Chair of Cyber-Physical Systems in Production Engineering
Department of Mechanical Engineering
Technical University of Munich

PID controller - "Drive along the wall"

denis.hoornaert@tum.de

August 19, 2021

Acknowledgement

The present lab has been made possible by the teaching resources provided by UPENN.

Objective

After having implemented different controllers following either some random decisions or simple intuition, it is time to introduce a controller aiming at getting as close as possible to a given objective by mitigating the error. To this end, the present lab asks the students to implement a PID controller.

Reminder

The PID controller or (Proportionate, Derivative and Integral controller) is a well know and widely used type of controller. The objective of the PID controller is to take the appropriate decision in order to reduce as much as possible the error based on the error itself, the cumulation of the previous errors and the rate of change. This error is set with respect to a given objective.

In theory, the PID function is expressed as follows:

$$u(t) = K_p e(t) + K_i \int_0^t e(t')dt' + K_d \frac{d}{dt}(e(t))$$

Where:

- $u(t)$ is the action taken at instant t in order to reduce the error
- $e(t)$ is the error at instant t between the current situation and the objective
- K_p is a constant weighting the proportional part
- K_i is a constant weighting the integral part
- K_d is a constant weighting the the derivative part

In our case, we will define the objective as *a ideal distance seperating the car and the wall that we wish to maintain* (i.e. the distance $|AB|$ in Fig. 1b). In other words, we aim at keeping a constant and arbitrary distance between the car and the wall through a lap (e.g. the middle of the road). In Fig. 1b, the virtual line drawn by keeping this distance is represented by the dashed line O, the current error of the car being ε. In consequnce, the action $u(t)$ at instant t will dictate the steering angle to be used in order to correct the error $e(t)$.

Let us considerate the case illustared by Fig. 1b where the car, located at the point A is driving away from the wall on the right. In this case as in any other, finding the distance $|AB|$ is equivalent to looking on the right for the smallest measurement. Once the distance has been found, the error can simply be computed using the equation 1.

$$e(t) = |OB| - |AB| \tag{1}$$

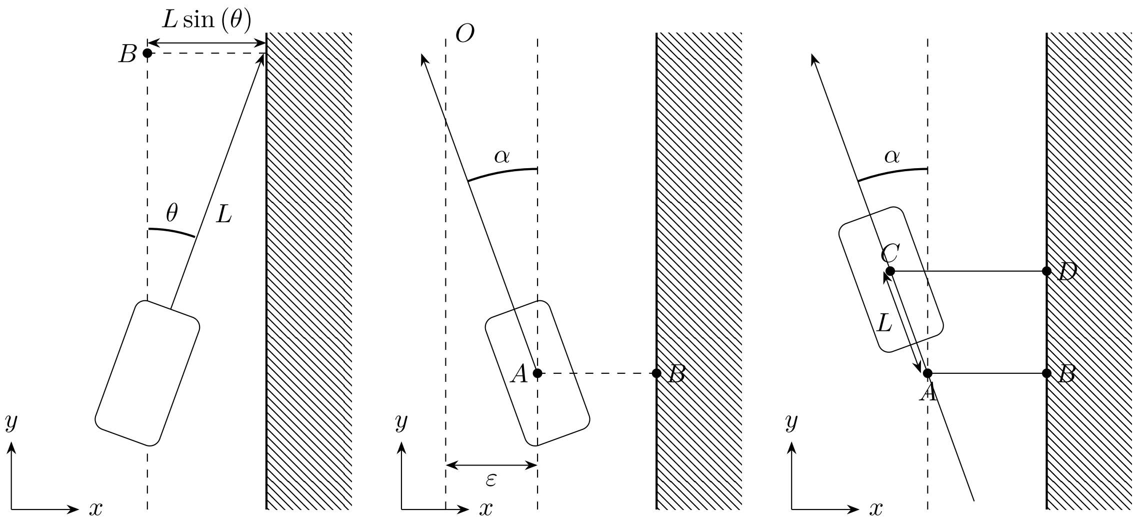
However, since the car is a (possibly) fast moving object, we are going to anticipate the next position of the car by considering its project as shown in Fig. 1c. In the latter, the projected position of the car is denoted by the point C, located at a distance $|CD|$ from the wall. The project can be done thanks to the angle *alpha*, the steering angle and the nominal speed of the car (noted L). The new distance $|CD|$ and the corresponding error can be computed using respectivelly equation 2 and 3.

$$|CD| = |AB| + L \times sin(\alpha) \tag{2}$$

$$e(t) = |OB| - |CD| \tag{3}$$

Finally, considering only the proportional and integral part, the PD controller is expressed as displayed in equation 4. Where $e(t)$ is the error at instant t and $e(t-1)$ is the error at the previous instant $t-1$.

$$u(t) = K_p e(t) + K_d(e(t-1) - e(t)) \tag{4}$$



(a) Simple project of the car (b) Geometric view of the car along a wall (c) Projected error of the car

Figure 1: Geometric view of the car along a wall

11.1

In the present lab, we ask you to implement the a PD controller, using the formula written above and to integrate it within the current project by connecting it to the `mux` node. As for the examples (Fig. 1a, 1b and 1c), we consider that the track's walls are flat.

Similarly to the *Follow the gap* controller, the range of results $u(t)$ of the PD controller must be mapped to the range allowed by the car's steering angle.

In order to ease the implementation, we suggest that the students consider a fixed speed, consider an objective distance of $1.5m$, implement only the proportional part of the controller and only thereafter, implement the derivative part.

For each of these implementations (i.e. proportional only and proportional plus derivative), study how the AEB system behaves. Does the latter interfere with the controllers? Discuss.

Obviously a fine tunning of the different parameters will have to be performed. More accurately, K_p and K_i will have to be determined.

11.2 Bonus

Integrate the *integral part of the PID* to what you already have and fine tune the K_i parameter. What is the outcome? Discuss.