FRTN05 Nonlinear Control and Servo Systems

Laboratory Exercise 1

Control of an Air Throttle with Dead-Zone

Johan Gagner, Rickard Bondesson, Bo Bernhardsson, Dept. of Automatic Control

Revised by Dan Henriksson, December 2006, Martin Ansbjerg Kjaer 2008, Andreas Stolt 2014

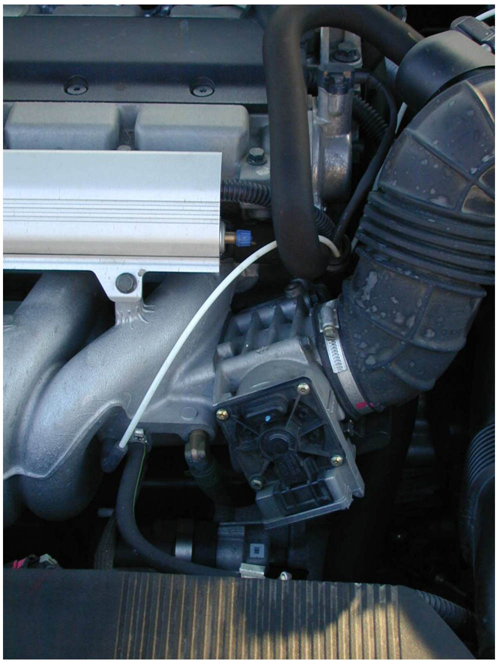

Figure 1 Throttle unit in a Volvo engine (the same that is used in the lab). By controlling the angular position of the circular plate inside the pipe, different amounts of air are fed into the engine. A fast throttle unit improves the possibility to do efficient and reliable engine control. The position of the throttle in the overall Volvo engine system is shown in Figure 15 at the end of this manual.

1. Introduction

The goal of this laboratory exercise is to control the air flow through a throttle unit. Throttle units are used for controlling the airflow to combustion engines. The model that you will work with was used in Volvo cars between 1999 and 2001, e.g., S60, V70, S70, and XC70. The main focus of this laboratory exercise will be on nonlinearities that affect the throttle control.

1.1 Contents

The throttle has severe nonlinearities which complicates good air flow control. Therefore, we first design an accurate position control to cope with the nonlinearities in the throttle, and then close an extra flow control loop around the position control. This gives the cascaded structure as shown in Figure 2. The main effort will be put on the position control, since this part is the most problematic.

The laboratory exercise consists of two parts:

1. Perform simulations in Matlab/Simulink to explore the system nonlinearities, the position control loop, and possible limit cycles. These simulations can be done at the department or at any computer with Matlab, Simulink and the Control System Toolbox.

2. Try position and flow control on the real throttle unit.

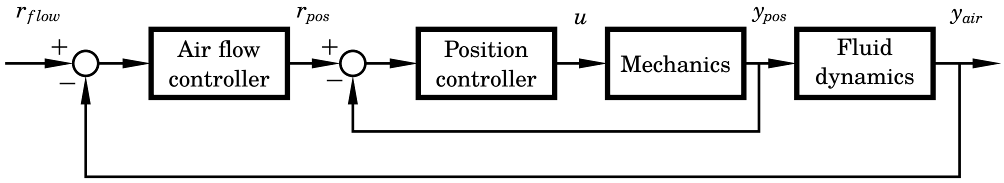

Figure 2 Block diagram of the cascaded control structure.

1.2 Preparations

Read through this manual and complete Assignments 1-5.

2. Background and Modeling of the throttle

2.1 About the Throttle

The throttle unit (see Figure 1) controls the air flow into the engine and is a crucial part of the engine control system. The position of the throttle in the overall Volvo engine system can be seen in Figure 15 at the end of this manual. The plate of the throttle is controlled by a DC motor and the plate position is measured with a potentiometer. The measured angle is approximately 0 degrees when the throttle is closed and 90 degrees when it is fully open.

A servo-controller has been designed to keep the throttle plate at the desired angle. The plate angle set-point is calculated by the electronic Engine Control Module of the car as a function of the accelerator pedal position, i.e., there is no mechanical connection between the accelerator pedal and the throttle.

The constructors have introduced a feature called the *limp-home mode* for the case that the control electronics, for some reason, would stop working. If there is a failure in the throttle controller, two springs will move the plate to a specific angular position. At this angle a small amount of air feeds the engine, which makes it possible to drive the car to the nearest workshop. While such a feature saves the customer, and Volvo, from a lot of trouble if anything goes wrong, it also creates a nonlinearity in the control loop, which decreases the total performance of the engine unless compensated for by the controller.

2.2 A Simple Model of the Nonlinearity

Two opposite directed torsion springs with different work ranges are used for the limp-home feature. When there is no torque asserted by the motor, the springs move the plate to the limp home position. The limp home position is also referred to as the *dead-zone position*, since the springs create a dead-zone nonlinearity around this angular position.

If the electric time constant in the throttle motor is neglected, the effect of the springs can be modeled as a relay acting around the limp home position, see Figure 3. The nonlinearity is defined by the dead-zone position, θ_0, and the spring torques, T_u and T_l. The spring torques usually differ between the upper ($\theta > \theta_0$) and lower ($\theta < \theta_0$) regions, i.e., $|T_u| \neq |T_l|$.

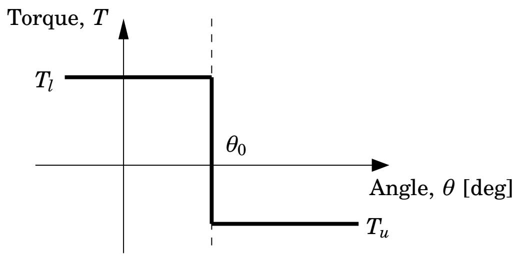

Figure 3 The springs are modeled as a relay acting around the dead-zone position, θ_0. The spring torques (T_l and T_u) are different (in sign and magnitude) above and below the dead-zone position.

In reality, the torques asserted by the torsion springs increase with the throttle angle. However, the effect of the dead-zone nonlinearity is only relevant for angles close to the dead-zone position, which makes the approximation with constant torques accurate enough for the purpose of this laboratory exercise.

Figure 4 shows a static model of the open-loop response of the throttle unit. The dead-zone nonlinearity makes the system insensitive to small input signals, since the control input must overcome the spring torques before affecting the output. Outside the dead-zone, the stationary process output θ, changes linearly with the input. The static gains of the system above and below the dead-zone are denoted k_u and k_l.

2.3 The Closed-Loop System

Figure 5 shows the closed-loop system including a linear controller, $C(s)$, the linear system model, $P(s)$, and the springs that cause the dead-zone nonlinearity. The springs act as an input disturbance on the process. Design of the controller, $C(s)$, is not part of this laboratory.

The total input to the system is given as $u = u_d + T(\theta)$, where u_d denotes the control signal from the linear controller and $T(\theta)$ is the additional spring torque as given by Figure 3.

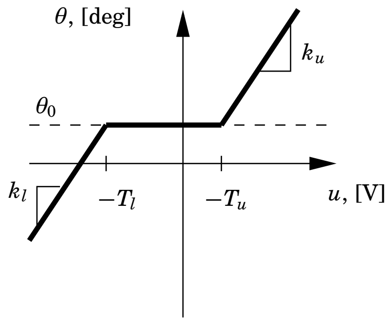

Figure 4 Open-loop response of the throttle unit with dead-zone. The input must overcome the spring torques before the plate can move. After that, the output changes linearly with the input, where k_u and k_l are the static gains of the system above and below the dead-zone.

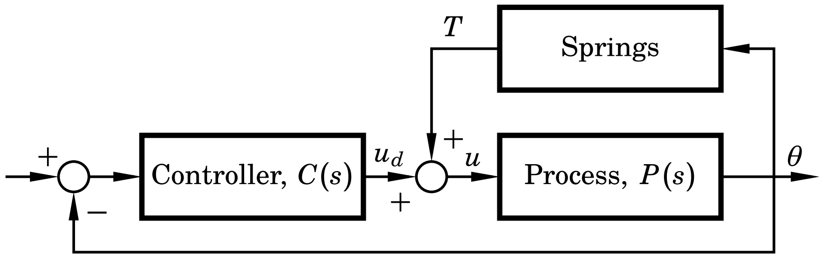

Figure 5 The spring nonlinearity affecting the closed-loop system. The spring torque is fed back with a positive sign to the input of the process.

2.4 Dead-Zone Compensation

The effect of the springs can be cancelled by modifying the control signal to the throttle motor according to Figure 6. The compensator adds a spring compensation, $-\hat{T}$, to the control signal from the linear controller. The spring compensation is based on estimates $\hat{T}_l$, $\hat{T}_u$ and $\hat{\theta}_0$, as shown in Figure 7. With the compensation, the total input to the system becomes

$$u = v + T = u_d + T - \hat{T} \tag{1}$$

With good estimates of the dead-zone parameters the compensator makes much of the trouble due to the springs disappear. If the compensation is perfect, $\hat{T} = T$ and $u = u_d$.

Estimation of the Spring Torques Good control performance requires a compensator with accurate estimates of the spring torques. There are many ways to estimate the parameters of the springs. The parameter values can be calculated before the engine starts or they can be updated during operation of the throttle.

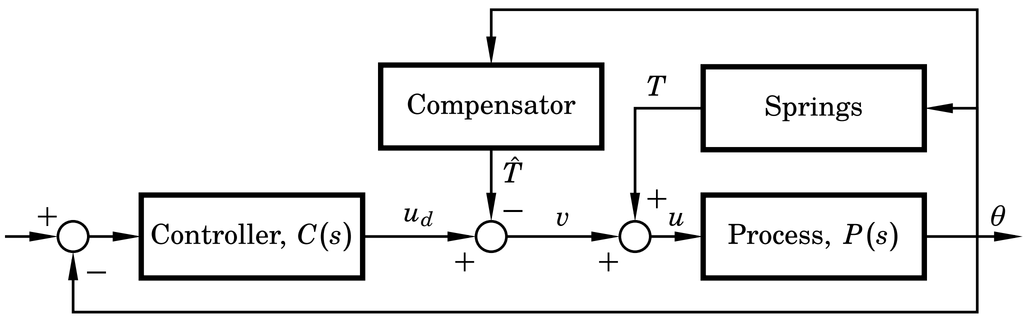

Figure 6 The springs and the compensation affecting the closed-loop system. The resulting input to the system is, $u = u_d + T - \hat{T}$. If the compensation is perfect, $\hat{T} = T$ and $u = u_d$.

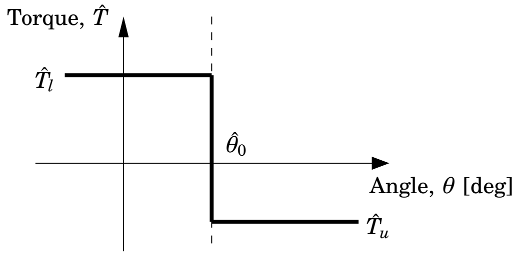

Figure 7 The compensation is based on dead-zone parameter estimates, $\hat{T}_l$, $\hat{T}_u$ and $\hat{\theta}_0$. Compare with the spring characteristics in Figure 3.

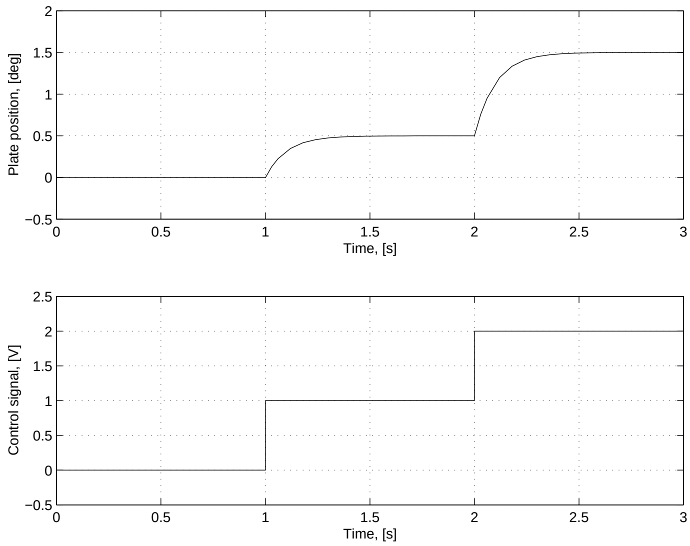

Figure 8 The spring torques, T_l and T_u, can be determined by examining a series of open-loop step responses. Two step responses are needed in each direction. Use $u > 0$ to determine T_u and $u < 0$ to determine T_l. In this figure we have $\theta_0 = 0$, which we will also assume in the simulation part of this laboratory.

Assignment 1: *How could the spring torques, T_l and T_u, be estimated with a series of open-loop step responses (similar to Figure 8). Find a formula for T_u and T_l. Remember that the stationary process output is given by $\theta = k_u \cdot (u + T_u)$ if $u > -T_u$ and $\theta = k_l \cdot (u + T_l)$ if $u < -T_l$.*

Artificial Nonlinearity Introduced by $\hat{\theta}_0 \neq \theta_0$ Now assume that we have obtained good estimates of the spring torques, $\hat{T}_l$ and $\hat{T}_u$, but that the estimated dead-zone position, $\hat{\theta}_0$, differs from the true position, θ_0. The compensation and the springs will then introduce a new nonlinearity, $T(\theta) - \hat{T}(\theta)$, as given by Figure 9. The effect of this nonlinearity will depend on the offset, $|\hat{\theta}_0 - \theta_0|$.

We will use describing function analysis to analyze possible limit cycles that could arise due to this artificial nonlinearity. Oscillations in the air flow are unacceptable in a car engine due to: increased wear of the throttle, worse fuel economy, and traffic safety.

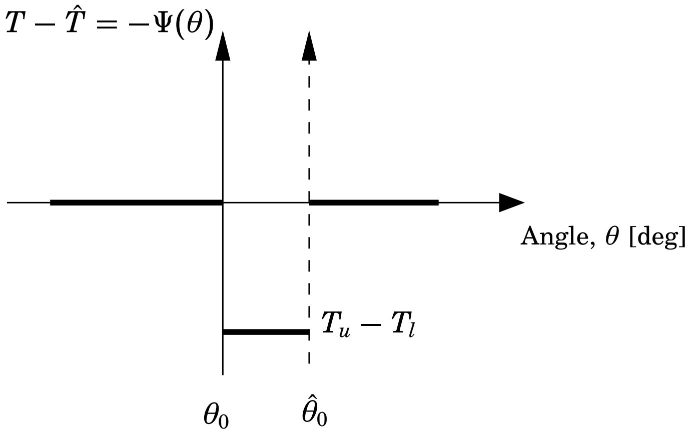

Figure 9 The artificial nonlinearity resulting from imperfect compensation of the spring torques, when $\hat{\theta}_0 \neq \theta_0$.

Figure 10 shows how the closed-loop system is transformed to fit describing function analysis of this artificial nonlinearity. Note that the theory of describing function analysis assumes that the nonlinearity is fed back with a minus sign. Therefore, the following analysis will be performed on the function, $\Psi(\theta) = -\big(T(\theta) - \hat{T}(\theta)\big)$.

Assignment 2: *Assume that the nonlinearity, $\Psi(\theta(\varphi))$, is fed with a sinusoidal input $\theta = \theta_0 + A \cdot \sin\varphi$. Compute the output of the nonlinearity as a function of $\varphi \in [0, 2\pi]$. Separate the computation into two cases, $A < |\hat{\theta}_0 - \theta_0|$ and $A > |\hat{\theta}_0 - \theta_0|$.*

Hint: $\hat{\theta}_0 > \theta_0$ and $\hat{\theta}_0 < \theta_0$ will lead to different outputs of the nonlinearity. It might be useful to define the angle $\varphi_F = \arcsin \frac{|\hat{\theta}_0 - \theta_0|}{A}$.

Assignment 3: *Calculate the describing function*

$$N(A) = \frac{b_1 + i \cdot a_1}{A} \tag{2}$$

for the nonlinearity Ψ, where

$$\begin{aligned} a_1 &= \frac{1}{\pi}\int_0^{2\pi} \Psi(\theta(\varphi)) \cdot \cos\,\varphi \; d\varphi \\ b_1 &= \frac{1}{\pi}\int_0^{2\pi} \Psi(\theta(\varphi)) \cdot \sin\,\varphi \; d\varphi \end{aligned} \tag{3}$$

Note: Since the nonlinearity is not odd there will be a constant-term, a_0, in the Fourier series. This will, however, not affect the amplitude and frequency of possible limit cycles.

Assignment 4: *Draw $-1/N(A)$ in the Nyquist plot of the linear system, $\frac{P(s)}{1+P(s)C(s)}$, in Figure 11. What conclusions can be drawn from this plot? When will there be limit cycles? Where should the Nyquist plot of $\frac{P(s)}{1+P(s)C(s)}$ cross the real axis to reduce the amplitude of the oscillations?*

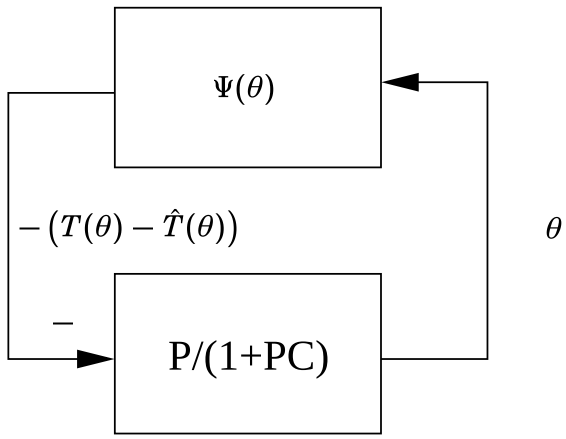

Figure 10 Transformation of the closed-loop system to fit the describing function analysis. Note that in the theory of describing function analysis, the nonlinearity is fed back with a minus sign. We will therefore analyze the nonlinearity, $\Psi(\theta) = -\big(T(\theta) - \hat{T}(\theta)\big)$.

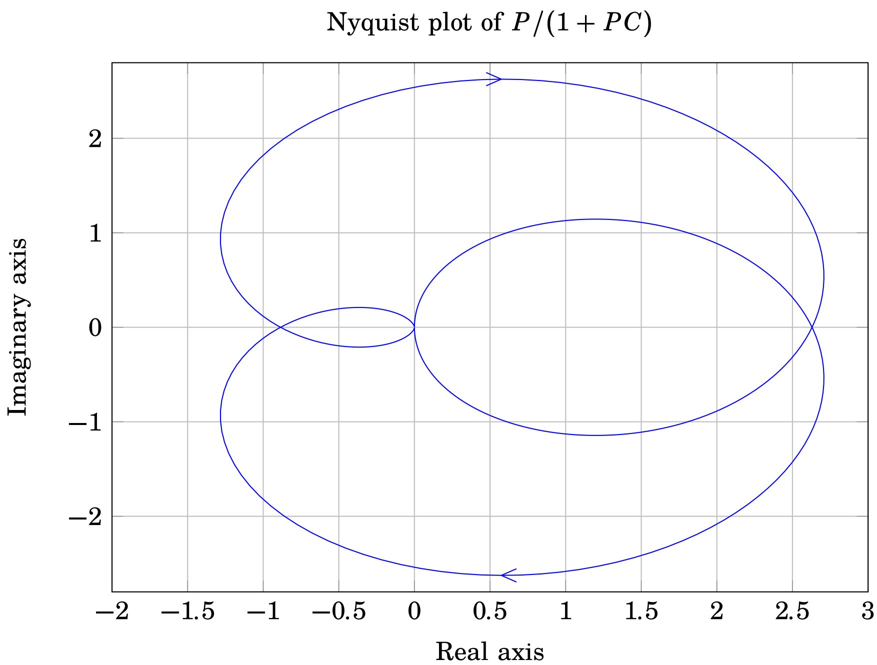

Figure 11 Nyquist curve for the closed-loop transfer function $P(s)/(1+P(s)C(s))$. Fill in $-1/N(A)$.

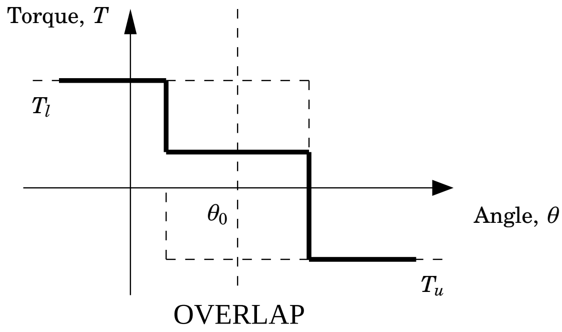

Figure 12 Overlap of the springs results in a modified spring model.

2.5 Overlapping Springs

The two springs not only create a dead-zone but they also introduce another, unwanted, feature. Due to limited manufacturing precision, the springs overlap in a small region around the dead-zone, as shown in Figure 12. This feature complicates the spring compensation.

Assignment 5: *Assume that the simple spring compensation of Figure 7 is used with* $\hat{\theta}_0 = \theta_0$. *Draw* $T(\theta) - \hat{T}(\theta)$ *for the case of overlapping springs. What will the problem be when passing through the dead-zone (sketch a step response)?*

There is no specification of how large the overlap is or even if it exists on every throttle unit. This implies that the feature cannot be included in the dead-zone model. Instead, a fix for the problem is used (investigated in Assignment 10).

3. Simulations

To do the following Matlab exercises you have to download some files as mentioned above. Put the files in the directory where you will run. Start Matlab and type `initlab1sim` at the Matlab command prompt. This initializes the simulation environment and gives access to necessary Matlab variables.

3.1 Spring Torque Estimation

Assignment 6: *Open the model* `lab1sim.mdl` *and copy the throttle block without overlapping springs to a new empty Simulink model. Add step signal sources and perform experiments to estimate the spring torques above and below the dead-zone. Use the results from Assignment 1. The dead-zone position in the simulations is $\theta_0 = 0$. Note that θ has units of degrees.*

3.2 Limit Cycle Oscillations

It is important to know if the control loop of the nonlinear system results in oscillations. Describing function analysis predicts the amplitude and frequency of possible limit cycles. The oscillations mainly depend on how accurate the dead-zone position estimate, $\hat{\theta}_0$, is. In the Matlab workspace the variable for true dead-zone position, θ_0, is denoted `DZPOS` and the dead-zone position estimate, $\hat{\theta}_0$, is denoted `DZPOS_hat`. T_l and T_u are denoted `TL` and `TU` in the Matlab workspace. The estimates used in the compensation, $\hat{T}_l$ and $\hat{T}_u$, are denoted `TL_hat` and `TU_hat`.

Assignment 7: *Open* `dfa.m`. *in a text editor (e.g. the Matlab editor) and fill in the results from Assignment 3 in the file. Try a few different values of the dead-zone position offset, $|\hat{\theta}_0 - \theta_0|$ (by changing the variable* `DZPOS_hat`*), and determine the predicted amplitude and frequency of the oscillations.*

Assignment 8: *Open the model* `lab1sim.mdl`. *Use the throttle model without overlapping springs and use the compensated control signal from the controller block. Try moving the throttle above, below and through the dead zone to generate oscillations. Try different values of* `DZPOS_hat` *and compare the amplitudes and frequencies with the results from the describing function analysis in Assignment 7.*

3.3 Overlapping Springs

Assignment 9: *Change the process in the model to include overlapping springs. Also, make sure that* `DZPOS_hat=DZPOS` *to avoid effects of errors in the compensation. Does the plate move through the dead-zone as predicted in Assignment 5? Is there a difference in behavior between large and small steps? Why/why not?*

A way to reduce the effects of the overlapping springs is to track the angular acceleration, $\ddot{\theta}$, of the plate and modify the compensation algorithm. Notice that the overlap cannot be modeled into the compensator since it is too hard to measure the width. Instead, a fix is applied if the characteristic symptoms occur.

Assignment 10: *Activate the fix in the model and move the throttle through the dead zone. Can you see any difference in behavior? Try different widths of the overlap with and without the fix activated. Can you figure out how the fix works?*

Note: *The workspace variable,* `OVERLAP`*, determines the width of the overlap around the dead-zone position in the simulations, see Figure 12. The interval in which the fix is active can be changed with the variable,* `fixwidth`*. The active interval is then, [*`DZPOS_hat`$\pm$`fixwidth/2`*]*

4. The Real Throttle

Warning*: The motor is very strong, and the throttle plate has sharp edges. Do not put your fingers near the plate when the throttle is running.*

Matlab must be run on the computer physically connected to the throttle. The input signal from the potentiometer should be connected to the AD-converter port 0. The output signal to the motor should be connected to the DA-converter port 0.

The real process is a bit more complex than the model used in the simulations. Since the characteristics of the throttle differ highly below and above the dead-zone, one controller for each area is used. This makes it necessary to implement bumpless transfer between controllers. Since the controllers have integral action, there is also need for an integrator anti-windup feature. The controller structure is shown in Figure 13 and Figure 14.

There are four throttle units and their respective spring compensation parameters and dead-zone position differ slightly. Good values of the parameters are loaded to the Matlab workspace by the function `initthrottle` as mentioned below. The spring compensation variables are called, `TU_hat`, `TL_hat`, and `DZPOS_hat` respectively, and can be changed for testing inaccurate spring compensation affects the performance.

Assignment 11: *To initialize the real-time environment type* `initlab1real` *at the Matlab command prompt followed by* `initthrottle(i)` *where* `i` *is the number of the throttle unit (1-4). Open the model* `lab1real.mdl`*. Try moving the throttle above, below and through the dead-zone. Can you see other nonlinear phenomena affecting the real throttle?*

5. Air Flow Control

So, finally it is time to design the air flow control. The control problem is now to use the flow measurements to calculate a reference to the position controller, to obtain the desired air flow.

We will take a more heuristic approach to this problem, and not use too much time on the modelling. However, from thermodynamics we know that flow and pressures often have nonlinear relationships, especially when exposed to valves. First we try to identify if the system has some static nonlinearity.

Assignment 12: *Use the position control from the previous assignment. Try different reference values for the position, and observe the air flow level in steady state. Is the relation linear? If not, find a function that approximates the relation. Also, find the inverse function, to compensate for the static nonlinearity.*

Hint: The following matlab commands can be used to fit the parameters of the nonlinear function $y(x) = \beta_1 x^2 + \beta_2 x + \beta_3$:

```
% First define the function
func = @(beta,x) beta(1)*x.^2 + beta(2)*x + beta(3);

% Initial guess of parameters
beta0 = [1;1;1];

% Call nlinfit to optimize for the parameters
% X and Y are vectors with data
beta = nlinfit(X,Y,func,beta0)
```

Now that the relation between the position reference end the air flow is linear (at least in steady state), we can close a loop to control the air flow.

Assignment 13: *Use the output from the inverse nonlinearity to design a feedback loop, where a reference to the air flow can be defined, acting on the position reference for the position control. You can design your own controller, or use the PI controller found in* `PIcontroller.mdl`.

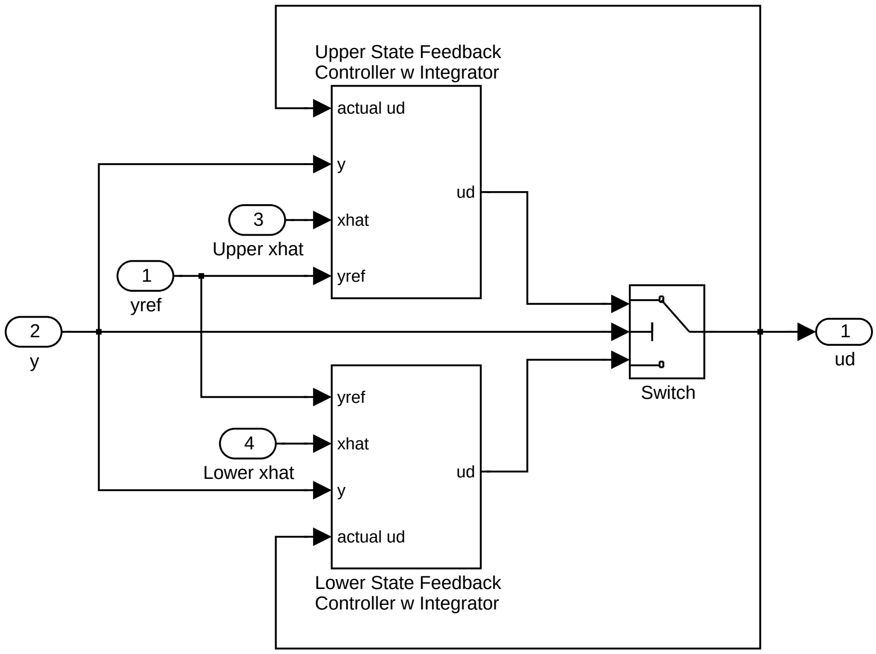

Figure 13 Two linear controllers are needed in the controller structure, one for operation above the dead-zone and the other for operation below the dead-zone. Bumpless transfer is implemented to prevent transients when switching between the controllers.

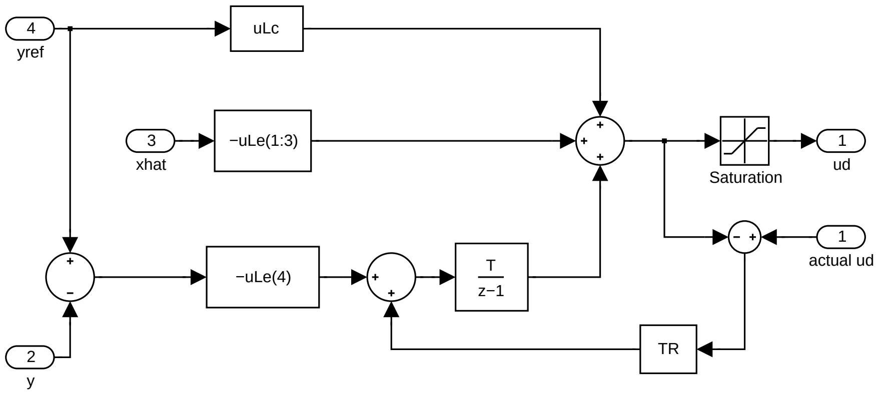

Figure 14 A close up of the upper state feedback controller with integral action in Figure 13. Tracking of the actual control signal implements integrator anti-windup and bumpless transfer.

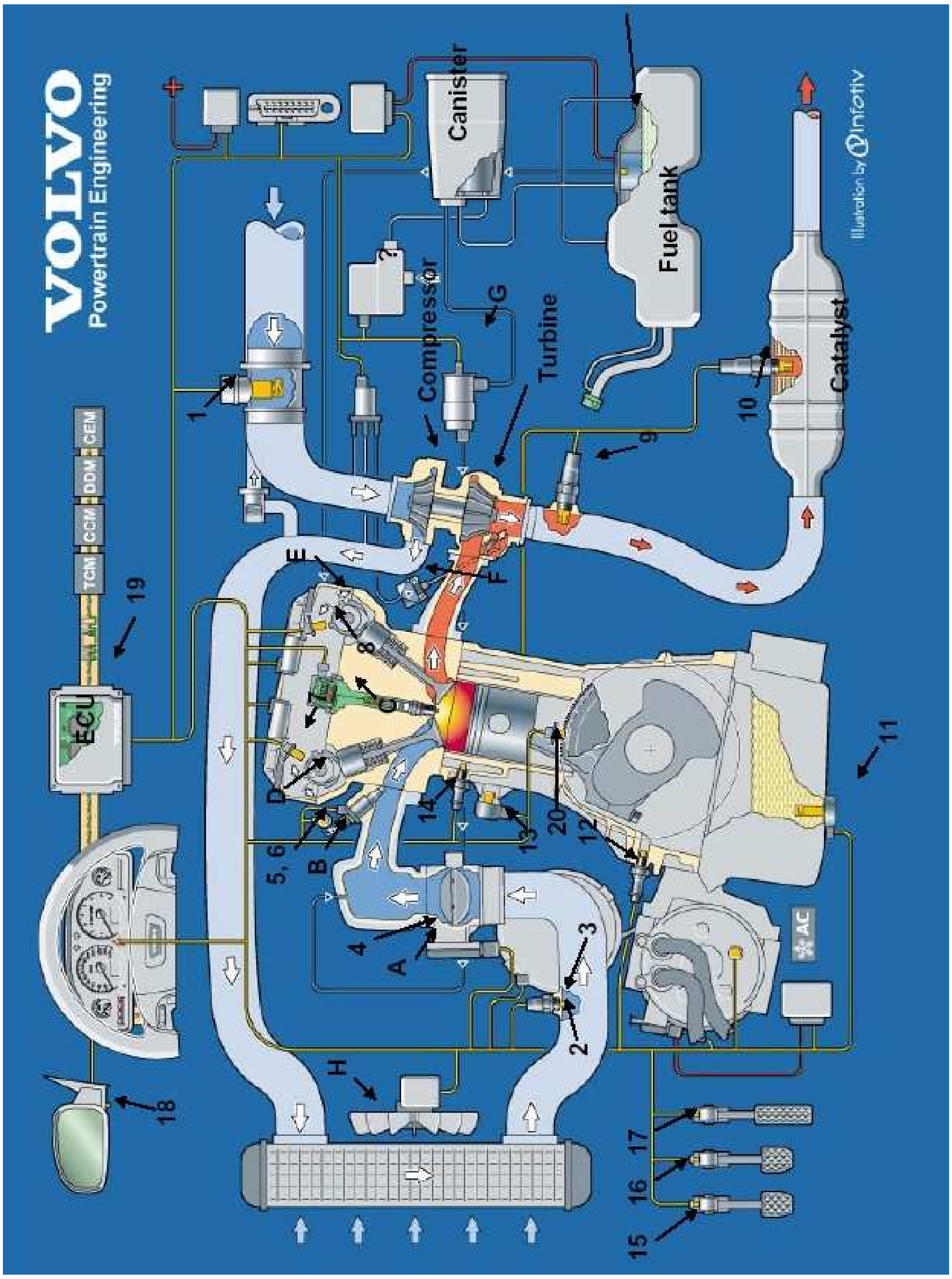

Figure 15 The position of the throttle unit (**4**) in the overall Volvo engine system.