ME131 Vehicle Dynamics and Control

HW2: Systems / Introduction to ROS

Assigned: 2/6/2019 Due: 2/13/2019, 11:59pm (On bCourses)

Please submit your homework solutions on bCourses as a single PDF of your solutions. When videos are required, please only submit the link as part of the solution PDF document. Late homeworks will not be accepted.

Problem 1 PI Controller Design (30pt)

Consider the standard closed-loop system shown in Figure 1 below. The transfer functions of the plant P, and of the controller C, are

$$P(s) = \frac{1}{s-1}, \quad C(s) = \frac{K_P s + K_I}{s}$$

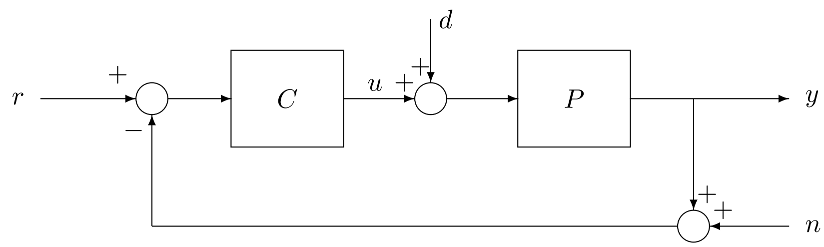

Figure 1: The standard closed-loop system.

1.1 (10pt) Let $K_P = 2$ and $K_I = 1$.
Fill in the table with the correct **steady-state** response of the closed-loop system output.
Write your steps in the blank pages next.

(Note, $\mu(t)$ denotes the unit step function, i.e., $\mu(t) = 0$ for $t < 0$ and $\mu(t) = 1$ for $t \geq 0$)

Input r(t)	Input d(t)	Input n(t)	Closed-loop response at steady state y_{ss}
$2\mu(t)$	$\mu(t)$	$3\mu(t)$	
$2\sin(5t)$	$\sin(t)$	$3\mu(t)$	

1.2 (10pt) Design the controller gains, K_P and K_I so that the unit-step forced response of $G_{r\to y}(s)$ satisfies the following performance specifications:

- Percent overshoot $PO \approx 10\%$.
- 5% settling time $t_s \leq 20$ sec.
- Rise time 5 sec $\leq t_r \leq 10$ sec.
(Use the second order approximation for t_r: $t_r \approx \frac{0.8+1.1\xi+1.4\xi^2}{\omega_n}$, where ξ is the damping ratio and ω_n is the natural frequency of the system.)

1.3 (10pt) Write a Matlab function implementing the PI controller in discrete time with the following format:

u = controller(e, Δt)

Submit your controller.m file.

Lab Deliverables

1. (10pt) Python function for Forward Euler discretized kinematic bicycle model from Sec.2.c. Submit the entire `system_model_simulation.py` file.
2. (10pt) Python function for the adaptive cruise controller from Sec.2.d. Submit your final `controller.py` file after tuning your gains in the ROS simulator.
3. (10pt) The simulated vehicle's speed plotted against the reference velocity, from Sec.2.j.
4. (10pt) Video from Sec.3.a showing the simulated BARC vehicle controlled from your keyboard.
5. (10pt) Rqt_graph during teleoperation, along with a short description of your system's components, from Sec.3.b.
6. (10pt) The teleoperation launch file from Sec.3.c.