TSFS02 — Vehicle Dynamics and Control

Computer Exercise 3: Electronic Stability Control (ESP)

Division of Vehicular Systems
Department of Electrical Engineering
Linköping University
SE-581 33 Linköping, Sweden

Contents

1 Introduction

The main purpose of a stability control system is to aid the driver in critical situations, such as, quick avoidance maneuvers. These systems are often denoted differently for different manufacturers, however, some of the more common abbreviations are ESP (Electronic Stability Program), ESC (Electronic Stability Control), DSC (Dynamic Stability Control), AYC (Active Yaw Control), and VDC (Vehicle Dynamics Control). Throughout this document the abbreviation ESP is used.

The stability control system should compensate under- and oversteer by applying a yaw moment, i.e., a turning moment about the vertical axis. This moment is usually created by braking the individual wheels, but could possibly also be achieved through active steering or applying a driving moment. In the vehicle, these systems are integrated in a control hierarchy together with the ABS, traction control, and other driver aid and safety systems. However, in this exercise you should consider only the ESP.

The main objective in this exercise is to develop and implement your own stability control system, and verify its functionality, using the simulation and modeling tool MATLAB .

1.1 Examination

To pass this exercise you should have fulfilled the following:

- Solved the preparation tasks.
- Solved all the computer exercise tasks.
- Answered all questions, with motivated and thoughtful answers.

The examination is done by presenting your results and answers to a course assistant at the schedule exercise session. To speed up the examination process, it might be a good idea to present the tasks as you complete them, instead of saving them all to the end.

2 Preparation Tasks

The tasks in this section, i.e., Task 1–6, are preparation tasks that should have been solved before starting with the computer exercise tasks in Section 3. Please verify with an assistant that you have solved these tasks correctly before starting with Task 7–8 (since your code implementations will be based on the equations you derive in the preparation tasks).

2.1 Controller Design

The goal for the ESP is to support the driver in steering the vehicle such that it behaves as the driver expects. This behavior can be formulated in terms of yaw rate Ω_z and body slip β. If the desired and the actual behavior differs too much, the controller intervenes by applying a yaw moment to help the vehicle turn, or straighten up. This yaw moment ΔM can thus be formulated as a function of Ω_z and β, written as

$$\Delta M = \Delta M(\beta_{nom} - \beta, \Omega_{z,nom} - \Omega_z) \tag{1}$$

where $\Omega_{z,nom}$ and β_{nom} are the nominal values, representing the desired or expected behavior of the vehicle. A simple proportional control can be formulated as

$$\Delta M = k_1(\beta_{nom} - \beta) + k_2(\Omega_{z,nom} - \Omega_z) \tag{2}$$

where k_1 and k_2 are tuning parameters.

Task 1

Determine appropriate expressions for $\Omega_{z,nom}$ and β_{nom}. All included parameters and variables should be explained (for example with figures). Motivate the assumptions you have made and why your expressions are appropriate for describing the desired vehicle behavior.

Task 2

Design a yaw-moment controller, creating a ΔM that is supposed to affect the vehicle. A suggested layout is given by (2). The available sensor signals and known parameters are listed in Appendix A.

Task 3

Given your controller design, what signs (positive or negative) should your design parameters have? If you have chosen the design in (2), it is the sign of k_1 and k_2 that are of interest. (A first guess of the absolute values of k_1 and k_2 is 5000.)

2.2 Controller Actuation

The calculated yaw moment ΔM can be actuated in different ways. Here we will be using a common method, where the wheels are individually braked to create the desired yaw moment.

Task 4

With the yaw moment ΔM known (from Task 1–3), describe and motivate how the controller decides which wheel to brake in different driving situations. How is the decision affected by the current under-/oversteering state of the vehicle, or which direction the vehicle is turning?

Hint: When applying a braking force (a longitudinal force) at the wheel, the lateral forces will also be affected (think about the friction ellipse). Use this correlation when deciding which wheel to brake.

Task 5

The yaw moment ΔM is now known and you have chosen which wheel to brake (in Task 4). Then the controller needs to know the amplitude of the braking force F_b, that is necessary to generate the yaw moment ΔM. Derive expressions for the brake force F_b as functions of the yaw moment ΔM.

Hint: You may have to use different expressions depending on which wheel you brake.

Task 6

In Task 5, you have computed the desired braking force F_b to generate the expected yaw moment. Now you need to transfer the braking force as the wheel torque, since the vehicle model takes the wheel torque as the direct control input.

Hint: The wheel torque generated based on the braking force is always negative. In addition, you also need to consider the tolerance of the wheel torque.

3 Computer Exercise Tasks

In the following tasks you should implement and evaluate an ESP based on the controller schemes that you established in the preparation tasks.

Task 7 – ESP implementation

Open MATLAB and navigate to the folder that you download from the course homepage. Implement your ESP controller in the `MyESP.m` file.

Task 8 – ESP evaluation

In the same folder you will find three prepared experiments. Use these to demonstrate that your controller operates as intended and that it actually aids the driver when needed. The three prepared tests are a J-Turn maneuver, slalom track, and double lane-change track:

- J-Turn maneuver (open `Task_8_Jturn.m`): A predefined steering signal is fed to the steering tires, turning with a constant rate to a maximum value before decreasing it (resulting in a vehicle path

similar to a J).

- Slalom track (open `Task_8_Slalom.m`): A predefined steering signal is implemented to drive around cones in a slalom pattern.
- Double lane-change track (open `Task_8_DoubleLaneChange.m`): A predefined steering signal is implemented to avoid an obstacle on the road.

For each task, first activate your controller by setting the value `ESP_mode` to `ON`, run the simulation then the results will be automatically saved in `result_ON.mat`. Then deactivate your controller by setting the value `ESP_mode` to `OFF`, the results will be automatically saved in `result_OFF.mat` by running the simulation. Finally open the file `makeplot.m` and visualize the results for comparison.

It is not a requirement that you use all tests to demonstrate your controller, use those that you think best demonstrate its function.

You should clearly demonstrate that your controller is working properly. Do this with data plots and possibly also visualizations, combined with discussions around what is seen in these figures and what you find interesting.

A Vehicle Parameters and Sensors

Table 1 specifies some useful vehicle parameters, and in Table 2 the different measurement signals, available for the ESP controller implementation, are listed.

Table 1 Vehicle parameters.

Parameter	Description	Value	Unit
m	Total vehicle mass	2100	kg
h	CoG height	0.50	m
L	Wheelbase	2.80	m
l_1	CoG to front axle	1.423	m
t_w	Wheel track width	1.6	m
R_w	Wheel radius	0.3	m
$C_{\alpha,f}$	Front axis cornering stiffness	100703.6788	N/rad
$C_{\alpha,r}$	Rear axis cornering stiffness	108361.2599	N/rad

Table 2 Vehicle sensors for the ESP controller.

Variable	Description	Unit
a_y	Lateral acceleration	m/s^2
v_x	Longitudinal velocity	m/s
v_y	Lateral velocity	m/s
Ω_z	Yaw rate	rad/s
δ_{sw}	Hand wheel steering angle	rad