Uppsala University
Dept. of Information Technology
Systems and Control

Introduction to Computer Control Systems

Computer exercise 1

Modeling, analysis, and PID control

Reading instructions: Glad-Ljung, Chapters 2, 3, 4, 6, and 11.

Name		Assistant's comments
Program	Year of reg.	
Date		
Passed prep. ex.	Sign	
Passed comp. ex.	Sign	

1 Introduction

This computer simulation exercise provides an introduction to the process lab, exercise 2. A nonlinear model of the robotic vehicle is considered and linearized at working points. Some properties of the linearized model are studied in comparison with the nonlinear one. Further, the linear model is sampled to enable digital implementation. Then properties of the model are revisited. Finally, the closed-loop system comprising the model of the plant and a PID controller is simulated for different sets of the controller parameters to illustrate expected performance of the control system.

To be well-prepared for this computer exercise, one should have read the chapters 2, 3, 4, 6, and 11 in the course book by Glad-Ljung thoroughly.

For the computer exercise, access to Matlab and Control System Toolbox is necessary. There is also a number of `.m`-files that are needed.

2 Modeling of the vehicle

The model of the system in this section is essentially the same as the one that will be implemented in the process labs. Figure 1 shows a sketch of the built vehicle. The vehicle could be separated into several basic parts, namely the brick, the motors, the wheels, and the sensor. Each of them has its own moment of inertia.

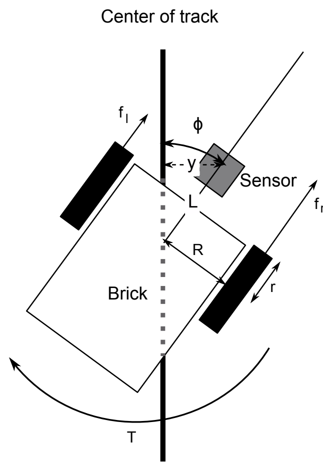

Figure 1: Robot viewed from above.

The vehicle will be given a constant forward speed v while the angle ϕ to the track axis is the controlled variable. To turn by a certain angle, the motors at the sides of the brick should be given an appropriate input signal Δu. For instance, to turn to the right side, the input voltage on the left motor u_l has to be higher than the input voltage on the right motor u_r. This relation could be

Variable	Description	Unit
T	Torque	Nm
R	Wheel axis raduis	m
ϕ	Angle to track	rad
$\tau_{r,l}$	Torque on left and right wheel	Nm
$\theta_{r,l}$	Rotation angle of left and right wheel	rad
I	Moment of inertia for robot around the center of rotation	

Table 1: List of variables

described as follows

$$\begin{aligned} u_l &= u_0 + \frac{\Delta u}{2} \\ u_r &= u_0 - \frac{\Delta u}{2} \end{aligned}$$

with u_0 representing the offset voltage corresponds to the constant velocity v. The distance between the center of the track and the center of vehicle's rotation is given by y. Thus, a model representing the relation between the input signal u and the output signal y is evaluated

$$\begin{aligned} u &= \frac{\Delta u}{u_{max}} 100 \\ &= K_\Delta \Delta u \end{aligned}$$

where the input signal u is the percentage of maximum voltage that can be supplied to the motors.

There is a number of possible models that can describe the behavior of the above described vehicle. One of the easier ways to perform modeling is by separating the translational and rotational movement.

Translational movement

The vehicle's translational movement is in the direction perpendicular to the track. This is given by two distances y_1 and y_2 as shown in Fig. 2. The distance y_1 will be the integrated velocity component perpendicular to the track while y_2 is the distance component perpendicular to the track due to the sensor rotation by the angle ϕ. It can be expressed by the following equations

$$\begin{aligned} y(t) &= y_1(t) + y_2(t) \qquad (1) \\ y_1(t) &= \int_0^T v \sin(\phi(t)) dt \\ y_2(t) &= L \sin(\phi(t)) \end{aligned}$$

The translational velocity of the vehicle is proportional to the offset voltage u_0

$$v = K_u u_0.$$

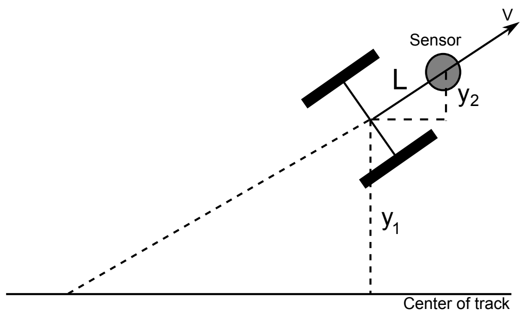

Figure 2: Translational part y_1 and rotational part y_2 of the total distance to the center of track y.

Rotational movement

From mechanics, the angle of the robot around the center of rotation could be defined as follows

$$\phi = \frac{r(\theta_l - \theta_r)}{2R}. \tag{2}$$

There are two ways to express the torque acting on the vehicle. The first way is by defining the difference between the torques on each wheel

$$T = \frac{R(\tau_l - \tau_r)}{r}$$

The other way is by defining moment of inertia of the system around a rotating body

$$T = I\ddot{\phi},$$

which would lead to the following relation between the torques on each wheel and the angle of the vehicle

$$\tau_l - \tau_r = \frac{rI}{R}\ddot{\phi} \tag{3}$$

The input signal given to the model is the difference between the voltage level of right and left motor. A model for a DC motor that describes the relation between the input voltage and the output torque is derived below. The total torque τ_{tot} that the motor will exert is proportional to the input voltage, *i.e.* $(\tau_{tot})_{r,l} = K_\tau u_{r,l}$. To produce an angular acceleration, the wheel requires a torque of $\tau_w = I_w\ddot{\theta}$. Furthermore, there is a torque required to account for the friction and back emf of the motor that is proportional to the angular velocity of the wheel $\tau_f = K_b\dot{\theta}$. The remainder of the torque τ will act on the load, *i.e.* move the vehicle. This gives the relation (dropping the indexes l and r for left and right motors).

$$\begin{aligned}
\tau_{tot} &= \tau_w + \tau_f + \tau \\
\tau &= \tau_{tot} - \tau_f - \tau_w \\
&= K_\tau u - I_w\ddot{\theta} - K_b\dot{\theta}
\end{aligned} \tag{4}$$

Putting together (2), (3) and (4) yields

$$\begin{aligned}\frac{rI}{R}\ddot{\phi} &= -I_w(\ddot{\theta}_l - \ddot{\theta}_r) - K_b(\dot{\theta}_l - \dot{\theta}_r) + K_\tau(u_l - u_r)\\ &= -\frac{2RI_w}{r}\ddot{\phi} - \frac{2RK_b}{r}\dot{\phi} + K_\tau \Delta u.\end{aligned}$$

Moving the terms related to ϕ to the left-hand side of equation results the differential equation

$$T\ddot{\phi} + \dot{\phi} = Ku \tag{5}$$

with

$$\begin{aligned}K &= \frac{K_\tau \tau}{2RK_bK_\Delta}\\ T &= \frac{1}{K_b}(\frac{r^2 I}{2R^2} + I_w).\end{aligned}$$

The output of the system is the distance from the center of the track to the sensor which according to (1) is given by

$$y(t) = L\sin(\phi(t)) + \int_0^t v\sin\phi(\tau)d\tau. \tag{6}$$

The nonlinear system could be represented in state space form as follows

$$\begin{aligned}x &= \begin{bmatrix}x_1 & x_2 & x_3\end{bmatrix}^T = \begin{bmatrix}\phi & \dot{\phi} & \int_0^t \sin\phi(\tau)d\tau\end{bmatrix}^T\\ \begin{bmatrix}\dot{x_1}\\ \dot{x_2}\\ \dot{x_3}\end{bmatrix} &= \begin{bmatrix}x_2\\ -\frac{1}{T}x_2\\ \sin(x_1)\end{bmatrix} + \begin{bmatrix}0\\ \frac{K}{T}u\\ 0\end{bmatrix}\\ y &= \begin{bmatrix}L\sin(x_1) + vx_3\end{bmatrix}\end{aligned} \tag{7}$$

Exercise 2.1: ***(Preparation exercise)*** *Derive the linearized model from given model (7) around the equilbrium point* $\phi = \phi_0$ *and* $u = 0$ *where* $\sin(\phi_0) = 0$. *(Hint: Glad-Ljung 11.4)*

Answer:

Create a continuous system **sys_c** in state space form using the command **ss()**. Run the command **step()** to see the step response of the system. Use the following values for the physical parameters, $T = 0.15$, $K = 0.0262$, $L = 0.075$, $v = 0.1$.

Exercise 2.2: *Choose different equilibria point for linearization, compare the step response of those systems and also with the nonlinear system using the command* `open_sim()`*. Explain the differences in these responses.*

Answer:

The next step would be to discretize the model in order to design the controller in discrete time. The sampling time would highly influence the response of the system.

Exercise 2.3: ***(Preparation exercise)*** *Derive the state space equation in discrete time for the system in exercise 2.1. (Hint: Glad-Ljung 4.2)*

Answer:

Convert the model to discrete-time **sys_d** using the command **c2d()**, try different sampling period, for instance 1s, 0.1s, and 0.03s.

3 System analysis

This section is about analyzing properties of the system, namely the stability. Stability of the system is defined by the location of its poles. In state space form, the poles appear as the eigenvalues of the state matrix. Use the command **pole()** or **eig()** to see the position of the poles in complex plane.

Exercise 3.1: *Where are the poles of the discretized systems located? How does it correlate to system stability? How does sampling period affect the position of the poles?*

Answer:

4 Closed loop systems

This section deals with the introduction of a controller into the system to make the closed loop system follow a certain reference value, which is often called trajectory tracking control. A commonly used PID controller is implemented here. It consists of three blocks connected in parallel, namely the proportional block, the integral part, and the derivative block. Each block has a parameter that can be tuned so that the control performance of the closed-loop system fulfills the specified criteria.

There are several ways to find the initial parameter values of PID, *e.g.* Ziegler-Nichols method, relay method, etc. After initial parameter values have been found, manual tuning has to be performed on those parameters to obtain the desired performance of the actual closed-loop system.

The general idea how to tune PID parameters is as follows. To obtain faster response of the closed-loop system, the proportional gain needs to be increased. The integral part is used to remove the steady state error. The derivative block is used to increase or decrease the damping factor, depending on the demanded response of the system.

Open the file `tracker_PID.m` to simulate the closed loop system with PID controller. Use the command `PID_reg(Kp,Ki,Kd,Ts)` to build the appropriate PID controller and use the command `closed_loop()` to evaluate the closed-loop system `cl_sys`. Detail information about this command is available at the appendix. Run the command `lin_close()` to run the closed-loop simulation of trajectory tracking control.

4.1 Proportional control (P-control)

Proportional control produces a control signal proportional to the output error

$$u(t) = K_p e(t)$$

where $e(t) = r(t) - y(t)$ and $r(t)$ is the reference signal.

Exercise 4.1: *Try to tune the proportional parameter K_p in the PID controller with $K_i = 0$ and $K_d = 0$ such that the closed loop system has good tracking performance in terms of speed of the convergence speed,* i.e. *the settling time is around 5 seconds. How would the value K_p influence response of the system? Is there any steady state error?*

Answer:

Exercise 4.2: *Now try the same proportional control on the disturbed system with a simulated disturbance acting on the vehicle's torque. Is there any steady state error?*

Answer:

4.2 Proportional control with integration (PI-control)

From the previous task, it is known that proportional control could not be able to remove the steady state error, thus integration is implemented at the controller to compensate the steady state error as follows

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau$$

Exercise 4.3: *By choosing K_p from previous exercise and $K_d = 0$, try to tune the the integral parameter K_i to eliminate the steady state error on the disturbed system. What is the influence of K_i in the response?*

Answer:

4.3 Proportional control with integral and derivative action (PID-control)

The implementation of integration would pretty much eliminate the steady state error in the closed loop system. However, this would also lead to an overshoot in the system response. Hence, to dampen this oscillation, the rate of change of the control signal could be reduced by letting it be dependent on the rate of change of the output error

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

Exercise 4.4: *Use the same value of K_p and K_i from previous exercise, try to tune the derivative parameter K_d of PID controller to reduce the overshoot at most 40% of the steady state value. How would K_d influence the closed loop response?*

Answer:

Exercise 4.5: *Implement the PID controller with the nonlinear system both disturbed and undisturbed. Are there any significant differences between the responses of linearized and nonlinear system?*

Answer:

Exercise 4.6: *Using the same value of K_p, K_i, and K_d, do the simulation of closed-loop system with different sampling period. Does it have any influence on the closed-loop response?*

Answer:

A List of matlab commands

`ss(A,B,C,D)` creates a continuous system in state space form.

`c2d(sys,Ts)` converts the continuous system `sys` into discrete time with sampling time `Ts`.

`step(sys)` produces the step response of `sys`.

`open_sim(sys,u,Tend)` compare step response of linearized system `sys` with the nonlinear system with step size `u` for simulation period `Tend`.

`eig(A)` calculates eigenvalues of `A`.

`pole(sys)` calculates the poles of system `sys`

`lsim(sys,u,t)` gives the response of `sys` to the input signal `u` within the time vector `t`.

`PID_reg(Kp,Ki,Kd,Ts)` evaluates the discrete-time PID controller.

`closed_loop(sys,reg)` evaluate the closed-loop system from open-loop system `sys` and regulator `reg`

`lin_close(sys_d,reg,Tend,Ts)` simulate a step input signal 10cm on linear system `sys_d` controlled by PID controller `reg` for simulation period `Tend` and sampling period `Ts`.

`lin_close_dist(sys_d,reg,Tend,Ts)` simulate a step input signal 10cm on linear system `sys_d` with simulated disturbance controlled by PID controller `reg` for simulation period `Tend` and sampling period `Ts`.

`nonlin_close(reg,Tend,Ts)` simulate a step input signal 10cm on nonlinear system controlled by PID controller `reg` for simulation period `Tend` and sampling period `Ts`.

`nonlin_close_dist(reg,Tend,Ts)` simulate a step input signal 10cm on nonlinear system with simulated disturbance controlled by PID controller `reg` for simulation period `Tend` and sampling period `Ts`.