# EN530.678 Nonlinear Control and Planning in Robotics
## Homework #6

March 14, 2018

Due: March 28, 2018 (before class)

Prof: Marin Kobilarov

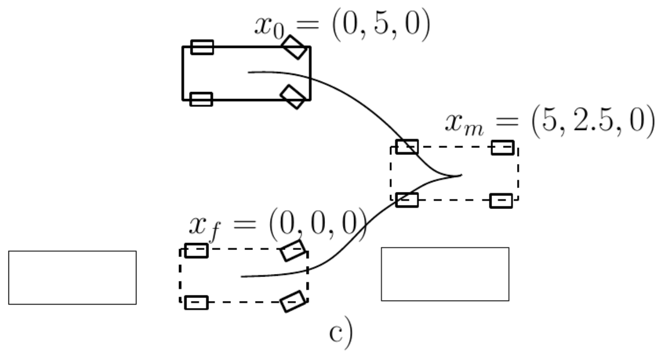


Figure 1: Parallel parking for the car.

1. Consider a simplified kinematic model of a car-like robot with configuration $x = (x_1, x_2, x_3)$, where $x_1, x_2$ denote the position and $x_3$ the orientation. The vehicle is controlled with forward velocity $u_1$ and steering angle $u_2$. The equations of motion $\dot{x} = f(x, u)$ are given by

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} \cos x_3 u_1 \\ \sin x_3 u_1 \\ u_1 \frac{\tan u_2}{\ell} \end{pmatrix}$$

where the constant $\ell > 0$ denotes the distance between the axles.

   (a) Assume that the car is required to track a desired feasible reference trajectory $x_d(t)$ with associated desired inputs $u_d(t)$ (e.g. such as those computed through differential flatness). Derive the linearized dynamics along the reference trajectory, i.e. in the form:

$$\dot{e}(t) = A(t)e(t) + B(t)(u(t) - u_d(t)),$$

   where the error $e(t)$ is defined as $e(t) \approx x(t) - x_d(t)$ and where $A(t) \triangleq \partial_x f(x_d(t), u_d(t))$, $B(t) \triangleq \partial_u f(x_d(t), u_d(t))$. Show that the error dynamics is controllable, along trajectories for which $u_1(t) \neq 0$. (see Appendix for the required definition of controllability).

(b) Implementation. Write a Matlab script `car_flat_care.m` which implements the steps given below. A file `uni_flat_care.m` (which demonstrates these functions for a related system) is provided as a reference which you can use for you own implementation if you choose to.

   i. Generate a parking maneuver (see Figure 1c) in flat output space $y = (x_1, x_2)$ and use the explicit expressions derived in the previous homework (problem 3b) to compute the state trajectory $x(t)$ and required inputs $u(t)$ from $y(t)$. As in the previous assignment, the car is required to perform a parallel parking maneuver starting at state $x_0 = (0, 5, 0)$, going forward to state $x_m = (5, 2.5, 0)$ and backing-up to state $x_f = (0, 0, 0)$. Assume that each segment takes time $T = 10$ s. When designing the paths, you can assume that the magnitude of the initial and final velocity of the car is $|u_1| = 1$ m/s.

   ii. Implement a linearization-based tracking controller using part 1a and follow the generated path. Start at a "perturbed" initial state $\widetilde{x}_0 = (0.25, 5.25, 0.1)$ and show that your controller stabilizes to the desired trajectory. Plot the flat output trajectory relative to the generated maneuver, and plot the control trajectory.

   iii. Inject Gaussian noise in the controls along the path and comment on the performance. Plot the flat output trajectory relative to the generated maneuver, and plot the control trajectory.

2. Recall the two-link manipulator with dynamics given in Lecture Notes #2. The file `arm_test.m` implements the ODE and simulates the computed torque law for this system. Extend this code as follows:

   (a) Compute a trajectory using the trajectory generation routine implemented in the previous homework (problem 4a). Denote the resulting trajectory by $q_d(t)$ and the associated feedforward control by $u_d(t)$.

   (b) Add a small disturbing external force to the dynamics which will result in deviation from the reference path (i.e. $u_d(t)$ alone cannot follow $q_d(t)$ exactly). Apply the computed torque law to employ feedback and track the trajectory $q_d(t)$. Plot $q(t)$ relative to $q_d(t)$, and plot $u(t)$ relative to $u_d(t)$.

Note: Upload your code and plots as a .zip file using `https://goo.gl/forms/1jxhjYUIlKqYdi7g1`; in addition attach a printout of the code and all plots to your homework solutions.

# Appendix

**Controllability of Time-varying systems.** A linear control system $\dot{x}(t) = A(t)x(t) + B(t)u(t)$ with $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ is controllable on $[t_0, t_f]$ if $A(t)$ and $B(t)$ are smooth and

$$\text{rank}[B_0(t)\ B_1(t)\ \cdots\ B_{n-1}(t)] = n, \text{ for all } t \in [t_0, t_f],$$

where the maps $B_i : [t_0, t_f] \to \mathbb{R}^{n\times m}$ are defined recursively according to

$$B_0(t) \triangleq B(t), \quad B_i(t) \triangleq \dot{B}_{i-1}(t) - A(t)B_{i-1}(t).$$