

TSFS06 LAB EXERCISE 2

Diagnosis of Fuel Evaporative System

April 8, 2019

1 OBJECTIVE

The objective with this laboratory exercise is to read, understand, and implement an algorithm described in a scientific publication. The implemented algorithm should also be evaluated based on experimental data. In this case, the application is detection of air leakage in a fuel tank in an automotive vehicle and the paper is “*Leakage Detection in a Fuel Evaporative System*” [1].

The description of the exercise starts with a short introduction to fuel evaporative diagnosis in Section 2. Available data and Matlab files is summarized in Section 3. A description of preparation exercises, the laboratory assignment, and the report is given in Section 4. Appendix A contains the Matlab skeleton files that is used in the exercise.

2 BACKGROUND

Emission based legislation requires that fuel can not evaporate in to the ambient air. This has been addressed by a so called *purge system*, where fuel vapor is led from the fuel tank into a *carbon canister*. The carbon canister is a container with active carbon that, through a *diagnostic valve*, is open to the ambient air. Evaporated hydro carbons are collected in the carbon canister and the canister is gradually filled. When the canister is full and cannot contain further fuel vapor, the canister need to be *purged*. this is done regularly through a *purge valve* that is located between the canister and the under pressure side of the intake manifold of the automotive engine. If the cleaning valve is open, air is drawn trough the carbon canister, with the hydro carbons, into the engine and can be combusted together with the normally injected fuel. When the canister is empty, the purge valve is closed and is again ready to contain hydro carbons. A schematic figure of the purge system is shown in Figure 1.

Since the system is emission related, the purge function is under law required to be supervised, i.e., the vehicle must itself detect any faults, set a diagnostic trouble code in the control system, and notify the vehicle owner to go to the workshop. A common setup to implement such supervision is to utilize two extra components, a *diagnosis valve* and a *pressure sensor*. The diagnosis valve is used to open and close the canal between the carbon canister and the environment. The pressure sensor is mounted in the fuel tank. The diagnosis valve is normally open and the purge valve closed. Any fault that affect the emissions above a certain level must be detected and the fault to monitor in this exercise is a leakage of fuel vapor from the fuel tank. In automotive diagnosis, this is referred to as fuel evaporative diagnosis.

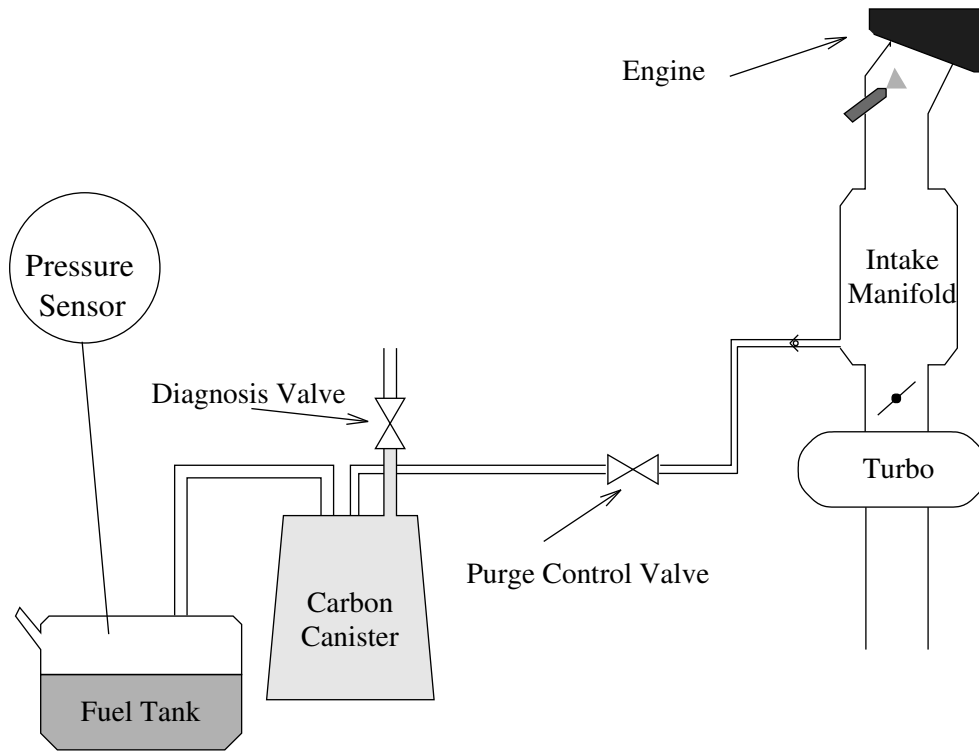


Figure 1: Purge system for automotive vehicle.

3 DATA SEQUENCES AND SKELETON FILES

In the exercise, measured data sequences are used that describes four different faulty case where the known hole diameter is 0.5, 1, 3.5 and 5 mm respectively. There is also a data sequence describing a system with no leakage. To evaluate your detection system, there is a sixth data sequence with an unknown fault status.

The data sequences can be downloaded from the course web-page. The file with the data is called `diagnosisFile.mat` and contains the variable `diagnosisSequences` which is a data structure in Matlab. The structure contains 6 fields, where each field also is a data structure. The data is stored according to the following table:

Field	Leakage
F00: [1x1 struct]	No leakage
F05: [1x1 struct]	0.5 mm leakage
F10: [1x1 struct]	1 mm leakage
F35: [1x1 struct]	3,5 mm leakage
F50: [1x1 struct]	5 mm leakage
Fxx: [1x1 struct]	Unknown case

To, for example, access data for the fault free case, the variable `diagnosisSequences.F00` is used at the Matlab prompt. All fault cases in turn contains the variables:

Variable	Unit	Description
time	s	Time
y	Pa	Measured tank pressure relative the ambient pressure
D	-	Diagnosis valve position
P	-	Purge valve position
Ts	s	Sample time
k1	-	Temperature and pressure dependent evaporation constant

On the course web-page there are two skeleton files to download to help structure your implementation. The file `lab2_main.m` is the main file and `Leakage_test.m` is

a function where you should implement your detection algorithm. These files are also included in Appendix A.

4 LABORATORY EXERCISE

The laboratory exercise should be performed in groups of maximum two persons. Below is specified the exercises that has to be solved and the are to be reported to pass the laboratory exercise.

4.1 Requirements on the diagnosis system

The developed detection system should be able to detect leakages in a fuel tank and should be implemented according to the description in [1]. Also, the solution should handle that the pressure sensor has an unknown bias and that the noise environment may vary between different data sets.

4.2 Report requirements

To understand the model and the detection method, the paper [1] has to be read thoroughly before the laboratory session. There are also a number of preparatory exercises related to the paper and the method you shall implement in Section 4.3. You should be able to orally report these exercises at the laboratory session.

The exercise is examined by a written report and there you shall discuss the relevant preparatory exercises in Section 4.3, the exercises in Section 4.4, and any questions and conclusions that arise during the laboratory work. The report shall also include comments to design choices made, how well the implemented algorithm manage to detect the different data sequences provided and described in Section 3. The calculations for preparatory exercises 3, 7, and 8 do not need to be demonstrated in the final written report.

The report shall be written as an independent document, i.e., it should be possible to read and understand the report without this laboratory memo or the paper [1] before. Further, it is not acceptable to write the report as a list of answers to the questions in this laboratory memo. Also, it can be tempting to include the source Matlab code in the report to explain what has been done. The Matlab code is *not* considered an explanation, explanations to what you have done shall be clear from the text. If you do want to include the code in an appendix, that is of course fine.

Suggestions for sections in the report is introduction/background, problem formulation, method/technical solution, analysis, discussion, and conclusions. It is not intended that the report shall be long, instead try to keep it short enough to make it a high quality report. Also, it is not allowed to use paragraphs from other texts, e.g., this text or paper [1]. One exception is Figure 1 in this material that is available together with the data sequences from the course web-page. If you use the figure in your report, there shall be a reference and a statement that you use the figure with permission.

Deadline for handing in the report: See course web-page <http://www.vehicular.isy.liu.se/Edu/Courses/TSFS06/>.

4.3 Preparatory exercises

The preparatory exercises should be done before the laboratory session.

1. Read CARB (California Air Resources Board) regulations [2] with respect to evaporative diagnosis ("Evaporative System Monitoring" on page 37) to understand what emission related diagnostic regulations look like.
2. Read the paper [1] carefully.
3. Derive the model of the fuel tank (9) using (2), (7) and (8) in [1].
4. In [1], leakage is detected by first lowering the pressure in the tank and then observe the pressure behavior. Another method could be to increase the pressure in the tank and then, in the same way, observe the pressure trace. Name advantages and disadvantages with respectively method.
5. The data sequences in Figure 2 and 3 in [1] can be separated into different intervals depending on the position of the two valves D (diagnosis valve) and P (purge valve). For example, the sequence starts with the diagnosis valve open and the purge valve closed which is one out of four possible combinations. Equation (9) only describes the pressure behavior in some of these four valve configurations. Which? Also state for which intervals that the partial air pressure p_a can be assumed constant. Are the answers to these questions dependent on if there is a leakage or not, and if so how?
6. Based on the discussions in the previous preparation exercise, write down a control strategy for the two valves such that the bias error can be compensated and leakage detected.

Leading question: Is it appropriate to identify the bias and detect leakage in the same valve configuration, and if so under which circumstances? Can the valves be controlled such that unknown model parameters are decoupled? Related to the physical process.

7. Do exercises 3.4-3.7 in the lesson course material.
8. To formulate a detection test using linear regression, you need a time-discrete version of equation (9) which is given by (10). Perform a time-discretization using an Euler-forward derivative approximation.
9. In the model there is an unknown term $k_1(p_f^0 + p_a - p_{\text{amb}})$.
 - If the diagnosis valve is open and the tank pressure has converged to a stationary value, which value has then the term $k_1(p_f^0 + p_a - p_{\text{amb}})$?
 - If the purge valve has been opened to lower the tank pressure in the tank and then closed again, the diagnosis valve is closed, and there is no leakage, which value then has the term $k_1(p_f^0 + p_a - p_{\text{amb}})$?
10. Formulate a test quantity T to detect leakages as described in [1]. Note, the noise size/characteristics may vary between different data sets. How is the noise variance σ^2 estimated when there is a leakage?
11. The test quantity T is assumed to, in the fault free case, be χ^2 distributed with the number of degrees of freedom equal to the number of rows in Φ minus the number of estimated parameters, see Section 4 in [1].

Important: Which assumptions has to be fulfilled for T to be χ^2 distributed and how can these assumptions be corroborated?
12. Describe how the threshold can be determined to achieve a given false alarm probability.

4.4 Laboratory exercises

The exercises are divided into two parts, leakage detection and hole size estimation.

4.4.1 Exercise 1: Leakage detection

The exercise is to design a detection system for leakages based on a pressure sensor with unknown bias. For the exercise there are two files, `lab2_main.m` and `Leakage_test.m`, downloadable from the course web-page that you can use to organize your code and solution. These files are also included in Appendix A. The following bullet list can help structure the work.

1. Compensate for bias errors in the measurements.
2. Describe which parts of the data sequences that is used in the detection. It is not mandatory to automatically extract the parts of the data for detection, it is OK to do it by hand.
3. In [1], the parameter k_1 depends on temperature T and the amount of fuel in the tank V , and is assumed to be known. Here, k_1 is available for each data set, see Section 3.
4. Implement the leakage detection algorithm described in [1]. Note that the selected data intervals are design parameters in your solution. How should they be selected?
5. Verify/corroborate that the assumptions made in the paper is appropriate. Hint: Matlab functions `normplot` and `xcorr` could be useful.
6. Determine thresholds such that the false alarm is approximately 1%. Hint: Matlab function `chi2inv` could be useful.
7. Test your implementation with suitable data sequences and verify that the system reacts as expected.
8. Test your implementation on the data sequence `diagnosisSequences.Fxx` and report the results.

4.4.2 Exercise 2: Hole size estimation

Given that a leakage has been detected, it could be of interest to estimate the hole diameter, something that is *not* discussed in the paper [1]. In this exercise you should analyze if it is possible to estimate a hole diameter when a leakage has been detected, based on the model that is included in the paper. The following bullet points can help you organize the work.

1. Formulate a model for the relationship between k_2 and the hole diameter.
2. Formulate an estimation model for the parameter k_2 .
3. Test with suitable data sequences.
4. Analyze the performance of the estimator and comment which uncertainties there might be in your proposed method.

A APPENDIX

A.1 lab2_main.m

```
%% *** TSFS06 Laboration 2 Skeleton main file: Diagnosis of Purge System ***
clc
clear
close all

%% *** Data Loading ***
load diagnosisFile.mat
```

```

%% *** Plot data ***
d = diagnosisSequences.F10;

figure(1);
plot( d.time, d.y);

figure(2);
plot( d.time, d.D);
hold on;
plot( d.time, d.P,'r');
legend('Diagnosis valve position','Purge valve position');

%% *** Set index intervals for all datasets
diagnosisSequences.F00.biasIdx = []; %(Samples [start end] for bias estimation)
diagnosisSequences.F00.diagIdx = []; %(Samples [start end] for diagnosis sequence 1)
diagnosisSequences.F00.diagIdx2 = []; %(Samples [start end] for diagnosis sequence 2)

diagnosisSequences.F10.biasIdx = [];
diagnosisSequences.F10.diagIdx = [];
diagnosisSequences.F10.diagIdx2 = [];

diagnosisSequences.F35.biasIdx = [];
diagnosisSequences.F35.diagIdx = [];
diagnosisSequences.F35.diagIdx2 = [];

diagnosisSequences.F50.biasIdx = [];
diagnosisSequences.F50.diagIdx = [];
diagnosisSequences.F50.diagIdx2 = [];

diagnosisSequences.F05.biasIdx = [];
diagnosisSequences.F05.diagIdx = [];

diagnosisSequences.Fxx.biasIdx = [];
diagnosisSequences.Fxx.diagIdx = [];

%% *** Detection results ***
pfa = 0.01; % Probability of false alarm

d = diagnosisSequences.F00;
res00a = Leakage_test( d, d.biasIdx, d.diagIdx, pfa) ;
res00b = Leakage_test( d, d.biasIdx, d.diagIdx2, pfa) ;

d = diagnosisSequences.F05;
res05 = Leakage_test( d, d.biasIdx, d.diagIdx, pfa) ;

d = diagnosisSequences.F10;
res10a = Leakage_test( d, d.biasIdx, d.diagIdx, pfa) ;
res10b = Leakage_test( d, d.biasIdx, d.diagIdx2, pfa);

d = diagnosisSequences.F35;
res35a = Leakage_test( d, d.biasIdx, d.diagIdx, pfa) ;
res35b = Leakage_test( d, d.biasIdx, d.diagIdx2, pfa);

d = diagnosisSequences.F50;
res50a = Leakage_test( d, d.biasIdx, d.diagIdx, pfa) ;

```

```

res50b = Leakage_test( d, d.biasIdx, d.diagIdx2, pfa);

d = diagnosisSequences.Fxx;
resxx = Leakage_test( d, d.biasIdx, d.diagIdx, pfa) ;

%% *** Assumptions validation

```

A.2 Leakage_test.m

```

function res=Leakage_test( data, biasIdx, diagIdx, pfa)
% Leakage_test - Detect leakages in purge system
%
% Inputs:
% 1) data      (Measured data)
% 2) biasIdx   (Samples [start end] for bias estimation)
% 3) diagIdx   (Samples [start end] for diagnosis sequence)
% 4) pfa,     (Probability of false alarm)
%
% Outputs:
% res is a struct variable which includes:
% 1) T, (The test quantity)
% 2) J, (The threshold value)
% 3) R, (The residual vector)

% Compute test quantity, threshold, and residual.
% Write, and use, the three functions
% 1. EstimateBias()
% 2. EstimateVariance()
% 3. TQCalc()

T = 0;
J = 0;
R = 0;

% Return results
res.T = T;
res.J = J;
res.R = R;
end

function bhat = EstimateBias(y)
% Estimate bias
bhat = ;
end

function s2 = EstimateVariance(y)
% Estimate variance

s2 = ;
end

function [T, R, J] = TQCalc(y,s2,pfa,k1,Ts)
% Compute test quantity, residual, and threshold

```

```
R = ;  
T = ;  
J = ;  
end
```

REFERENCES

- [1] Mattias Krysander and Erik Frisk, *Leakage Detection In a Fuel Evaporative System*, Control Engineering Practice, 2009
- [2] Californian Air Resources Board, §1968.2. *Malfunction and Diagnostic System Requirements—2004 and Subsequent Model-Year Passenger Cars, Light-Duty Trucks, and Medium-Duty Vehicles and Engines*, http://www.arb.ca.gov/msprog/obdprog/section1968_2_clean2013.pdf, April 2014