”Follow me” controller

Course: Grundlæggende Indlejrede Systemer (GiS)
Modul 3, Regulering af dynamiske systemer
Group: Group N
Date:
Names: Anders Hvidgaard Poder
Henrik Arentoft Dammand

1. Introduction

Create a regulator that allows a car to follow another car at a predefined distance.

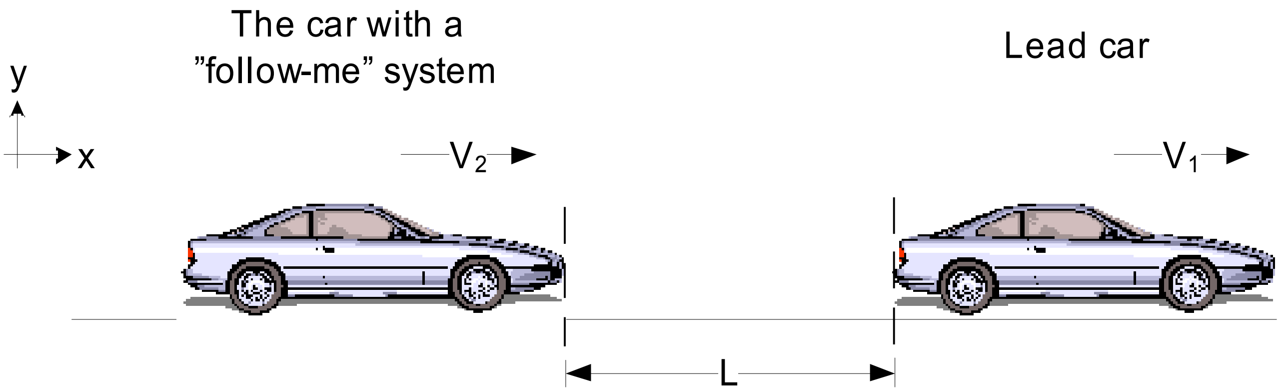


Figure 1, Overall view of the “follow me” system

The “follow me” system illustrated in Figure 1 consists of two vehicles. The front car leads the way while the rear car automatically follows the lead car at a fixed distance. The “follow me” system continuously measure the distance to the lead car and automatically make adjustments to uphold the fixed distance. The “follow me” system depicted in Figure 1 can be model as shown in Figure 2.

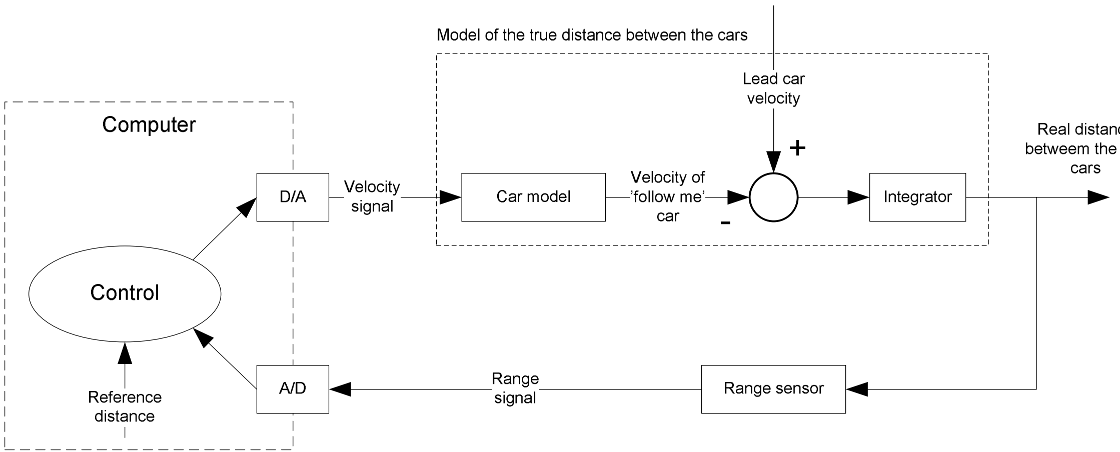


Figure 2, the overall regulator.

The computer block contains the control system can only make adjustment if it has a reference distance to compare with the measured distance and the difference is used to adjust the velocity. The A/D converter and D/A converter in Figure 2 are seen as ideal with no delay and a perfect zero-order-hold and reconstruction filter (which is not practically possible). The same is true for the sensor which instantly returns the exact same value as the actual value. The "follow me" controller is structured so it follows the principles of a basic feedback control system. Thereby, we can analyze and optimize the feedback control system if we know each blocks transfer function.

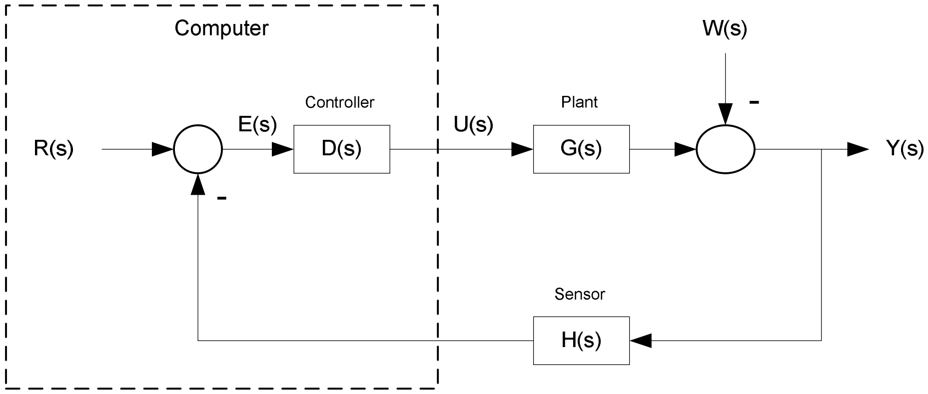
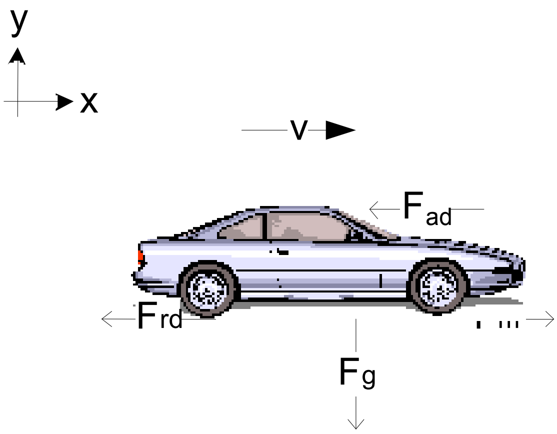


Figure 3, Basic feedback control system

Each block consists of a "Laplace" transformed transfer function, which is a mathematical model of the "follow me" car, the relationship between the lead car and "follow me" car, sensor and controller.

2. Model



- Frd: Force asserted on the car from rolling drag
- Fad: Force asserted on the car from wind resistance
- Fm: Force asserted on the car from the motor via the wheels
- Fg: Gravitational force
- v: Velocity of vehicle due to the sum of all forces not being 0

Figure 4, free body diagram of the "follow me" vehicle

In the free body diagram above is only shown the overall forces affecting the system. Many, many has been ignored because they are negligible. A complete list will not be supplied, as it is almost always possible to come up with some that has not been considered. To give a few examples may be mentioned

- The capacitance generated between the windings in the engine – considered to have a negligible effect compared to the inductance and resistance.
- The initial friction resistance, which is non-linear and lager than the friction resistance lim car velocity -> 0.
- The temperature dependency of the tires spring effect.
- The loss in the transfer of torque from the engine shaft to the wheels.
- And so on and so on.

And this does not even include the assumptions made in the assignment (ignoring hills and changes in the wind, which would make the wind resistance non-linear).

Finally, since there is no changes in the road (no hills, no bumps) and we ignore the breaking and acceleration effect on the suspension (since the car has no breaks, we can simply assume that it has four-wheel-drive, and then the acceleration and breaking should be evenly distributed), we can ignore the effect of the tires and suspension, making the model much simpler.

Having no bumps, hills or turns (centrifugal force is relative to mass and speed), and considering the tires ideal, the gravitational force, being directly proportional to the other forces, irrelevant. This means that Fg can be ignored.

The contribution from the rolling drag (F_{rd}) can be calculated by

$$F_{rd} = C_{rr}F_N = C_{rr}mg\cos\theta \qquad \text{eq. 1}$$

C_{rr} is the dimensionless rolling resistance coefficient. On a smooth road and low-resistance car tires the C_{rr} coefficient is low. In order to further simplify the model we assess the contribution from rolling drag to be negligible; hence we can eliminate F_{rd}.

This leave air drag F_{ad}, which acts as an opposite force to the vehicle’s moving direction and is proportional to the velocity of the vehicle. For a matter of simplicity we make the air drag linear proportional to the velocity.

$$F_{ad} = bv \qquad \text{eq. 2}$$

Where b is the air drag constant. The air drag equation is appropriate at relative slow speed (i.e. low Reynolds number, $R_e < 1$). However, we assume the air drag equation is valid for all speeds.

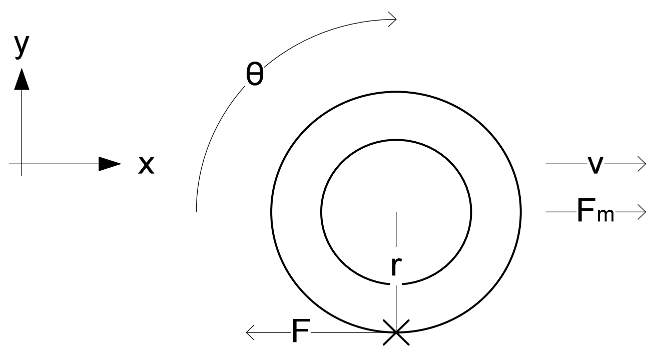
Using Newton’s second law that states that the net force applied to the vehicle produces a proportional acceleration we have:

$$F = ma \qquad \text{eq. 3}$$

Where F is the total force asserted on the mass (with sign and direction) so the equations can be collected to:

$$ma = F_m - F_{ad} \Leftrightarrow m\frac{\partial v}{\partial t} = F_m - bv \qquad \text{eq. 4}$$

The force F_m is the net force from the vehicle’s wheels



From the free body picture of a wheel we can establish an equation that gives the net torque applied to the wheel. The force F is equal with F_m with opposite sign i.e. $F_m = -F$.

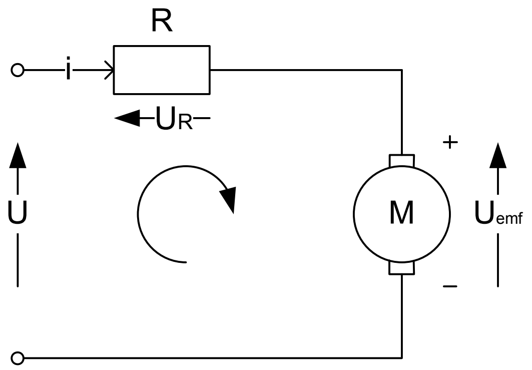
$$-Fr + J\frac{\partial \omega}{\partial t} = \tau_m \Leftrightarrow F_m r + J\frac{\partial \omega}{\partial t} = \tau_m \Leftrightarrow F_m r + J\frac{1}{r}\frac{\partial v}{\partial t} = \tau_m$$

$$F_m = -J\frac{1}{r^2}\frac{\partial v}{\partial t} + \frac{1}{r}\tau_m \qquad \text{eq. 5}$$

We combine eq. 4 with eq. 5 and we have:

$$m\frac{\partial v}{\partial t} = -J\frac{1}{r^2}\frac{\partial v}{\partial t} + \frac{1}{r}\tau_m - bv \qquad \text{eq. 6}$$

The torque t_m in eq. 6 is the contribution from the vehicle's DC motor. A simplified model of a DC motor is shown below



The DC motor's torque τ_m is a linear proportional to the current i and k_t is motor torque constant and we have:

$$\tau_m = k_t i \qquad \text{eq. 7}$$

We can insert eq. 7 into the angular momentum equation (eq. 6) and we have:

$$J\frac{\partial \omega}{\partial t} = J\frac{1}{r}\frac{\partial v}{\partial t} = k_t i \qquad \text{eq. 8}$$

The DC motor's angular velocity is defined as a linear voltage difference u_{emf}, proportional to the angular velocity ω of the armature shaft and k_e and we have:

$$u_{emf} = k_e\frac{\partial \theta}{\partial t} = k_e\omega \qquad \text{eq. 9}$$

By using Kirchoff's laws the following equation can be derived

$$-u + Ri + u_{emf} = 0$$

$$-u + RJ\frac{1}{k_t r}\frac{\partial v}{\partial t} = -k_e \omega$$

$$-u + RJ\frac{1}{k_t r}\frac{\partial v}{\partial t} = -k_e\frac{1}{r}v$$

$$RJ\frac{1}{k_t r}\frac{\partial v}{\partial t} = u - k_e\frac{1}{r}v$$

$$J\frac{1}{r}\frac{\partial v}{\partial t} = \frac{k_t}{R}u - \frac{k_e k_t}{Rr}v \qquad \text{eq. 10}$$

Hence, we can in eq. 10 substitute $J\frac{1}{r}\frac{\partial v}{\partial t}$ with torque τ and we have:

$$\tau = \frac{k_t}{R}u - k_e\frac{k_e k_t}{Rr}v \qquad \text{eq. 11}$$

In eq. 11 the torque from the DC motor has been found and can be inserted into eq. 8 and we have:

$$m\frac{\partial v}{\partial t} = -J\frac{1}{r^2}\frac{\partial v}{\partial t} + \frac{1}{r}\left(\frac{k_t}{R}u - k_e\frac{k_e k_t}{Rr}v\right) - bv$$

$$m\frac{\partial v}{\partial t} = -J\frac{1}{r^2}\frac{\partial v}{\partial t} + \frac{k_t}{Rr}u - \frac{k_e k_t}{Rr^2}v - bv$$

$$\left(m + J\frac{1}{r^2}\right)\frac{\partial v}{\partial t} = \frac{k_t}{Rr}u - \left(\frac{k_e k_t}{Rr^2} + b\right)v \qquad \text{eq. 12}$$

Equation 12 is reduced to interacting forces as a change of velocity over time and voltage. For simplicity we replace all the constants in reduced form:

$$b_1 = m + J\frac{1}{r^2}$$

$$a_1 = \frac{k_e k_t}{Rr^2} + b$$

$$a_2 = \frac{k_t}{Rr}$$

The parameters are explained below:

m Mass of the vehicle

J	The net angular inertia of the vehicle
r	Radius of the wheel
R	Electrical resistance in the motor
k_e	Angular velocity constant
k_t	Torque constant
b	Wind resistance coefficient

We can rewrite equation 12 in reduced form and we have:

$$b_1 \frac{\partial v}{\partial t} = a_2 u - a_1 v \qquad \text{eq. 13}$$

Equation 13 in Laplace transformed version we have:

$$b_1 s V_{(s)} = a_2 U_{(s)} - a_1 V_{(s)}$$

$$a_2 U_{(s)} = (a_1 + b_1 s) V_{(s)} \qquad \text{eq. 14}$$

The equation 14 can be rewritten and we find the vehicle's transfer function

$$\frac{V_{(s)}}{U_{(s)}} = \frac{\frac{a_2}{b_1}}{\frac{a_1}{b_1} + s} \qquad \text{eq. 15}$$

We substitute the constants and we have:

$$K_c = \frac{a_2}{b_1} \qquad \text{eq. 16}$$

$$A_c = \frac{a_1}{b_1} \qquad \text{eq. 17}$$

We can reduce equation 15

$$\frac{V_{(s)}}{U_{(s)}} = \frac{K_c}{A_c + s} \qquad \text{eq. 18}$$

3. Simulation

In section 2 we were able to make a very simple mathematic model of "follow me" car and describe the model in S-plan. Before we can use the model we need to realise some constants we can insert into the model. We have tried to pick some real data from a DC-motor and a medium car.

Mass of the car	m = 1250 Kg
Wheel radius	r = 0.325 m
Angular inertia	$J = J_{wheel} + J_{car} = 72kg * (0.325m)^2 / 2 + 1250kg * (0.325m)^2 = 136Kgm^2$

We don’t have a clue about the air drag coefficient b, but we guess the coefficient is relative small so we have made an assumption that the air drag coefficient b is equal to C_w, which is a dimensionless coefficient for relative large velocity.

Air drag coefficient b = 0.36

Now, we need to pick some real constants for the DC-motor and we have found some we think are sensible.

DC-motor resistance	R = 0.118 ohm
Torque coefficient	K_t = 5.9 Nm/A
U_{emf} coefficient	K_e = 5.9 V/RPM

We can calculate K_c and A_c coefficients for the transfer function and we have

$A_c = 1.1$
$K_c = 0.06068$

We can hereby insert the coefficients into the eq. 18 and we have

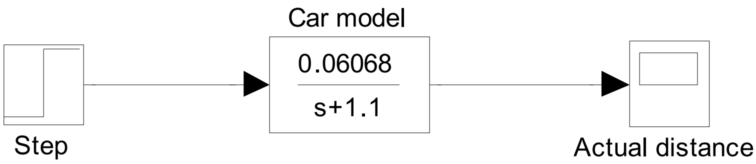


Figure 5, the car model in s-plan with coefficients

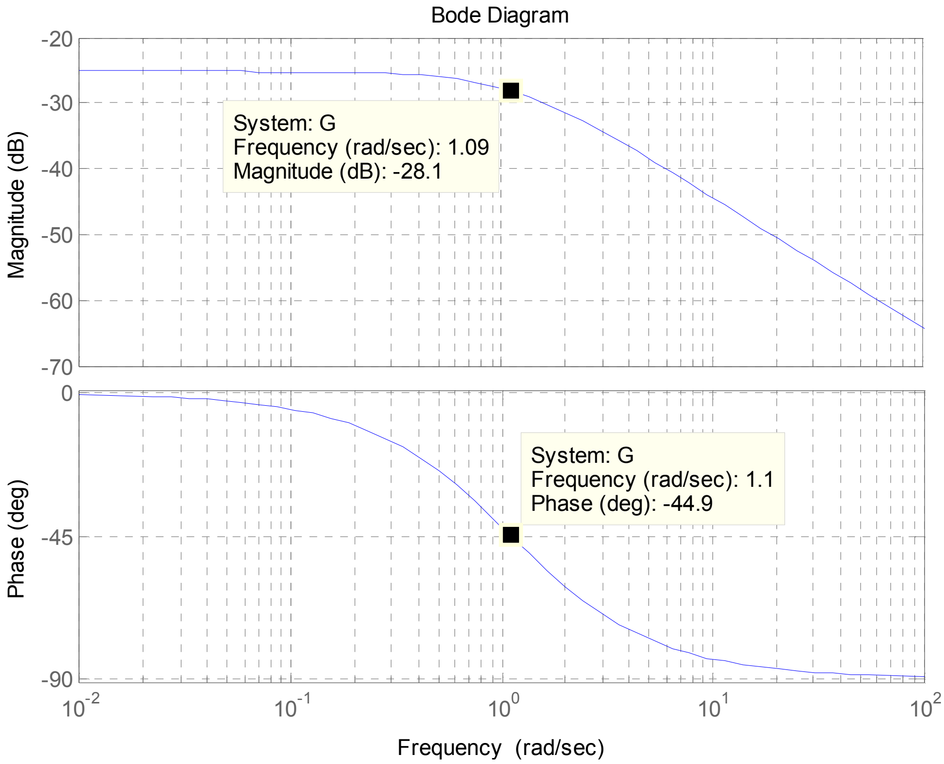


Figure 6, bode-plot of car model

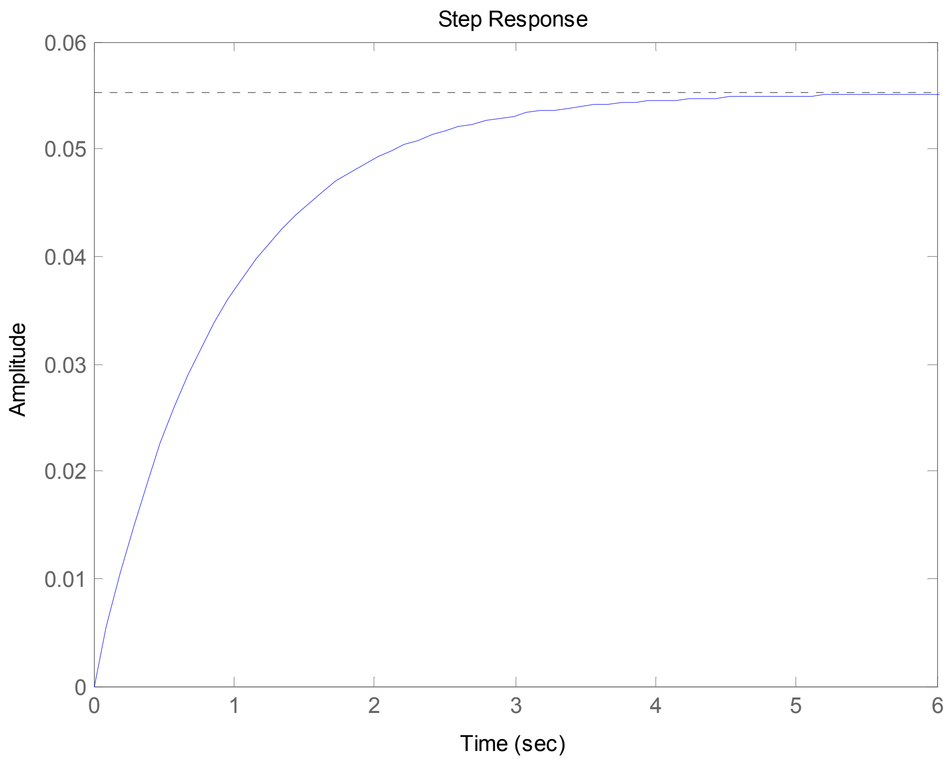


Figure 7, step response of the car model

4. Designing the regulator

We need to choose a regulator that can meet our requirements. Ideally we want to keep the distance perfectly, but that is theoretically impossible, even with perfect acceleration.

Even if we figure that there are no delay between the actual distance of the lead car and the regulator (this can be made small with a good sensor and fast sampling with good accuracy). And even if we choose a first order system with no delay (only dependent on the measured distance, not on the change) and have a perfect theoretical DAC. Then it is not possible to follow the lead car perfectly because the acceleration of the following car cannot be infinite, which would be required. We can however realise that it is a requirement that the maximum acceleration of the lead car must be smaller than the minimum acceleration of the "follow-me" vehicle.

This will allow us to catch up to the lead vehicle no matter how fast it accelerates (both positive and negative (breaking - so we also need better breaks)).

Assuming this is true then our job is to reduce the response time of the regulator as much as possible. Unfortunately this comes at a cost. If we want to quickly react to changes in the other cars velocity (and thereby its distance to us), we must quickly accelerate to catch up, and then, when we have caught up, decelerate, so we can keep the right distance. This is naturally possible, but would not make for very pleasant driving, as we would either be accelerating as much as possible or decelerating as much as possible - if you have ever tried to drive with an inexperienced driver who is afraid of the speeder, or one who mixes up the speeder and break, you know how that feels.

Another matter is one of safety. If we are to follow at 3 meter distance, then it is not good if we overshoot this distance by e.g. 4 meters, as this means we have hit the lead vehicle. Based on this we can set a requirement that we must never have an overshoot of more than the reference distance (1).

To make this more specific an overshoot of 30% (leaving some for inaccuracies in the model, coefficients and sampling) would be a good maximum value.

Requirement 1: Maximum overshoots 30%

Another important aspect is stability. Stability of a first order system can be directly seen on a bode plot, where it is important that we have a phase-margin above 0 (i.e. phase less than -180 degrees). A good rule of thumb is to keep a phase-margin of at least 45%, which also leave some for inaccuracies in the model, coefficients and sampling, and we will use this rule of thumb.

Requirement 2: Minimum phase margin 45 degrees.

As the overshot is closely related to the phase margin it may not be possible to meet both of these requirements or one may infer the other. We will not know this until we begin the design.

Then there is the aspect of keeping the distance to the lead car accurate when the lead car is not making changes to its speed. This is called Steady-state error, and can be directly read from the regulators response to a unit step. A unit step is defined as the value changing from 0 to 1 instantaneously, and then maintaining the 1. Even if the lead car makes multiple changes to its speed, but eventually remains at one speed, it will have the same steady-state error. For some systems it is not required to meet the desired value exactly, but we make this requirement.

Requirement 3: Steady-state error in response to a step = 0.

There is a very simple way of assuring this, and that is to make the system a type 1 system. Other systems are type 0 (a P system - a simple feedback with amplification), which do not have a 0 steady state error, and type 2, which has a steady-state error of 0 for both step and ramp. We have, however, no reason to have a steady-state error of 0 for a ramp, as it does not matter if we lag or lead a little when the lead car accelerate or decelerate, as long as we find the right distance eventually (and it is always better to choose as simple a system as possible).

This can be realised with many different regulators, but keeping to the three main-stream regulators; P, PI and PID, we can see that we cannot use a P-regulator, as it is a type 0 system (it has a steady-state error != 0). We will however, for experiments sake, show how it could be used to implement the regulator, even if it will not be able to meet the stead-state error. Figure 8 shows a block diagram of a P-regulator.

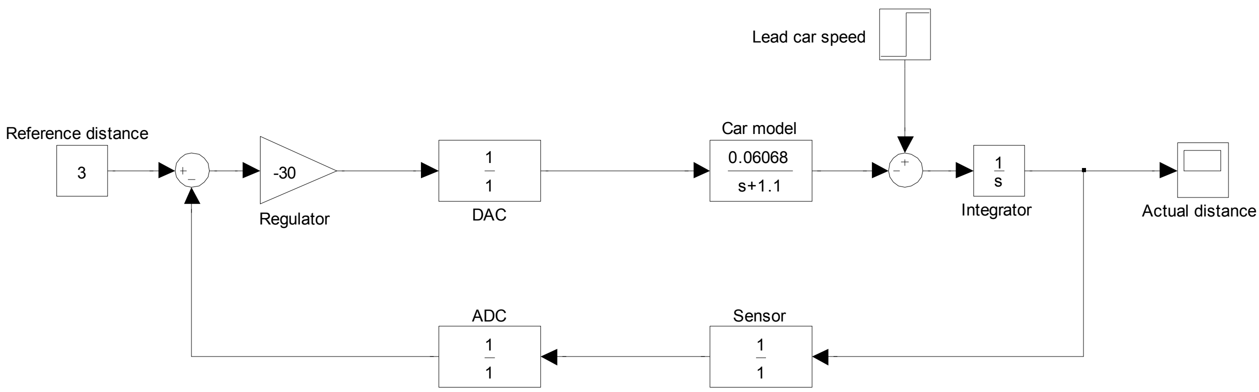


Figure 8, block diagram of a P-regulator

Many different gain parameters were attempted. The first thing we notice is that a negative gain is required for the system to be stable. This is due to the system itself: "If we are too far away we must try and reduce the distance". We then attempted to change the value of the gain, and the higher the gain the faster the system reached its steady-state, and the close it was to the correct value, but the higher overshot it had. At Gain = -1 it took about 80 seconds to reach steady-state and with no overshot, but a final value of approx. 22, or a steady-state error of 22 - 3 = 19 meter or 533%. At a gain of -10 it took only about 10 seconds to reach steady-state and we have an overshot of about 7%, and a steady-state error of 1.75. Since we allowed 30% overshot, we can have a gain of approximately -30, which gives us an overshot of 30% and a settling time of about 7 seconds with a steady state error of 0.6. This is shown in Figure 9.

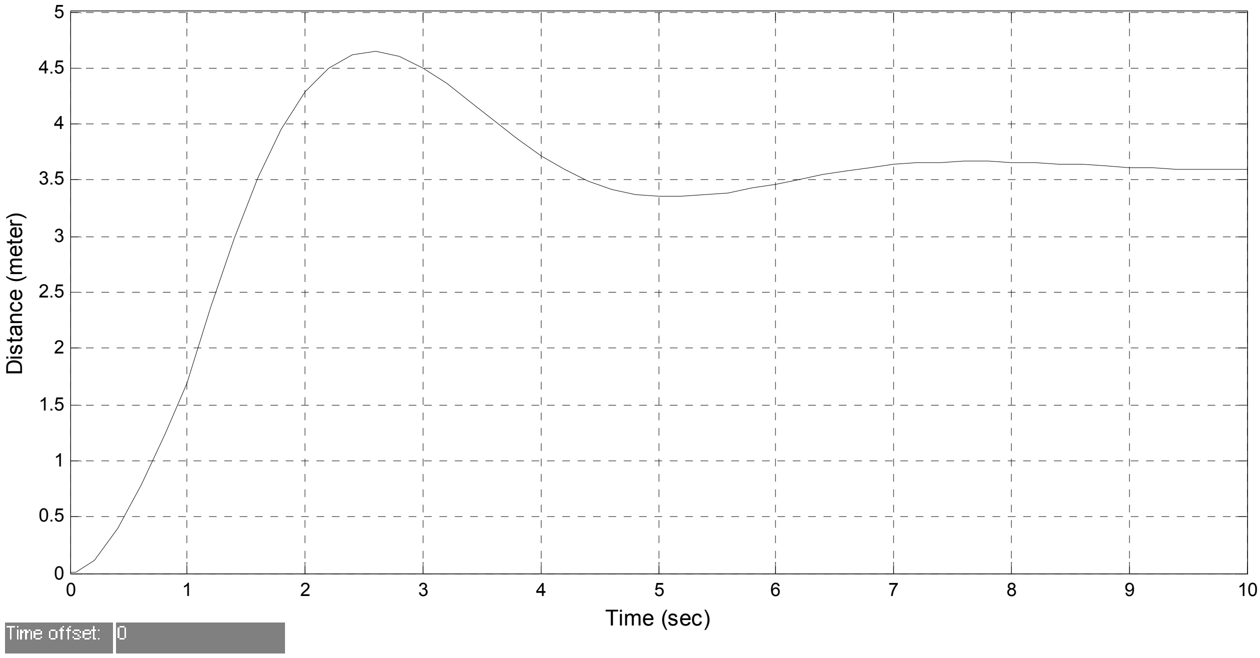


Figure 9, step response for pure gain regulator

In order to get a type 1 regulator we need an integrator in the regulator. This is represented as 1/s in Laplace, and unfortunately has the consequence that the phase-response is moved down 90 degrees, wreaking havoc on our phase margin. To counter-act this effect we can add a zero in the equation, as this will move the phase-response up 90 degrees. Naturally these 90 degrees up or down are not in the entire phase-band, but only for a certain interval. Here an important stability rule comes in; if the highest frequency where the gain is 1 is higher than -180 degrees the system is stable. Even if there are lower frequencies where the phase-margin is negative and the gain is > 1 - this may seem counter-intuitive, but it is true none the less.

In order to choose the coefficients of the regulator, under the constraints above, and with a fourth requirement

Requirement 4: As fast a rise time (i.e. response time) as possible

We have to follow the below steps:

1. Realise that in order to have a steady-state error of zero we require to have a type 1 system, and that means having an integrator in the regulator. See Figure 9 of pure P-regulator for proof.
2. Having a pure integrator in the system will lower the phase-margin to a level where it is not possible to achieve the min 45 degree phase margin.
3. Adding a zero will lift the frequency response by 90 degrees in a "bubble", and as long as this "bubble" has the cross-over-frequency and it is above -135 degrees (required for 45 degrees phase-margin), then the system is stable.

4. We can now choose the highest possible P that will ensure a max. 30% overshot and have the best possible rise-time. This means choosing the P as high as possible, i.e. exactly where the 30% is just met.

First we add the pure integrator (K_i*1/s), as shown in Figure 10.

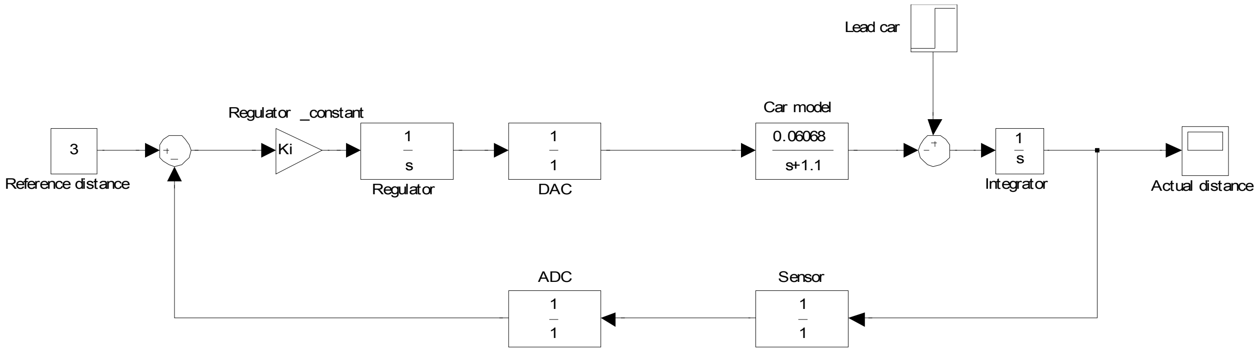


Figure 10, integrator replace the pure gain regulator

Then we create a bode-plot of the open-loop system. This is done by first realizing that K_i must be negative, i.e. larger distance must result in increased power to the engine and not reduced, as is the "raw" functionality of the system. However open-loop transfer functions use a positive proportional gain, so we must use a P = -K_i. Second we must set the reference and disturbances to 0 and then multiply the transfer functions in the forward and feed-back path, giving us:

(V_{For}) (Regulator) (Model) (Integrator) (Feed-back path)

$$\left(0 \quad - \quad \left(-P\frac{1}{s} \quad * \quad \frac{0.06068}{s+1.1}\right)\right) \quad * \quad \frac{1}{s} \quad * \quad 1 \Rightarrow$$

$$P\frac{0.06068}{s^3+1.1s^2} \qquad \text{eq. 19}$$

This can also be seen in Figure 11.

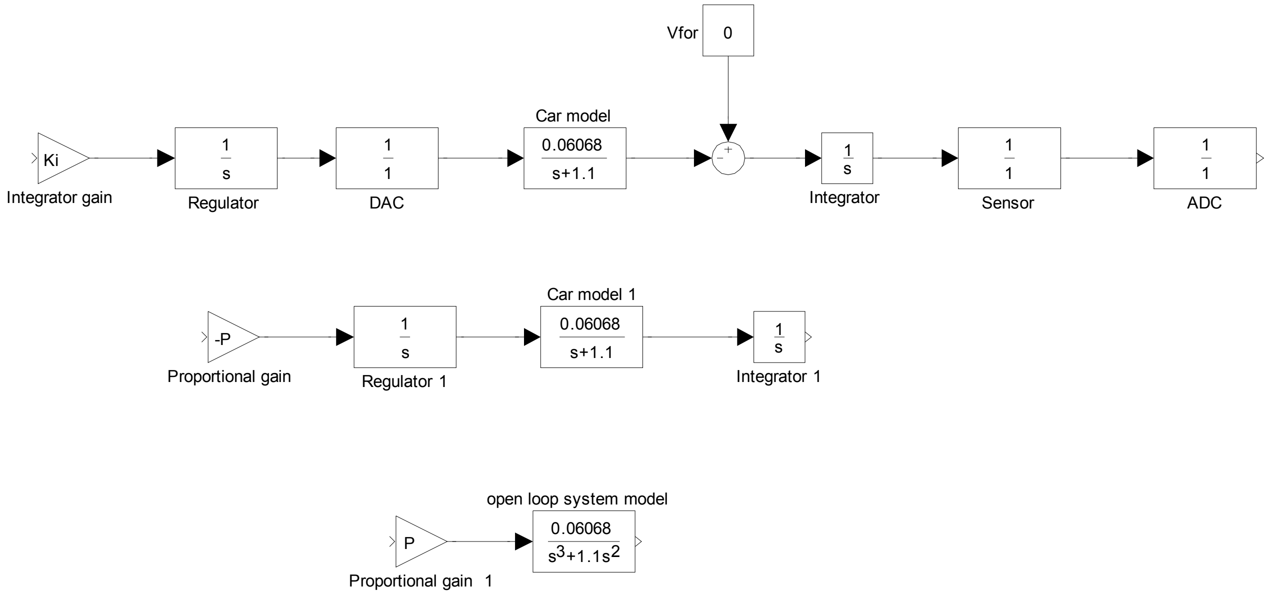


Figure 11, open-loop system of a PI regulator

This is possible as -P can then be written as -1 * P, and superposition allows us to interchange the multiplication term. We realize that designing with integration and stability the value of P is of little importance in the beginning as we focus solely on the phase-characteristic, which is independent of P. We therefore choose a P of 1. This gives the following plot in Figure 12.

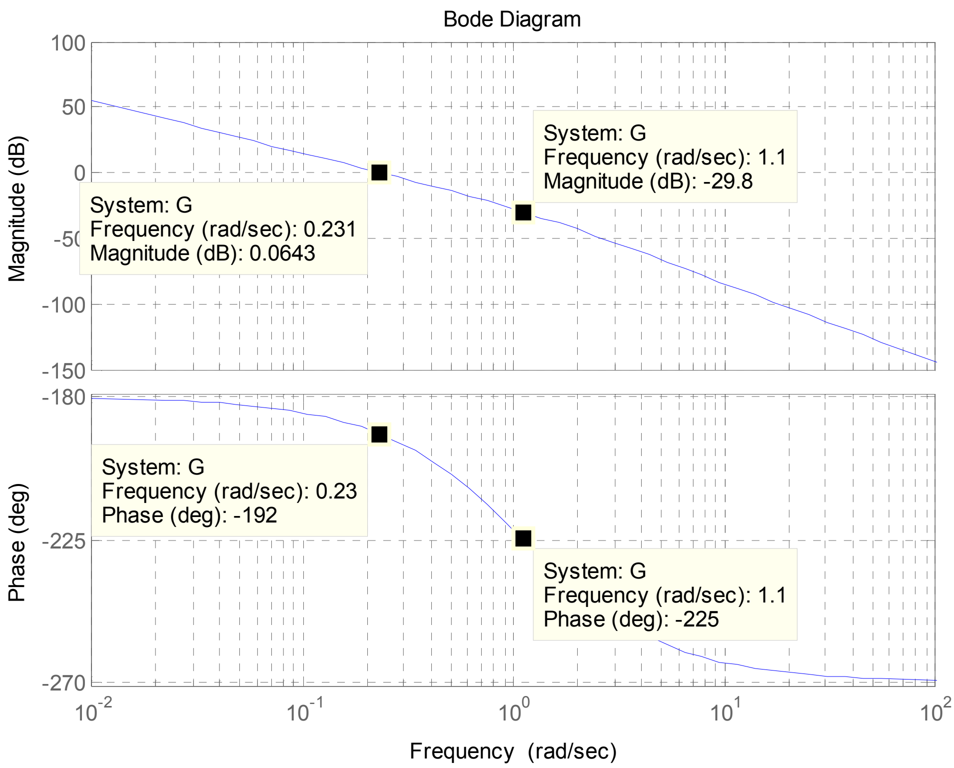


Figure 12, Bode plot of open loop system of a PI regulator

Here it may be seen that the open-loop system as a phase margin of -12, which is clearly unstable. It can further be seen that the system never reach a phase > -180 degrees, so the system will be unstable for any value of P. It can furthermore be seen that increasing the phase margin by 90 degrees is sufficient to make the system stable. Increasing the phase can be achieved by adding a zero.

Here it is important to realize that a zero do not increase the phase by 90 degrees over the whole frequency span, it will decrease with lower frequencies. In fact where we place the zero this decrease will be exactly 50%, or a phase increase of 45 degrees.

With this knowledge we can determine the lowest possible frequency at which the zero may be placed. We know it must have a phase margin of at least 0 degrees in order to be stable (the 45 degree requirement comes later), i.e. a phase of -180 degrees. We also know that at the zero we will achieve a 45 degree phase lift, so we need to find the location where a 45 degree increase will give -180 degrees, or a current phase of -225.

In the bode plot we can determine this to be at 1,1 rad/s. This naturally do not meet our 45 degree phase margin requirement, but finding the highest possible value of a that does meet this requirements is not so easy - actually the calculations have a level of complexity that makes it beneficial to use educated guessing and tweaking. We do however know that it must be less than 1,1 rad/s.

Before we begin determining the highest value of the zero we will just pause a little on the subject of why we want to do that. Since we can control the gain we can naturally create a 45 degree phase margin anywhere in the valid interval that we determine in a minute. Secondly we can choose to look at the cross-over frequency, as we want this to be as high as possible. Unfortunately the cross-over frequencies for different values of the zero is almost identical, so for this reason we choose based on the gain. We want the zero that creates the highest gain, as this will help suppress changes in the system (response time). Changes (noise) are suppressed by a factor of 1 + the open loop gain, and this happens to be the zero with the lowest possible frequency (highest possible value).

Now, to determine this value we have to try different value for the zero. We begin by an educated guess of a zero a factor of 10 below the minimum zero stability-frequency found above of 1,1 rad/s, so we begin in 0,11 rad/s.

Using Matlab's sisotool it is possible to graphically (drag and drop) modify the location of the zero and then see the effects of the frequency response. This makes it easier to determine the "highest possible" value of the zero. We also have to remember that it does not have to be the highest, as the lower values work just fine, they may just not be as good at keeping up, but as good can be a very small difference if we are talking about two zero values that are close together - there is such a thing as a "good enough" zero.

Using this tool we determine the zero to be at 0,19 rad/s, meaning that for any value in the interval]0;0,19] it is possible to have a P that gives us a phase margin of 45 degrees - for the reasons mentioned above we choose 0,19 rad/s as our zero, giving us the open-loop transfer function:

$$P\frac{0.06068s + 0.1153}{s^3 + 1.1s^2} \qquad \text{eq. 20}$$

From this we can determine the P required to achieve a 45 degree phase margin, and this turns out to be P = 8. Putting this into the transfer function we get the bode plot in Figure 13.

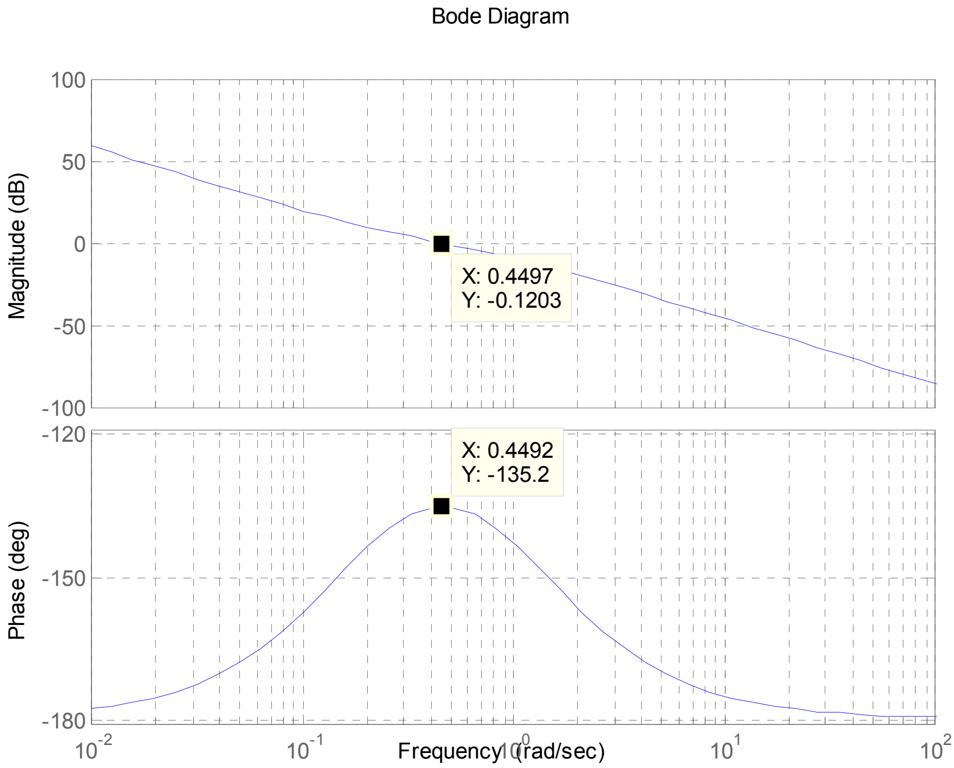


Figure 13, bode plot of a PI-regulator

We place this in the model and attempt to modify the gain to achieve an overshot of max. 30%. We realize that this is not possible, as we do not accelerate fast enough to keep up with the lead car's step speed (infinite acceleration), which is also unrealistic. We therefore only look at the second loop in the figure, i.e. how much the distance swings under the curve. This gives us a gain of -150 (we remember from earlier that the gain has to be negative). The actual overshot will be smaller, as it is not possible for the lead car to accelerate as a step, only as a ramp.

In Figure 14 may be seen the step response of the system, where the 30% "overshot" and the zero steady-state error are apparent.

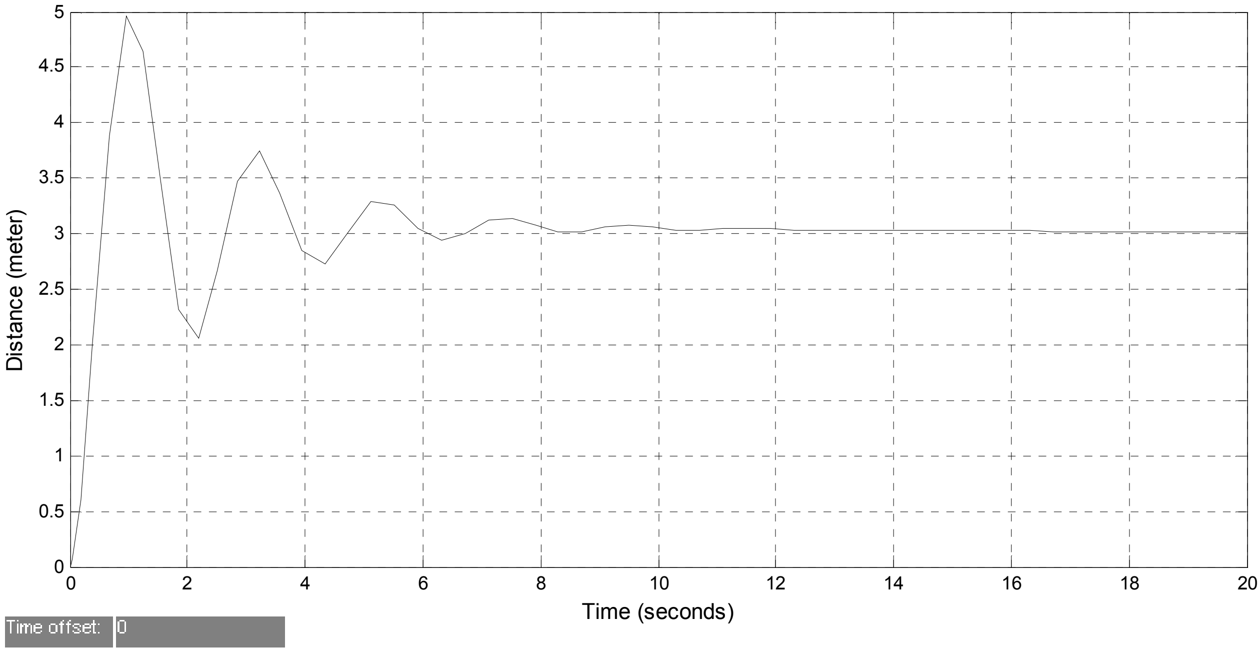


Figure 14, step response of PI-regulator

This regulator actually follows the lead car quite well, but you may be thinking: "I can follow just as well as the regulator".

The problem is that the system does not take the lead car's change in speed into account. If the lead car changes speed fast, we should accelerate fast to catch up, and if it only slowly changes speed, we should also only alter our speed slightly in order not to over-shoot too much.

This may be resolved by adding a D (Differentiator). In order to do this we are going to rewrite the regulator to show the individual parts P, I and D. Furthermore Simulink do not permit a transfer function with more zeros than poles, so we have to give the differentiator a pole somewhere where it will not do any damage - we chose a pole in 100 rad/s. The resulting regulator may be seen in Figure 15.

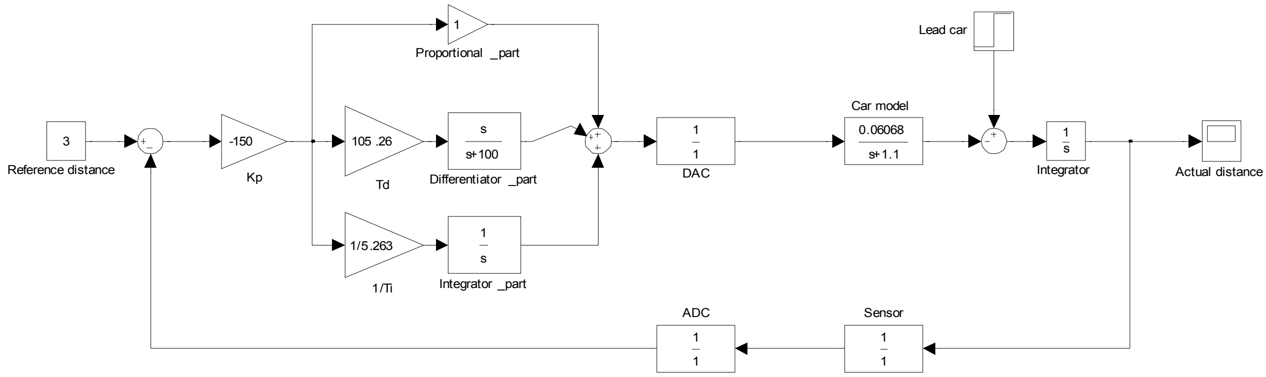


Figure 15, block diagram of the PID regulator system

Here we have already assigned values for K_P, T_d and T_i, so we better devote a little time to how we got these values. A rule of thumb is to ignore the differentiator (set T_d to 0) and tune the regulator, and then Td can be found as 20 times T_i. Luckily we have already tuned the PI regulator, so we can just use the values from there. Here we have to remember that we rewrote the layout of the regulator, so we have to change the format of

-150 * (s+0,19)/s to the form of K_p * (1 + 1/T_i * 1/s). This is done simply by multiplying:

$$\frac{-150s-28{,}5}{s} \Rightarrow \frac{-150s}{s}+\frac{-28.5}{s} \Rightarrow -150+(-28.5)\frac{1}{s} \Rightarrow -150*\left(1+0.19*\frac{1}{s}\right) \Rightarrow$$

$$-150*\left(1+\frac{1}{5.26316}*\frac{1}{s}\right)$$,i.e. K_p = -150 and T_i = 5,263

Using the rule of thumb this makes T_d = 5,263 * 20 = 105,26

When we simulate the unit step we can see a massive improvement in response time and overshot, as can be seen in Figure 16.

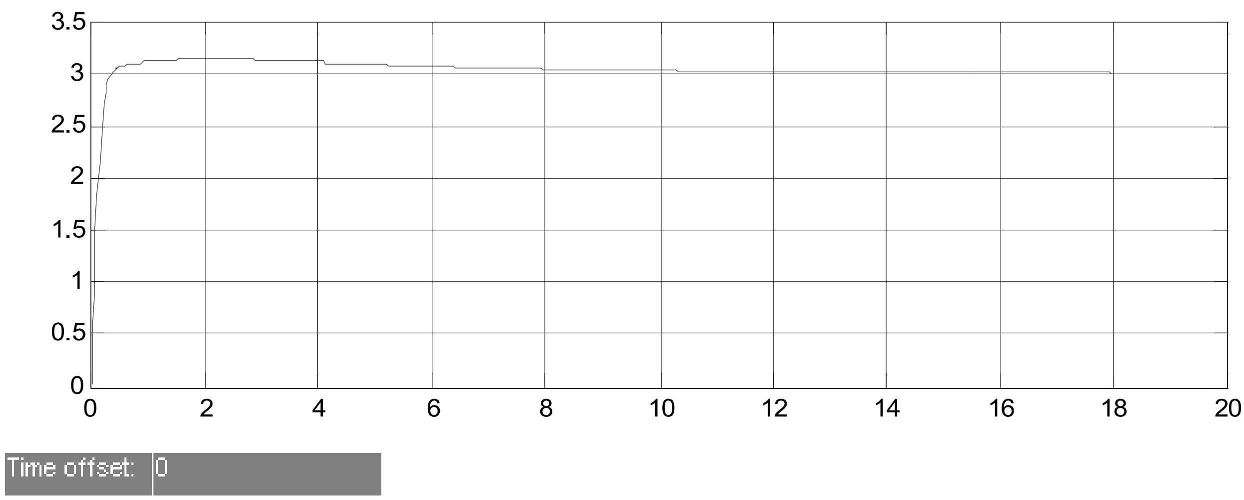


Figure 16, block diagram of the PID regulator system

The rise-time is almost halved and there is no oscillation at all, only an initial overshot way below the 30%. The disadvantage of the D is that it becomes highly sensitive to rapid changes in speed. Naturally a car is not capable of changing speed as a step, or having samples jump many km/h from one to the other. However, this may happen in case of an error.

This problem of error tolerance may be resolved by adding in a low-pass filter. As we know the lead-cars maximum acceleration, we also know the maximum angle of the velocity-ramp that this would generate. Based on this we can determine the frequency content of this ramp, which will give up the highest possible frequency that may realistically, be part of the system. If any frequency above this is found in the sampled distance, then it must be an error. To imagine this think of a step; In order to create a theoretical step you must include all frequencies up at infinity. The more vertical a curve, the higher the frequencies it contains.

Removing these errors is done simply by adding a low-pass filter after the ADC, which will insure that any incorrect values will be removed. This filter is easy to design, but will not be shown here.

5. Testing the regulator

When we simulate our PI regulated system with the lead-car supplied, see figure bb, we get the distance-chart shown in figure mm. If we can see that we generally follow the lead car quite well (once we catch up. The lead car starts at 80km/h and we start at 0, so it is kind of unfair to the regulator).

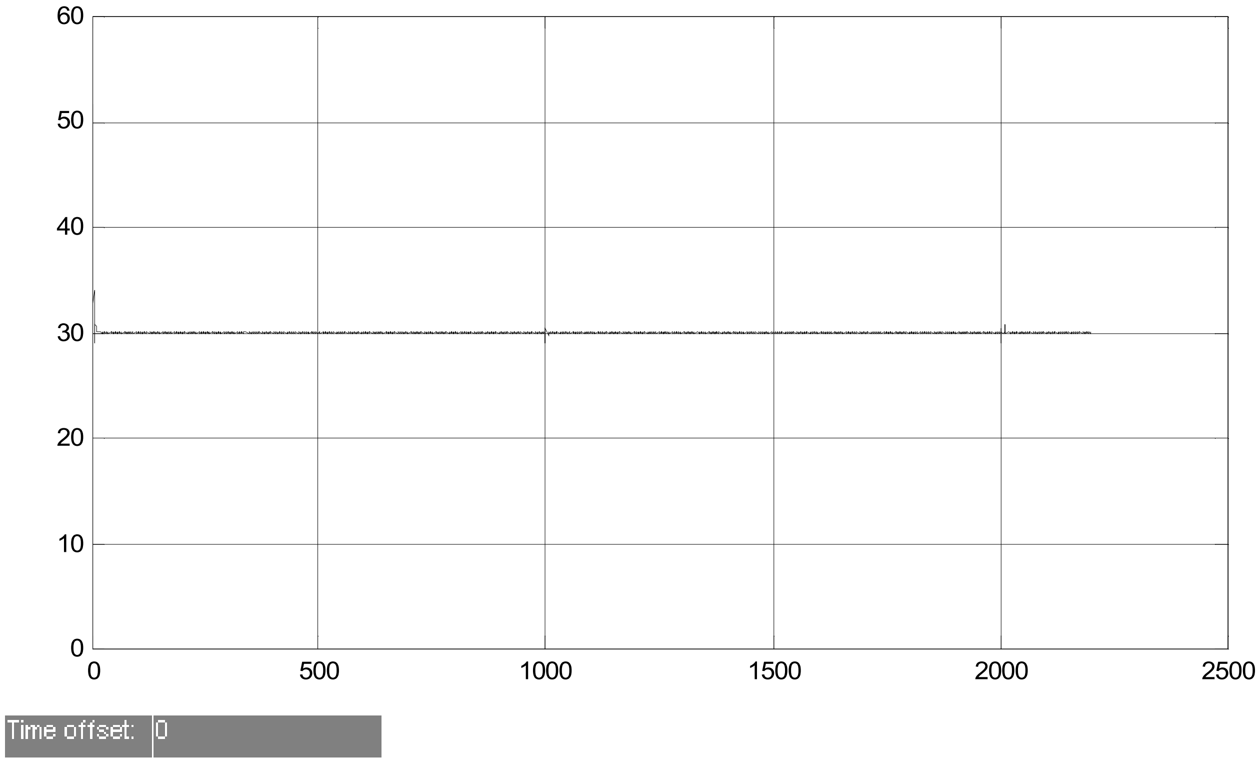


Figure 17, test run of PI regulator on realistic lead car

If we zoom in on the different parts where a change happens:

- 0s (we start at 80 km/h)
- 300s (we accelerate as a ramp)
- 500s (we stop accelerating and maintain a speed of 90km/h)
- 1000s (we instantly change to a speed of 80km/h)
- 2000s (an incorrect reading of 72km/h is supplied)
- 2010s (an incorrect reading of 88km/h is supplied)

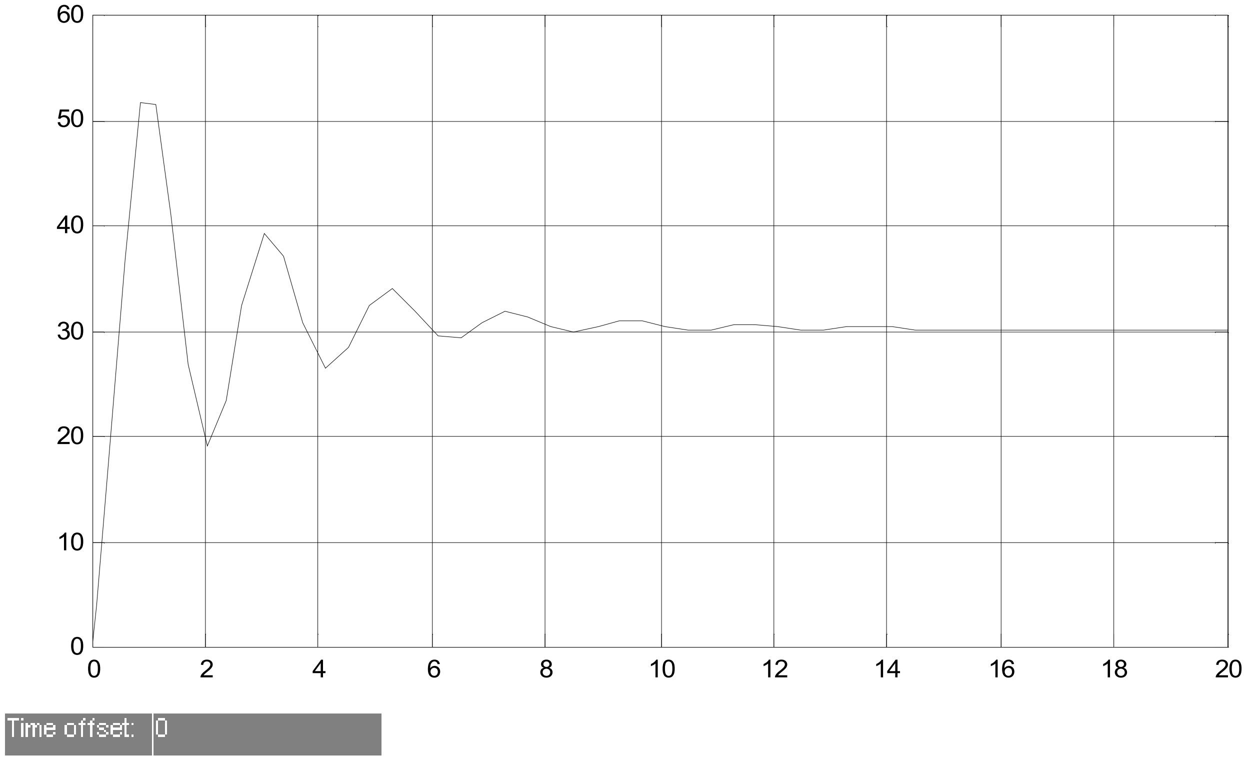


Figure 18, initial test interval: zero to 80 km/h, PI-regulator

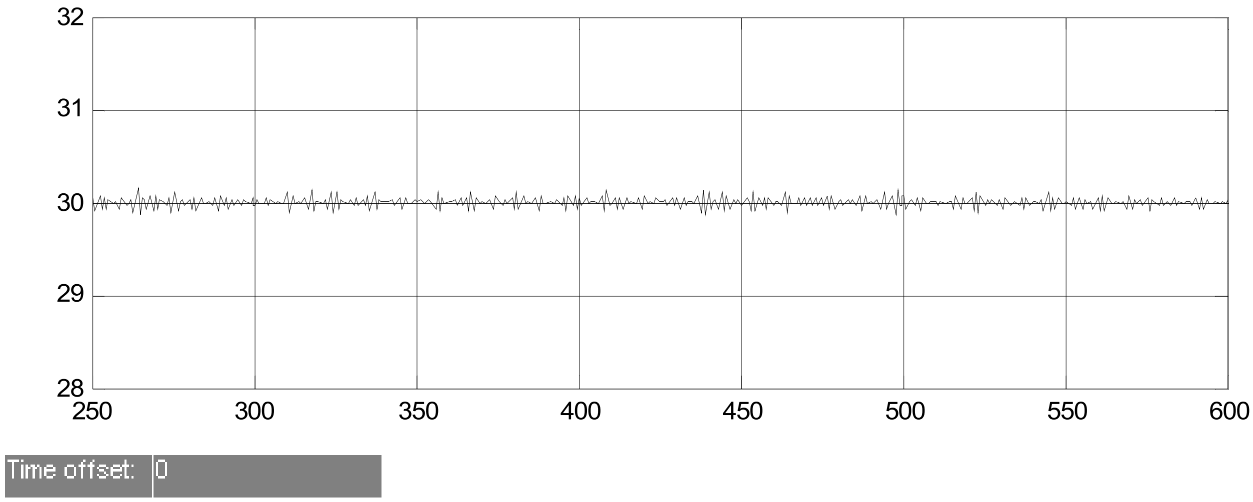


Figure 19, lead car accelerates: + 10 km/h from 300 - 500, PI-regulator

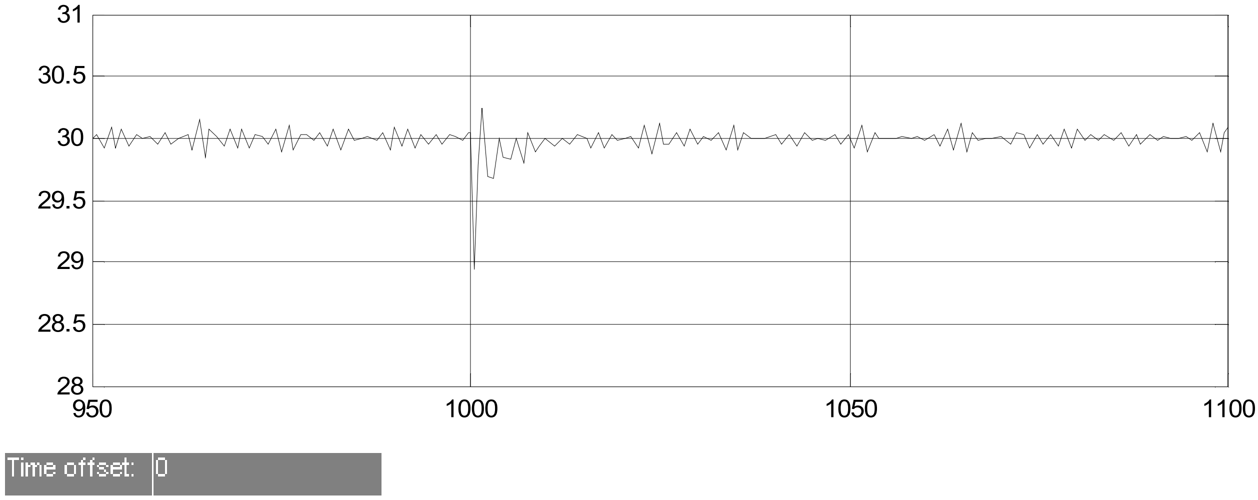


Figure 20, lead car instantly decelerates: - 10 km/h, PI-regulator

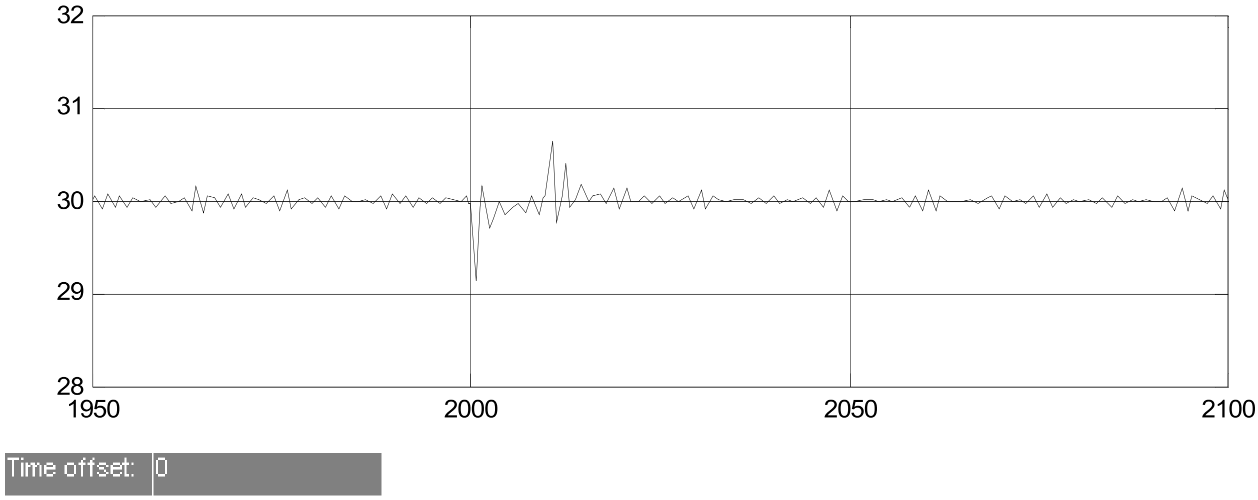


Figure 21, Invalid sensor reading: +/- 8 km/h, PI-regulator

As it may be seen in the above 5 figures (Figure 18 - 21), which shown a zoom of different parts of the graph, the regulator performs as expected.

Figure 18 shows that the initial jump to 80km/h causes an oscillation similar to the one for the unit step, but with the unit (1) replaced by 80km/h. We can also see that we have a settling time of about 15 seconds (within 1% of the final value). As for the rise-time we have to realize that the first time we pass the 30m mark is because the lead car moves away from us rapidly from a distance of 0m. The second crossing of 30m is because we have caught up in speed and have closed the gap. This means we can calculate the time we reach 80km/h as the time we start getting closer to the lead car, and at approximately 1 second. Naturally this is an unrealistic value for 0 - 80km/h in the real world, but for a 90 horse-power ideal DC-engine, with ideal traction, it is not so unrealistic - even one that is PI-regulated.

As mentioned before an instant acceleration (or deceleration) is not practically possible, so how does our regulator handle a gentler acceleration from 80km/h to 90km/h over 3 min and 20sec? We can see in Figure 19 that it handles this quite well. The change occurs in 300 and 500, and it is not possible to discern any fluctuation due to acceleration/deceleration from the general noise in the system. We can also see that the noise in the system causes an error of about +/- 10cm. This is due to our models extremely good acceleration/deceleration abilities. When the lead car accelerates slowly we can catch up so fast it just appears as noise. We do, however, not have a regulator that guarantees zero steady-state error for a ramp input.

Well then, what about an instant deceleration? We have previously mentioned that this is impossible, but it is similar to a unit step response, as may be seen from Figure 20. Here is may also be seen that this instant jump 10km/h only results in a error of up to 1m, or 3% of the reference.

Finally there is the sensor error. In Figure 21 we can see that we react to the sensor error almost as much as we did for the instant drop 10km/h. This is again because of our rapid acceleration. We are able to adjust quite a lot based on just that single sample. Here a lower acceleration would actually have been preferable.

There are other ways to counteract these sensor errors, as mention above in the design. Adding a low-pass filter would prevent this "over reaction", as only realistic readings would result in regulator changes. Naturally this would cause a slower rise-time to a step (instant change), but this is impossible anyway.

Finally there is the PID regulator, where we ran in to a bit of a problem. Simulink will not allow us to simulate the full 2200 or so seconds in one go. It just results in garbage. This makes the simulation difficult, as we can only look at 100s at a time. We are not going to show all of these sub-plots, but merely show the initial 0 - 100s plot, so it is possible to compare to the PI-regulator. This is shown in Figure 22.

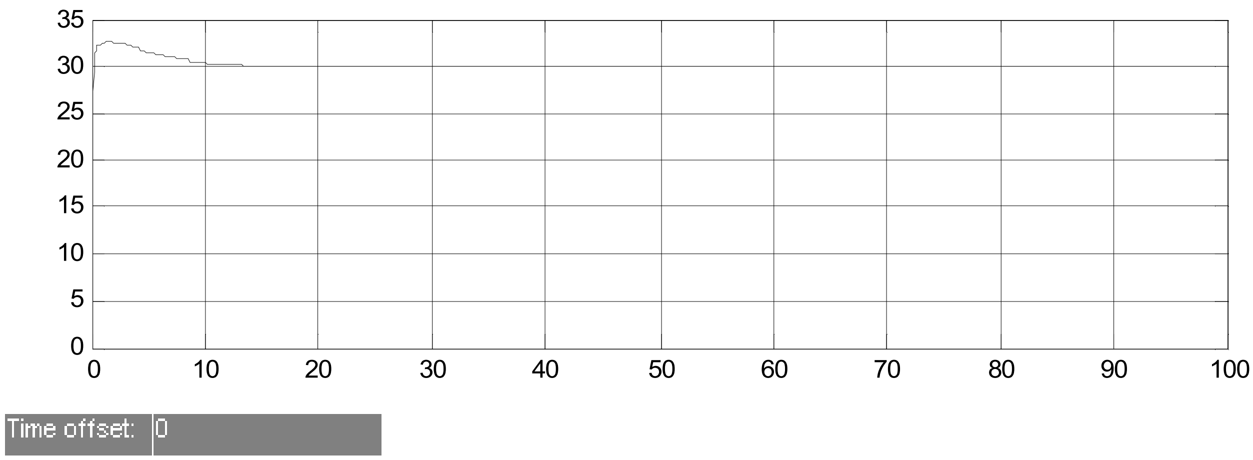


Figure 22, initial test interval: zero to 80 km/h, PID regulator

Here it can be seen, as expected, that our response time, overshot and settling time is greatly reduced. With a settling time of less than 10 seconds and no overshot at all, we follow the other car remarkably. It can also be seen just how powerful an ideal car with an ideal 90 horse-power DC engine is.

If we look at the sensor error, where the differential part can cause the most damage, we can see that we get an error of about 25cm from this, which is not bad, but this error can be greatly reduced by introducing the aforementioned low-pass filter.

6. Conclusion

All in all we feel that it has been shown that the PI and PID regulators have earned their popularity and can certainly be used to fulfill our requirements - even with a much smaller DC-engine. We have also shown that given the specification for a system it is possible to create a realistic model of this system which may then be used to design ones regulator. Finally we have shown how to tune a PI and PID regulator for optimal performance.