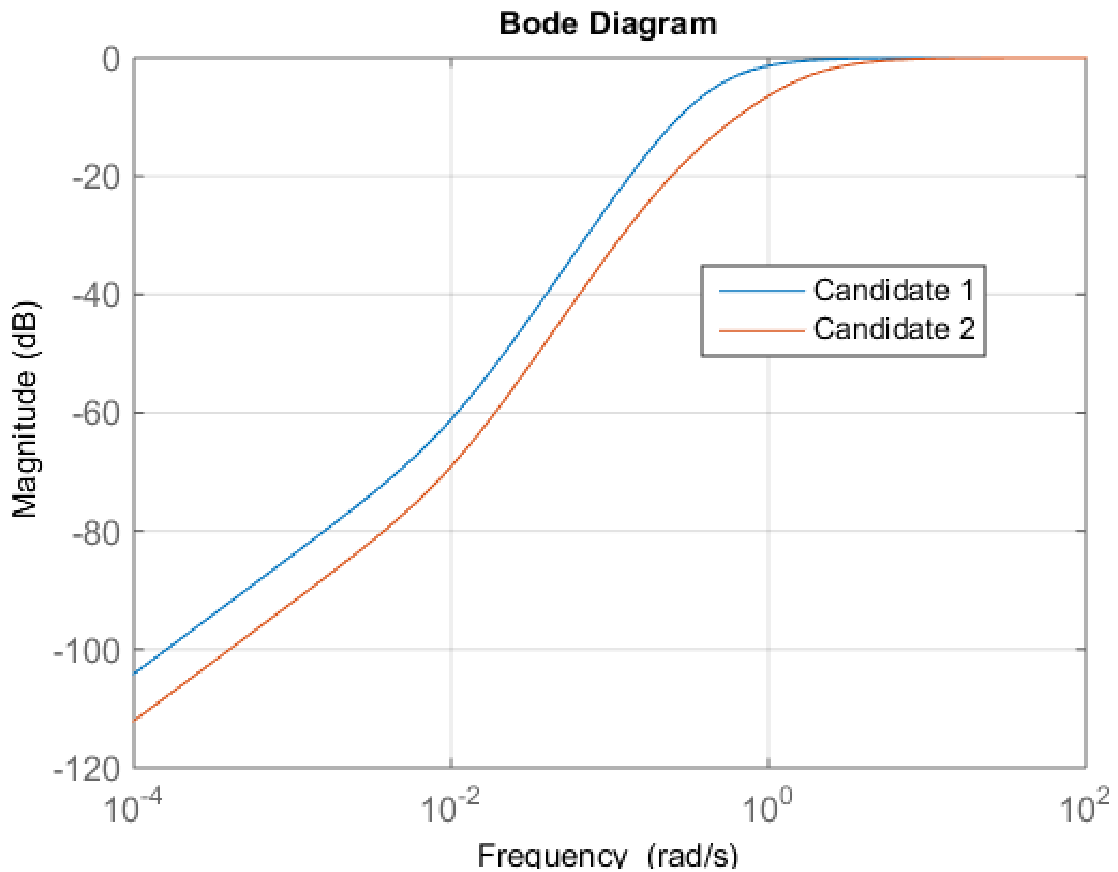
Bode plots of relevant transfer functions

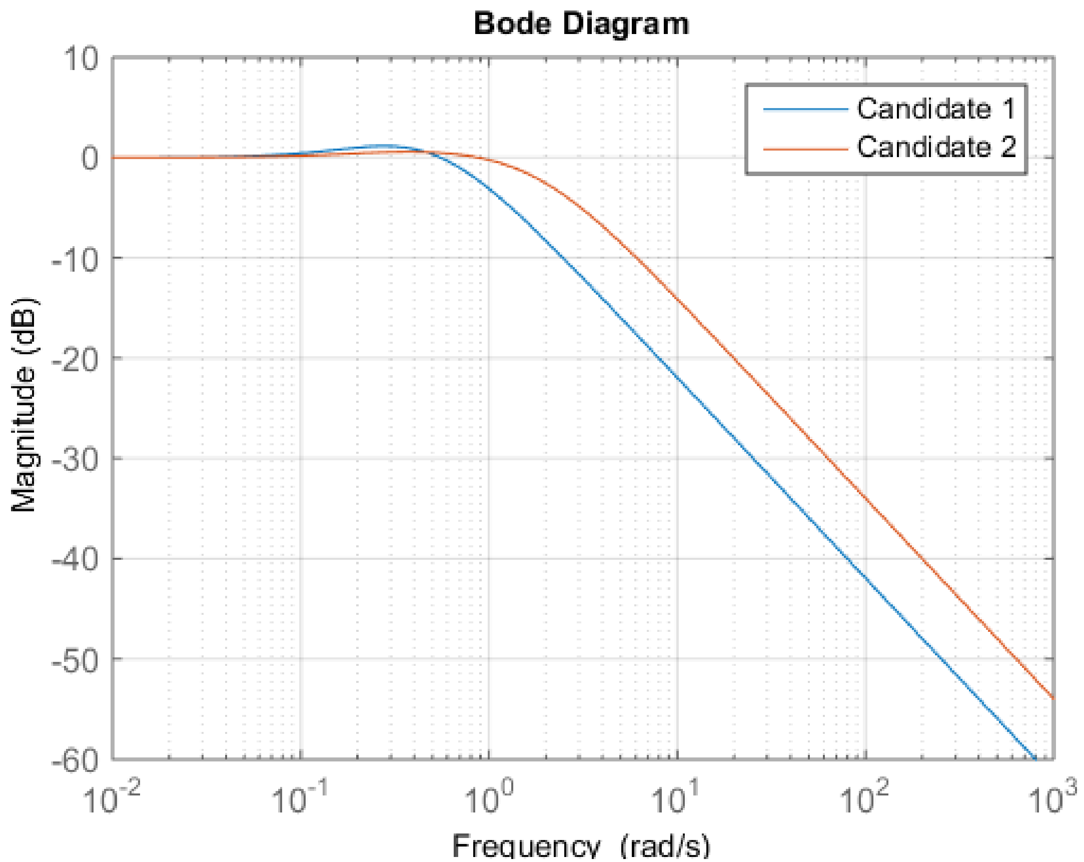




Designing a Cruise Control System to Achieve Frequency Domain Objectives – Assessment

Bode plots of relevant transfer functions





Conclusion: Both solutions achieve the required objectives…neither one is better than the other

Limitations of Classical Control

Classical control design techniques focus on choosing **a design** (out of **many possibilities**) that **achieves some set of objectives** – No mechanism exists for selecting the **best** design

Optimal control design focuses on choosing the **best controller** out of the set of available controllers that satisfy some constraints

General optimal control problem statement: *Select the control signal (or control trajectory) that* ***minimizes/maximizes an objective function subject to a system model and constraints***

- Key ingredients for optimal control: Objective function, dynamic model, and constraints

Optimal Control Example – Cruise Control

Formal control problem:

Minimize $$J(u(t); v(0)) = \int_0^{t_f} \left((v(t) - v_{des})^2 + K(u(t))^2 \right) dt$$

Objective function

Subject to:

$$\left.\begin{aligned} \dot{x} &= v \\ \dot{v} &= -0.01v + 10u \end{aligned}\right\} \text{Dynamic model}$$

Constraints

$$u_{min} \le u(t) \le u_{max}, 0 \le t \le t_f$$

Continuous control input constraint

$$v_{min} \le v(t) \le v_{max}, 0 \le t \le t_f$$

Continuous state constraint

$$\left.\begin{aligned} v(t_f) &= v_{des} \\ x(t_f) &= x_{target} \end{aligned}\right\} \text{Terminal state constraints}$$

Optimal Control Example – Cruise Control

1) Objective function = cost function:

$$J(u(t), v(0)) = \int_0^{t_f} \left[(v(t) - v_{des}(t))^2 + K(u(t))^2 \right] dt$$

decision variable

initial state.

Note: Continuous-time cost functions are commonly referred to as cost ***functionals***. Mathematically, functionals are functions of functions, and $u(t)$ is a function of time.

subject to:

2) Dynamic model: $\frac{v(s)}{u(s)} = \frac{10}{s + 0.01}$

$$\dot{v} + 0.01v = 10u$$

$$\Rightarrow \dot{v} = -0.01v + 10u$$

Optimal Control Example – Cruise Control

3) Constraints:

$$v_{min} \le v(t) \le v_{max}, \quad 0 \le t \le t_f$$

$$u_{min} \le u(t) \le u_{max}, \quad 0 \le t \le t_f$$

$$v(t_f) = v_f$$

$$x(t_f) = x_f$$

Common Ingredients in Optimal Control

Decision variable – The variable that is actually ***optimized*** (in the case of the cruise control example, this is the control input *trajectory*, $u(t)$)

Objective function – A function that is to be **minimized** or **maximized** over some **time window** (from 0 to t_f in the previous example)

- When the objective function is to be minimized, it is called a *cost function*; when it is to be maximized, it's called a *fitness function* or *reward function* (or just "objective function")

Initial condition – The value of the system states at the beginning of the time window over which the objective function is evaluated (in the cruise control example, this is $v(0)$)

Dynamic model – A state space model that describes how the system evolves

Constraints – Limits on the states and control signals over the prescribed time window (when constraints are only imposed at time t_f, they are referred to as ***terminal constraints***

Optimal Control Implementations

Offline optimization – At time $t = 0$, optimize the ***entire control trajectory***, $u(t)$**:**

$$u^*(t) = \arg\min_{u(t)} J(u(t), \mathbf{x}(0)) \quad \text{where} \quad J\big(u(t), \mathbf{x}(0)\big) = \int_0^{t_f} (g(\mathbf{x}(t), u(t))dt + h(\mathbf{x}\big(t_f\big))$$

subject to constraints

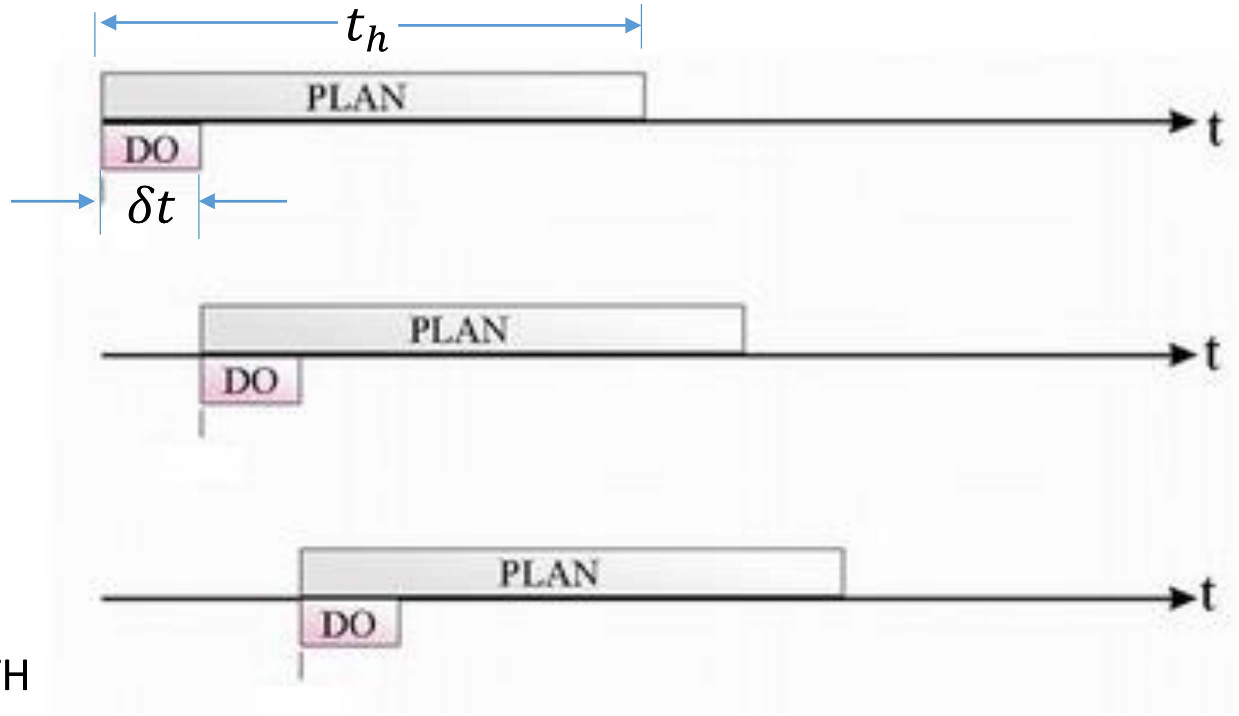
Online (receding horizon) optimal control – also known as model predictive control (MPC):

- At time 0, compute the control trajectory that minimizes $J\big(u(t), \mathbf{x}(0)\big) = \int_0^{t_h} (g(\mathbf{x}(t), u(t))dt + h(\mathbf{x}\big(t_f\big))$, subject to constraints
- A little bit later, at time δt, compute the control trajectory that minimizes $J\big(u(t), \mathbf{x}(\delta t)\big) = \int_{\delta t}^{\delta t + t_h} (g(\mathbf{x}(t), u(t))dt + h(\mathbf{x}\big(t_f\big))$, subject to constraints
- Repeat every δt time units

Receding Horizon Control - Interpretation

Online (receding horizon) optimal control – also known as model predictive control:

- At time 0, compute the control trajectory that minimizes $J\big(u(t),\mathbf{x}(0)\big) = \int_0^{t_h}(g(\mathbf{x}(t),u(t))dt + h(\mathbf{x}(t_f))$, subject to constraints
- A little bit later, at time δt, compute the control signal that minimizes $J\big(u(t),\mathbf{x}(\delta t)\big) = \int_{\delta t}^{\delta t+t_h}(g(\mathbf{x}(t),u(t))dt + h(\mathbf{x}(t_f))$, subject to constraints
- Repeat every δt time units



Source: ETH

Notes About Optimal Control Implementations

MPC (receding horizon control) is <u>much</u> more common in practice than offline optimization of $u(t)$

<u>BUT</u> The main mathematical operation for MPC is the same as in offline optimization of $u(t)$, namely minimization of a cost functional (subj. to a model & constraints)

…and we will focus in this course mostly on that main mathematical operation (but will have some lectures dedicated to specific considerations when you implement your optimization in a receding horizon manner.

Challenge with Optimal Control

UNC CHARLOTTE

Classical controllers are *closed form* – Given knowledge of the controller inputs, the output can be computed through a given equation, with no other information.

Example: $u(t) = k_p e(t) + \int_0^t k_i e(\bar{t}) d\bar{t}$ …knowing $e(t)$, $u(t)$ can be immediately computed

Optimal controllers are *generally not closed form* – The controller output cannot, in general, be computed immediately from the controller inputs

General form of the control signal ($u(t)$) under optimal control:

$$\left. \begin{aligned} u^*(t) &= \arg\min_{u(t)} J(u(t); \mathbf{x}(0)) \\ \text{subject to:} \quad & \begin{gathered} \dot{\mathbf{x}} = f(\mathbf{x}, u) \\ u(t) \in U, 0 \le t \le t_f \\ \mathbf{x}(t) \in X, 0 \le t \le t_f \\ \mathbf{x}(t_f) \in X_f \end{gathered} \end{aligned} \right\}$$

Key question: How can $u^*(t)$ be computed from the typically infinite number of candidate control trajectories ($u(t)$) that satisfy constraints?

General Tools for Optimal Control

Challenge: Computing $u^*(t)$ from an infinite set of possibilities is challenging and sometimes impossible

Techniques for arriving at approximations (and occasionally exact solutions) for u^* are the subject of this course…

Common approaches to optimal control design (sometimes used in combination):

- ***Discrete time approximation:*** Model the system in **discrete time** and partition the optimization time window (from 0 to t_f) into a ***finite number of time steps***…converts the problem into a finite-dimensional optimization
- ***Local optimization (using convex optimization tools):*** Make an initial guess at the optimal control trajectory, $u_0(t)$. Optimize perturbations/variations, denoted by $\delta u(t)$, around this initial guess, where $u(t) = u_0(t)+\delta u(t)$
- ***Pontryagin's minimum principle:*** Derive an optimal parametric form for the optimal control input trajectory, then optimize coefficients to the parametric form, e.g., $u(t) = A\sin(\omega t)$, where A and ω are parameters to be optimized

Approximate Semester Timeline

Remainder of August:
Mathematical preliminaries

September:
Local, convex optimization tools for discrete time systems

Early October:
Global discrete time optimization via dynamic programming

…

…

Late October:
Discrete time optimization tools for model predictive control

Early November:
Introduction to continuous time optimization tools

Remainder of Semester:
Use of Pontryagin's minimum principle for continuous time optimal control

Preview of next lecture (and beyond)

UNC CHARLOTTE

Topics for lecture 2:

- Setting up optimal control problems (with examples)
- Well-posed, ill-posed, and poorly defined optimal control problems

Beyond next lecture, we will start to look at finite-dimensional optimal control problems in discrete time:

- Converting continuous time systems to discrete time models
- Equivalence between finite-dimensional design optimization and discrete time optimal control problems
- Convexity