Contents

[MBD: Designing a complex automotive system 1](#_Toc59133989)

[Name: Rasika Shejul | Unique ID: 2005457 1](#_Toc59133990)

[TOPIC: Modeling a fault-tolerant fuel control system 1](#_Toc59133991)

[Data inspector 3](#_Toc59133992)

[Solver selection strategy 3](#_Toc59133993)

[MATLAB function block 3](#_Toc59133994)

[Look-up table 3](#_Toc59133995)

[Result and conclusion: 4](#_Toc59133996)

[References: 7](#_Toc59133997)

# MBD: Designing a complex automotive system

# Name: Rasika Shejul | Unique ID: 2005457

# TOPIC: Modeling a fault-tolerant fuel control system

This model represents a fuel control system for a gasoline engine. The system is highly robust in that individual sensor failures are detected and the control system is dynamically reconfigured for uninterrupted operation. The ideal (i.e. stoichiometric) mixture ratio provides a good compromise between power, fuel economy, and emissions. The target air-fuel ratio for this system is 14.6. Typically, a sensor determines the amount of residual oxygen present in the exhaust gas (EGO). This gives a good indication of the mixture ratio and provides a feedback measurement for closed-loop control. If the sensor indicates a high oxygen level, the control law increases the fuel rate. When the sensor detects a fuel-rich mixture, corresponding to a very low level of residual oxygen, the controller decreases the fuel rate.

Sensors used: throttle position sensor, engine speed sensor, EGO sensor, MAP sensor

Output: constant AFR

This example shows how to combine Stateflow with Simulink to efficiently model hybrid systems. Traditional signal flow is handled in Simulink while changes in control configuration are implemented in Stateflow.

**Working of each block:**

1. Fuel rate control:

* Chart: control logic

A single Stateflow chart, consisting of a set of six parallel states, implements the control logic in its entirety. The four parallel states shown at the top of Figure 4 correspond to the four individual sensors. The remaining two parallel states at the bottom consider the status of the four sensors simultaneously and determine the overall system operating mode. The model synchronously calls the entire Stateflow diagram at a regular sample time interval of 0.01 sec. This permits the conditions for transitions to the correct mode to be tested on a timely basis.

* Subsystem: Airflow calculation and fuel calculation.

The Airflow Calculation block is the location for the central control laws. This block is found inside the Fuel rate control subsystem. The block estimates the intake air flow to determine the fuel rate which gives the appropriate air/fuel ratio. Closed-loop control adjusts the estimation according to the residual oxygen feedback in order to maintain the mixture ratio precisely.

The fuel subsystem (within the Fuel rate control subsystem) sets the injector signal to match the given airflow calculation and fault status. The first input is the computed airflow estimation. This is multiplied with the target fuel/air ratio to get the commanded fuel rate. Normally the target is stoichiometric ratio.

1. Engine gas dynamics:

* Mixing, combustion

The nonlinear oxygen sensor (EGO Sensor block) is found inside the Mixing & Combustion block within the Engine Gas Dynamics subsystem. EGO Sensor is modeled as a hyperbolic tangent function, and it provides a meaningful signal when in the vicinity of 0.5 volt. The raw error in the feedback loop is thus detected with a switching threshold. If the air-fuel ratio is low (the mixture is lean), the original air estimate is too small and needs to be increased. Conversely, when the oxygen sensor output is high, the air estimate is too large and needs to be decreased. Integral control is utilized so that the correction term achieves a level that brings about zero steady-state error in the mixture ratio.

The normal closed-loop operation mode, LOW, adjusts the integrator dynamically to minimize the error. The integration is performed in discrete time, with updates every 10 milliseconds. When operating open-loop however, in the RICH or O2 failure modes, the feedback error is ignored and the integrator is held. This gives the best correction based on the most recent valid feedback.

# Data inspector

The model loads necessary data into the model workspace from data.m. The model logs relevant data to MATLAB workspace in a data structure and streams the data to the Simulation Data Inspector. Logged signals are marked with a blue indicator while streaming signals are marked with the light blue badge

# Solver selection strategy

Since the system continuous type of model, Variable step solver is used.

ode45 performs well with most ODE problems and should generally be your first choice of solver. ode45 (Dormand – Prince) solver is used for nonstiff type of models and gives medium accuracy.

# MATLAB function block

MATLAB functions: theta and ratio, EGO.

Functions ->open in MATLAB.

# Look-up table

Lookup table blocks use arrays of data to map input values to output values, approximating mathematical functions. To approximate a function in N variables, use the [n-D Lookup Table](https://localhost:31515/static/help/simulink/slref/ndlookuptable.html) block.

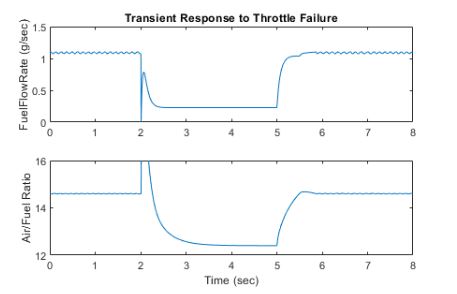
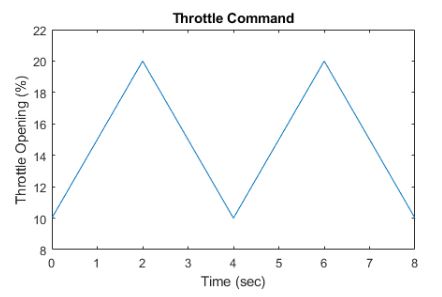
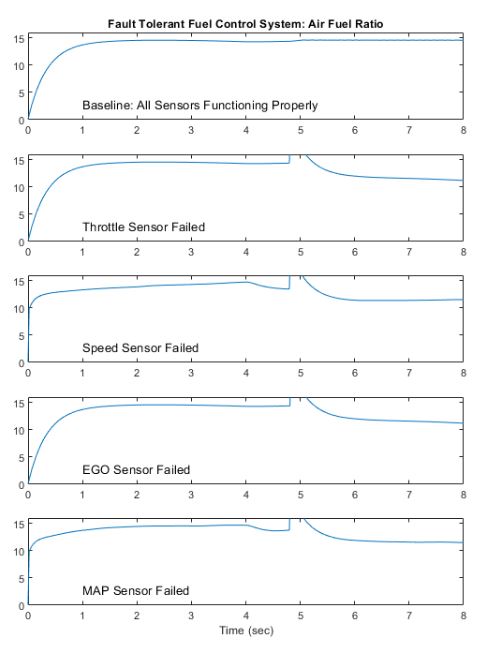
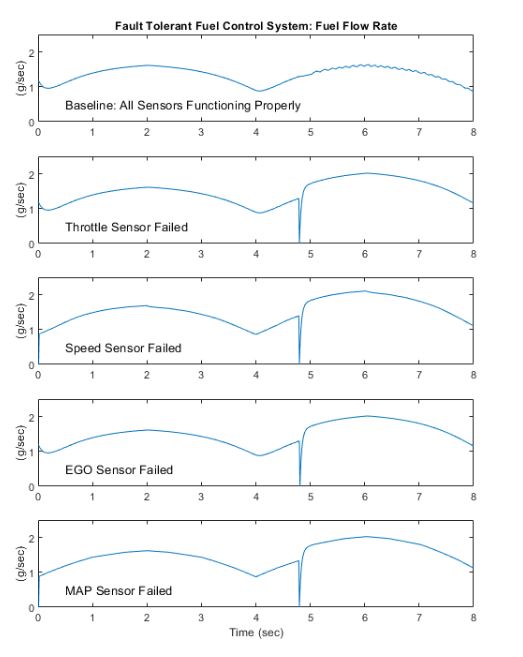
# Result and conclusion:

IDEAL RESULT:

The simulation is run with a throttle input that ramps from 10 to 20 degrees over a period of two seconds, then goes back to 10 degrees over the next two seconds. This cycle repeats continuously while the engine is held at a constant speed so that the user can experiment with different fault conditions and failure modes. Click on a sensor fault switch in the dashboard subsystem to simulate the failure of the associated sensor. Repeat this operation to slide the switch back for normal operation.

Figure 1 compares the fuel flow rate under fault-free conditions (baseline) with the rate applied in the presence of a single failure in each sensor individually. In each case note the nonlinear relationship between fuel flow and the triangular throttle command (shown in Figure 3). In the baseline case, the fuel rate is regulated tightly, exhibiting a small ripple due to the switching nature of the EGO sensor's input circuitry. In the other four cases the system operates open loop. The control strategy is proven effective in maintaining the correct fuel profile in the single-failure mode. In each of the fault conditions, the fuel rate is essentially 125% of the baseline flow, fulfilling the design objective of 80% rich.

Figure 2 plots the corresponding air/fuel ratio for each case. The baseline plot shows the effects of closed-loop operation. The mixture ratio is regulated very tightly to the stoichiometric objective of 14.6. The rich mixture ratio is shown in the bottom four plots of Figure 2. Although they are not tightly regulated, as in the closed-loop case, they approximate the objective of air/fuel (0.8\*14.6=11.7).

The transient behavior of the system is shown in Figure 4. With a constant 12 degree throttle angle and the system in steady-state, a throttle failure is introduced at t = 2 and corrected at t = 5. At the onset of the failure, the fuel rate increases immediately. The effects are seen at the exhaust as the rich ratio propagates through the system. The steady-state condition is then quickly recovered when closed-loop operation is restored.

# References:

1. MATLAB documentation for topic modeling a fault tolerant fuel control system.