Practical Exercise

Modeling and Control of Hybrid Systems (sc4160)

2014 – Version 3

Delft Center for Systems and Control, Delft University of Technology

---

General remarks

- This exercise consists of two parts. Each part has several steps that are outlined in a road map. You should follow this road map and present the results in a clear and concise report. In this report you should clearly explain and motivate all the choices you have made while solving the practical exercise.
  In your report you should also add one evaluation and conclusions section of max. 1.5 pages, in which you briefly outline the main insights you have obtained while making this assignment.
  You should also add the MATLAB files you have written in an appendix to the report.

- The deliverable of this assignment is a written report of max. 50 pages (including cover pages and appendices) about the assignment that also contains the *appendix* with your m files. Make sure to mention your group number and student number on the cover page. The report is to be emailed as a **pdf** file to
  `m.hajiahmadi@tudelft.nl`
  The *hard* deadline for emailing the report is Friday, June 20, 2014 at 17.00 p.m.

- You will be graded on the contents and the presentation of the report, on the originality[1] of your answers, and on the correctness, *efficiency*, and readability of the MATLAB files (i.e., do not forget to include explanatory comments in your MATLAB files).

- We recommend you to keep the computations symbolic or analytic as long as possible and not to hardcode any of the parameters in your MATLAB programs (instead, write one separate MATLAB function or script that defines the parameters) so that you can easily take other parameter values, longer control horizons, other reference signals, etc. into account.
  Furthermore, since each step of this assignment depends on the preceding ones, we recommend that after Steps 1.2, 1.5, 1.6, 2.2, and 2.3 of the road map, you check your intermediate results with the teaching assistant (during the office hours or via email).

[1] I.e., a correct answer that differs from the answers given by the other groups will be graded higher than a correct answer that is an almost literal copy of the answer of another group.

1 Part 1: Adaptive cruise control

1.1 Set-up

We consider an adaptive cruise control (ACC) application in which 2 cars are driving one after the other (see Figure 1).

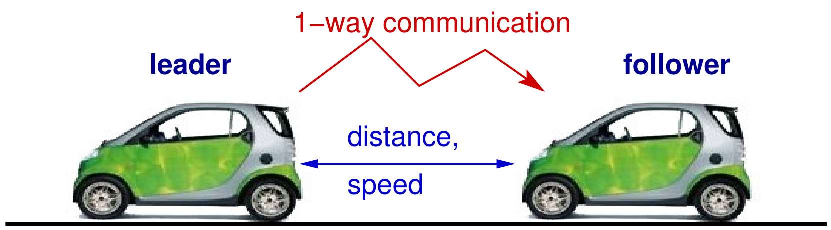

Figure 1: ACC set-up considered in the practical assignment.

In general, the aim of ACC is to ensure a minimal separation between the vehicles (i.e., distance keeping) and a speed adaptation (i.e., the speed differences between the vehicles should be kept as small as possible). In this exercise we will — for the sake of simplicity — only consider the speed adaptation control and we assume that the leading vehicle communicates its speed to the following vehicle, which then has to track this speed as well as possible.

For the vehicle dynamics we consider a simplified model in which at time t the following forces act on the vehicle (which has mass m):

- the "driving" force $F_{\text{drive}}(t)$, which is proportional to the throttle input $u(t)$: $F_{\text{drive}}(t) = bu(t)$,
- a dynamic friction force $F_{\text{friction}}(t)$, which is proportional to the square of the speed $v(t)$ of the vehicle: $F_{\text{friction}}(t) = cv^2(t)$.

Braking will be simulated by applying a negative throttle. We will assume that the vehicles drive in the forward direction, so the speed will always be nonnegative. For passenger comfort we also include a maximal acceleration/deceleration: $|a(t)| \leqslant a_{\text{comf,max}}$. The parameters of the vehicle are given in Table 1.

Parameter	Value	Units
m	850	kg
c	0.4	kg/m
b	3700	N
$u_{\max}$	0.9	—
$u_{\min}$	-1	—
$a_{\text{comf,max}}$	2.5	m/s^2

Table 1: Parameters of the vehicle.

1.2 Tasks & Road map

Step 1.1: Note that as we are only considering the speed adaptation and as the leading vehicle communicates its speed to the follower, we only have to consider the following vehicle.
The continuous-time model for the position $x(t)$ and speed $v(t)$ of the following vehicle in the state-space form is formulated as:

$$\frac{d}{dt}\begin{bmatrix} x(t) \\ v(t) \end{bmatrix} = \begin{bmatrix} v(t) \\ \frac{1}{m}\underbrace{bu(t)}_{F_{\text{drive}}} - \frac{1}{m}\underbrace{cv^2(t)}_{F_{\text{friction}}} \end{bmatrix}$$

Given the maximal throttle input $u_{\max}$ and the maximal braking input $u_{\min}$, determine the maximal speed $v_{\max}$ and the maximal acceleration $a_{\text{acc,max}}$ and deceleration $a_{\text{dec,max}}$ of the vehicle.

Step 1.2: Construct a piecewise affine (PWA) approximation P with 2 regions of the friction force curve $V : [0, v_{\max}] \to \mathbb{R} : v \mapsto v^2$ as follows. We want a perfect match for $v = 0$ and $v = v_{\max}$. This implies that we still have two degrees of freedom, i.e., the coordinates (α, β) of the middle edge point of the PWA curve (see Figure 2). Now determine α and β such that the *squared* area between V and P (i.e., the squared area corresponding the hashed region in Figure 2) is minimized, or equivalently, such that

$$\int_0^{v_{\max}} \big(V(v) - P(v)\big)^2 \, dv$$

is minimized.
Note: Use analytic computations (by hand or by using Mathematica or Maple, etc.) to determine the optimal α and β (and *not* numerical computations/optimization).

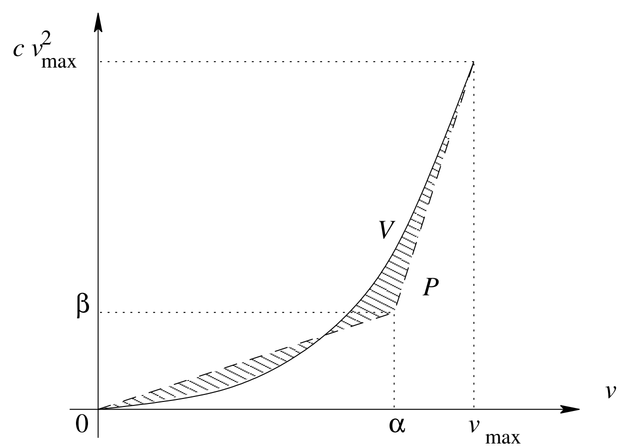

Figure 2: The quadratic function V and its PWA approximation P.

Step 1.3: Now approximate the friction force using the PWA function P instead of the quadratic function V. Compare the output of the resulting continuous-time PWA model with that of the original model for a sinusoidal throttle input. Can you explain where the differences — if any — come from?
Note: By properly selecting the initial speed you should be able to guarantee that the speed never becomes negative and that both regions of the PWA model are visited.

Step 1.4: Now discretize the PWA model of the vehicle using a sample step T with $T = 0.125$ s and a forward Euler rule for the discretization.

Step 1.5: Transform the discrete-time PWA model of Step 1.4 into an MLD model.
As we are only considering speed tracking in this assignment and not distance keeping, the position of the car will not influence the performance (i.e., the control objective) in any way. Hence, for the transformation of the discrete-time PWA model into an MLD model, the only state variable that should be considered is the speed.

Step 1.6: Now we design an MPC controller for the MLD model using the implicit MPC approach. The performance J should be a trade-off between the tracking J_{track} (i.e., the difference between the speed and the reference speed communicated by the leading vehicle) and the input energy J_{input} (for groups 1, 2, 7, and 8), the smoothness of the throttle signal (for groups 3, 4, 9, and 10), and the smoothness of the derivative of the throttle signal (for groups 5, 6, 11, and 12). More specifically, we have

$$J(k) = J_{\text{track}}(k) + \lambda J_{\text{input}}(k)$$

with $J_{\text{track}}(k)$and $J_{\text{input}}(k)$ as given in the following table:

Group number (modulo 12)	$J_{\text{track}}(k) = \|\tilde{v}(k) - \tilde{v}_{\text{ref}}(k)\|_1$ and $J_{\text{input}}(k) =$	Group number (modulo 12)	$J_{\text{track}}(k) = \|\tilde{v}(k) - \tilde{v}_{\text{ref}}(k)\|_\infty$ and $J_{\text{input}}(k) =$
1	$\|\tilde{u}(k)\|_1$	7	$\|\tilde{u}(k)\|_1$
2	$\|\tilde{u}(k)\|_\infty$	8	$\|\tilde{u}(k)\|_\infty$
3	$\|\Delta\tilde{u}(k)\|_1$	9	$\|\Delta\tilde{u}(k)\|_1$
4	$\|\Delta\tilde{u}(k)\|_\infty$	10	$\|\Delta\tilde{u}(k)\|_\infty$
5	$\|\Delta^2\tilde{u}(k)\|_1$	11	$\|\Delta^2\tilde{u}(k)\|_1$
6	$\|\Delta^2\tilde{u}(k)\|_\infty$	12	$\|\Delta^2\tilde{u}(k)\|_\infty$

where $\tilde{v}(k) = [v(k+1) \ldots v(k+N_{\text{p}})]^{\text{T}}$, $\tilde{v}_{\text{ref}}(k) = [v_{\text{ref}}(k+1) \ldots v_{\text{ref}}(k+N_{\text{p}})]^{\text{T}}$, and $\tilde{u}(k) = [u(k) \ldots u(k+N_{\text{p}}-1)]^{\text{T}}$ with N_{p} the prediction horizon. Note that for a signal $s(\cdot)$ we have $\Delta s(k) = s(k) - s(k-1)$ and $\Delta^2 s(k) = \Delta s(k) - \Delta s(k-1) = s(k) - 2s(k-1) + s(k-2)$.
In order to get a well-defined objective function groups 3, 4, 5, 6, 9, 10, 11, and 12 may assume that $u(k_0-2) = u(k_0-1) = 0$ where k_0 corresponds to the first sample step of the total simulation period $[0, T_{\text{end}}]$ (cf. Step 1.8).
Write a MATLAB file that computes the optimal MPC input sequence for a given sample step k for values of N_{p} and N_{c} (i.e., the control horizon) up to 6, and for arbitrary values of λ. Note that a discretized version of the comfort constraint $-a_{\text{comf,max}} \leqslant a(t) \leqslant a_{\text{comf,max}}$ should also be taken into account!
Also note that due to the approximation made in Step 1.2, which is only valid for nonnegative speeds, we should also *explicitly* add the constraint $v(t) \geqslant 0$.

Hints & notes:

- Note that by introducing one or more dummy variables optimization problems of the form $\min_{\theta\in\mathbb{R}^n} \|\theta\|_1$ subject to $A\theta \leqslant b$ or $\min_{\theta\in\mathbb{R}^n} \|\theta\|_\infty$ subject to $A\theta \leqslant b$ can be transformed into a linear programming (LP) problem:

  - Recall that for $\theta \in \mathbb{R}^n$, we have $\|\theta\|_1 = \sum_{i=1}^n |\theta_i|$. Now it is easy to verify that any optimal solution (ρ^*, θ^*) of the problem

$$\min_{\rho,\theta \in \mathbb{R}^n} \rho_1 + \cdots + \rho_n \quad \text{subject to } -\rho \leqslant \theta \leqslant \rho \text{ and } A\theta \leqslant b$$

    is also an optimal solution of $\min_{\theta \in \mathbb{R}^n} \|\theta\|_1$ subject to $A\theta \leqslant b$ (and vice versa if we set $\rho^* = \|\theta^*\|_1$).
  - Similarly, recall that for $\theta \in \mathbb{R}^n$, we have $\|\theta\|_\infty = \max_{i=1,\ldots,n} |\theta_i|$. Now it is easy to verify that any optimal solution (τ^*, θ^*) of the problem

$$\min_{\tau \in \mathbb{R},\theta \in \mathbb{R}^n} \tau \quad \text{subject to } -\tau \leqslant \theta_i \leqslant \tau \text{ for } i = 1,\ldots,n \text{ and } A\theta \leqslant b$$

    is also an optimal solution of $\min_{\theta \in \mathbb{R}^n} \|\theta\|_\infty$ subject to $A\theta \leqslant b$ (and vice versa if we set $\tau^* = \|\theta^*\|_\infty$).
- Using the hint above the MPC optimization problem at step k can be transformed into a mixed-integer linear programming problem (MILP). In order to solve this problem you will need an MILP solver, for which you could use one of the following options:
  - use the `mpt_solveMILP` command of the MPT toolbox ver. 2.6.3 (see `http://people.ee.ethz.ch/~mpt/2/downloads/`).
    To install and activate this toolbox, see the instructions at
    `http://control.ee.ethz.ch/~mpt/docs/install.php`
    For step 2 of the installation procedure we recommend you to use the
    `addpath(genpath(...))`
    approach. Note that this command should be typed every time you (re)start MATLAB and want to use the Multi-Parametric Toolbox. The command to solve MILP problems is `mpt_solveMILP`
    Note that there is an on-line reference guide for the Multi-Parametric Toolbox at
    `http://control.ee.ethz.ch/~mpt/docs/`
    The main page for `mpt_solveMILP` is
`http://people.ee.ethz.ch/~mpt/2/docs/refguide/mpt/solvers/mpt_solveMILP.html`
  - use the `glpk` function of the MPT toolbox ver. 3 (for MATLAB R2011a and newer) (see
    `http://people.ee.ethz.ch/~mpt/3/`) To install and activate this toolbox, see the instructions at
    `http://control.ee.ethz.ch/~mpt/3/Main/Installation`
  - If you are using Matlab R2014a, you can use the function `intlinprog`
  - use the `cplex` command of the TOMLAB cplex toolbox (note that you need a license, see `http://tomopt.com/tomlab/`). With this command you can solve MILP problems (see also the `milp_solve_tomlab_cplex` script on the course website).
- As we want you to get some insight into the hybrid MPC method and the relation with integer programming, you are *not* allowed to solve the entire exercise using the `mpt_control` or `mpt_optControl` commands (or related commands) of the MPT toolbox or the Hybrid Toolbox (see `http://cse.lab.imtlucca.it/~bemporad/hybrid/toolbox/`). However, feel free to compare the results obtained with your own programs to those obtained with the MPT toolbox or the Hybrid Toolbox, and to discuss the differences, if any.

Step 1.7: Write a MATLAB file to simulate the closed-loop behavior of the system (i.e., apply the receding horizon approach in which at each step the optimal MPC control input is recomputed and applied to the system) using the original continuous-time model as simulation model.

Step 1.8: Assume $\lambda = 0.1$ (in general, an appropriate value of λ is determined by taking into account the nominal values of J_{track} and J_{input} and by some tuning and iteration). Consider two combinations of N_{p} and N_{c}: first the combination $(N_{\text{p},1}, N_{\text{c},1}) = (5,4)$ which is the same for all groups, and another combination $(N_{\text{p},2}, N_{\text{c},2})$ that you may select yourself with $N_{\text{p},i} \in \{7,8,9,10\}$ and $1 < N_{\text{c},i} < N_{\text{p},i}$. For each combination run your program for the time interval $[0, T_{\text{end}}]$ with $T_{\text{end}} = 25$, for $v(0) = 0.9\alpha$ where α is the value found in Step 1.2, and for the speed reference signal v_{ref} which defined as follows (see also Figure 3):

$$v_{\text{ref}}(t) = \begin{cases} 0.85\alpha & \text{for } 0 \leqslant t \leqslant 3 \\ 1.2\alpha & \text{for } 3 < t \leqslant 9 \\ 1.2\alpha - \dfrac{1}{12}\alpha(t-9) & \text{for } 9 < t \leqslant 15 \\ 0.7\alpha & \text{for } 15 < t \leqslant 18 \\ 0.7\alpha + \dfrac{4}{15}\alpha(t-18) & \text{for } 18 < t \leqslant 21 \\ 0.9\alpha & \text{for } 21 < t \leqslant 30 \end{cases}.$$

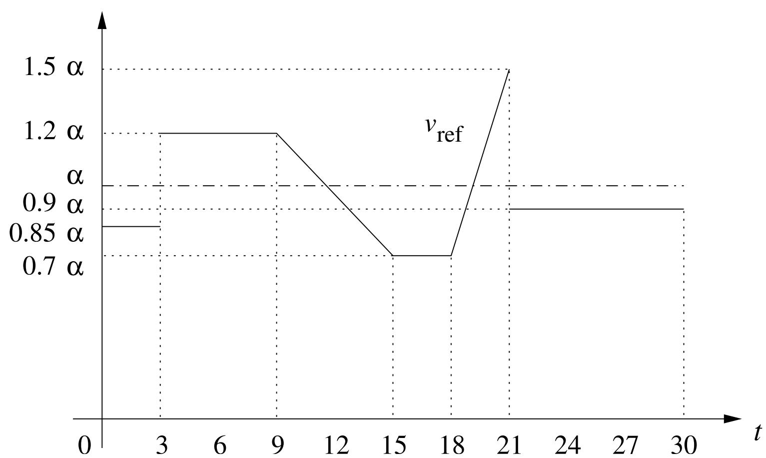

Figure 3: The reference speed signal to be used.

Make a plot of the evolution of the controlled closed-loop system in the (x,v) phase plane and of the evolution of x, v, the acceleration a, v_{ref}, $v - v_{\text{ref}}$, u and Δu over time. Compare the obtained trajectories and discuss the differences, if any.

Depending on your progress, the following step could also be performed (if you complete the bonus step, your report is allowed to count up to 55 pages):

Bonus step (if completed successfully, you will get up to 1 point extra): You will have noticed that computing the optimal MPC input using the `mpt_solveMILP` function requires quite some computation time, especially for large N_{p} and N_{c}. If this time is larger than the sampling time of the controller (in our case T =0.125 s) then the (basic) on-line MPC optimization approach is not feasible. One of the possible solutions is then to use the explicit MPC approach in which for each possible current state $v(k)$ and future reference vector $\tilde{v}_{\text{ref}}(k)$ the optimal MPC input $u^*(k)$ is computed off-line using multi-parametric mixed-integer linear programming, and stored in a look-up table (cf. page 112 of the lecture notes and the references [23, 24, 28, 29, 36] of the lecture notes).

Now you should apply the explicit MPC approach to the ACC example and repeat Steps 1.6–1.8 but now with explicit MPC instead of implicit MPC. In order to compute the explicit MPC solution you can use the `mpt_mpmilp` function of the Multi-Parametric Toolbox.

Compare the off-line and on-line computation times required for each approach with $N_{\text{p}} = 4$ and $N_{\text{c}} = 4$, and explain the results.

2 Part 2: Roll-angle control

2.1 Set-up

We consider the problem of controlling the roll angle of an aircraft.

The system has two distinct dynamics and the state space model of the plane is defined by:

$$\dot{x}(t) = A_{\sigma(t)}x(t) + B_{\sigma(t)}u(t) + H_{\sigma(t)}w(t) \quad (1)$$

$$y(t) = E_{\sigma(t)}x(t) + F_{\sigma(t)}u(t) + G_{\sigma(t)}w(t) \quad (2)$$

where $x \in \mathbb{R}^{n_x}$ is the state vector, $y \in \mathbb{R}^{n_y}$ is the output, $u \in \mathbb{R}^{n_u}$ is the controlled input, and $w \in \mathbb{R}^{n_w}$ is the disturbance vector (more information about the dynamics can be found in J.V. Vegte, *Feedback control systems*, page 381). The switching signal σ determines which subsystem governs the dynamics over time. Moreover, the system matrices are defined as follows:

$$A_1 = \begin{bmatrix} -5 & 0 \\ 0 & 8 \end{bmatrix},\ A_2 = \begin{bmatrix} 3 & 0 \\ 0 & -2 \end{bmatrix},\ B_1 = \begin{bmatrix} 4 \\ 0 \end{bmatrix},\ B_2 = \begin{bmatrix} 0 \\ 2 \end{bmatrix},\ H_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix},\ H_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

$$E_1 = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix},\ E_2 = \begin{bmatrix} 0 & 4 \\ 0 & 0 \end{bmatrix},\ F_1 = \begin{bmatrix} 0 \\ 2 \end{bmatrix},\ F_2 = \begin{bmatrix} 2 \\ 0 \end{bmatrix},\ G_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ G_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The aim is to design stabilizing controllers for the roll angle of the plane, based on linear matrix inequalities (LMIs). The designed controllers must be able to reduce effects of the disturbance w on the output y.

2.2 Tasks & Road map

Step 2.1: Motivate why a single state feedback controller would not realize our aims.
Hint: So motivate whether or not a linear state feedback controller with a constant gain can stabilize the system.

Step 2.2: Assume that $u(t), w(t) = 0, \forall t$. Design a switching law that stabilizes the switched system (using the multiple Lyapunov functions approach from the lecture notes, Section 5.2.2). Verify (using simulation for a few random initial points) that the designed switching law indeed asymptotically stabilizes the switched system.

Step 2.3: Now suppose that the system is affected by a disturbance signal with the following property:

$$w \in \mathcal{L}_2 \Longleftrightarrow \|w\|_2^2 = \int_0^{+\infty} w^{\mathrm{T}}(\tau)w(\tau)\mathrm{d}\tau < \infty. \tag{3}$$

Assume that the control input $u = 0, \forall t$. The goal is to design a stabilizing switching law that not only stabilizes the system, but also assures the inequality:

$$\sup_{w \in \mathcal{L}_2} \|y\|_2^2 - \rho\|w\|_2^2 < 0, \tag{4}$$

with $\rho > 0$ a scalar. The control design procedure starts with assuming that there exists a Lyapunov function V (with $V(x = 0) = 0$) with the derivative bounded by the following inequality:

$$\dot{V}\big(x(t)\big) < -y^{\mathrm{T}}(t)y(t) + \rho w^{\mathrm{T}}(t)w(t). \tag{5}$$

Integrating both sides from 0 to $+\infty$ gives:

$$V\big(x(+\infty)\big) - V\big(x(0)\big) < -\underbrace{\int_0^{+\infty} y^{\mathrm{T}}(\tau)y(\tau)\mathrm{d}\tau}_{\|y\|_2^2} + \rho \underbrace{\int_0^{+\infty} w^{\mathrm{T}}(\tau)w(\tau)\mathrm{d}\tau}_{\|w\|_2^2}$$

Now if the system is asymptotically stable when $w \equiv 0$, then $x(+\infty) = 0$ and $V(x(+\infty)) = 0$. Therefore, the left-hand side of the last inequality is zero if the initial state of the system $x(0) = 0$ and hence, the system has the property (4). Now defining the Lyapunov function as:

$$V = \min_{i \in \{1,2\}} x^{\mathrm{T}} P_i x, \tag{6}$$

you have to design a stabilizing switching law that also guarantees (4). (Hint: To this aim, you can follow the same idea as in Section 5.2.2 in the lecture notes, for the multiple Lyapunov functions approach. The difference is that you need to integrate (5) too). You should formulate the design problem as a linear matrix inequality (LMI) problem (assuming that the scalar variables $\beta_1 = \beta_2$ (see Section 5.2.2, Lecture notes) are fixed).

Step 2.4: Using the LMI obtained in the previous step, formulate an optimization problem in order to obtain the minimum possible ρ. For $\beta_1 = \beta_2$ ranging from 0 to 200, with steps of 5, solve the optimization problem. Report the minimum value obtained for ρ and also the corresponding matrices P_1 and P_2. Now implement the switching law using the matrices obtained. Assume $w(t) = t\exp(-0.5t)$, $\forall t \geq 0$, and $x(0) = 0$. Simulate the closed-loop controlled system (with $u \equiv 0$) for 20 seconds. Plot the evolution of the states x, the output y, and the switching signal σ over time. Using the trajectories of w and y, find the value of $\|y\|_2^2/\|w\|_2^2$. Compare it with the minimum ρ^* determined by solving the optimization problem.

Step 2.5: Now in order to further improve the performance, we introduce a state feedback controller $u = K_i x, i = 1, 2$. The gains K_i along with the switching input σ should be properly designed in order to stabilize the switched system and further, to assure (4). The task is to obtain a convex optimization problem subject to an LMI (with fixed β), in order to find the optimal gain matrices K_i, the matrices P_i and the minimum ρ. Hint: you can use the technique mentioned in Section 5.3.1 of the Lecture notes.

Step 2.6: Solve the convex optimization problem formulated in the previous step for fixed $\beta_1 = \beta_2$ ranging from 0 to 200, with steps of 5. Report the minimum ρ obtained and also, the corresponding matrices P_1, P_2, K_1, K_2. Furthermore, implement the state feedback controller $u = K_i x$ together with the switching law $\sigma(t) = \arg\min x^{\mathrm{T}}(t) P_i x(t)$ using the matrices obtained in the previous step. Assume $w(t) = t\exp(-0.5t)$, $\forall t \geq 0$, and $x(0) = 0$. Simulate the closed-loop controlled system for 20 seconds. Plot the evolution of the states x, the output y, the state feedback controller u, and the switching signal σ over time. Using the trajectories of w, y, find the value of $\|y\|_2^2/\|w\|_2^2$. Compare it with the optimal ρ^* determined by solving the optimization problem defined in the previous step.

Note: Do not forget to write a section about discussion on the obtained results and conclusions.