Robustness analysis of a spark ignition engine

Clément Roos
ONERA

The term spark-ignition engine refers to internal combustion engines, usually petrol engines, where the combustion process of the air-fuel mixture is ignited by a spark from a spark plug. This is in contrast to compression-ignition engines, typically diesel engines, where the heat generated from compression is enough to initiate the combustion process, without needing any external spark.

A block diagram of the linearized open-loop engine model is shown in Figure 1. It is composed of three main blocks: the filling dynamics of the manifold chamber, the combustion process and the rotational dynamics of the engine. The inputs are the duty cycle of the throttle valve $D(t)$ and the spark advance position $A(t)$. The outputs are the relative air pressure of the manifold $P(t)$ and the engine speed $N(t)$. Finally, $K_1, \dots, K_7$ are static gains and J corresponds to the polar moment of inertia. Note that $1/J$ will be replaced with K_8 in the sequel for the ease of notation.

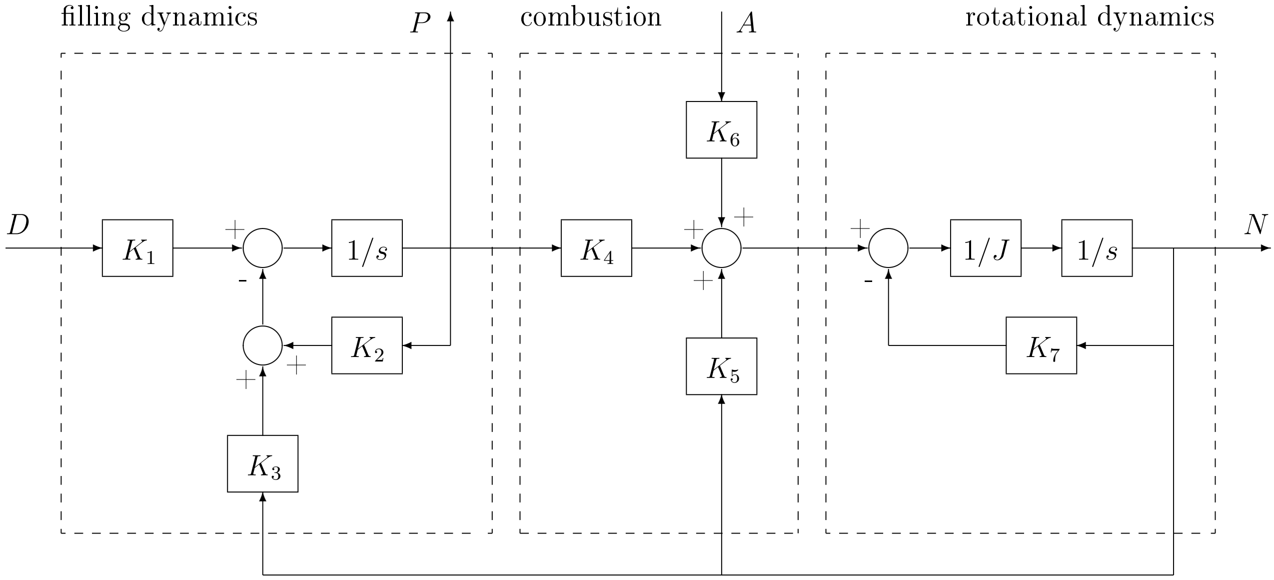

Figure 1: Block diagram of the linearized open-loop engine model

The engine can work at many different operating points. Three of them are considered here, which correspond to various working conditions. Point A represents the completely unloaded engine at idle speed. Point B represents a slightly loaded engine at idle speed. Finally, point C represents the case in which the first gear is engaged. Standard identification techniques are used to obtain three sets of parameters, which are given in Table 1.

Parameters	K_1	K_2	K_3	K_4	K_5	K_6	K_7	$K_8 = 1/J$
Point A	2.1608	0.1027	0.0357	0.5607	2.1999	3.8429	2.0283	1.0000
Point B	3.4329	0.1627	0.1139	0.2539	1.7993	1.8078	1.8201	1.0000
Point C	2.1608	0.1027	0.0357	0.5607	1.7993	3.8429	1.8201	0.1000

Table 1: Parameters values for the three operating points of interest

A controller is designed for operating point B, which is the most commonly occurring. The whole closed-loop interconnection is represented in Figure 2 and the controller gains are given below:

$$G = \begin{bmatrix} 0.0081 & 0.0872 & 0.1586 & -0.1202 \\ 0.0187 & 0.1826 & 0.0848 & -0.0224 \end{bmatrix}$$

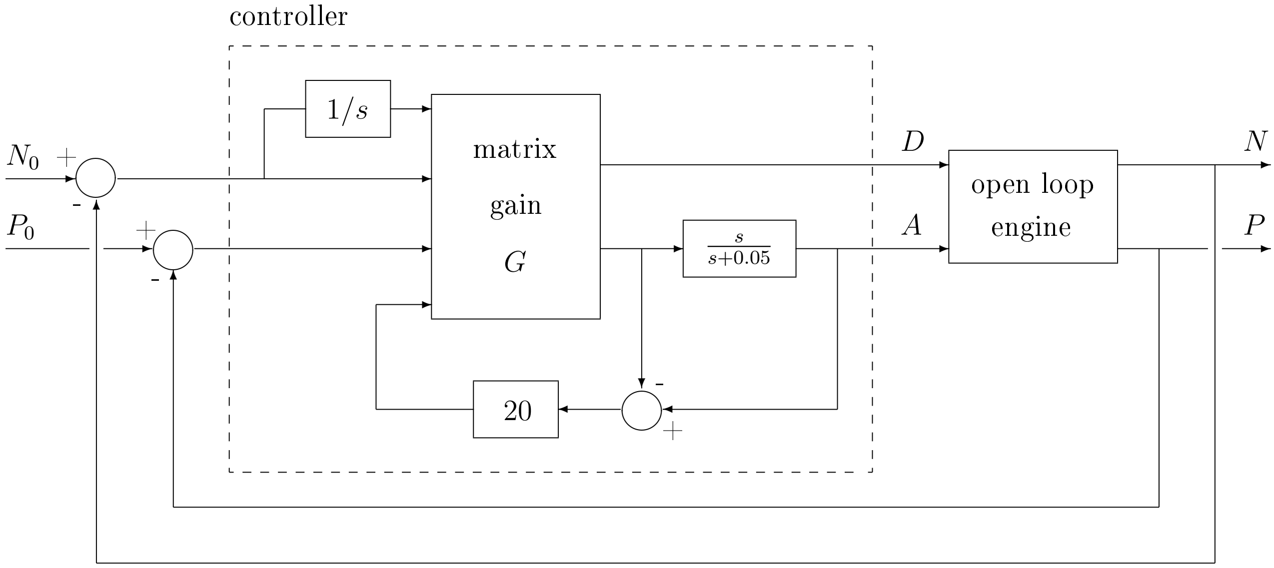

Figure 2: Block diagram of the closed-loop engine model

For each $i = 1, \dots, 8$, the values of K_i at operating points A, B and C are denoted K_i^A, K_i^B and K_i^C respectively. The assumption is made that during the normal operation of the engine, K_i may take all values between $K_i^- = \min(K_i^A, K_i^B, K_i^C)$ and $K_i^+ = \max(K_i^A, K_i^B, K_i^C)$. Hence, the range of variation of K_i is simply the interval $\left[K_i^-, K_i^+\right]$. As a result, the considered parametric domain $\mathbf{K}$ is a box defined as:

$$\mathbf{K} = \left\{(K_1, \dots, K_8) \in \mathbf{R}^8 : K_i^- \leq K_i \leq K_i^+\right\}$$

The objective is to determine whether the closed-loop interconnection of Figure 2 is stable for all $K \in \mathbf{K}$.

Instructions

You have to work in pairs. Evaluation will be performed on the basis of a report in pdf format, which should include all necessary comments, figures and pieces of Matlab code. All answers should be justified, but try to be concise!

In addition to standard Matlab tools, two libraries of the Systems Modeling Analysis and Control (SMAC) Toolbox are used in this work:

- the GSS library allows to build Linear Fractional Representations and can be downloaded at *http://w3.onera.fr/smac/gss*,
- the SMART library allows to compute bounds on the structured singular value μ and can be downloaded at *http://w3.onera.fr/smac/smart*.

There is no tricky question, even if you have not understood all the underlying theory. Nevertheless, you must be rigorous and make no mistake when handling the numerical values of the parameters, or you might quickly get into trouble. It is also important to read and understand the Matlab documentation before you use a function. For example, if you do not know how to use the function `fun`, first type `help fun` (or `help gss/fun` in case of an overloaded function).

1 Modeling

Q1.1 - By hand, compute the state-space representation of the open-loop engine shown in Figure 1 with state vector $x = [N\ P]^T$, input vector $u = [D\ A]^T$ and output vector $y = x$. Find the expressions of $\delta_1, \dots, \delta_7$ such that:

$$\dot{x} = \begin{bmatrix} \delta_5\delta_7 & \delta_4\delta_7 \\ \delta_3 & \delta_2 \end{bmatrix} x + \begin{bmatrix} 0 & \delta_6\delta_7 \\ \delta_1 & 0 \end{bmatrix} u \tag{1}$$

Using Matlab, define the parameters $\delta_1, \dots, \delta_7$ with their ranges of variation $[\delta_1^-, \delta_1^+], \dots, [\delta_7^-, \delta_7^+]$ and their nominal values $\dfrac{\delta_1^- + \delta_1^+}{2}, \dots, \dfrac{\delta_7^- + \delta_7^+}{2}$ (function `gss`). Remember that the gains $K_1, \dots, K_8$ can vary independently of each other. Create the open-loop parametric model (overloaded function `ss`).

Q1.2 - Compute a state-space representation of the controller shown in Figure 2. The state matrices can be obtained either by hand or by using Simulink and the function `linmod`.

Q1.3 - Create the closed-loop parametric model $\Sigma_1(\delta)$ represented in Figure 2 (overloaded function `feedback`). A Linear Fractional Representation `sys` is obtained, where the LTI model $M(s)$ and the structure of the block-diagonal matrix $\Delta = \text{diag}(\delta_1 I_{q_1}, \ldots, \delta_7 I_{q_7})$ are given by the fields `sys.M` and `sys.D` respectively. Give the order of $M(s)$ and the values of $q_1, \ldots, q_7$ (overloaded function `size`).

2 Stability analysis

Q2.1 - Check closed-loop stability at the operating points A, B and C (overloaded function `eval`).

Q2.2 - Evaluate the stability of the closed-loop parametric model $\Sigma_1(\delta)$ with 1000 simulations (function `dbsample`). Plot the eigenvalues of all samples on the same figure. In case an unstable sample is found, give the corresponding values of $\delta_1, \ldots, \delta_7$.

Q2.3 - Evaluate the stability of the closed-loop parametric model $\Sigma_1(\delta)$ with μ-analysis:

- Remove all exogenous inputs and outputs of $\Sigma_1(\delta)$ to obtain an LFR for stability analysis.
- Compute μ upper and lower bounds on the whole frequency range (functions `muub` and `mulb`). You can use the following Matlab code:

```
optub.lmi=1;optub.tol=0;
ubnd=muub(sys,[],optub);
[lbnd,wc,pert,iodesc]=mulb(sys);
```

- Conclude about the stability of the closed-loop model on the parametric domain $\mathbf{K}$. Note that the Δ matrix is normalized in `muub` and `mulb`, so stability is ensured on $\mathbf{K}$ if and only if the largest value of μ over the whole frequency range is less than 1. In case an unstable configuration is found, give the corresponding values of $\delta_1, \ldots, \delta_7$ (output argument `iodesc` of function `mulb`) and check your result using the overloaded function `eval`.

Q2.4 - Compare the results of Q2.2 and Q2.3.

3 Adjustment of the combustion chamber

Following the conclusions of Q2.2 and Q2.3, some adjustments are performed inside the combustion chamber. New values are identified for K_5 and K_6, as shown in Table 2.

Parameters	K_5	K_6
Point A	2.0183	4.4962
Point B	1.7993	2.0247
Point C	1.7993	4.4962

Table 2: New parameters values for the three operating points of interest

Q3.1 - Define the parameters $\delta_1, \ldots, \delta_7$ with their new ranges of variation. Create the new closed-loop parametric model $\Sigma_2(\delta)$.

Q3.2 - Evaluate the stability of the closed-loop parametric model $\Sigma_2(\delta)$ with μ-analysis. Conclude about the stability of the closed-loop model on the parametric domain $\mathbf{K}$.

4 Improvement of the analysis results

4.1 By applying branch-and-bound

The gap between the μ upper and lower bounds can be reduced by applying a branch-and-bound algorithm. The idea is to partition the uncertainty domain in more and more subsets until the gap between the highest lower bound and the highest upper bound computed on all subsets becomes less than a user-defined thresholdϵ. Such an algorithm is known to converge, *i.e.* ϵ can be reduced to an arbitrarily small value. However, it suffers from an exponential growth of computational complexity as a function of the number of real uncertainties. The choice of ϵ allows to handle the tradeoff between accuracy and computational time.

Q4.1 - Apply the branch-and-bound algorithm to the closed-loop parametric model $\Sigma_2(\delta)$ created in Section 3. You can use the following Matlab code:

```
optbb.maxgap=0.05;
[lbnd,wc,pert,iodesc]=mubb(sys,[],optbb);
```

Conclude about the stability of the closed-loop model on the parametric domain **K**.

4.2 By reducing the size of Δ

The gap between the μ upper and lower bounds strongly depends on the considered LFR. Although this is not an exact rule, the trend is as follows: the larger the size of Δ, the larger the gap.

Q4.2 - By carefully reading the description of the engine, determine whether the LFRs used until now are minimal. If no, determine the smallest possible values of $q_1, \ldots, q_7$ in the matrix Δ. Find a suitable factorization of equation (1), which leads to this minimal representation. Create the new open-loop parametric model using the numerical values of Section 3 and the corresponding closed-loop parametric model $\Sigma_3(\delta)$.

Q4.3 - Evaluate the stability of the closed-loop parametric model $\Sigma_3(\delta)$ with μ-analysis. Conclude about the stability of the closed-loop model on the parametric domain **K**. Compare the results with those obtained in Q3.2 and Q4.1.