PID CONTROL IN IS500/1000

PID control is a well defined method to create closed loop control for devices like boost control or AFR control etc. If you have a system that you would like to control to a specific value, you can measure the current output (AFR value or boost level) and take suitable action depending on what you, in each and every moment, want the signal to be.
You talk about the **Desired value** and the **Current value**. The difference between the two is the **Current error**.

The PID control loop defines how to adjust the input to the system depending on the current error to minimize the error as fast as possible. The parameters you can adjust is P=Gain, I=Sum of all errors, D=Sudden errors

In the examples below we try to control a PWM output which is connected to a boost control valve in a turbo charged application. At the same time the current turbo charge pressure is measured as input to the closed loop control with a MAP-sensor.

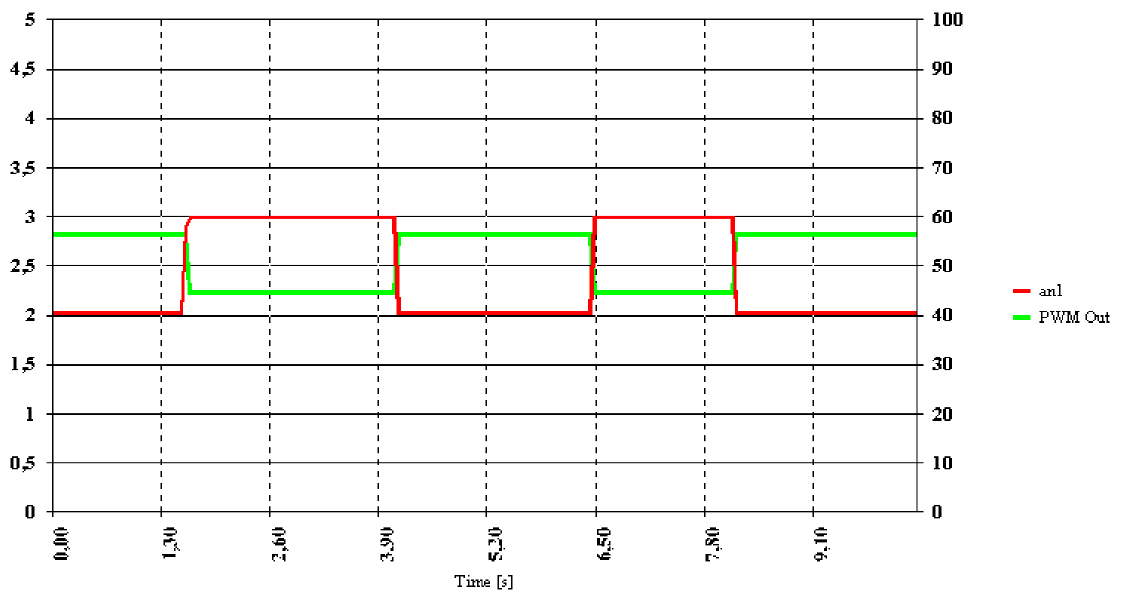
P

The **P-factor** adjusts the proportional part of the closed loop control. I.e if the output is too low, the PID control will increase the PWM-output with P*err and vice versa.

An error of 1 Volt, changes the PWM-signal with 1.25%

Example:
The boost is 0.2Volt too low (which is 0.1 bar with MPX-MAP sensor), the P-factor will increased the PWM signal with 0.25% if P=1. If the P-factor = 10 the PWM-signal will change with 2.5%

Example log



Desired boost is set to 2.5V.
PWM-Init is 100 %.
P-factor is 10
The measured boost is connected to Analog 1.

In the example the boost is sometimes 0.5V too low, and sometimes 0.5V too high.
When the boost is too high, the PWM signal is adjusted to 44%. (Increased by 10*0.5*1.25%=6%)
When the boost is too low, the PWM signal is adjusted to 56%.

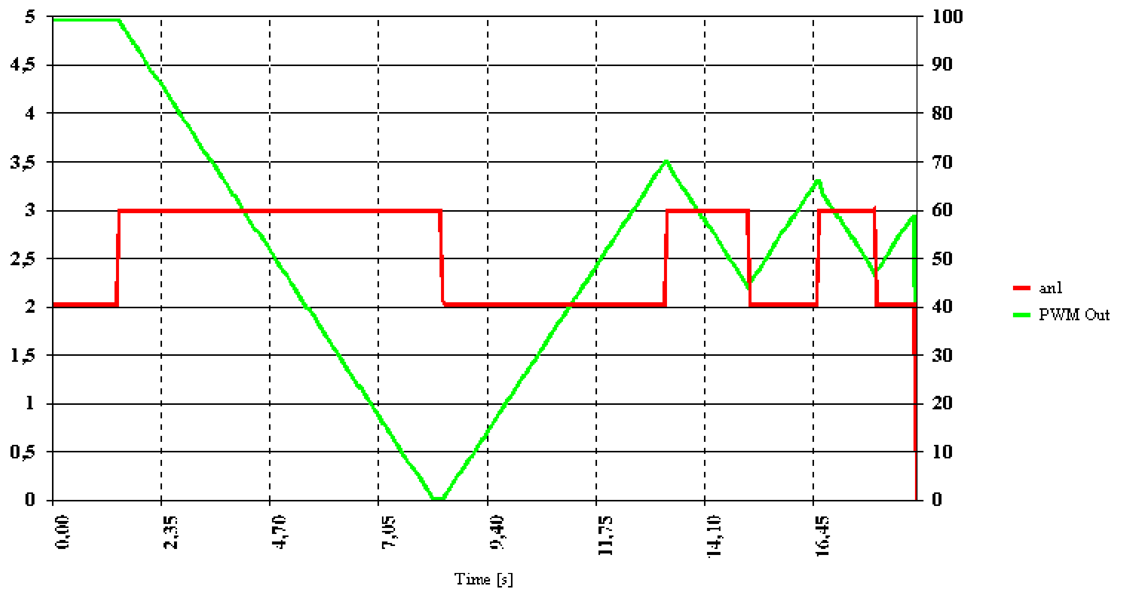
I

The I-factor compensates for long term errors, by summarize all the errors in the history and successive adjust the PWM-signal as long as the error continues to occur.

A long term error of 1 Volt, makes the PWM-signal to change with 3% per second.

Example:
If you have a long term error of 0.2 Volt, the PWM output will be adjusted with 0.6% per second.

Example log



Desired boost is set to 2.5V.
PWM-Init is 100%.
I-factor is 10
The measured boost is connected to Analog 1.

In the example the boost is sometimes 0.5V too low, and sometimes 0.5V too high.
When the boost is too high, the PWM signal is decreased with 15% per second. (Increased by 10*0.5*3%=15%)
When the boost is too low, the PWM signal is increased with 15% per second..

D

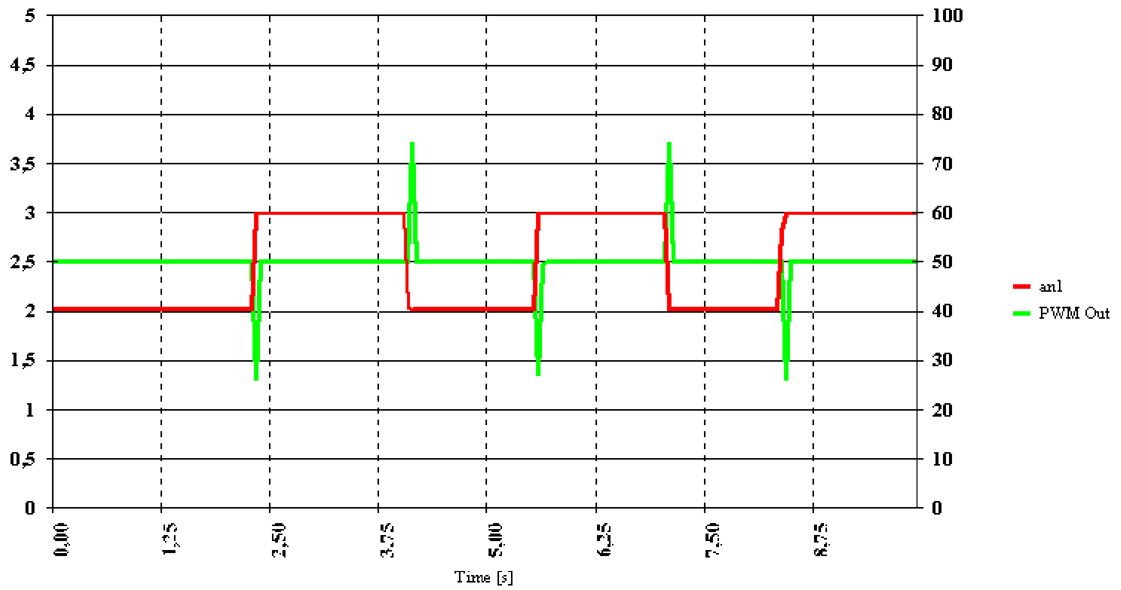
The D-factor compensates for rapid changes of the in-signal, by shortly change (or “brake”) the PWM-signal.

The boost must be within the range of maximum error of 0.63V below the desired value, to “brake” the sudden boost increase. The “brake” of sudden boost lost is always enabled

A sudden (<0.05sec) change of 1 Volt, for the boost signal gives a PWM change of 2.5%

Example:
If you suddenly have an error of the boost level of 0.2 V, The D-factor shortly compensates the PWM-output with 0.5%

Example log



Desired boost is set to 2.5V.
PWM-Init is 100%.
D-factor is 10
The measured boost is connected to Analog 1.

In the example the boost suddenly changes 1V up, and sometimes 1V down.
When the boost suddenly goes high, the PWM signal is decreased with 25%. (Decreased by 10*1*2.5%=25%)
When the boost suddenly goes low, the PWM signal is increased with 25%.

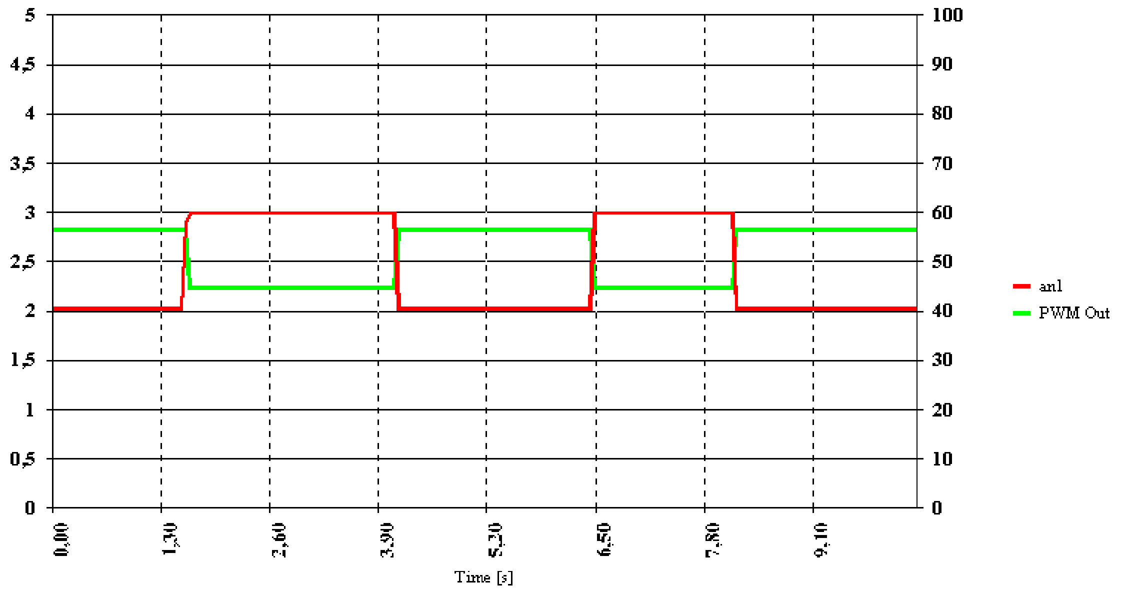
PWM-INIT

To make the PID control to work, you need a default value of the PWM-output signal where the closed control loop starts from. PWM-init factor sets the start value of the PWM-out, if maximal desired boost is selected.

PWM-Init 100% makes that the PWM-signal will use the same start value as the desired boost level in%.

Example:
If PWM-init is 100% and the desired boost level is 5 Volt, the default PWM-value will be 100%.
If the desired value changes to 4 Volt, the default PWM-value will be 80%.
I.e you select which PWM-signal the system should start with to reach the desired boost (after a few milliseconds the I-factor will correct it anyway if it was a bad start value)

Example log

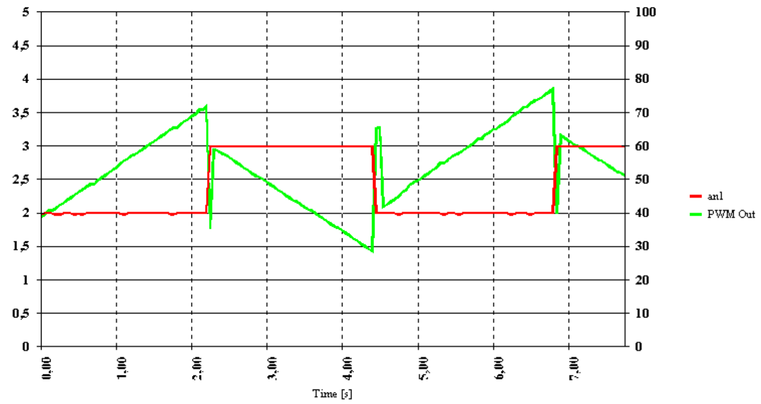


Desired boost is set to 2.5V.
PWM-Init is 100 %.
P-factor is 10
The measured boost is connected to Analog 1.

In the example the boost is sometimes 0.5V too low, and sometimes 0.5V too high.
The default PWM-signal will be 50% (100%*2,5/5=50%)
When the boost is too high, the PWM signal is adjusted to 44%. (Increased by 10*0.5*1.25%=6% from the PWM-init selected)
When the boost is too low, the PWM signal is adjusted to 56%.

PID

Example when all the PID-parameters are active and helps to get the desired boost level

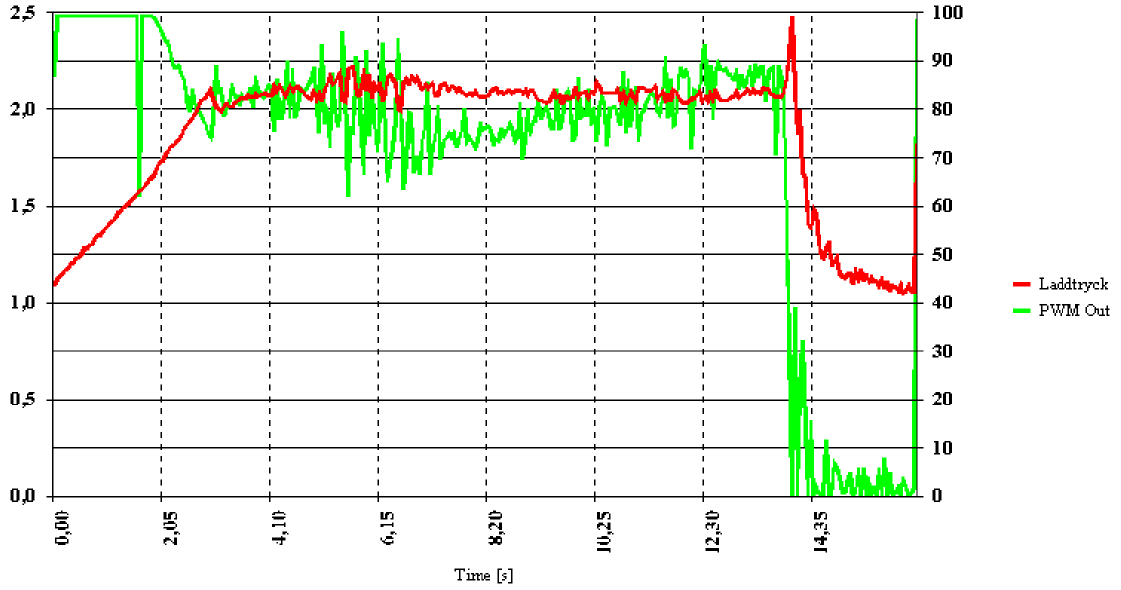


Desired boost is set to 2.5V.
PWM-Init is 50%.
P=10, I=10, D=10
The measured boost is connected to Analog 1.

In the example the boost suddenly changes 1V up, and sometimes 1V down.
When the boost suddenly goes high, the PWM signal is decreased with 25%. (Decreased by 10*1*2.5%=25%)
When the boost suddenly goes low, the PWM signal is increased with 25%.

In the example the boost is sometimes 0.5V too low, and sometimes 0.5V too high, compared to the desired boost level 2.5V.
When the boost suddenly goes too high, the PWM signal is decreased with 25% by the D-factor. At the same time the P-factor gives a general decrease of the PWM-signal because the value is too high.
The error stays at 0.5V too high, so the I-factor continue to lower the PWM-signal, because the error seems to stay.

Real life example



Red signal is boost level (bar)
Green signal is PWM-output

This is a real example from a turbo charged car, where the PID tries to control the boost to 2.1 bar after the throttle is pressed to the floor. The first 2 seconds, the PWM-out is 100% to try to get the boost up. When the boost starts to reach the 2.1 bar the PWM-signal is starting to get down, to avoid "shooting over". If the boost then gets too high, the PWM signal goes down, and if the signal goes too low it gets up again.
It is as simple as that…

How to adjust the parameters for a specific car

How to find the right P-I-D parameters can only be done by testing. Basically you like to have as high value as possible without having an unstable system which is oscillating, because the system is all the time over compensating for the current error.

A good procedure is:

1. Select
   a. PWM-init=100%
   b. P=0
   c. I=0
   d. D=0

2. Increase the **P-factor**, and make log files and observe if the **current boost** reaches the **desired boost** fast but without shooting over (overcompensating). With just the P-factor you will never reach exactly the desired value, but you should be stable just a little bit away. (You will always have a resting error)

3. To get rid of the resting error, you need to use the **I-factor**. Increase this as much as possible without having oscillation. You can decrease the P-factor a little, to be able to increase the I-factor even more.

4. To end with, start increasing the **D-factor** this as much as possible without having oscillation.