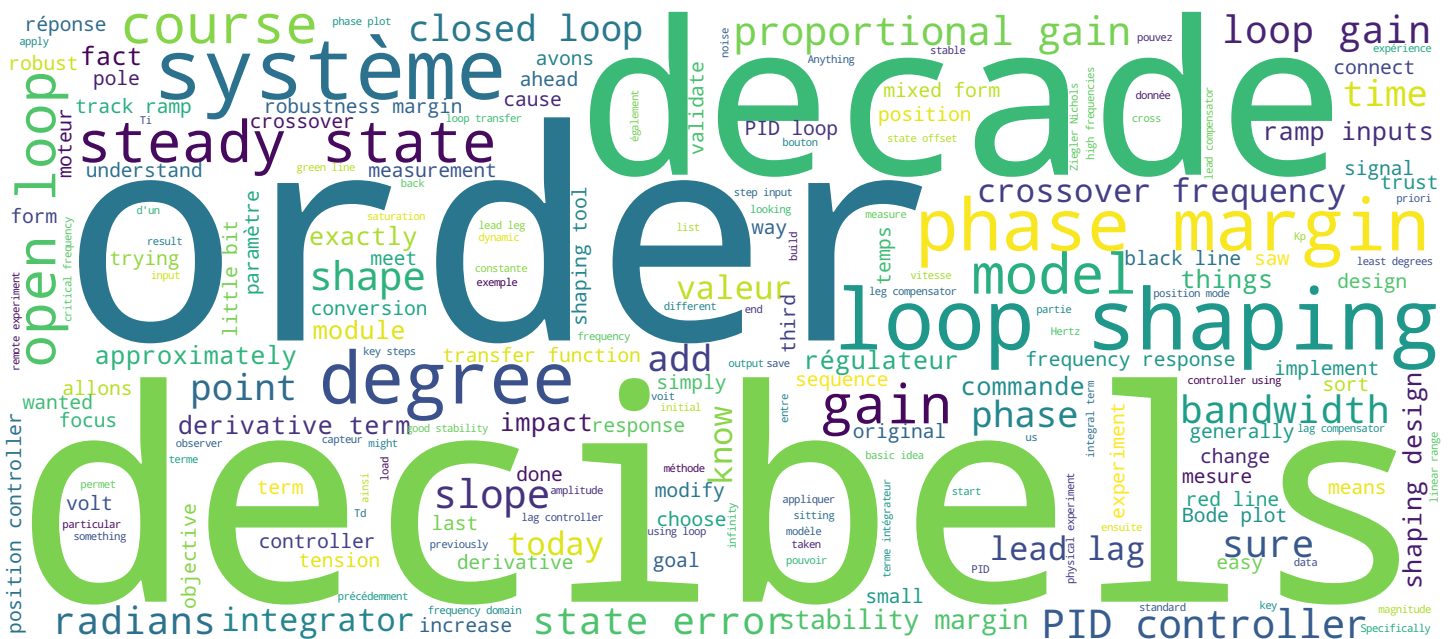


Position control using Loop Shaping

Christophe Salzmann & Colin Jones

Automatic Control Lab



Frequency domain controller design

1. Translate from objectives to loop shape
2. Shape the system response
3. Lead-Lag and PID equivalence
4. PID mixed and PID serial forms

TP Control Systems

Welcome to your hands-on session in which we're going to design a position controller using loop shaping. Our objective today is to do a frequency domain controller design, and there's going to be four key steps that we're going to go through. The first is we're going to translate from the objectives we're trying to achieve to an open loop shape, a shape that we want our loop gain to have in open loop. We're then going to take the model of our system and we are going to add a controller in order to modify that shape so that it matches the ideal shape that we laid out in the first point. The main tool that we use in order to do this shaping are, of course, a sequence of lead-lag controllers. Today, we're going to focus on the equivalence of a PID controller to a specific type of lead-lag controller. We're going to end up designing a PID controller. A PID controller designed in this way is going to be represented in a serial form. However, the PID controller that we implement, and in fact, most PID controllers, are implemented in mixed form. The last thing we need to do is to make this conversion, and then, of course, to validate our control design on the experimental platform.

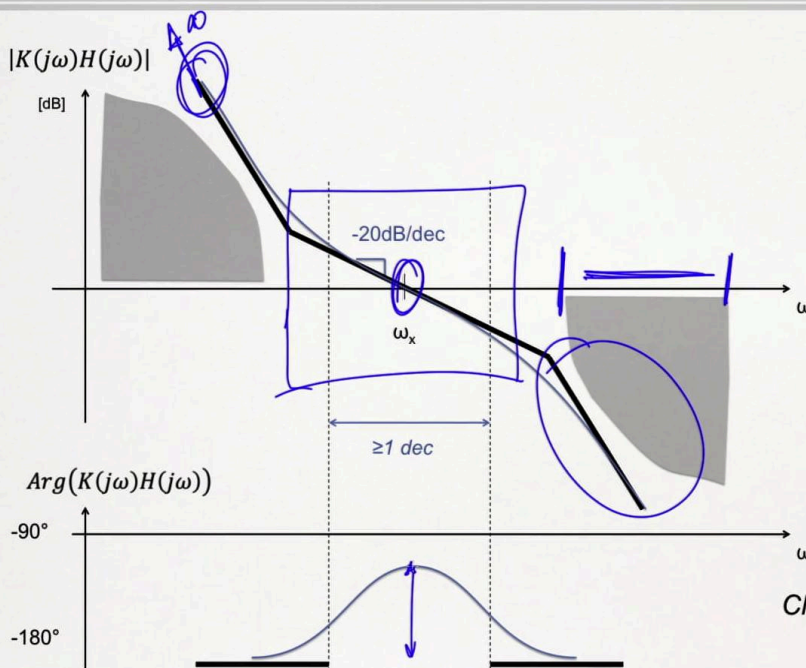
Notes

Summary



0m 07s

Loop shaping goals



- KH large for small ω (Steady-state error)
- KH small for large ω (Modeling errors, etc.)
- Slope of KH equal to -20dB/dec at crossover (Good stability margins)

Choose $K()$ to match the above requirements

TP Control Systems

We generally have three goals that we're trying to meet for the open-loop transfer function or the loop gain of our system. First, at low frequencies, we want our K times H , our loop gain, to be large. Generally, we want it to go to plus infinity as the frequency goes to zero. Of course, the purpose of this is this is exactly how we get zero steady-state error. Second, at high frequencies over here, we want our loop gain to become very, very small. This is primarily because at high frequencies up here, we generally don't have a good understanding of our system. Anything that we're amplifying over here is almost certainly just going to be noise or model errors or things that we don't really understand, and so we don't want our system to amplify those things, and so we want our gain to be low. The final point is that the crossover frequency in this area here, we want to have a good stability margin. The stability margin we're interested in here is if we're travelling through 0 decibels, the stability margin that we talk about is the phase margin. We want our phase margin to be at least 60 degrees, hopefully a little bit higher.

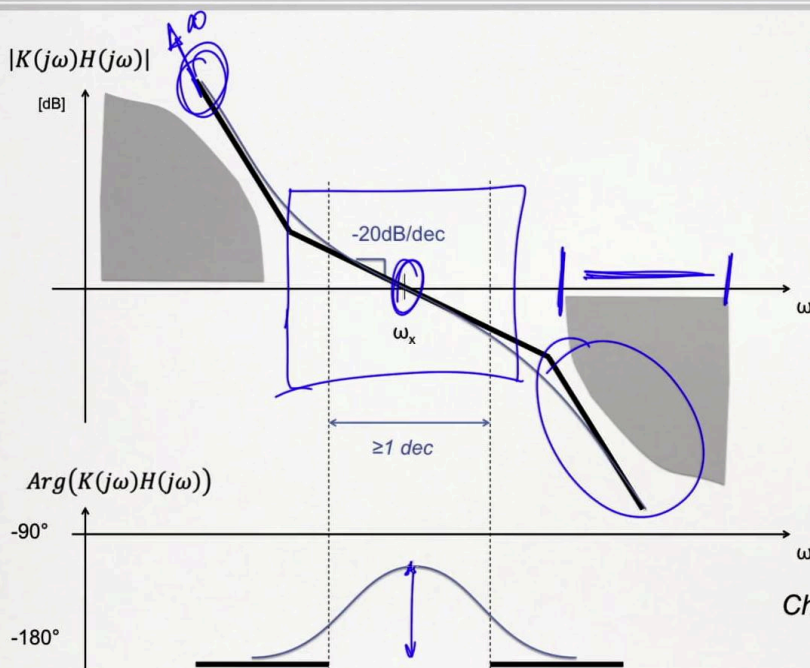
Notes

Summary



1m 14s

Loop shaping goals



- KH large for small ω (Steady-state error)
- KH small for large ω (Modeling errors, etc.)
- Slope of KH equal to -20dB/dec at crossover (Good stability margins)

Choose $K()$ to match the above requirements

TP Control Systems

What we know from the Bode integral formula is that if we have a slope of -20 decibels per decade, then our phase is approximately going to be about -90 degrees. We look for a slope of -20 decibels per decade at crossover in order to get a phase of -90 , which should give us a stability margin of about 60 - 90 degrees. Now, our goal is to start from the frequency response of the system that we have and then to choose a controller K , which will turn our initial system frequency response H , into one of exactly this sort of shape.

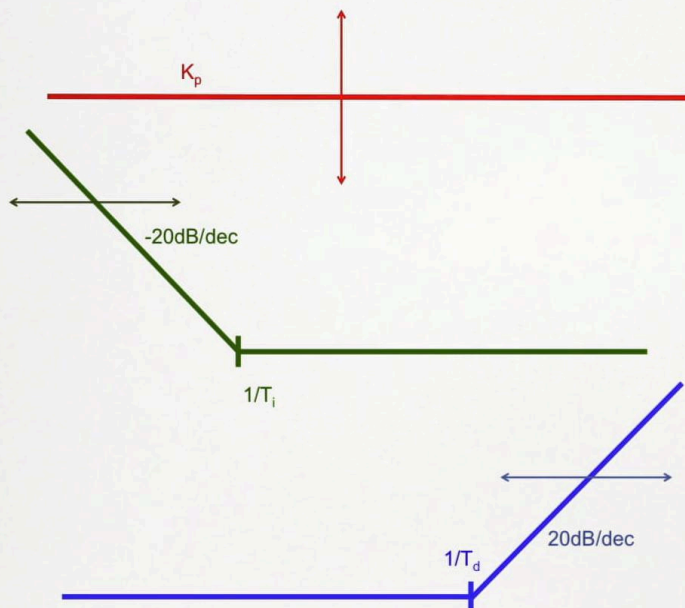
Notes

Summary



2m 35s

Lead/Lag and PID terms



PID

$$K(s) = K_P \left(1 + \frac{1}{T_i s} \right) (1 + T_d s)$$

$$= K_P \frac{s + 1/T_i}{s} (1/T_d + s)$$

Lead-Lag

$$C_{ar}(s) := \frac{s + b_{lead}}{s + a_{lead}} \cdot \frac{s + b_{lag}}{s + a_{lag}}$$

$$\underbrace{0}_{a_{lag}} \leq \underbrace{\frac{1}{T_i}}_{b_{lag}} \leq \underbrace{\frac{1}{T_d}}_{a_{lead}} \leq \underbrace{\infty}_{b_{lead}}$$

TP Control Systems

The tools that we have available in order to do this shaping is a sequence of lead-leg compensators, which take the form that you see down here. Now, today, what we're going to focus on is a very special class of lead-leg compensator, which is the PID controller. You can see if we write it here in series form, what we have is one lead compensator, which is our derivative term, and that is simply a zero at $-1/T_d$. The Bode plot here is going to be zero until we hit $-1/T_d$, at which point it's going to slope upwards at 20 decibels per decade. We have our integral term over here, which is $1/s$ at $-1/T_i$ and 1 -pole at zero. You can see we're going to be sloping down as a result of the pole at -20 decibels per decade until we hit our zero, which occurs at $-1/T_i$ right here, and then we level out at zero. Finally, of course, we have a proportional gain, which just at all frequencies moves this up and down. The sum of these three things is going to be the lead-lag compensator that we're going to look at today.

Notes

Summary



3m 16s

Loop shaping design

Idea

- Shape system response using lead/lag filters
- Match the ideal open-loop frequency response

Loop shaping steps

1. Cross over frequency
2. Phase margin
3. Ramp tracking without error

TP Control Systems

The basic idea behind loop shaping design is that we want to shape the system response by using these lead-lag filters, and the shape that we're looking for is the ideal open-loop frequency response that we saw previously. Now, today we're going to focus on three key steps. First, we're going to modify the gain in order to choose our crossover frequency. Then we're going to look at the phase margin in order to make sure that we have a sufficient phase margin that our system will be robust. Finally, we want to be able to track ramp inputs today, and so we're going to add additional lead compensators in order to be able to track ramp inputs.

Notes

Summary



4m 24s

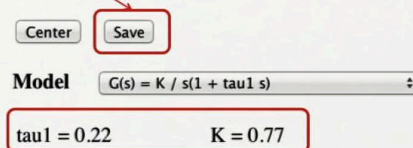
Experiment 1: System model in position

Using the remote experiment tool

- Apply a step (in position and open loop mode) within the linear range
- Save the measurements

Using the temporal fit tool

- Load the measurements and fit the model
- Save the **model** using the **save** button



Note: very similar to what has been done in M4

TP Control Systems

We're now ready for the first experiment. Here, all we need you to do is to build a model of our system in position. Go ahead and connect to the remote experiment tool, apply a step input in the linear range, save the measurements, and then open the temporal fit tool, load the measurements that you've taken and fit a model. Then make sure that you save the model. You should be able to do this very quickly, as we've done this previously in two other modules as well.

Notes

Summary



5m 08s

Experiment 2: Loop shaping

Using loop shaping, design a controller for **position** with the following specifications:

- No steady-state error for ramp inputs
- Phase margin larger $\geq 60^\circ$
- $\omega_b = 5 \text{ rad/s}$

Using the interactive PID loop shaping tool

- Load the previously saved **model**
- Select the desired **controller**

TP Control Systems

The second experiment is the main experiment today, and we want to do a loop shaping design. There are three objectives that we want to meet in our loop shaping design. The first is that we want to make sure that we have no steady-state error for ramp inputs, the second is that we want to have a phase margin larger than 60 degrees, and the third is that we want to have the bandwidth of our closed-loop system to be 5 radians per second, approximately. What we're going to use is our interactive PID loop shaping tool. Go to the loop shaping tool and load your previously saved model and select an appropriate controller from the list of controller structure.

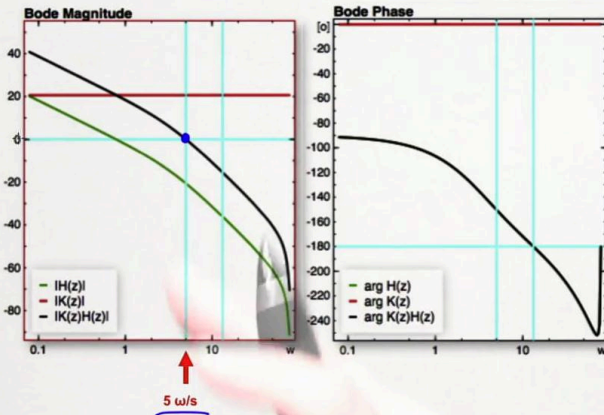
Notes

Summary



5m 34s

Experiment 2: Loop shaping, cont.'



Cross over frequency:

- Set the K_p to 1, T_i to ∞ (>9000) and T_d to 0
- Increase K_p until $\omega_0 = \omega_b$ (vertical cyan line)

With $\omega_b = 5 \text{ rad/s}$

TP Control Systems

The first thing that we want to look at is to choose the crossover frequency in order to set the bandwidth of our closed-loop system. We're going to do this by tuning just the proportional gain. Turn off your integrator and your derivative. You can do that by setting your integrator to a very large number and your derivative to a very small number. Then what we want to do is to increase the proportional gain until our crossover frequency, ω_{c0} , is approximately equal to the bandwidth that we wanted, which in this case is 5 radians per second. Of course, we recall that the crossover frequency of our open-loop system is approximately equal to the bandwidth of our closed-loop system. As we change the proportional gain, what we'll see is that the black line will move up and down. The green line is the original system dynamics. The red line is the controller or the compensator that we're adding, and the black line is the sum or the product of these two things. You can see that changing our proportional gain, the red line up and down, will have no impact on the phase plot. It's only going to impact the magnitude plot. We want to have 5 radians per second over here, and we want for that to go through the 0 dB point.

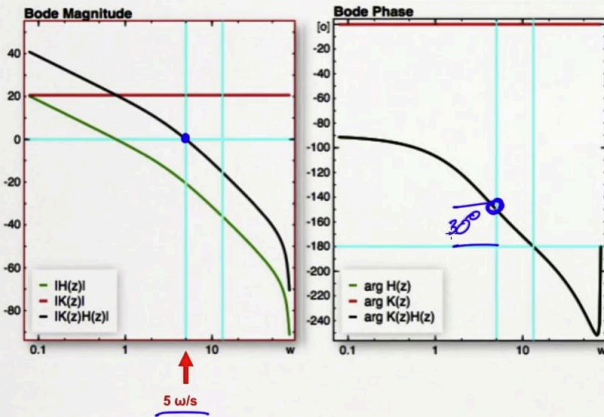
Notes

Summary



6m 13s

Experiment 2: Loop shaping, cont.'



Cross over frequency:

- Set the K_p to 1, T_i to ∞ (>9000) and T_d to 0
- Increase K_p until $\omega_0 = \omega_b$ (vertical cyan line)

With $\omega_b = 5 \text{ rad/s}$

TP Control Systems

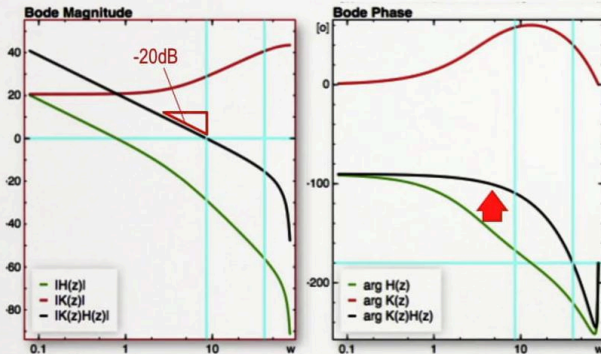
Just change the gain until you get that particular value. Now, the second thing we want to do is to make sure that we have sufficient robustness margins. We can see, of course, that our critical frequency is going to be 5 radians per second. But at 5 radians per second, we're sitting here at about -150 degrees. That means that our phase margin is currently only about 30 degrees, but we wanted a phase margin of at least 60 degrees.

Notes

Summary



Experimentation 2: Loop shaping, cont.'



Phase margin:

- Add derivative term, i.e. increase T_d from 0
- Make slope at $\omega_b = -20$ db (lift phase at ω_b towards -90)

TP Control Systems

In order to improve this situation, what you want to do is to add a derivative term. You can see this red line is now our proportional gain plus the derivative term. You can see that in magnitude, the impact is it causes a slope increase of 20 decibels per decade, and in phase, it causes an increase of about 90 degrees in the phase over some frequency range. The whole point is that we now want to see that our crossover slope, the slope at which we cross over zero, is about -20 decibels per decade. You can see that the impact that that's going to have on the phase is our phase margin goes from about 30 degrees now closer to about 90 degrees, which is exactly what we want to see.

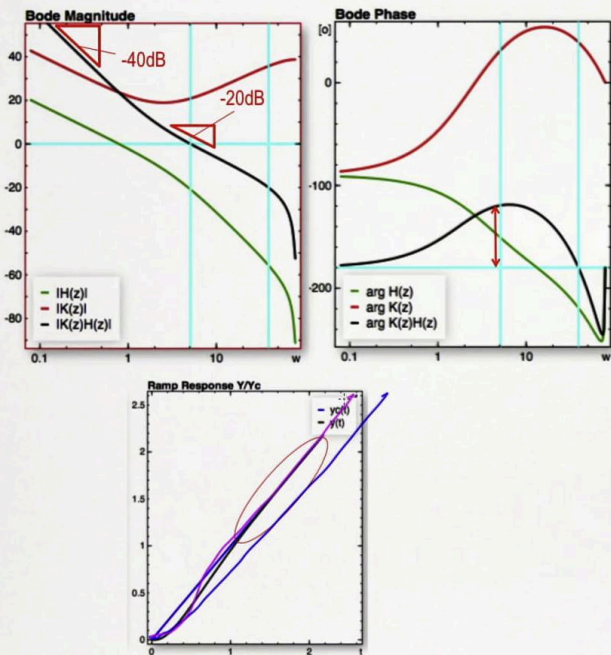
Notes

Summary



7m 57s

Experiment 2: Loop shaping, cont.'



Ramp tracking without error

- Two integrators \rightarrow initial slope of -40dB / dec
- Add integrator, i.e. decrease T_i from >9000
- Observe integrator effect when tracking a ramp
- If needed, adjust K_p for crossover frequency
- Check robustness margins

TP Control Systems

Now, the third thing that we want or the third criteria that we had to meet today was that we wanted to be able to track ramp inputs with zero steady-state error. Now our current system has one, you can see here the original system, the green line, has one integrator. The slope as we head toward zero is -20 decibels per decade. But we know that if we want to track a ramp with zero steady-state error, we need to have this be -40 decibels per decade. The way that we get this is that we add an integral term over here or a lag compensator, and that will increase our slope by an additional -40 decibels per decade, and we'll have this slope of -40 as we go toward zero, which of course, gives us zero steady-state error in response to a ramp. Have a look at the ramp response in the PID tuning tool, and you should see that before you add the integrator, you'll have a steady-state offset like this. After you add the integrator, you'll have no steady-state offset afterwards. The final thing you want to do is to check your robustness margins and make sure that we still satisfy the conditions that we had before. You'll notice that since we added our derivative term over here, that in fact, the bandwidth of your system will have increased a little bit.

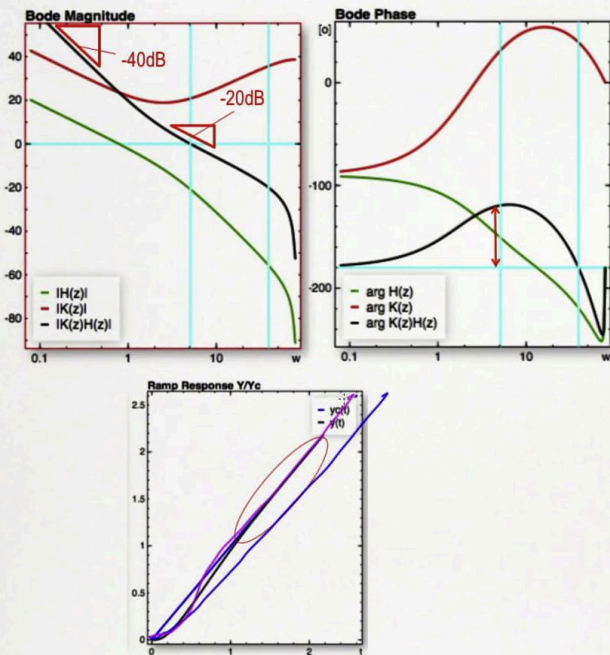
Notes

Summary



8m 41s

Experiment 2: Loop shaping, cont.'



Ramp tracking without error

- Two integrators \rightarrow initial slope of -40dB / dec
- Add integrator, i.e. decrease T_i from >9000
- Observe integrator effect when tracking a ramp
- If needed, adjust K_p for crossover frequency
- Check robustness margins

TP Control Systems

You might want to readjust your proportional gain just a little bit now in order to bring the bandwidth of your system back to 5 radians per second.

Notes

Summary



10m 05s

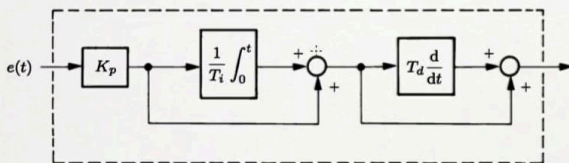
Experiment 2: Series to mixed form

- The implemented controlled is defined in the mixed form
- You need to convert from series to mixed form

$$K(s) = K_p \left(1 + \frac{1}{T_i s} \right) (1 + T_d s)$$



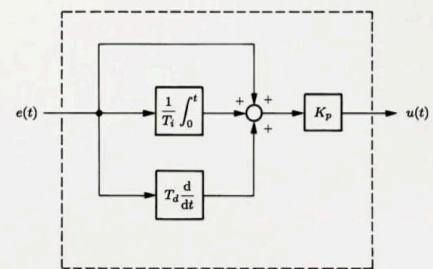
$$K(s) = K'_p \left(1 + \frac{1}{T'_i s} + T'_d s \right)$$



$$K'_p = K_p \frac{T_i + T_d}{T_i}$$

$$T'_d = \frac{T_i T_d}{T_i + T_d}$$

$$T'_i = T_i + T_d$$



TP Control Systems

Now, the last thing we want to do is to actually implement this controller on the physical experiment. However, our PID loops in the physical experiment are written in mixed form as we see over here. This is standard for most PID implementations. But since we've done a loop shaping and lead-lag design, our PID controller is in this form over here. What we need to do is to convert the gains that we see over here to the gains that we need over here. You can either go and do that algebra, or you can trust that we've done the algebra correctly over here. Make the conversion so we can actually implement this thing.

Notes

Summary



10m 16s

Experiment 3 – Validate your controller

- Connect to the system
- Select the **closed** loop and **position** modes
- Set the PID parameters according to the ones in mixed form
- Apply a ramp signal with an amplitude 2 [v], and a frequency of 0.1 [Hz]
- Verify that there is no tracking error

+

TP Control Systems

The final part of this experiment is to validate your controller. Go ahead and connect the system, select closed loop and position mode, and then set your PID parameters according to the ones that you found, and most importantly, make sure that they are now in mixed form because this is what the system is expecting. Now, apply a ramp signal with an amplitude of 2 volts and a frequency of 0.1 Hertz. The key thing you want to do is to verify that there is in fact no tracking error. You can also add mismatch, you can also add uncertainty, and ensure that in fact your system is still robust because we know we have a robustness margin of 60 degrees.

Notes

Summary



10m 54s

Conclusions

What you have learned today

- Loop shaping design is expressed in terms of frequency requirements
- Control objectives commonly described using Bode plot
 - KH large for small ω (Steady-state error)
 - KH small for large ω (Modeling errors, etc.)
 - Slope of KH equal to -20dB/dec at crossover (Good stability margins)
- Easy to sketch manually or interactively

TP Control Systems

What we looked at today was the design of a position controller using loop shaping. First thing we saw was that loop shaping design is expressed in terms of frequency requirements. What shape we need the open loop transfer function to have so that it has good properties in closed loop. Specifically, we want the open loop gain, K times H to be large for small ω . We do this so that we have zero steady-state error. Second, we want K times H to be very small for large ω , specifically at frequencies beyond which we don't trust our model or our sensors aren't working correctly and so on. We don't want to be amplifying noise. We don't want to be amplifying things that we don't understand. We want our loop gain to roll off these higher frequencies. Finally, since we want our system to be stable, and we want our system to be robust, we need to have that the slope of K times H is equal to -20 decibels per decade at crossover. If we do this, we know that we will have good stability margins. Finally, hopefully you saw that loop shaping or design using these sorts of techniques is very, very easy to do. Specifically, it's easy to do through a manual sketch or interactively through something like the PID design tool that you saw today.

Notes

Summary



11m 34s