LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY

LIGO Laboratory / LIGO Scientific Collaboration

LIGO-T050197-00-R 09/27/2005

IMPLEMENTING A MODAL CONTROL AND ESTIMATOR FOR A TRIPLE PENDULUM

L. Ruet, R. Mittleman, D. Ottaway

Distribution of this document:
Advanced LIGO Suspensions/Isolation Working Group

This is an internal working note
of the LIGO Scientific Collaboration

California Institute of Technology
LIGO Project - MS 18-34
1200 E. California Blvd.
Pasadena, CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project - NW17-161
175 Albany St
Cambridge, MA 02139
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

LIGO Hanford Observatory
P.O. Box 1970
Mail Stop S9-02
Richland WA 99352
Phone 509-372-8106
Fax 509-372-8137

LIGO Livingston Observatory
P.O. Box 940
Livingston, LA 70754
Phone 225-686-3100
Fax 225-686-7189

http://www.ligo.caltech.edu/

1 SUMMARY

2 INTRODUCTION

The advanced LIGO suspensions have been designed to provide a very good passive isolation at high frequencies. However, to get this performance and reduce the effect of thermal noise, the suspension's resonances need to have a very high Q at the lowest frequencies, which need to be damped. The damping strategy is to use active control to reduce the Q of the resonances and reduce the settling time after a shock. While this method works very well to damp the resonances, it also creates new problems by re-injecting the sensor noise. As a matter of fact, the platform the suspension sits on is so quiet that the sensor noise is bigger than the seismic noise at high frequencies. By using simple control, the feedback loop will re-inject and increase the noise in the pendulum at high frequencies

So the goal is to create an active control that can

- Damp the resonances
- Re-inject as little sensor noise as possible at high frequencies (above 10Hz)

Classic feedback method can be used to do that, but they require a lot of work and a complex process of filter design is needed, which can takes time and is not flexible. On top of that, the performances you can get are acceptable are not optimal.

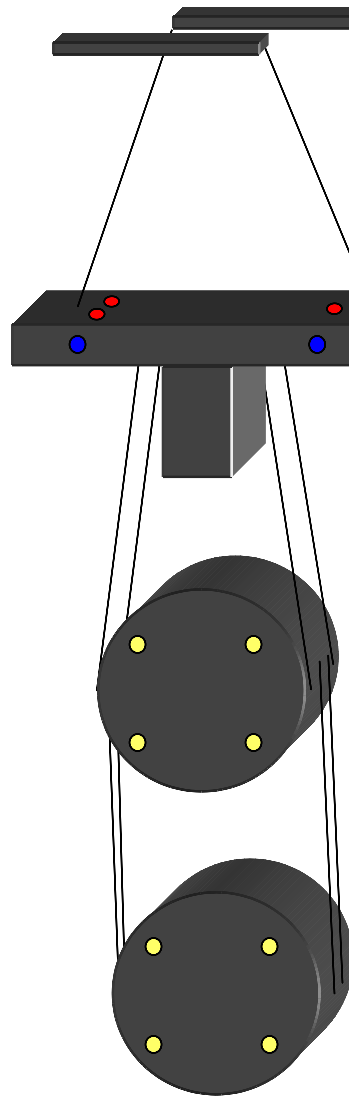
Another approach can be used: by using a modal decomposition, we can damp each mode independently and have more damping on the most energetic modes, while we have less damping but less sensor noise transmission on the highest weak modes. In order to work, we need to have the entire modal signal, we use an estimator to recover the signal we can't measure directly and we will see that this estimator can also be used to filter a part of the noise.

We will then apply this new method to the mode cleaner triple pendulum in advanced LIGO configuration and compare the performances with a classic feedback method.

In the last section, we will see that we have tested this new method on the triple pendulum we have in LASTI.

3 THE TRIPLE PENDULUM

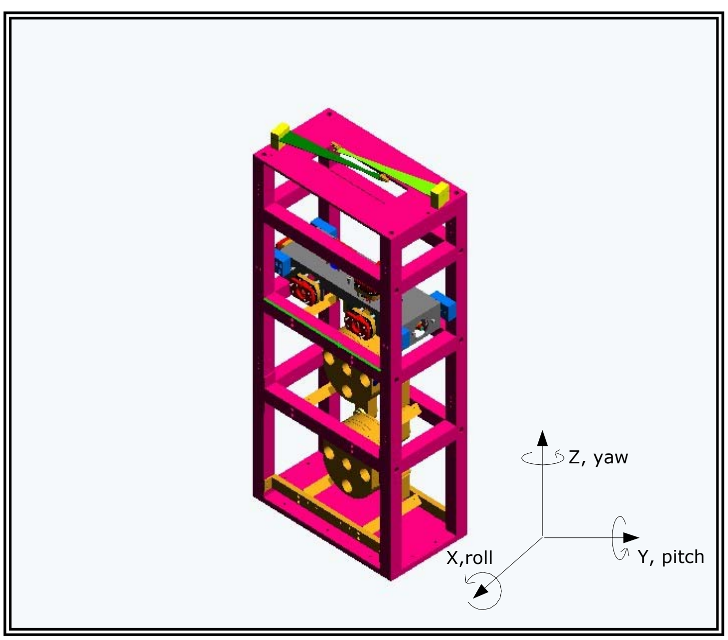
Before explaining the method for the control loop. It is important to know the system we are going to talk about. The example we will take in this document will be the triple pendulum. The method can be applied to any kind of suspension, but since we have a triple pendulum in LASTI, it is easier to test and verify our method on this suspension.



The triple pendulum is made of 3 masses of about 3Kg each. It gives 3 stages of passive isolation. The vertical isolation is provided by 2 stages of blades (top and first mass).

The triple pendulum also has 14 sensors and actuators. The 8 bottom sensors/actuators (yellow dots) are not used in the damping control. The 6 sensors/actuators on the first mass (red, blue and green) are used to sense and damp the pendulum in the 6 degrees of freedom.

The sensors/actuators are called OSEMS, they are both a shadow sensor and a coil/magnet actuator.



4 MODAL CONTROL

4.1 Introduction

Why would we like to use another method for the control? The main reasons are:

1. Make the control easier to design
2. Reduce the sensor noise injection

In order to do that, we want to work in a new basis where the equations of motion would be simpler.

The method consists of a modal decomposition. Instead of controlling the system in the "real" basis where (x1,x2,x3…) are the motion of each mass of the pendulum, we apply a mathematic change to work in a new, more simple basis called the modal basis (q1,q2,q3…). In this basis, the equations are decoupled, which provides 2 advantages:

1. The control design only consists in changing few gains for each mode
2. The control of each mode can be optimized so that the sensor noise injection is reduced

4.2 How, Mathematics

The idea is simple; we want to find the basis change that will decouple the equation of motion.

Let's start with the equation of free motion in the real (x) basis (M is the mass matrix and K is the stiffness matrix):

$$M\ddot{x} + Kx = 0$$

We apply the transformation $x = Xe^{i\omega t}$

$$\omega^2 MX = KX$$

$$M^{-1}KX = X\omega^2$$

Where ω^2 are the eigenvalue of $M^{-1}K$ and X are the eigenvectors of $M^{-1}K$.

We call ϕ the matrix formed by the eigenvectors X . By definition, calculating a matrix in its eigenvectors basis makes it diagonal, so let's now write the equation of motion in the new basis ϕ . We call q the new variable in this basis.

$$x = \phi.q$$

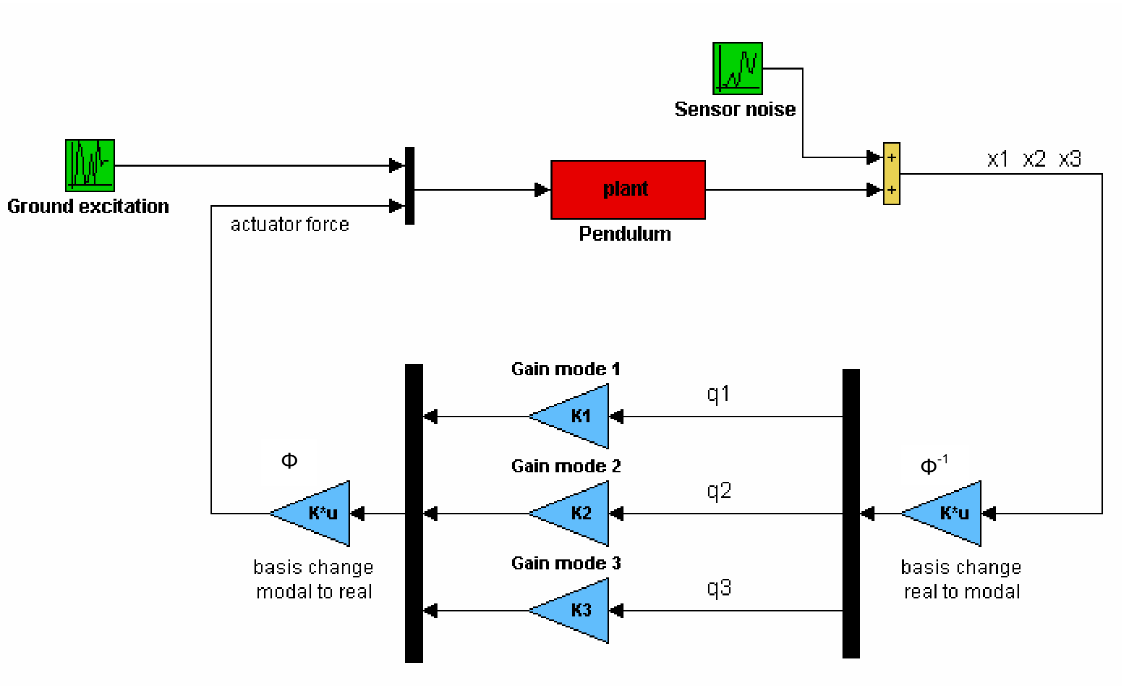
$$\ddot{q} + \phi^{-1}M^{-1}K\phi.q = 0 \quad \text{for a free motion}$$

$$\ddot{q} + \phi^{-1}M^{-1}K\phi.q = \phi^{-1}F \quad \text{for a forced motion}$$

(Since ϕ is orthogonal, $\phi^{-1} = \phi^t$ and we can use ϕ^t to save computation time)

having the computer applying the basis change inside a control loop is simple, as you can see on the diagram below :

- We change the basis by multiplying the real output by $\phi^{-1}(or\phi^t)$
- The new variables (q1,q2,q3,…) can be controlled one by one without having to worry about coupling.
- We go back to the real basis by multiplying the real data by ϕ to apply the force



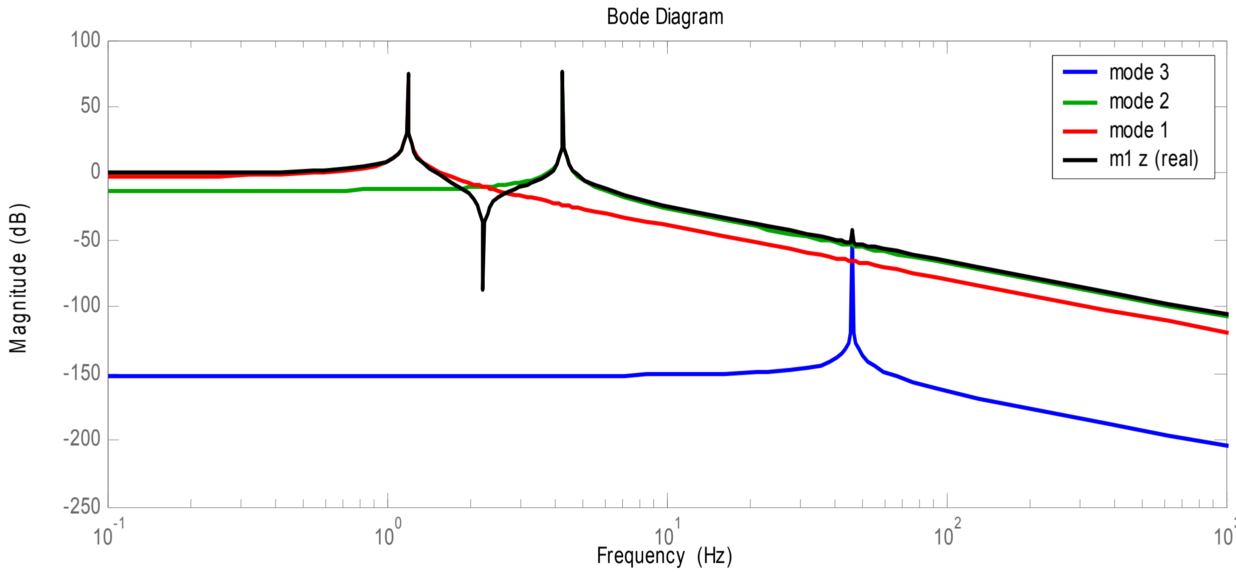
Modal control diagram

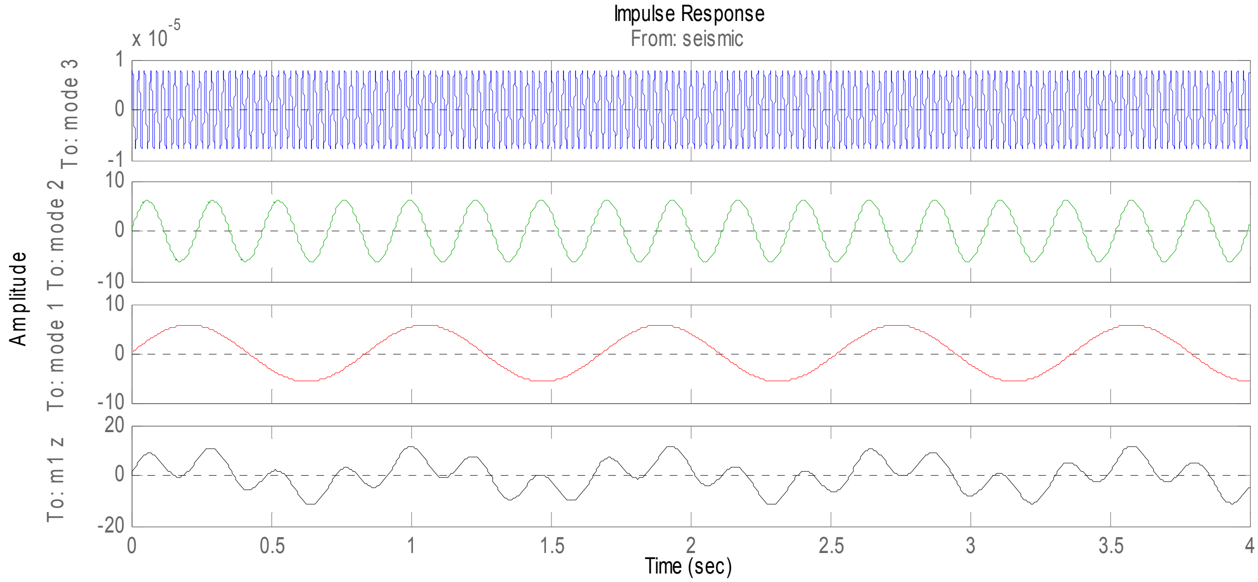
4.3 Example

In the following example, we will study the case of the vertical (x becomes z) motion of the mode cleaner triple pendulum; this is a 3 dof system with 3 resonances at 1.2 Hz, 4.8Hz and 46.3 Hz.

We use this plant in our diagram shown above and check the results of the basis change :

- We plot the Bode magnitude of the TF from ground to z1, q1, q2 and q3
- We plot the step response from ground to z1, q1, q2 and q3





Instead of having one complex 3dof system, we have 3 simple single dof systems that we can control separately. We are now be able to design a control more easily.

4.4 Modal control and sensor noise

We have seen how modal control makes the control easier to design. But we will also see that it enables you to practically eliminate the sensor noise injection while keeping a very good damping.

How can we do that ?

- The lowest modes are the dominant mode in the impulse/step response, thus the biggest gain are required on the lowest modes, but not on the highest modes.

- Because the lowest modes are at the lowest frequencies, they are very easy to filter with aggressive filter. Also if you look at the plots above, the lowest modes already naturally filter the high resonances.
- The strategy is to apply high gain and aggressive filtering on the lowest modes; his is easy and will provide a very good damping. For the highest modes, we can now use lower gain because they are not very important in the impulse response, it will at the same time decrease the sensor noise.

To conclude, the modal control enables us to use high gain on the lowest mode, while using lower gain for the highest modes (and thus decrease sensor noise injection). Each mode can also have its own filter optimized to reduce the sensor noise re-injection.

4.5 Conclusion

We have seen the modal control method has 2 main advantages

1. It makes control design easy
2. It helps to reduce sensor noise injection

However, it is important to remember it also have drawbacks:

1. It needs a good model to generate the eigenvector matrix
2. We need to measure as many degrees of freedom as the number of modes the system has (we need to know the “full state”). This is not always possible.

In our case, we don’t have as many sensors as degrees of freedom, and so it makes a direct modal decomposition impossible to achieve. To surmount this problem, we can use an estimator to reconstruct the full state of the system: This will be described in the section below.

5 ESTIMATOR

5.1 Introduction

We have seen previously that we can only use modal control if we can measure the full state; This is unfortunately not going to be the case in the LIGO suspensions. We need to find a way to generate these missing data.

The method we will use is called an estimator, we will reconstruct the full state by using a model that contains every degree of freedom, and by constantly updating the model with the measurements we can get.

Once we can generate the full state, we can use the modal control we saw above and take advantage of this method.

In this section we will not talk about modal control at all, the controller we describe can be any kind of controller

5.2 How does it work

The first idea is to use a model in parallel with the real plant; the model can generate as many outputs as you need to feed your control. Of course the output of the control can be re-injected into the model as known inputs.

However, we easily imagine that the model lacks one input that we don't know: the seismic noise (or excitation).

The second step is to add a loop to the estimator. We take the output of the estimator and keep the signal we can measure on the real plant. We then compare this signal to the real plant's one (subtraction). This gives us an error between the model output and the plant output. We can then feed this error to a feedback in the model to minimize it.

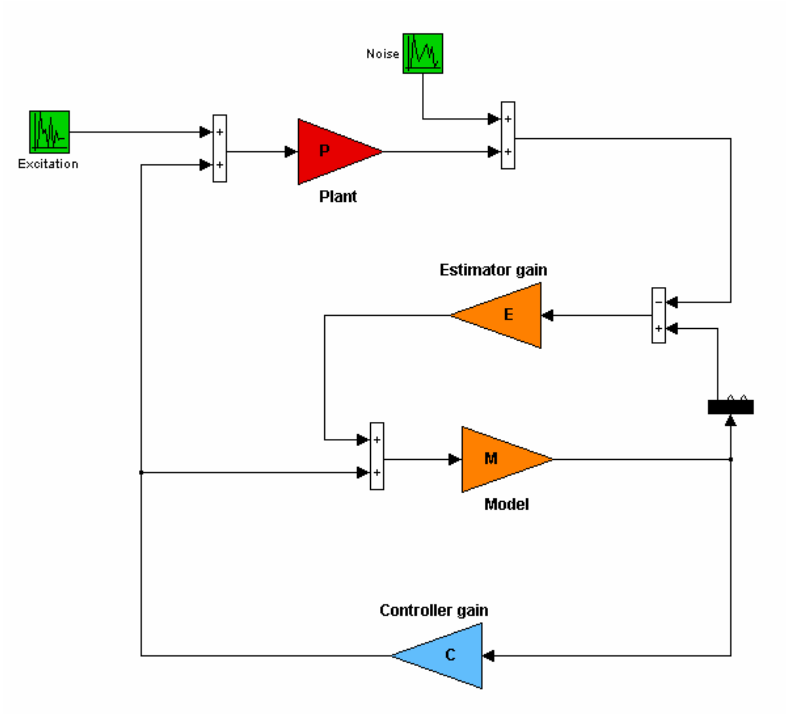
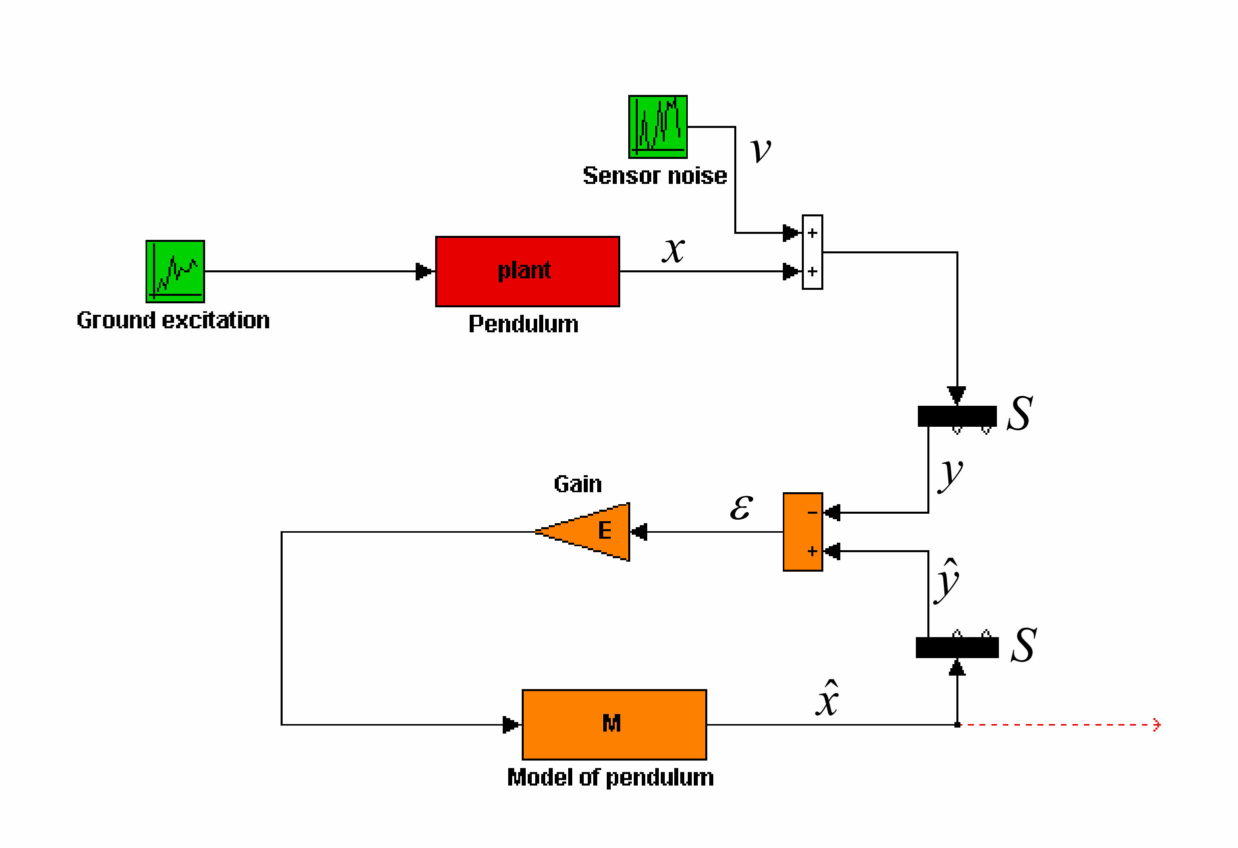


Diagram of an estimator
The controller C can be any kind of control it has an input and an output, this is all we need

This system can now work, the estimator uses the measurement to minimize the error between the model output and the real output, meanwhile, it also generates the other degrees of freedom we can't measure.

One might wonder if the output of the controller really needs to be injected into the model too. The system would work if you don't add this connection; the force will just be considered at another noise that the estimator loop will minimize. However, since we have this information, it is better to use it.

Let's now focus on the estimator only and forget the controller for a few moments:



- *The ^ sign is used for estimated state*
- *x is the complete state (for ex $m1_x$ $m2_x$ $m3_x$)*
- *y is the part of the state you can measure ($m1_x$)*
- *The gain E can be any kind of control (SISO, MIMO, filtering, LQ, pole placement, …)*
- *S is a simple matrix represented by the 2 black demux boxes. S is the matrix to go from the full state to the measured state. For example, for a 3dof system, S=[1 0 0] if we can only measure the first dof*

$$\varepsilon = \hat{y} - y = S.\hat{x} - S.x - S.v$$

$$\hat{x} = M.E.\varepsilon$$

$$\hat{x} = E.M.S.\hat{x} - E.M.S.x - E.M.S.v$$

$$\hat{x} = \frac{E.M.S}{E.M.S - 1}.x + \frac{E.M.S}{E.M.S - 1}.v$$

The reason it works is because the motion of each mode is coupled to the first mass (the suspensions have been designed for that), thus M is not diagonal and we can generate missing data with the model and signal we can measure.

Like we wrote above, the estimator feedback (called estimator filter or estimator gain) can be any kind of control, it can be SISO or MIMO.

In a MIMO estimator, the error that is calculated between the estimated and real output is distributed on every input of the model (every mass), It is also possible to combine multiple errors in case where we have more than 1 sensor. The MIMO estimator is more complicated and harder to “visualize” than a SISO is.

The reasons why we choose a SISO estimator are the following ones:

- It is easier to analyze
- We are using only one sensor on the triple pendulums
- The main perturbation comes from the frame (through first mass) so there is no real reason to distribute the error on the other degree of freedom
- IT lends itself to an intuitive understanding

We will probably need to study the MIMO estimator for more complex cases later (see conclusion) but the SISO estimator will work well enough for the triple pendulum.

5.3 Behavior for low and large gain

1. $\hat{x} \xrightarrow{EMS \to \infty} x + v$

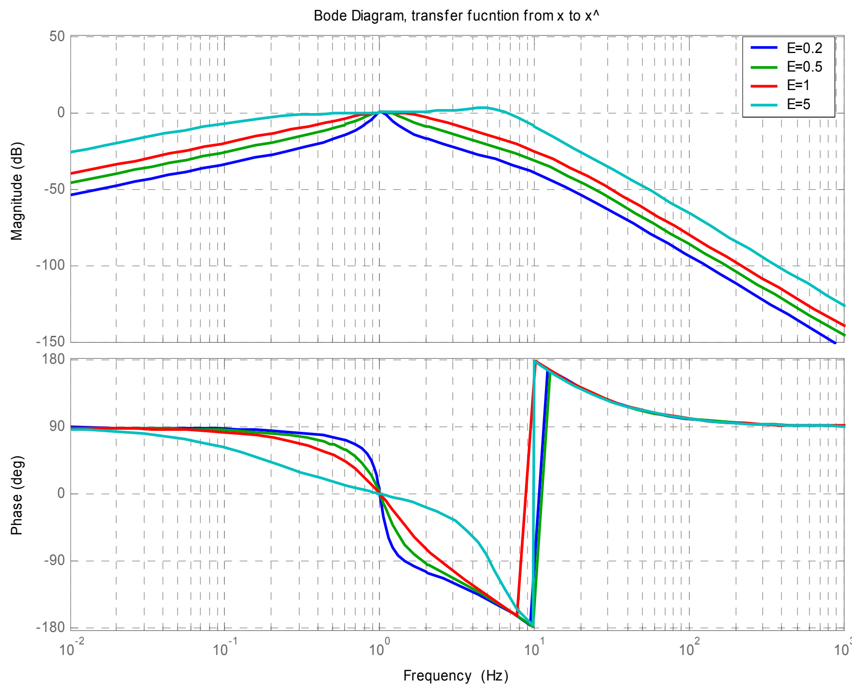
The estimated output is equals to the real output (with sensor noise), it is useful at the resonances, it means our estimator doesn’t modify the phase or the gain.

2. $\hat{x} \xrightarrow{EMS \to 0} E.M.S.x + E.M.S.v$

The noise and the real data are filtered but the phase is modified, this is interesting above the resonances when we don’t care about the phase.

It is now interesting to see what happened between those 2 limits and how the estimator behaves when we modify the value of the estimator gain E.

Let’s plot the transfer function from $x \ to \ \hat{x}$ for a single dof system (simple pendulum with a resonance at 1Hz) for different gains of E (we will use a very simple estimator filter for this example):



Behavior of the estimator for a 1dof system and different values of E
One dof system (pendulum) with resonance at 1hz

What can we see on this plot:

- The lower the estimator gain is, the lower the transmission at high frequency is, this is good for sensor noise filtering
- When the gain decreases, it creates a “bump” at the resonance and keeps the magnitude up, which means you don’t loose control ability. You do loose a bit of phase but at an acceptable rate
- The estimator basically creates a good filter that you can tweak just by turning one gain up or down and avoids spent time on complex filtering design. We have seen it for 1dof, and it would be even more useful for multi-dof (the estimator will create many bumps at the resonances while keeping the phase up, just by turning a single gain). No complex filter design is required.

5.4 Stability

The stability of the estimator alone is quite simple, you just want the estimator loop to be stable, and so you apply usual the rules to get a stable loop (either using Bode or Nyquist criteria).

5.5 Conclusion

We have seen how we could generate the full state by only using some measurements and a model of the system we want to control. It will be useful to combine it with the modal control and thus create a simple control loop which provides good damping and sensor noise filtering. We have seen that the estimator is a good filter by itself, by reducing the gain of the loop; we can reduce the transmission of the noise without reducing the gain at the resonance frequencies and without loosing too much phase. We will see that in detail in the next section.

6 CONTROL AND ESTIMATOR

6.1 Introduction

We have seen how the modal control works and what was the advantage of using such a method; unfortunately, it can be used only if you know the full state of the system. We have then seen a method to reconstruct the state with some measurements: the estimator. We will now focus on how to combine these 2 systems and build a whole control.

In this section we will not talk about modal control, the controller we describe can be any kind of controller

6.2 Mathematical loop model

We will calculate a mathematical model of the diagram shown below, the plant is driven by the excitation w and the output of the plant is added to the sensor noise v. This sum is then injected into the estimator and the output of the estimator $\hat{x}$ (reconstructed state) is injected in the controller C (which represent a full modal control) to generate the feedback force

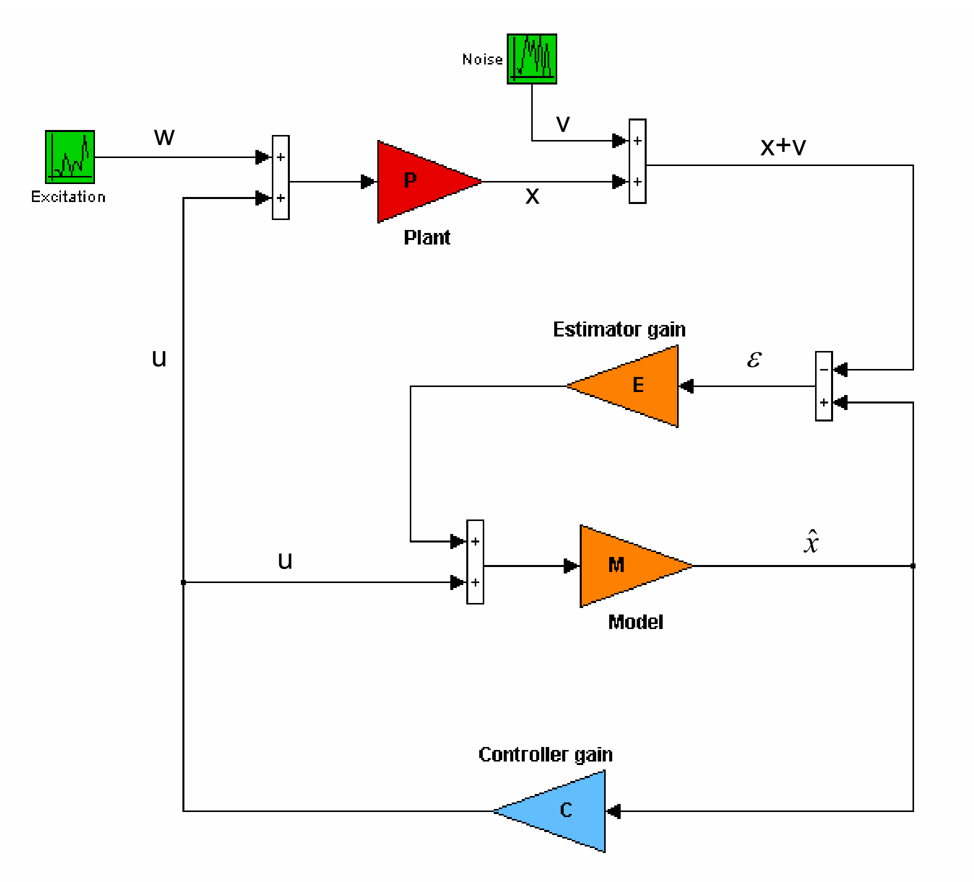


Diagram of the estimator + controller

We can now calculate the transfer functions between our 2 inputs (excitation w and sensor noise v) and the output x :

$$x = (w+u).P$$

$$x = (w + C.\hat{x}).P$$

$$\hat{x} = (u + E.\varepsilon).M$$

$$\hat{x} = (C.\hat{x} + E.(\hat{x} - x - v)).M$$

$$\hat{x} = \frac{E.M.(x+v)}{C.M + E.M - 1}$$

$$x = P.w + \frac{E.M.C.P.(x+v)}{C.M + E.M - 1}$$

$$\boxed{x = \left[\frac{C.M.P + E.M.P - P}{C.M + E.M - 1 - E.M.C.P}\right].w + \left[\frac{E.M.C.P}{C.M + E.M - 1 - E.M.C.P}\right].v}$$

We will see later that this model can be used to study the stability or the sensor noise transmission.

It is important to note that this model, while perfectly mathematically correct, can only be used in Matlab for one degree of freedom. Matlab has problems dividing transfer functions with the TF or ZPK form (it can be used for 1dof if done very carefully though). The best choice is the state-space notation that works for any kind of system. (see Appendix II)

6.3 Behavior with sensor noise

The model gives us x as a function of the seismic noise and of the sensor noise, let's now study its behavior at limits.

$E.M \to \infty$

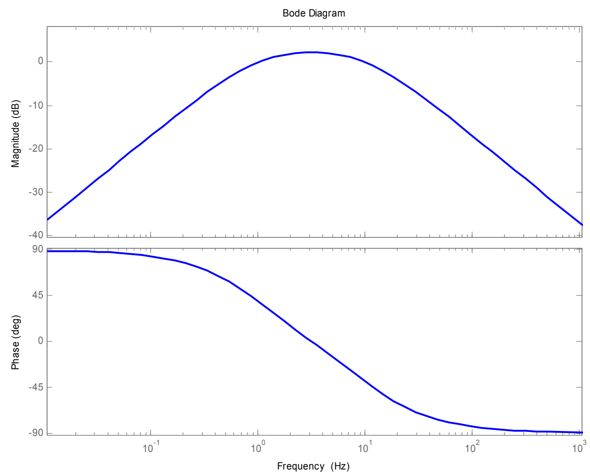
- $x = \left[\frac{P}{1 - C.P}\right].w + \left[\frac{C.P}{1 - C.P}\right].v$
- This is the behavior of a loop with no estimator

$E.M \to 0$

- $x \to \left[\frac{(C.M - 1).P}{C.M - 1}\right].w + \left[\frac{0}{C.M - 1}\right].v$
- $x \to [P].w + [0].v$

- When the estimator gain goes to 0, the loop is open and there is no more control, and no more sensor noise re-injected.

This result is very useful, it means that when M is big (at resonances), you keep a very good damping, and when M is small (high frequency), the sensor noise is filtered. You can amplify this effect with the estimator gain E, and you can have a filter that has a large gain at the resonances, and small at higher frequencies.
Let's design a simple E filter to have high gain at the resonance and lower gain at infinites :

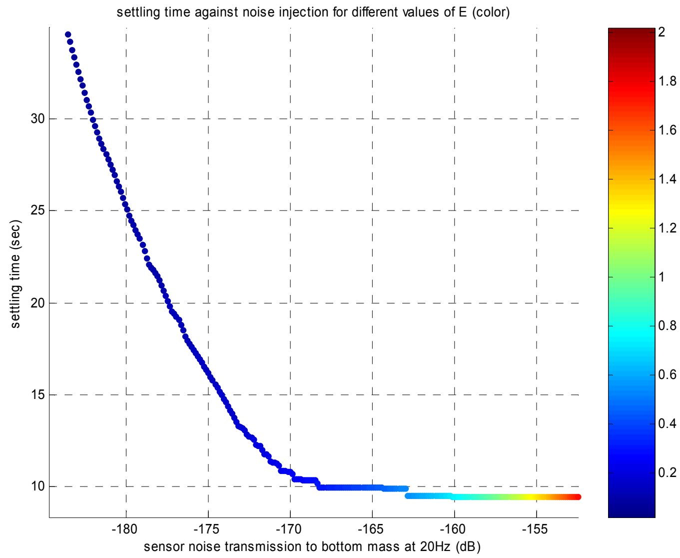


Estimator filter

The filter, like every other filter we will use in this document, are normalized so the gain equals 1 at the first resonance frequency of the plant.

We also need to choose the gain of this filter, in order to choose the best value of the estimator gain E, we can use the loop's model we showed above to create an interesting plot:

- The Y value is the settling time of x (seismic impulse to x response)
- The X value is the transmission from sensor noise v to x in dB at a given frequency (we choose 20hz)
- The color is the value of the estimator gain E



Noise transmission / settling time plot

3. *The Y value is the settling time of x (seismic impulse to x response) (would be close to infinite with no damping and infinite if the system is unstable)*
4. *The X value is the transmission from sensor noise v to x in dB at a given frequency (we choose 20hz)*
5. *The color is the value of the estimator gain E*

The plot is not continuous due to the method we use to calculate the settling time; it can sometimes "jump" from one to another period.

This plot will enable us to choose the best value for the estimator gain by optimizing sensor noise and damping..

6.4 Stability

The first important thing to keep in mind about stability is that it is not affected by sensor noise, even if it is large, it can't turn a stable control to an unstable one because the sensor noise transfer function and excitation transfer function share the same poles (see mathematics).

From the equation above, we can get the open loop transfer function so that we can use usual method to study the stability:

$$\frac{P}{1+L} = \left[\frac{C.M.P + E.M.P - P}{C.M + E.M - 1 - E.M.C.P}\right]$$

where L is the open loop transfer function (TF from w to u)

$$L = \frac{-E.M.C.P}{C.M + E.M - 1}$$

The system is unstable if L=-1 (we understand this is one criteria and that this is not the only one but of you design "normal" filters with no positive poles, you can use this one)

The entire loop is stable as long as $C.M + E.M - 1 - E.M.C.P \neq 0$

This is unfortunately difficult to solve, but if we consider that M=P (Model = Plant)

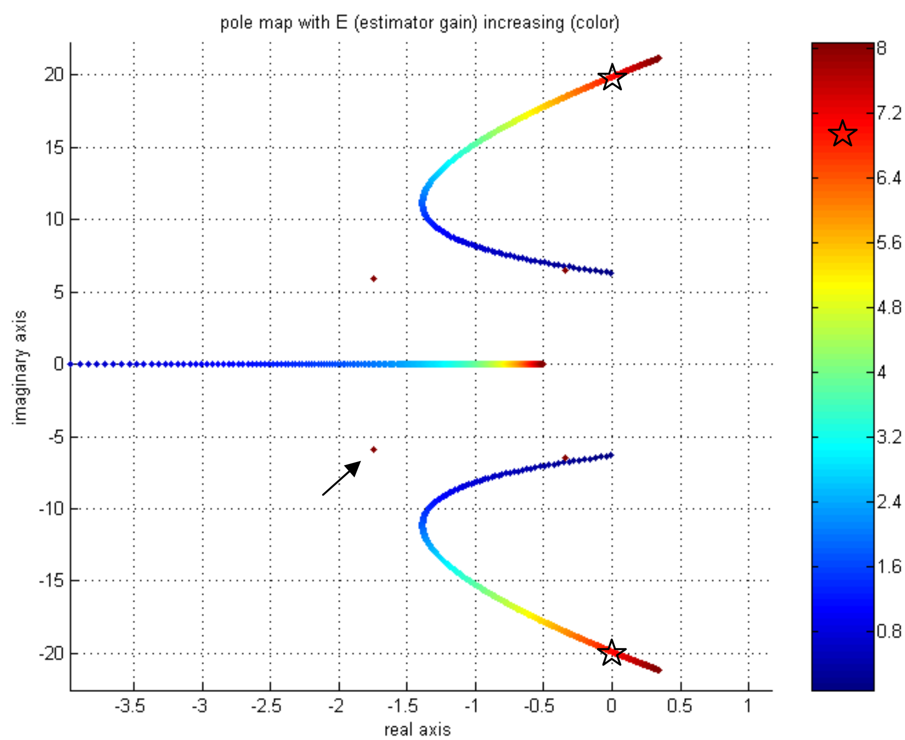
$C.[P] + E.M - 1 - E.M.C.P = (1 - E.M).(1 - C.P)$ (We just replaced one M by a P)

It is easier to solve, it means that the 2 inner loops have to be stable (loop with no estimator and internal estimator loop). We can write the 2 first stability criteria:

- $1 - E.M \neq 0$ (the inner estimator loop must be stable)
- $1 - CP \neq 0$ (the loop with no estimator must be stable)

These rules are the only one if the plant and the model are the same. However, since we want to be able to study the stability when the model and the plant are different (in case we have a mismatch), we need to find another method.

A more global approach is to study the location of the poles of the closed loop in a real/imaginary map. If every pole has a negative real part, then the loop is stable.



Pole map of the closed loop with E increasing (the color gives the value of E)

- *Plant and model are the same*
- *E shape is the one described in section 5.3*
- *Some of the poles are constant (arrow) (they don't change with E), they are the poles of the filters*
- *Poles with positive real part represent unstable loops (star)*

This plot enables us to see very quickly what is the maximum gain for E (color), in this example (1 dof) the poles start to be positive when E goes above 7.

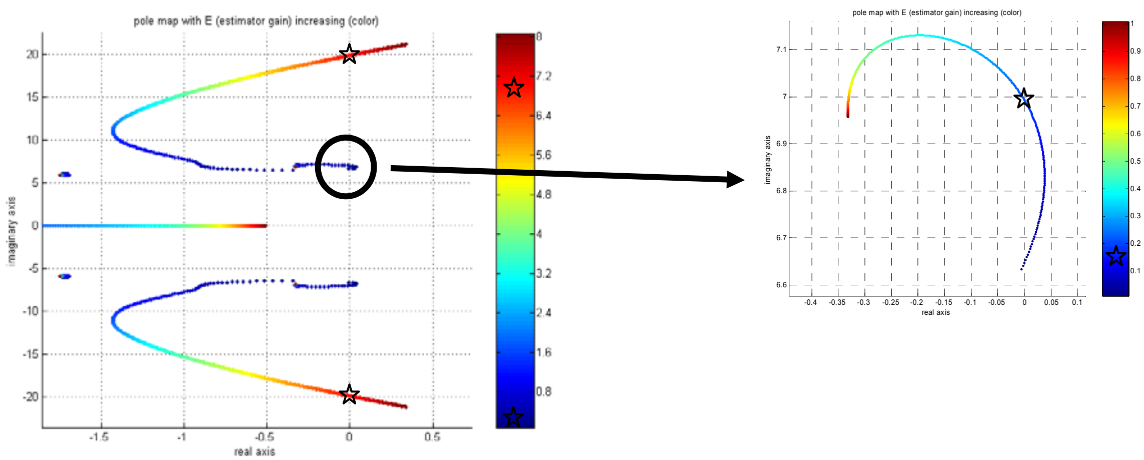
6.5 Error in the model for a 1dof system

For a single dof system, we can easily study what happens when the model is not perfect. We know (it has been checked) that the Q of the model resonances don't have to match the plant's one really well. The other mismatch you can have on the model is the frequency of the resonance and we can study what happens when the plant and model resonances do not match.

We will plot the stability plot (pole map) for 2 different cases. Since we design our loop using the model, we will use the single-dof model we used before (one resonance at 1Hz) and modify the plant.

1. Plant resonance is 10% higher than the model's one :

    - Stability :

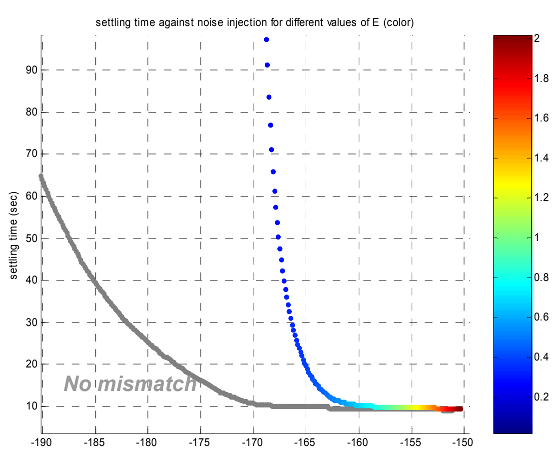


Pole map for different values of E
Plant's resonance freq is 10% higher than model's one

Zoom

The error affects the stability for low values of the estimator gain, we see that the estimator gain can't be lower than 0.3 or the loop will be unstable.

- performances :

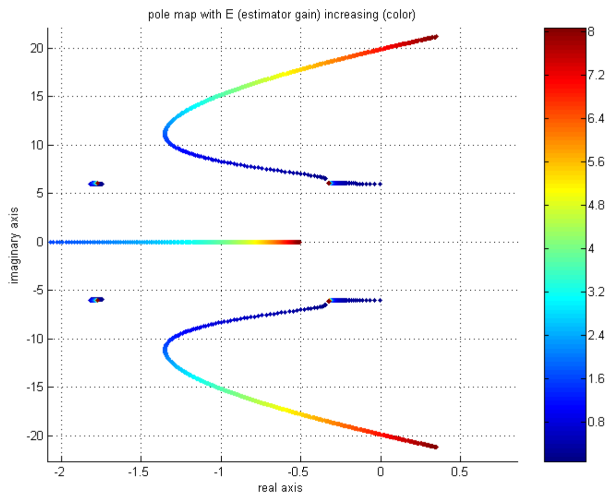


Noise/damping plot, Plant's resonance freq is 10% higher than model's one
No model mismatch case is in grey

As you can see, if you compare with the same plot we made above (see section 6.3), the noise/damping performance is not really affected (0.5 sec difference) above E=0.8. However, below this value, the performance of the loop becomes really bad. The difference between the model and the plant can't be compensated by the estimator feedback because the gain is too small, and the damping is inefficient (and like we saw above it even becomes unstable for the lowest values).

2. Plant resonance is 10% lower than the model's one

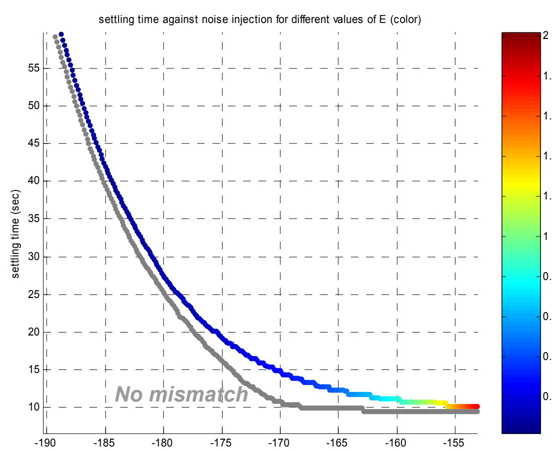
- Stability :



Pole map for different values of E
Plant's resonance freq is 10% smaller than model's one

Even if it creates new poles (the model's ones and the plant's ones are not the same anymore), a negative error doesn't make the loop unstable whatever the gain of the estimator gain is (the limit of the gain is still the same as with no error)

- performances :



Noise/damping plot, Plant's resonance freq is 10% smaller than model's one
No model mismatch case is in grey

However, the noise/damping performances are decreased by this negative error. We see that the damping is not as good as without error when E is small, it can be explained by the same reasons we saw above.

3. Conclusion

What we learnt here on a single dof system is that for both the stability and the noise/damping performance, a too low value of the estimator gain is dangerous, low value of this gain will make the loop unstable for positive error, low values will also make the damping less efficient whatever the error is. In the future we will pay attention to choose estimator gain that is not too small to avoid all those problems.

6.6 Error in the model for multi dof systems

Even though we understand the stability and performances well when we have a wrong model for a single dof system, it is very hard to generalize to a multi dof system. The main reasons for this problem are :

1. How to quantify a model mismatch for a multi dof system (all resonances off, only one, random error…) ?
2. Having the frequencies of the model match doesn't guaranty that the modal shapes match too, it adds another possible mismatch.

We didn't find a good and clear solution to those problems, we know our conclusions for 1 dof can be translated to the multi-dof case (choose a estimator gain that is not too small), but we can't quantify it and we don't know if other problem can happen (A good solution to study a multi-dof would be to use a Monte Carlo simulation, see Appendix IV , this work is in progress)

However, We know which parameters need to be good and which don't need to be good :

- The resonances frequencies (can be checked with characterization) need to be good
- The modal decomposition (not hard if model is complete) needs to be correct
- The gain (at 0Hz) needs to be good (like for every other control, and it is very easy to adjust in few seconds with a measurement)
- The Q doesn't need to be very good

We have thought about ways to improve and adapt the model so that it matches the plant better by using a method to minimize the error between the model's transfer functions and the measured transfer functions (see Appendix I). It is important to note that every time we tried the loop on the triple pendulum in LASTI (with or without a modal matching method), we never had any instability.

6.7 Conclusion

We have seen how to create a mathematical model of the loop with an estimator and modal control. This enables us to predict the damping performances, the stability and the noise performances of the loop and see how to optimize it.

We have seen that we need to choose the estimator filter so that we get higher gain at the resonances and the lowest gain where we want to filter the noise. A simple filter with 1 zero and 2 poles works well for that.

We have then studied the influence of the estimator gain on the noise/damping performances, by plotting the settling time, the sensor noise transmission and the value of E on a same plot, we can choose the best value of E.

We also have worked on stability. If the model and the plant are the same, then the stability is very easy to check. We can also check the influence of a bad model on the stability and the performances on a single dof system, we have learned that we shouldn't choose a too low value for E, but it is a bit more difficult to quantify this influence on a multi-dof system. However, we know which parameters need to match and we have thought about way to improve the model.

We have only used very simple shape for the estimator filter E, more discussion about more complicated filters can be found in the conclusion of the document.

We will now try this new loop on the MC triple pendulum in the advanced LIGO environment.

7 MODAL CONTROL AND ESTIMATOR FOR ADV. LIGO SUSPENSION

7.1 Introduction

We have seen earlier how the modal control and the estimator loop works and how to study with the damping, the stability and the sensor noise with this new kind of loop.

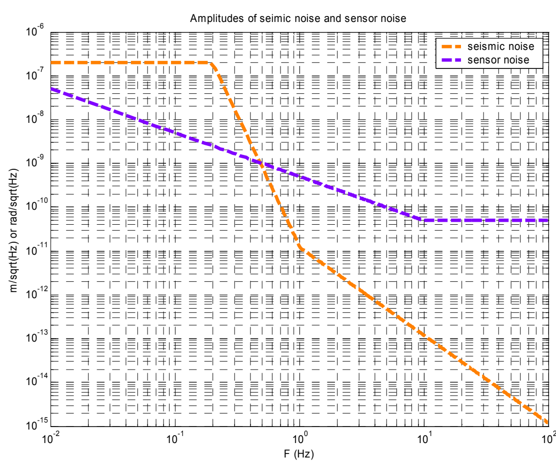
The next step, which is shown in this document, has been to use the method in order to design the mode cleaner triple pendulum loop for the advanced LIGO conditions, which means using the seismic noise expected from HEPI, the SEI and the sensor noise of the OSEMS used on the triple pendulum. We will explain in detail how to build the loop to match a 10sec damping requirement while minimizing the sensor noise re-injection in the pendulum. We will also test this loop and filters on the triple pendulum to check the loop's performances on the real plant.

7.2 Advanced LIGO environment and requirements

7.2.1 Noise inputs

This study will use the expected seismic noise level on the optical table after the 2 stages of isolation, HEPI and the internal seismic isolation. The same seismic noise level is used for every degree of freedom

The sensor noise produced by the shadow sensor has been measured and can be modeled as shown on the figure:



Seismic noise on Adv LIGO optical table
Sensor noise of the Osems

Although the seismic noise and sensor noise should be different for the angular dofs. We decided to keep the same levels too simplify the study (the main goal is not to give performances but to explain the method and compare it with classic feedback which will use the same noise inputs).

The inertial/relative sensor case will not be studied here because we want to keep this example simple. The sensors we choose are inertial. In reality you should add the sensor noise and the seismic noise to simulate the real sensor noise. But we consider this seismic part to be negligible (it is only important in low frequencies, where sensor noise is not a problem).

7.3 Goal

7.3.1 Damping

The settling time to an impulse excitation should be 10sec +/- 10%. We define the settling time as the time the bottom mass takes to come back below +/-5% of the maximum amplitude.

We use the same damping goal for the 6 degrees of freedom, as well as for the classic feedback control.

(We arbitrarily chose 10sec for the purpose of this document. Since this parameter has not be chosen yet.)

7.3.2 Sensor noise

Since the pendulum is sitting on a very quiet platform and the sensors generate a noise higher than this physical excitation at high frequency, the feedback control is going to re-inject a part of this sensor noise into the pendulum actuators and thus it will increase the noise amplitude at the bottom mass. The purpose of the control loop is to match the damping requirement while minimizing the sensor noise re-injection above 10 Hz.

The noise performances requirements for a mode cleaner triple pendulum are given below (see T010007-02) :

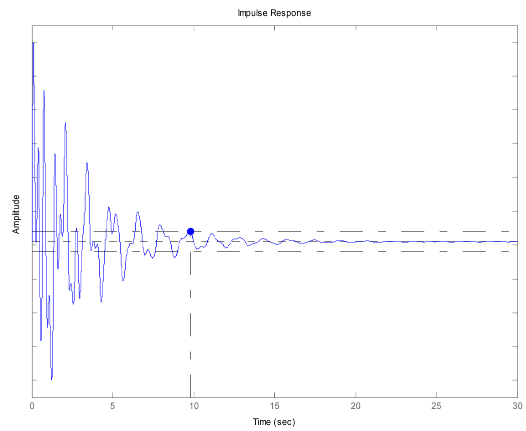
Direction	At 10 Hz $m/\sqrt{Hz}$ or $rad/\sqrt{Hz}$	At 100 Hz $m/\sqrt{Hz}$ or $rad/\sqrt{Hz}$
X	3e-17	3e-19
Pitch	3e-14	3e-16
Yaw	3e-14	3e-16
Y	3e-14	3e-15
Roll	3e-14	3e-15
Z	3e-14	3e-15

7.4 Usual feedback control strategy

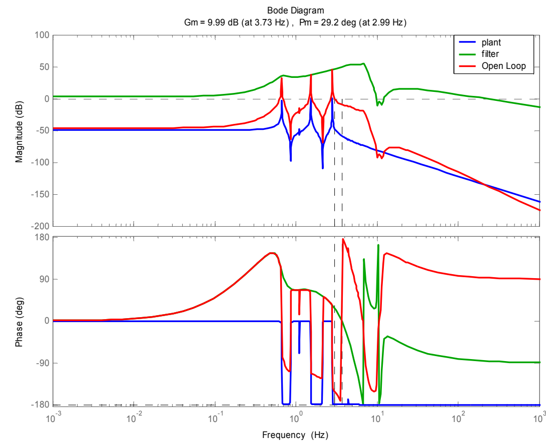
The usual strategy of control is to measure the first mass motion, filter this signal and re-inject it into the actuators of the first mass. Since the system has many degrees of freedom and modes, the design of the filters is not trivial.

The filters we will use here have been designed in Glasgow, and then modified to be as efficient as possible for the Mode Cleaner suspension. Even if there might be slightly better solutions and filters, we consider that the time spent on this design and the limitations of such a method allow us to think those filters are a good enough for our study.

Below is the example for the filter in the X direction, You'll notice the filter has been designed to minimize the noise above 10Hz while keeping a good phase margin and gain on the lowest modes to optimize the damping :



X direction, impulse response with classic feedback

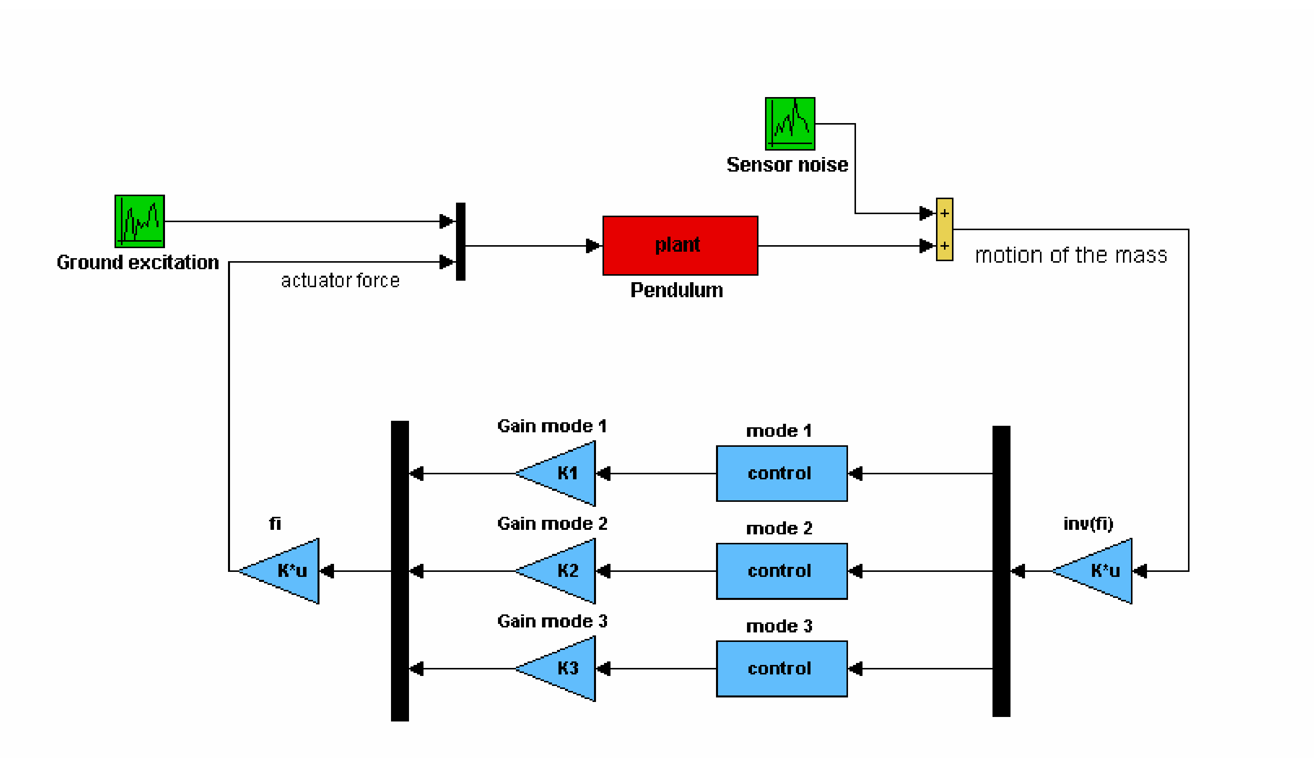


Plant, filter and open loop for the X control of a classic feedback

7.5 Designing the filters

7.5.1 Control filters

Here is a reminder of the modal control loop for a 3dof system.



Schematic of a modal control loop

As you can see, the 3 modes are controlled in different controllers, with different gains and filters. The modal control split every mode into a different “small controller”, one per mode. Those filters need to provide damping for a single dof system, which is easy to design. The filter will be parameterized with the resonance frequency of the mode so that each filter is optimized for the mode it is supposed to control.

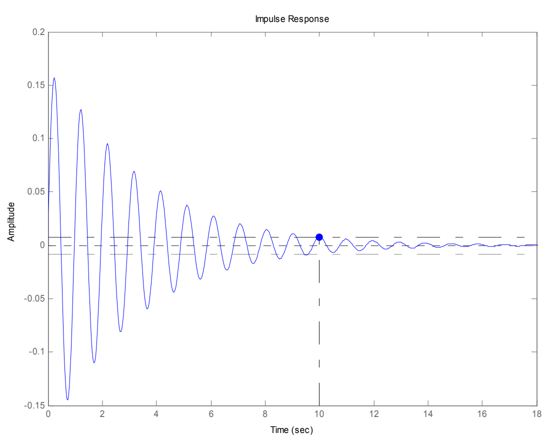
The requirements of such a filter are :

1. Large gain and good phase margin at resonance for an efficient damping
2. Good filtering after the resonance to reduce sensor noise injection

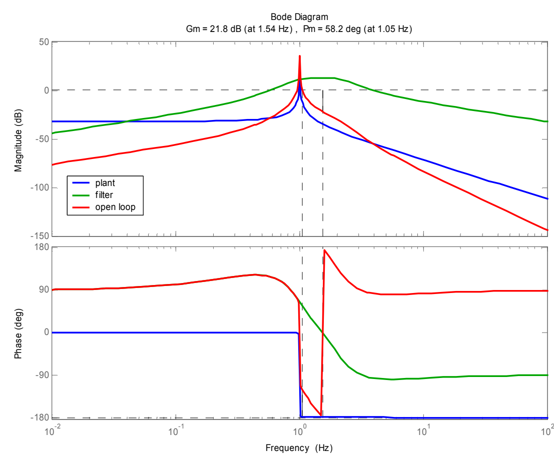
We could probably spend several hours optimizing such a control, using complex filters. However, in this document, the choice has been made to keep the filter simple and robust.

The filter we chose uses a zero at 0Hz to gain phase and reduce DC control, and then a complex pair of poles at twice the resonance frequency to filter the higher frequencies by keeping a good phase margin. A very small bump is then added on the resonance to increase the gain without affecting the phase too much.

Below is an example of this filter on a simple dof system, with a resonance at 1Hz :



Impulse response for a single mode



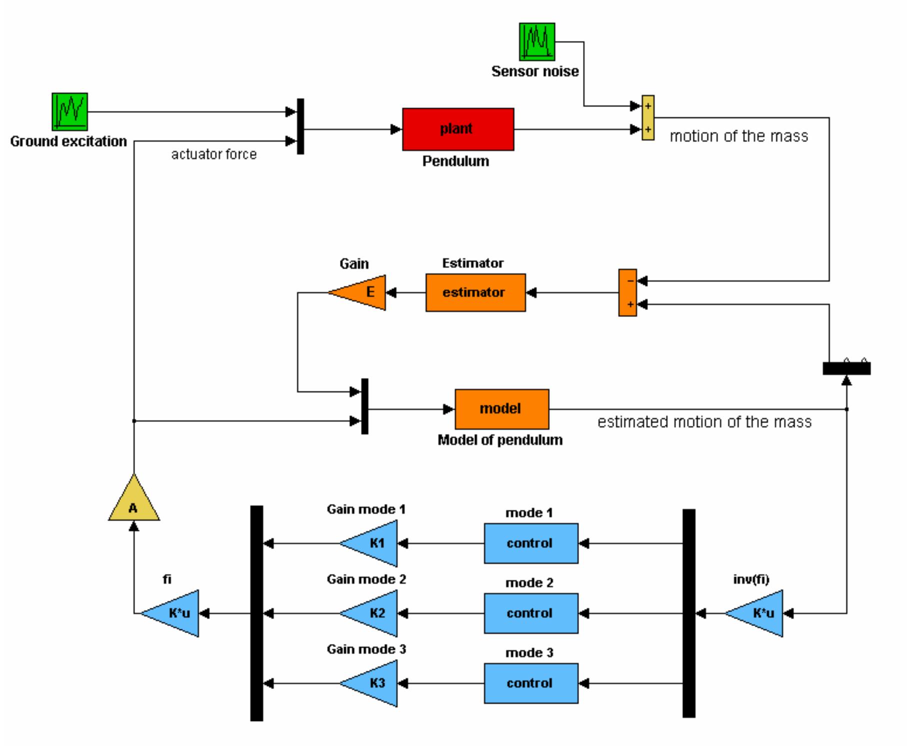
Plant, filter and open loop for a 1dof system damping

As you can see, the phase margin is large and the unstable point is far (50% more than the resonance frequency), this is important if the model doesn't match the plant.

We will use this filter for each mode, remember that the filter is parameterized so it moves with the mode frequency.

7.5.2 Estimator filters

Below is a schematic of the estimator + modal control loop. We are now focusing on the estimator filter.

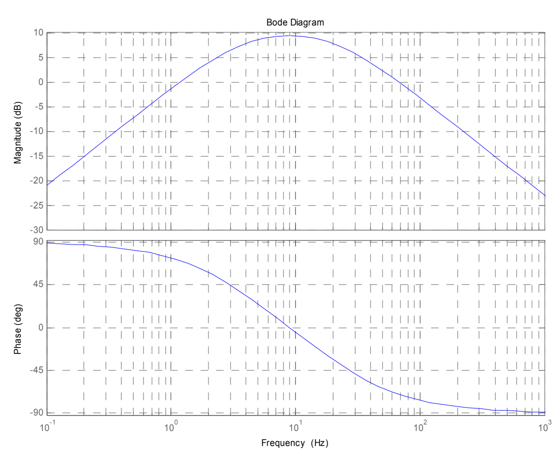


Schematic of the loop with the modal control and the estimator

- *Plant is in red*
- *Estimator is in orange*
- *Modal control (and modal state, everything else being real state) is in blue*
- *The yellow A matrix is the matrix to keep only the signal you can actuate, for example for 3 dof with only one actuator at the top it would be A=[1 0 0;0 0 0;0 0 0]*

Like we have seen before, the estimator filter needs to have a large gain where the resonances are and a small gain where we want to filter the sensor noise.

We also want to use the same filter for every degree of freedom to make the control simpler. Last, but not least, the inner loop of the estimator needs to be stable, that means that we need to have a decent phase margin at the highest resonance, the highest resonance we will have to control is 5.7 Hz, we have to choose a simple filter that gives a bit of phase at this frequency :



There is no doubt that one could design a better filter. We want a large gain at the resonances while keeping the inner loop stable and robust. However, we didn't want to make a complex filter here, because we want to show that the main advantage of this loop is to get a good performance with a very simple design.

7.6 Vertical (Z) direction, control design

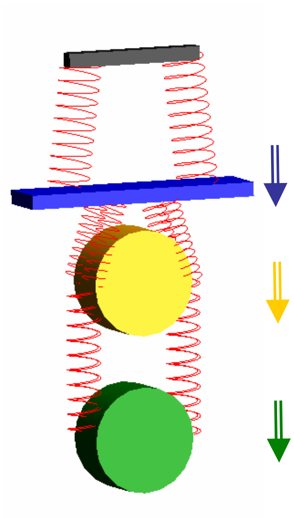
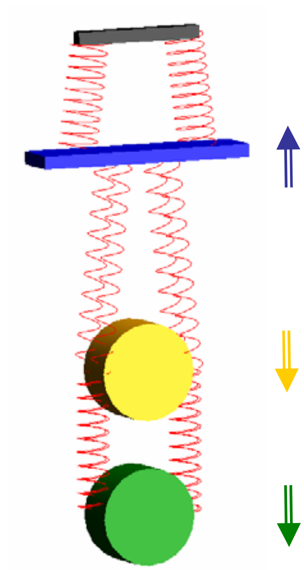
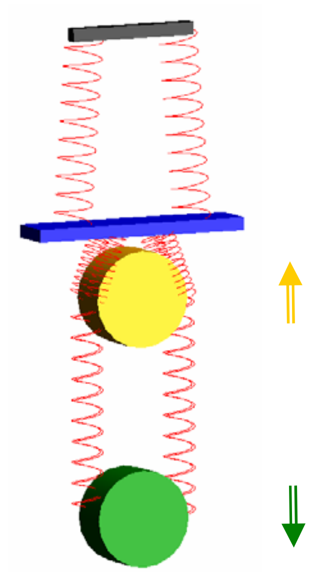
This first part shows how to design the control for the vertical direction; it will be the most detailed part in order to explain every step of the control design

7.6.1 Introduction

We only use the model of the vertical direction here, this is a 3dof plant with

- 3 outputs: m1_z, m2_z and m3_z
- 4 inputs: forces on m0-z (frame motion), m1-F_z, m2-F_z, m3-F_z

The resonance frequencies and mode shapes are shown below :

Mode1 at 1.18Hz	Mode2 at 4.25Hz	Mode3 at 46Hz
		

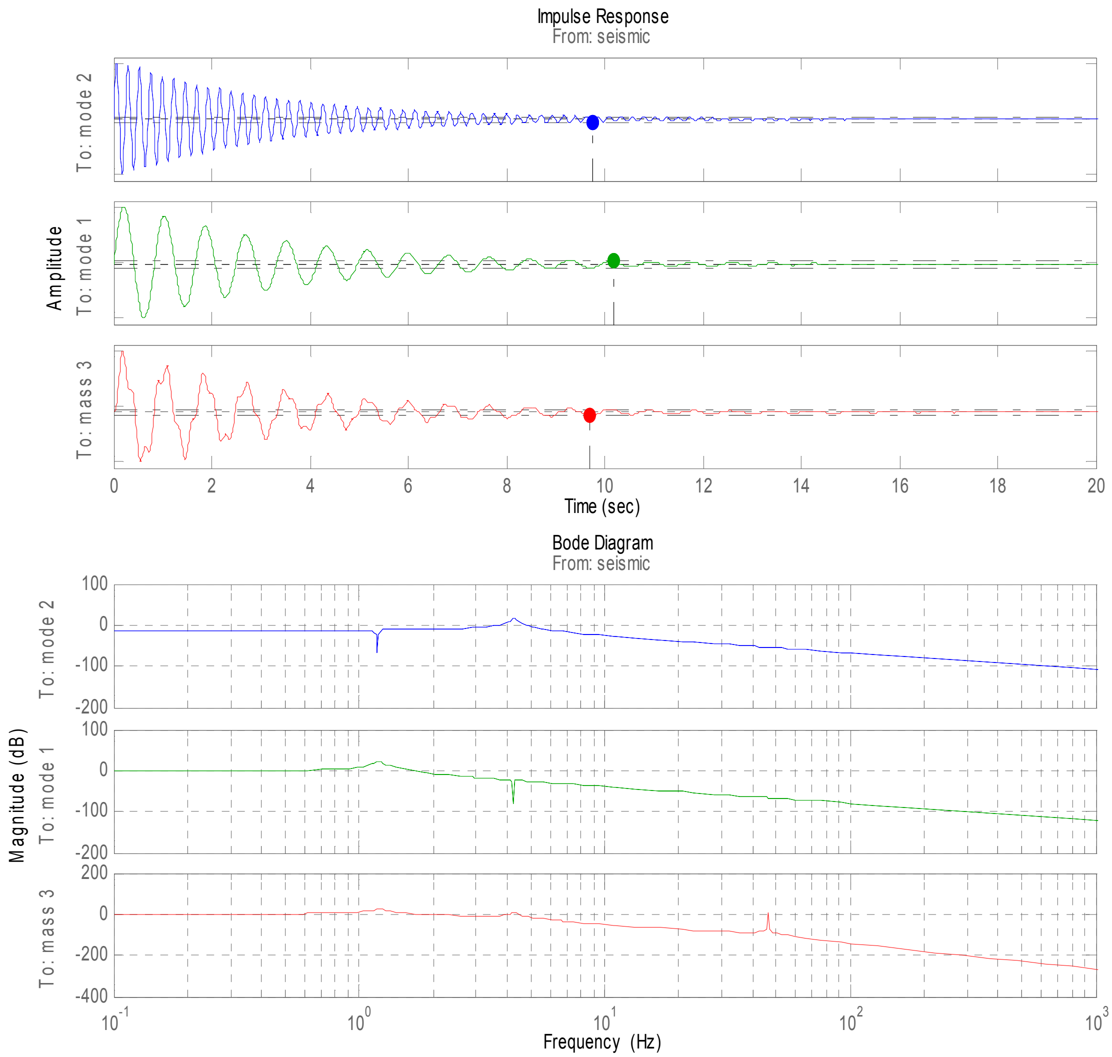
Because its frequency is very high and weakly coupled to the top mass, the 3rd mode won't be controlled. We will only focus on the first mode and second mode damping.

7.6.2 Modal control, no estimator

We have seen above how to design our filters, each controller has a filter like the one we described above which is dependant of the mode frequency. The only thing we have to do now is to turn gains up or down to match the required damping. Since we want to explain the vertical control step by step, we will first only care about the damping and we will deal with the sensor noise injection later.

We choose the 2 gains (3rd mode is not controlled) so that the settling time of each mode is 10sec. The gains are :

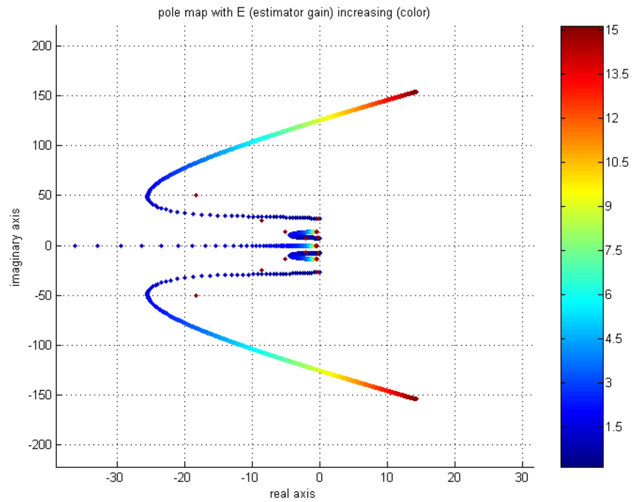
- Mode 1 : K=65
- Mode 2 : K=4.5
- Mode 3 : K=0



Z direction : Impulse response of every modes and of the bottom mass

7.6.3 Adding the estimator

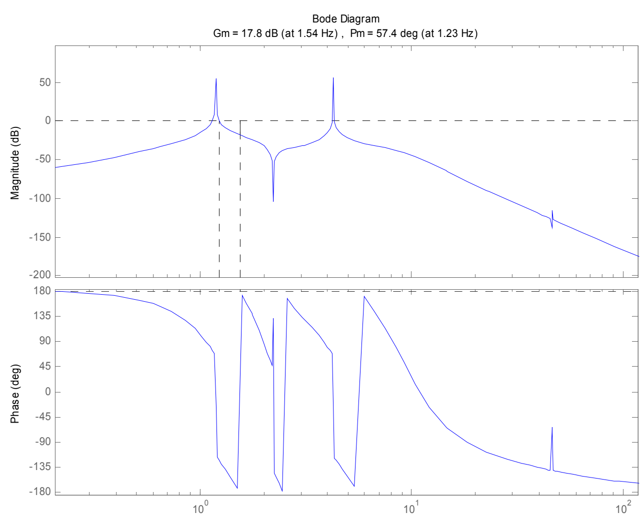
We now add the estimator (see diagram in 7.5.2) .We want to choose a gain to keep our system stable. In order to do that, we plot the poles of the closed loop for different values of the estimator gain E, poles with positive real part are unstable.



Z direction, pole map for different values of E (color)
There is one "shape" (like a loop) for each resonances, the little things we see close to 0 are other shapes like the biggest one but the biggest one is the most important one in term of stability (it is the first to go unstable)

We see that we can choose E between 0 and 9. If we choose the most stable point, then a good choice for E is E=2.

We can now check the phase margin of the open loop for the entire control loop (modal control + estimator). This transfer function below is the open loop of the vertical plant + estimator + modal control :

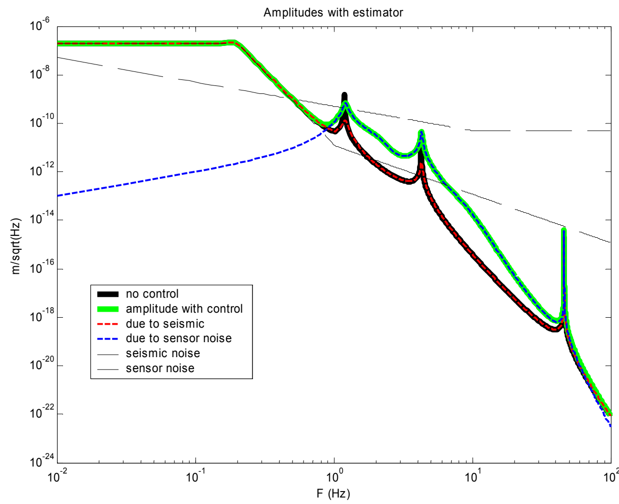


Z direction, phase margin of the open loop modal control + estimator

7.6.4 Analyzing the result

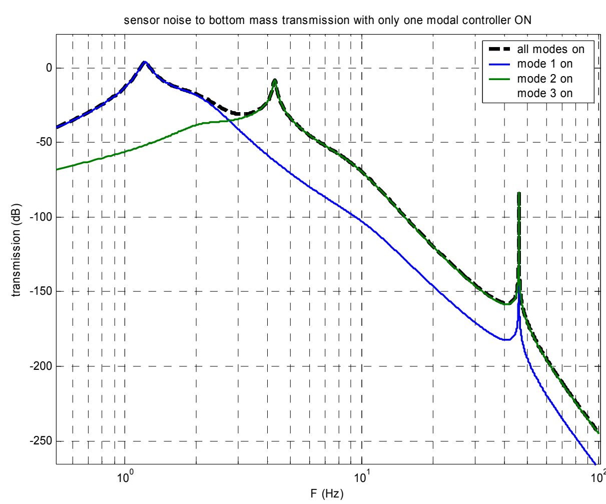
Now that our loop has been done, we need to analyze its performances and see if we can improve it, especially by reducing the sensor noise injection.

The first way to check the result is to plot the amplitude of the bottom mass motion. We can check the influence of the sensor noise injection at high frequency:



Z direction, amplitudes of seismic noise and sensor noise
*Amplitude of bottom mass motion due to seismic (TF*seismic amplitude) in red*
*Amplitude of bottom mass motion due to sensor noise (TF*sensor noise amplitude) in blue*
Total amplitude of bottom mass (RMS of blue and red) in green

This is definitely not the best we can do, the sensor noise brings a lot of noise between 8Hz and 45Hz. We can check how each mode contributes to the noise re-injection :



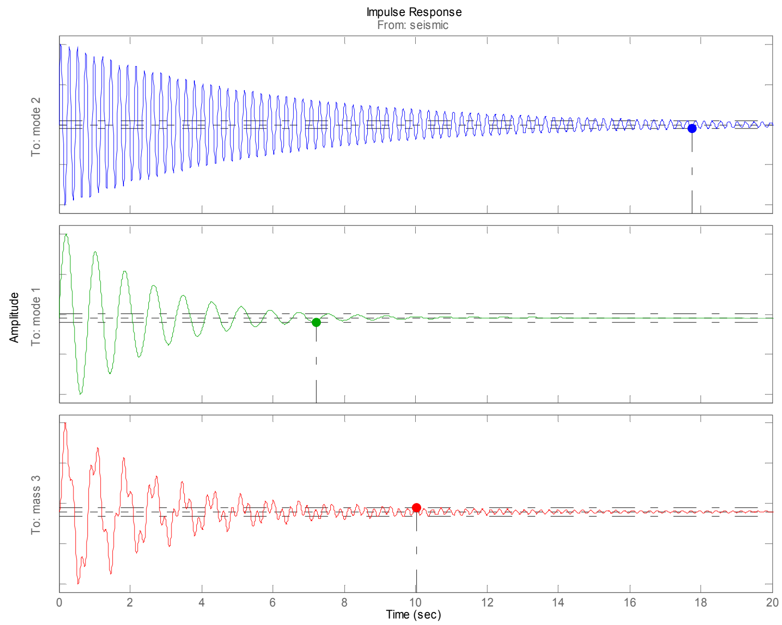
Z direction, participation of each modes in the sensor noise transmission

We can easily see that the first mode doesn't carry a lot of noise; the second mode however, produces about 10 times more noise, the third modes brings nothing since the gain is 0. To improve our control, the first thing to do is to balance the gains of the different modes

We will change our strategy and apply more gain on the first mode while we reduce the gain on the second one.
The new gains we choose are :

- Mode 1 : K=80
- Mode 2 : K=3
- Mode 3 : K=0

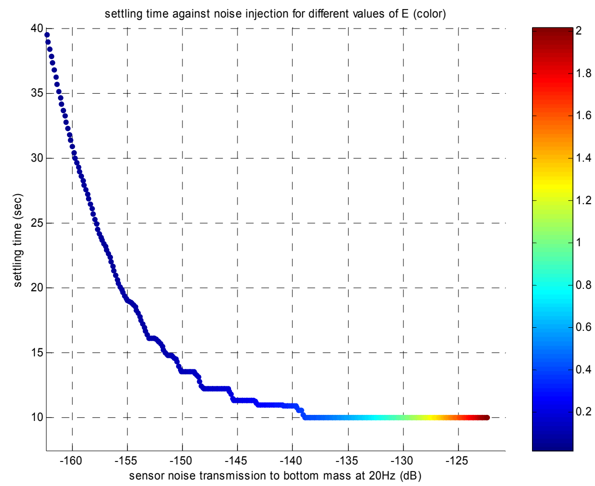
This gives the following damping:



Z direction : Impulse response of every modes and of the bottom mass with the new gains

We see that the bottom mass damping still match the 10sec, but this time we have more gain on mode 1 and less on mode 2.

The last step to reduce the noise injection is to play with the estimator gain. We can for example plot the settling time against the noise injection for different values of the estimator gain :

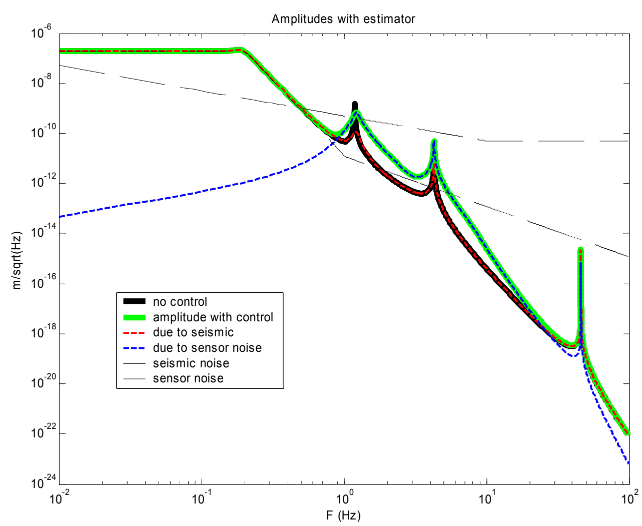


Z direction, settling time and sensor noise transmission at 20Hz for different values of E

Looking at this plot, we see that a value of 0.8 is a good choice to reduce the sensor noise transmission without affecting the damping. (Also remember the conclusion we had about stability before, a too small value for E can decrease the performances if the model is not perfect).

- New value of the estimator gain : E=0.8

Now that we have chosen our new values, we can plot the amplitude again and see the improvement, the noise at 10 Hz has been reduced by a factor of 3 since our last result

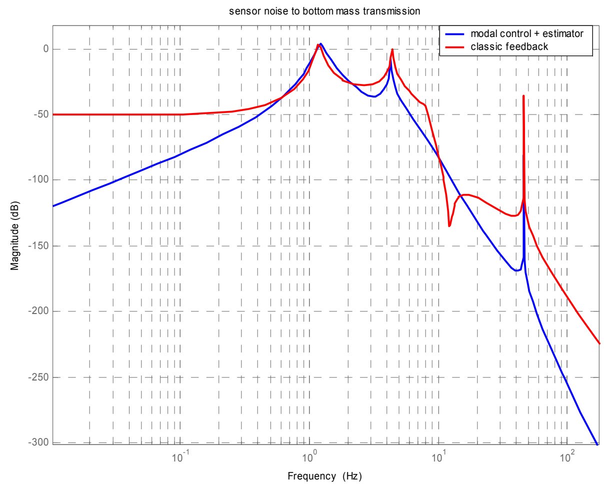


Z direction, amplitudes with the new gains

7.6.5 Comparison with classic feedback

The result is pretty good, the sensor noise becomes negligible above 30 Hz and we easily meet the noise requirement at 10Hz.

It is now the time to compare the sensor noise transmission to the bottom mass for a classic feedback and for our modal control:



Z direction, comparison of sensor noise to bottom mass transmission

We can see the modal control is more efficient everywhere except between 10Hz and 15Hz, above 15Hz, the difference is big and increases with the frequency. At 20 Hz, the modal control provides about 20dB less transmission than the classic feedback.

In case we would like to have better performance at 10Hz, it would actually be really easy to change the filter used for each mode by allowing more transmission in low frequency and designing a hole at 10 Hz. We didn't make this choice here to keep our system simple (see Appendix III for better results with more aggressive filters). The modal control + estimator performance doesn't come from a tedious work on filter tweaking, but from the method itself.

7.7 *Yaw control design*

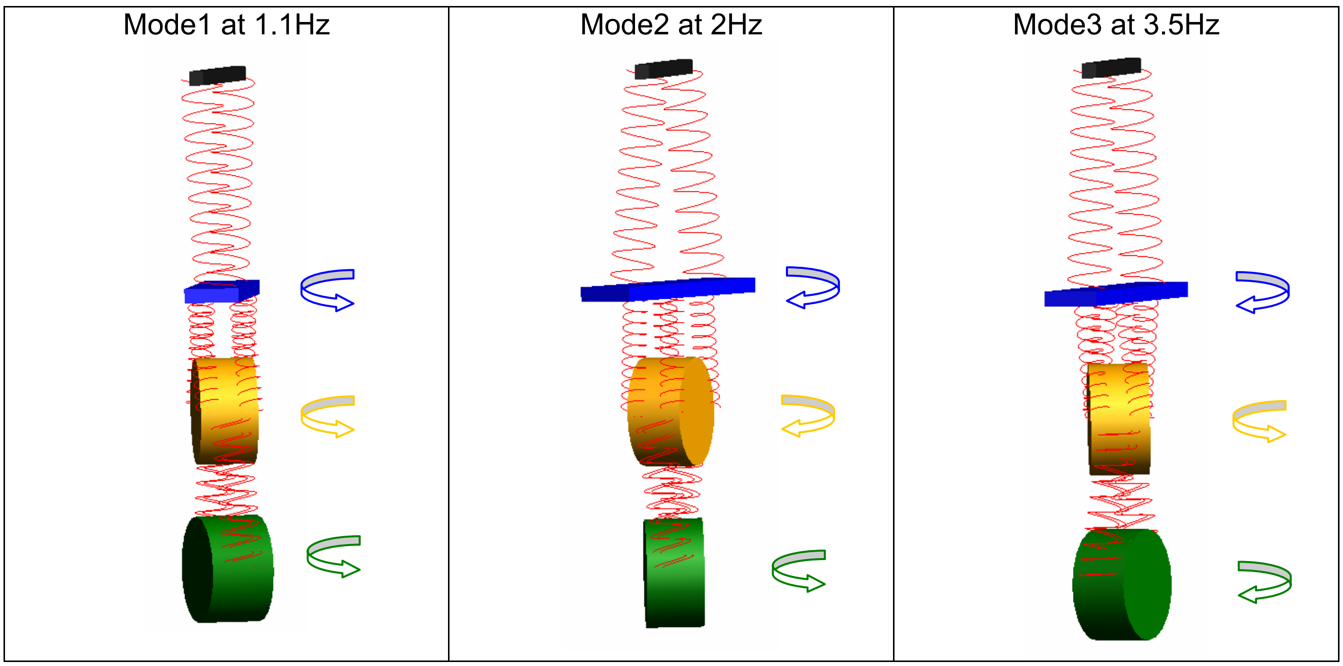
We won't explain the design of the yaw control in detail since it is the same method we used for the vertical control. We will only show the important plots and variables:

7.7.1 Introduction

We only use the model of the yaw direction here; this is a 3dof plant with

- 3 outputs m1_yaw, m2_ yaw and m3_ yaw and
- 4 inputs forces on m0-T_{yaw} (frame motion), m1-T_{yaw}, m2-T_{yaw}, m3-T_{yaw}

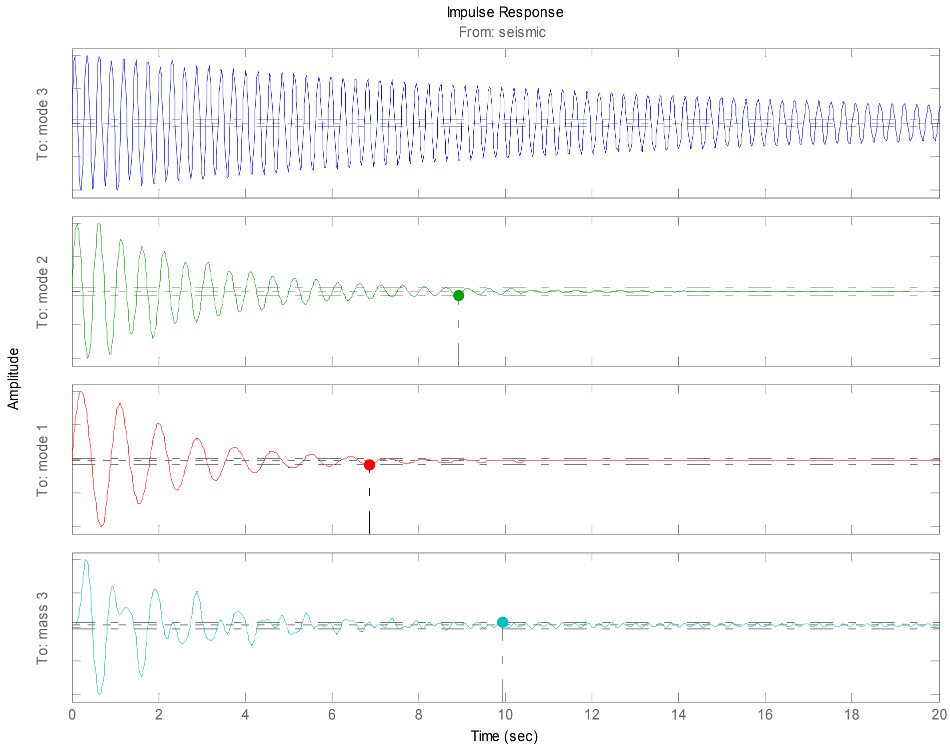
The resonance frequencies and mode shapes are shown below :



7.7.2 Gains of the modal control

We know now that we can increase gain on the lowest modes and decrease the one of the highest mode to get the best efficiency.

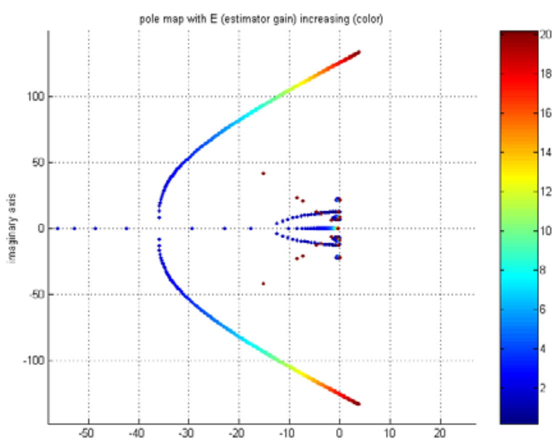
- Mode1 : K1=4e-1
- Mode2 : K2=6e-2
- Mode3 : K3=4e-2



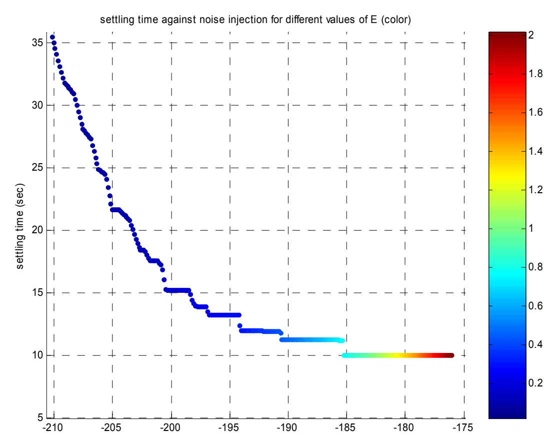
Yaw direction, impulse response of every modes and of the bottom mass

7.7.3 Adding the estimator

We then add the estimator and check the stability and the damping/noise plot for different values of the estimator gain:



yaw direction : pole map for different values of the estimator gain E



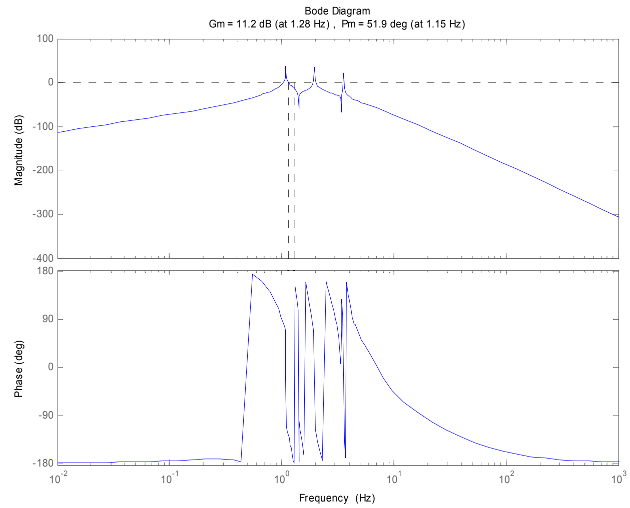
yaw direction, settling time and sensor noise transmission at 20Hz for different values of E

Looking at these 2 plots, we choose

- E=1

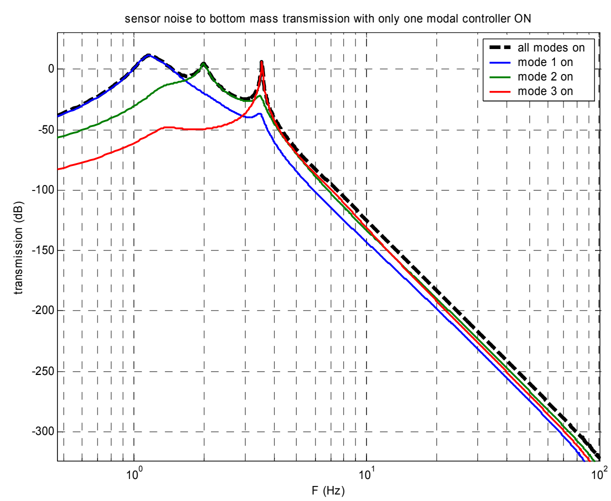
7.7.4 Checking

We can now check the phase margin stability of the whole loop :



yaw direction, phase margin of the open loop modal control + estimator

And we can also plot the influence of each modes on sensor noise to bottom mass transmission :

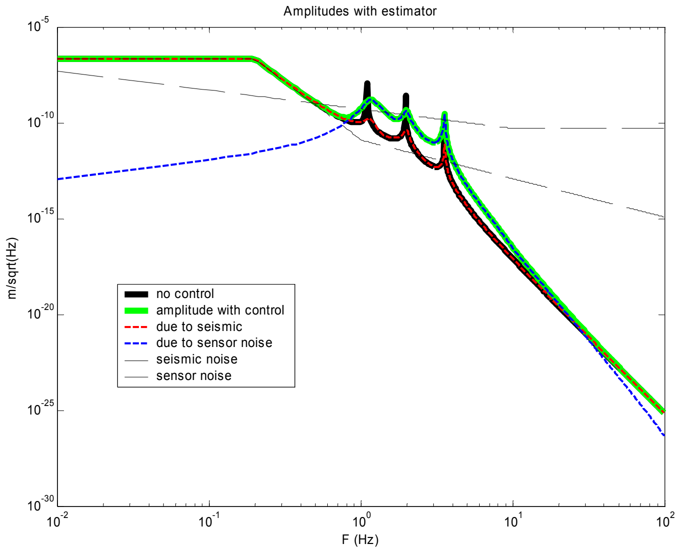


Yaw direction, participation of each modes in the sensor noise transmission

As we can see, we can't really balance the mode's gains better, since every modal controller transmits about the same amount of sensor noise.

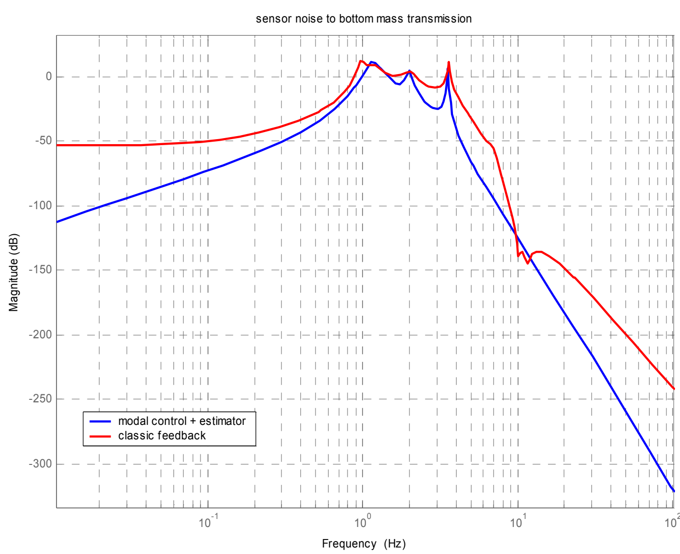
7.7.5 Results

We can now check the result by plotting the amplitude of the bottom mass yaw motion in advanced LIGO environment :



yaw direction, amplitudes

And compare the sensor noise transmission with the classic feedback :



yaw direction, comparison of the sensor noise to bottom mass transmission

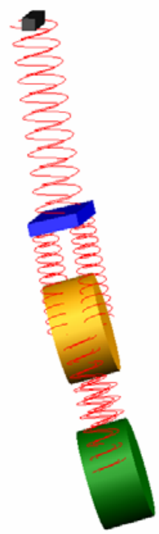
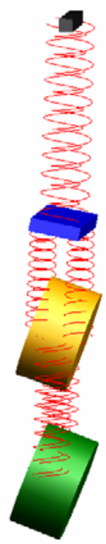
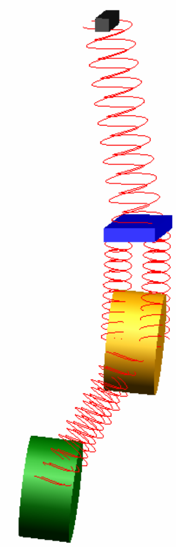
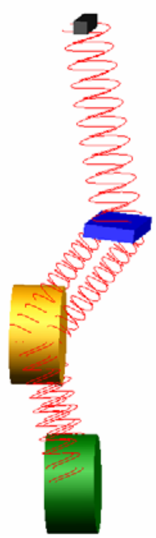
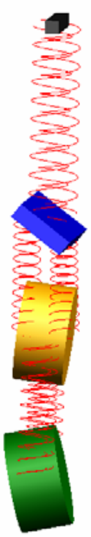
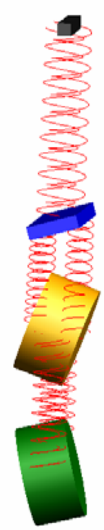
7.8 Longitudinal (X) and pitch control

7.8.1 Introduction

We use the model of the longitudinal/pitch direction here, this is a 6dof plant with

- 6 outputs : m1_x, m2_x and m3_x, m1_pitch, m2_pitch and m3_pitch
- 8 inputs : forces on m0-x and m0-pitch (frame motion), m1-F_x, m2-F_x and m3-F_x, m1-T_{pitch}, m3-T_{pitch} and m3-T_{pitch}

The resonance frequencies and mode shapes are shown below :

Mode1 (X) at 0.65Hz	Mode2 (pitch) at 1.15Hz	Mode3 (X) at 1.5Hz
		
Mode4 (X) at 2.8Hz	Mode5 (pitch) at 4.1Hz	Mode6 (pitch) at 5.7Hz
		

Because the X and pitch direction are strongly coupled to each other (for example you see the first mode has a lot of pitch in it). The modal decomposition is not orthogonal and some of the X and pitch modes are very coupled.

The transformation from real basis to modal basis is still good and coupling is not bad at this point (it just means we will use X motion **and** pitch motion to generates one mode)

However, the transformation from modal to basis (to apply the force) is not good. For example, If we control mode1, because the directions are coupled, we will actuate on both pitch and X. Even if it is not wrong mathematically speaking, we would prefer to avoid that to make the control simpler and to keep X and pitch controls separated.

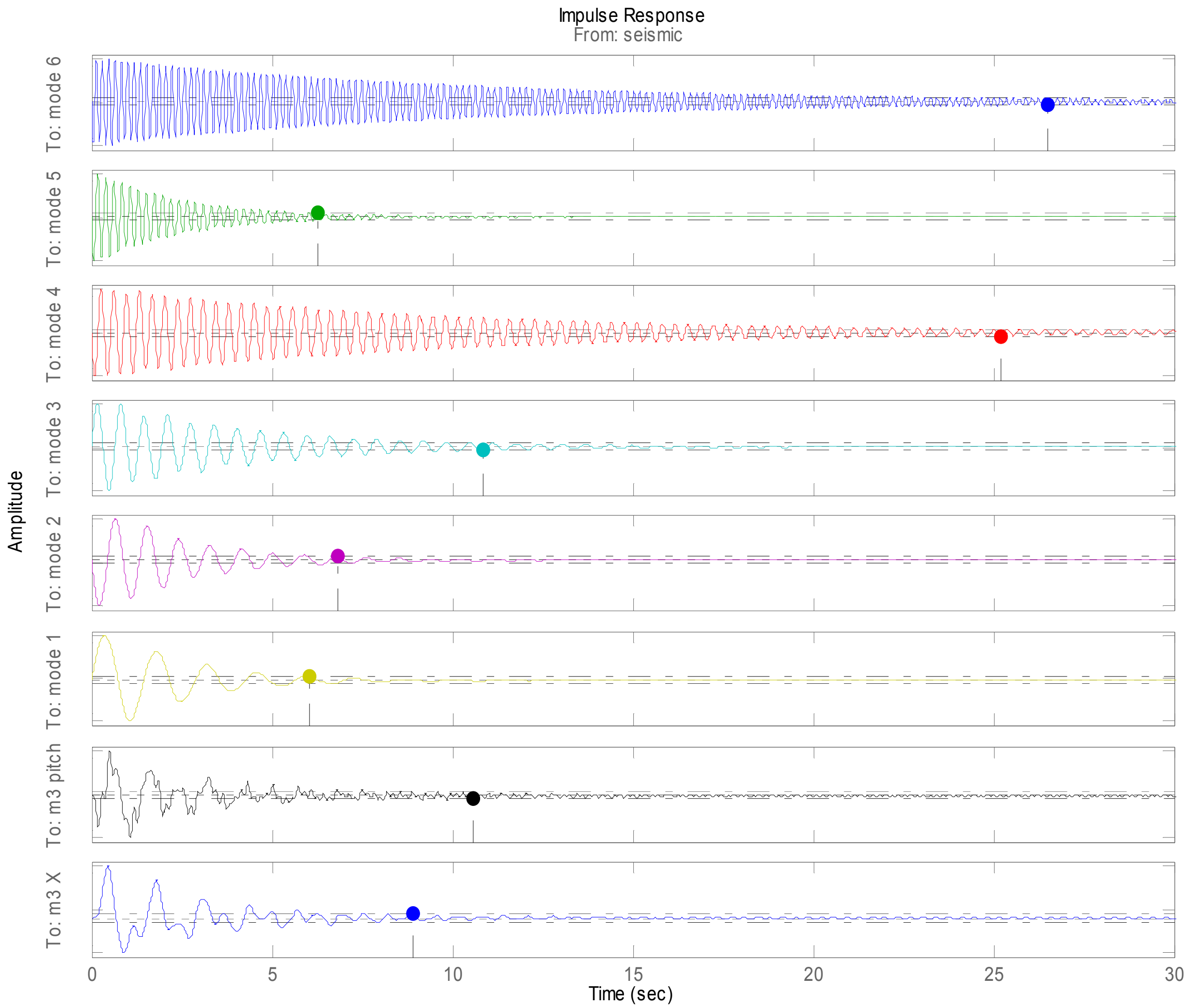
We have decided to manually force the modes actuation to be decoupled, it means that we forbid the pitch actuator to control the X modes (modes 1, 3 & 4) (and same for X actuator with pitch modes). It is a very simple change and provides excellent results as well as a simpler system to design. This is also what have always been done in classic feedback methods (where the 6 dofs are considered independent even if they are physically not)

7.8.2 Modal control gains

We want to have a large gain for the lowest modes and a small gain for the highest modes in order to optimize the ratio damping/noise. We choose the following gains :

- Mode1 : K1=120
- Mode2 : K2=8e-2
- Mode3 : K3=6
- Mode4 : K4=1
- Mode5 : K5=2e-3
- Mode6 : K6=2e-3

Here is are the impulse result for a impulse on M0_X and M0_pitch (at the same time)

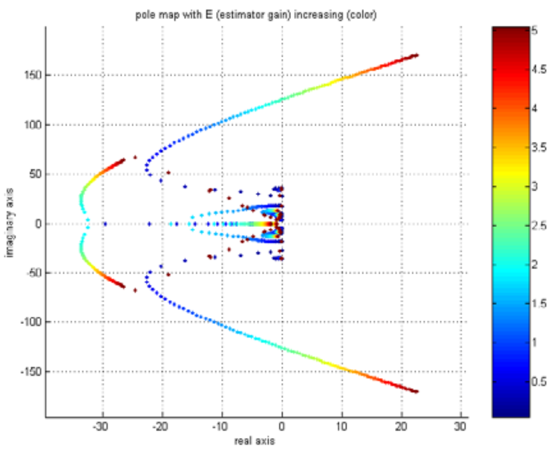


X/pitch direction, impulse response of every modes and of mass3 pitch and X

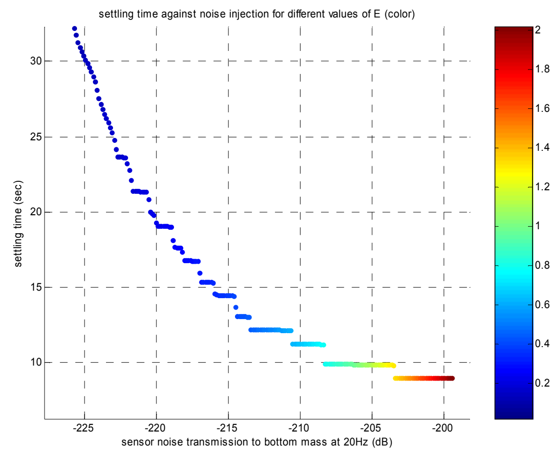
You'll notice that the damping on mode5 is very quick and might wonder why we didn't choose to decrease the gain. The damping on this mode is really efficient because this mode is strongly coupled to the mass1 (see pictures above) and thus the damping will be very strong. Another reason is that we want to keep K5 and K6 the same for safety reasons. Those 2 modes almost have the same frequency, if we have a plant-model mismatch; it is very likely that one of the modes will drive the other one and vice-versa. Giving those modes the same gain reduces the risks of unstability in case of model mismatch (It is like we control both modes together).

7.8.3 Adding the estimator

The estimator gain has the same effect on the settling time/noise plot for X and pitch, there is no need to plot both. Below is the pole map to check the stability and the damping/noise plot to choose the optimal value of the estimator gain E.



X/pitch direction, pole map for different values of the estimator gain E

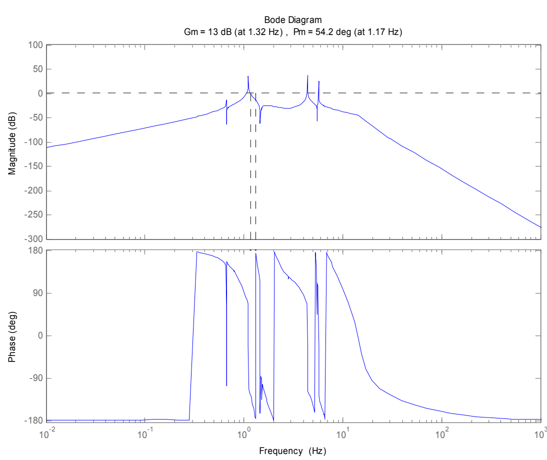


X direction, settling time and sensor noise transmission at 20Hz for different values of E

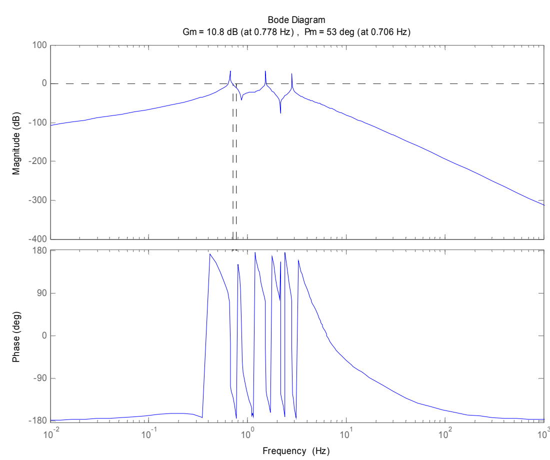
- We choose E=1

7.8.4 Checking

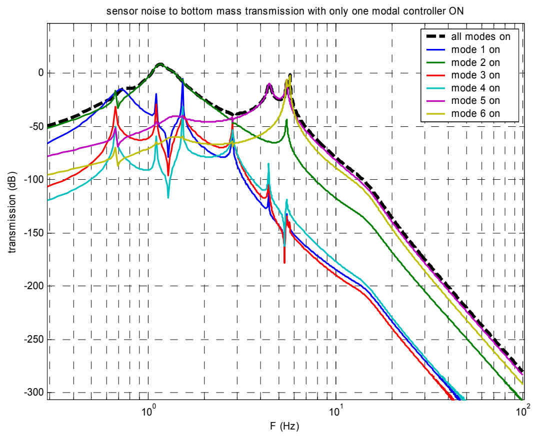
We can now check the phase margin and the influence of every mode on the sensor noise transmission:



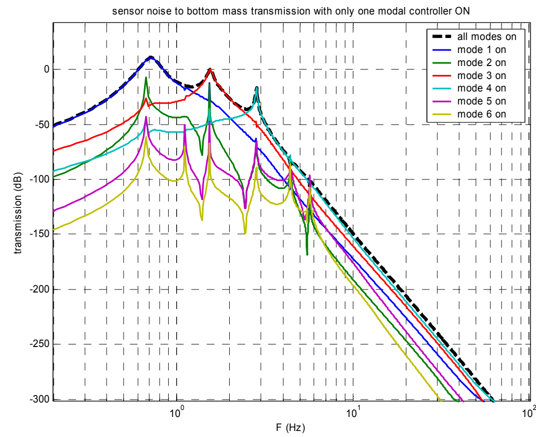
Pitch direction, phase margin of the open loop



X direction, phase margin of the open loop

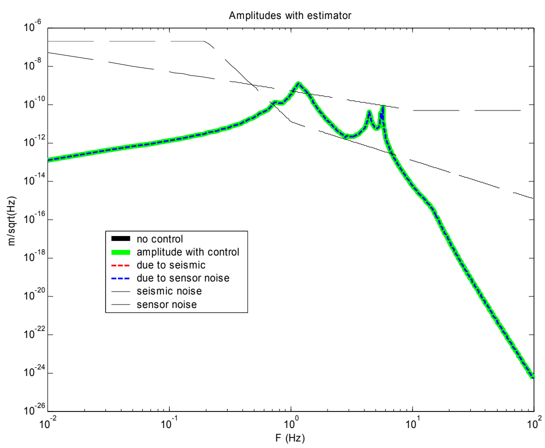


Pitch direction, participation of each mode in the sensor noise transmission



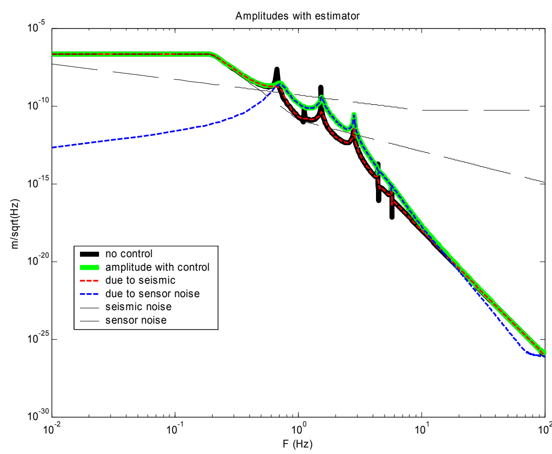
X direction, participation of each mode in the sensor noise transmission

7.8.5 results

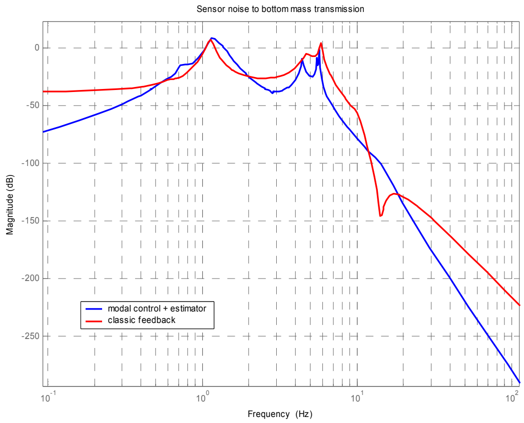


Pitch direction : amplitudes with seismic noise in Pitch direction

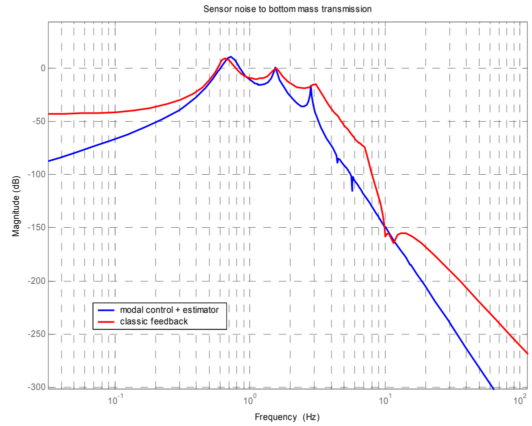
When you drive the frame in pitch, the pendulum doesn't move at all (due to the design). The only excitation is the sensor noise



X direction : amplitudes with seismic noise in X direction



pitch direction, comparison of sensor noise to bottom mass transmission



X direction, comparison of sensor noise to bottom mass transmission

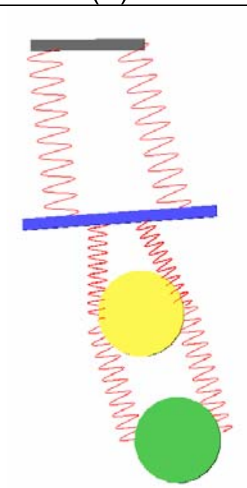
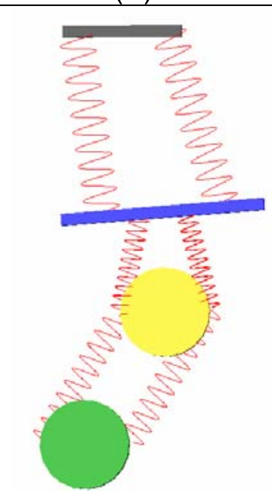
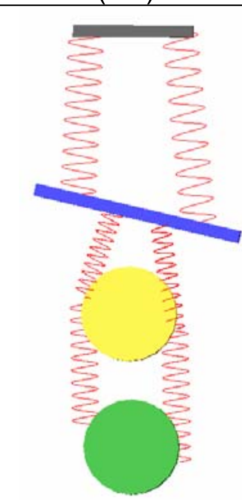
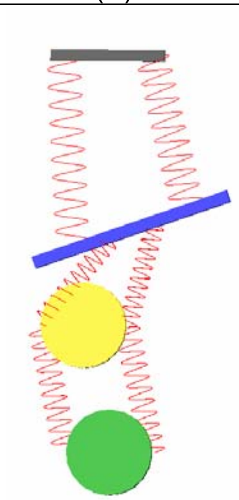
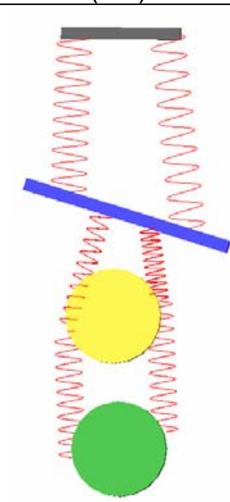
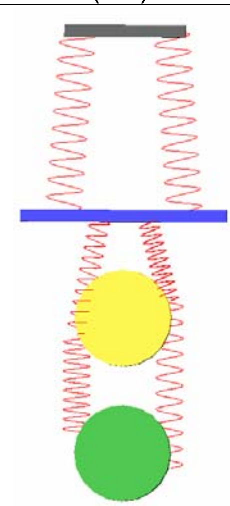
7.9 Transverse (Y) and roll control

7.9.1 Introduction

We use the model of the transverse/roll direction here, this is a 6dof plant with

- 6 outputs m1_y, m2_y and m3_y, m1_roll, m2_ roll and m3_ roll
- 8 inputs forces on m0-y and m0-roll (frame motion), m1-F_y, m2-F_y and m3-F_y, m1-T_{roll}, m2-T_{roll} and m3-T_{roll}

The resonance frequencies and mode shapes are shown below:

Mode1 (Y) at 0.67Hz	Mode2 (Y) at 1.5Hz	Mode3 (roll) at 2.2Hz
		
Mode4 (Y) at 2.7Hz	Mode5 (roll) at 3.6Hz	Mode6 (roll) at 50Hz
		

Once again, the modes and direction are not perfectly decoupled, we will use the same method we used in the longitudinal and pitch control to reduce the coupling between the 2 directions.

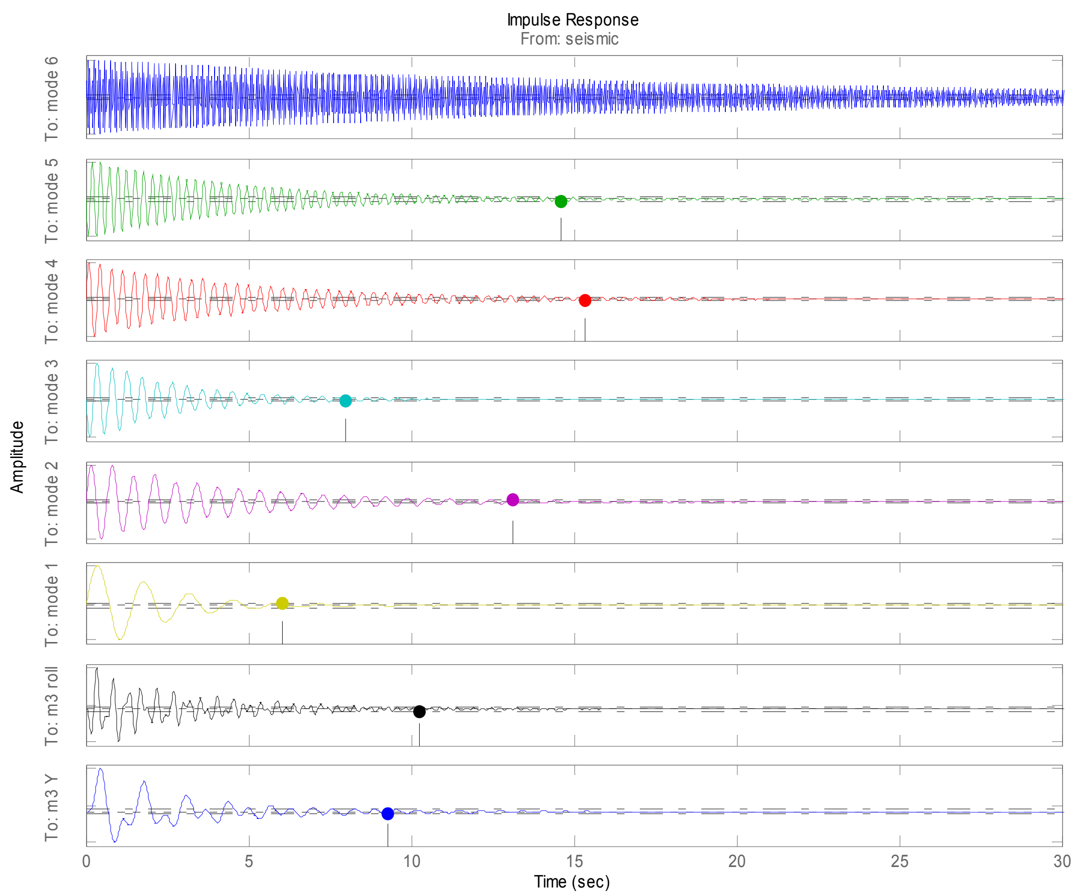
Like for the vertical control, the 6th mode won't be controlled because it has a very high frequency and it is not coupled to the first mass.

7.9.2 Modal control gains

We want to have a large gain for the lowest modes and a small gain on the highest modes in order to optimize the ratio damping/noise. We choose the following gains :

- Mode1 : K1=120
- Mode2 : K2=5
- Mode3 : K3=5e-2
- Mode4 : K4=2
- Mode5 : K5=1.5e-2
- Mode6 : K6=0

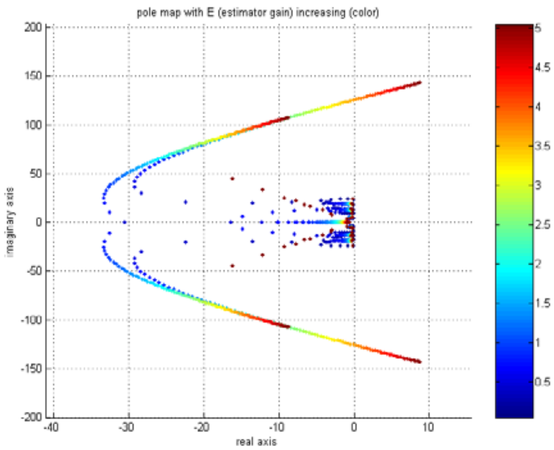
Here is are the impulse result for a impulse on M0_Y and M0_roll (at the same time)



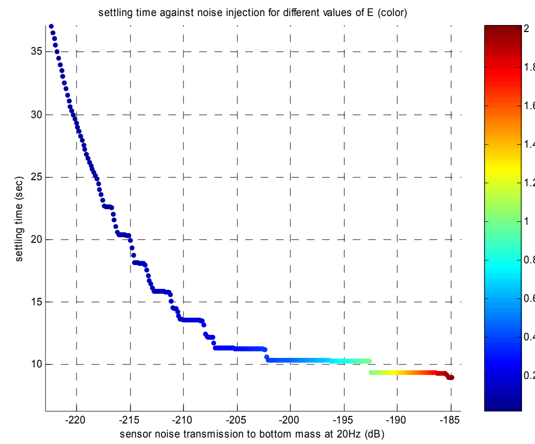
Y/roll direction, impulse response for every gain and for mass3 Y and roll

7.9.3 Adding the estimator

Below is the pole map to check the stability and the damping/noise plot to choose the optimal value of the estimator gain E.



Y/roll direction, pole map for different values of E

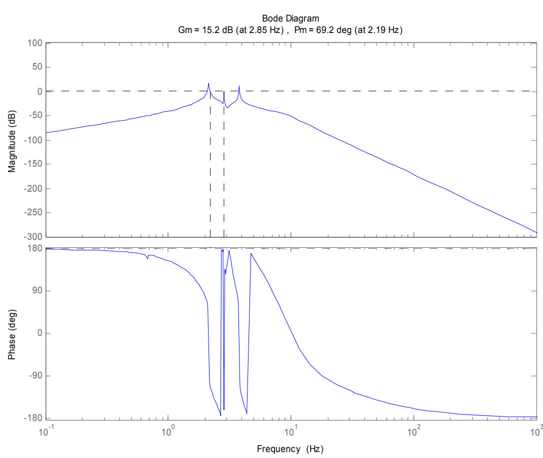


Y direction, settling time and sensor noise transmission at 20Hz for different values of E

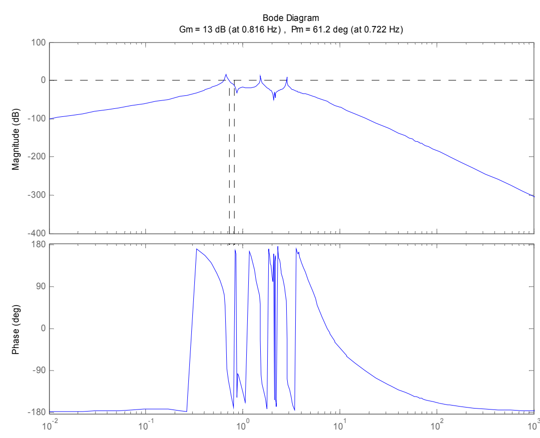
- We choose E=0.8

7.9.4 Checking

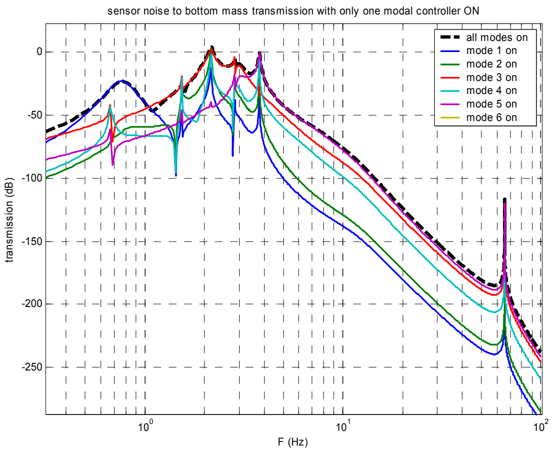
We can now check the phase margin and the influence of every mode on the sensor noise transmission:



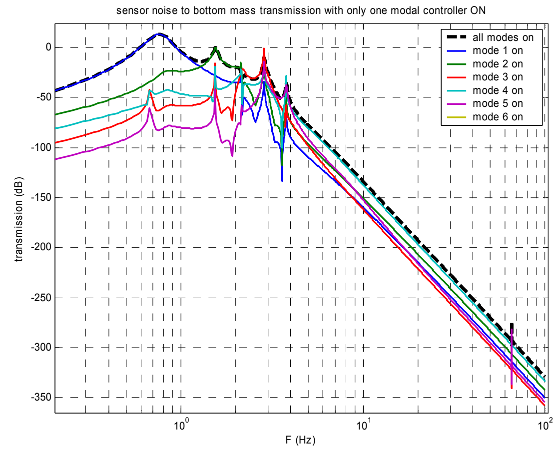
Roll direction, phase margin of the open loop



Y direction, phase margin of the open loop

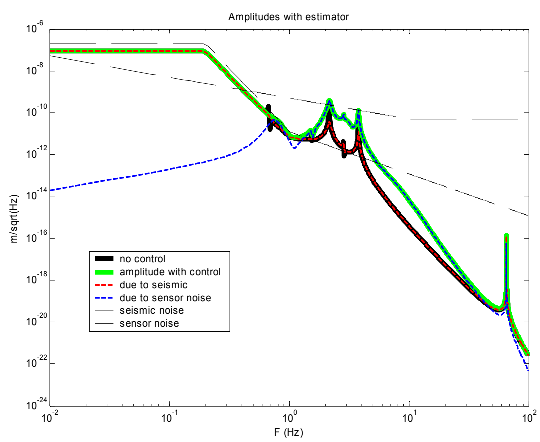


Roll direction, participation of each mode in the sensor noise transmission

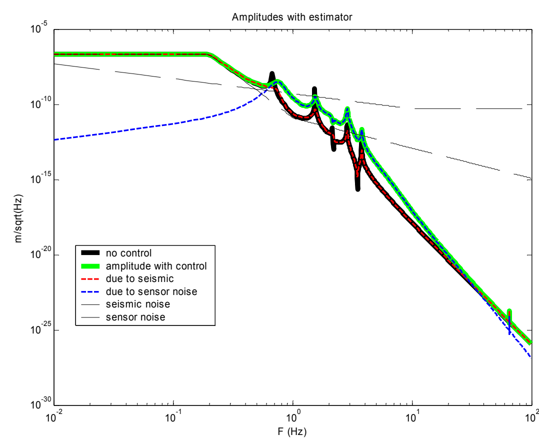


X direction, participation of each mode in the sensor noise transmission

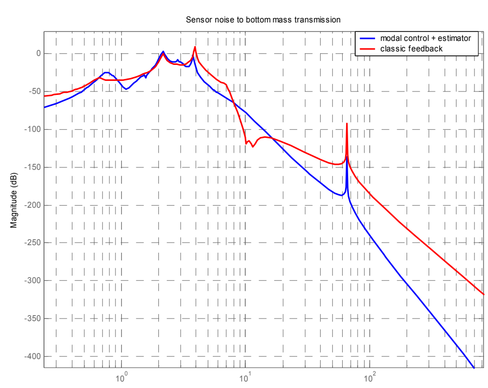
7.9.5 results



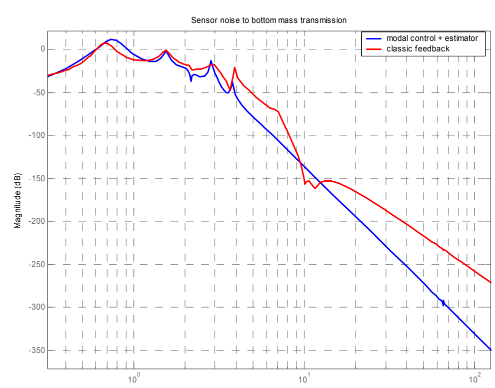
Roll direction, amplitudes with Seismic noise in roll direction



Y direction, amplitudes with seismic noise in Y direction



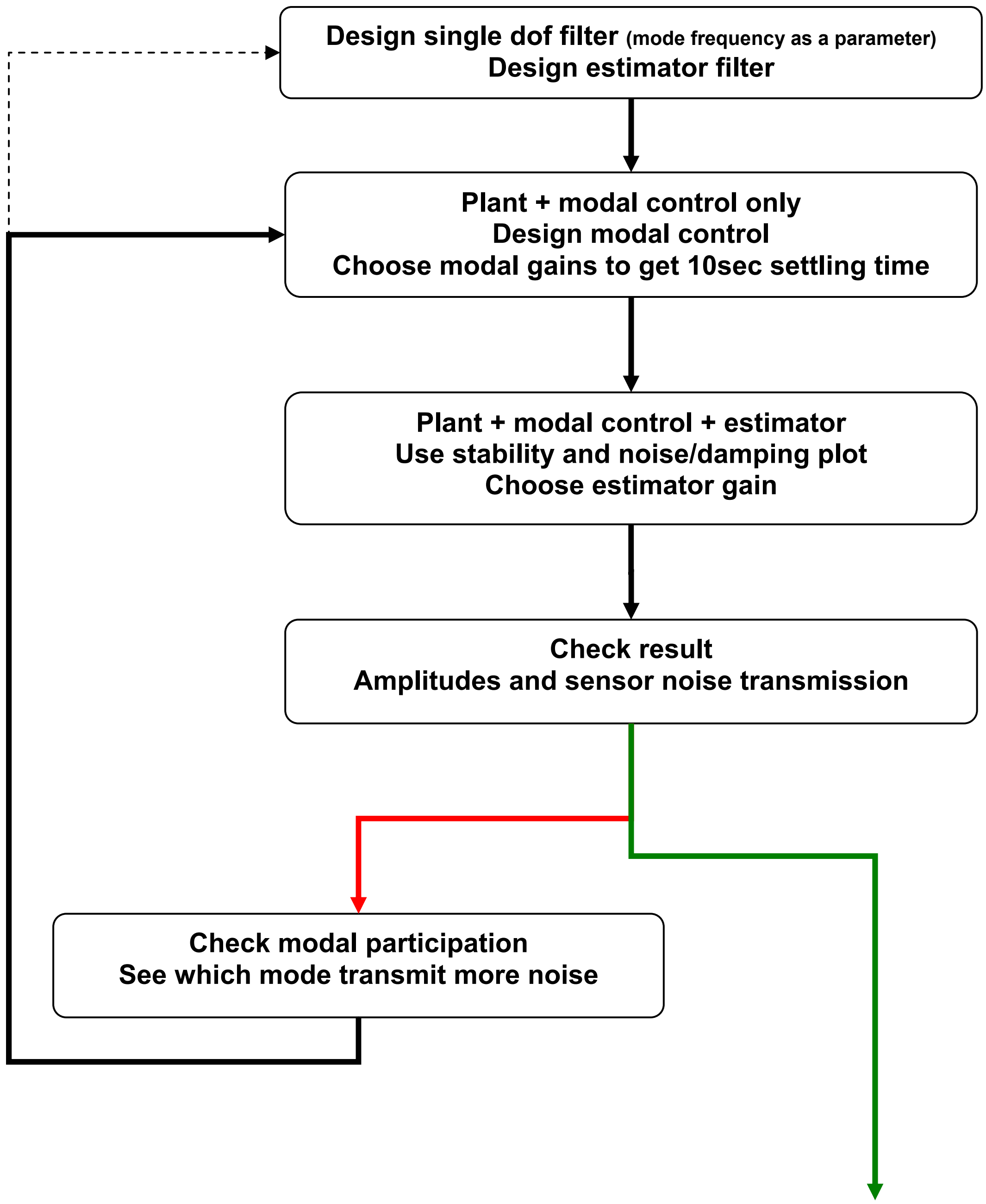
Roll direction, comparison of sensor noise to bottom mass transmission



Y direction, comparison of sensor noise to bottom mass transmission

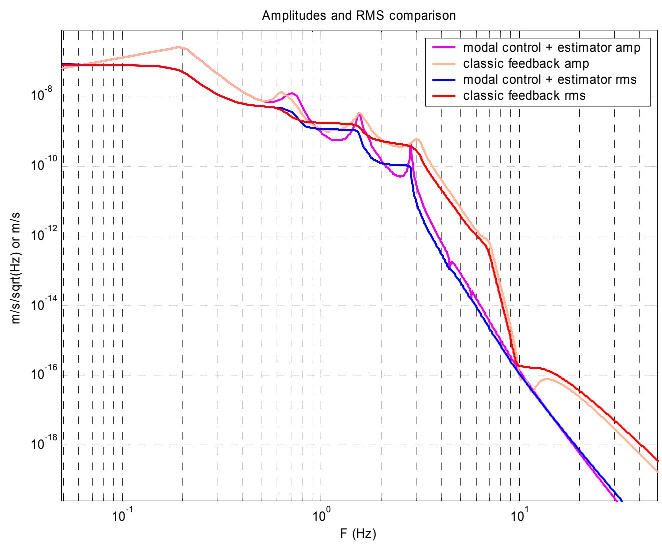
7.10 Summary

Here is a flow chart that summarizes the process of designing a modal control and estimator loop

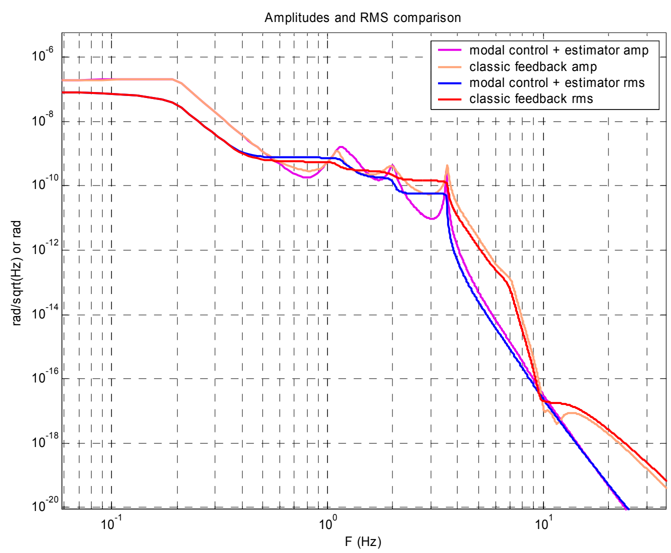


7.11 Locking performances (RMS)

Before we conclude about the design of this modal control and estimator loop. It can be interesting to look at the performances of the new loop in term of locking performances and compare it to the classic feedback. We will plot the RMS velocity of the X motion and the RMS angle of the yaw :



X direction
Amplitudes and RMS velocity comparison



Yaw direction
Amplitudes and RMS angle comparison

7.12 Conclusion

We have seen how easy and “automatic” the design of the damping loop was with this new method. By modifying a few gains, the designer can add damping or reduce sensor noise as needed. Moreover, we see that the performances are good and better than a classic feedback method. The difference in sensor noise transmission can reach up to 30dB at 100Hz. We chose to use simple filter for this example, we are confident that more aggressive filters can give better results. Designing more aggressive modal filters is the first thing to try while a more aggressive estimator filter requires is a bit more complicated and would require more work.

The next step was to test our loops on the triple pendulum we have in LASTI in order to validate the results.

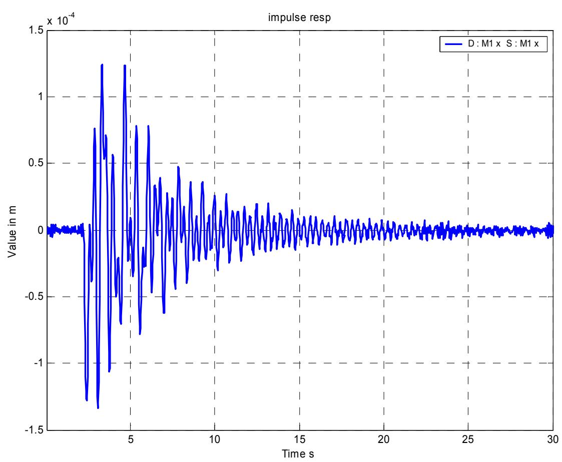
8 EXPERIMENTAL RESULTS

Now that we have created our control loop for the 6 degrees of freedom, we want to check that it works on the real pendulum we have in LASTI. The control system is programmed into the dSpace computer using Simulink

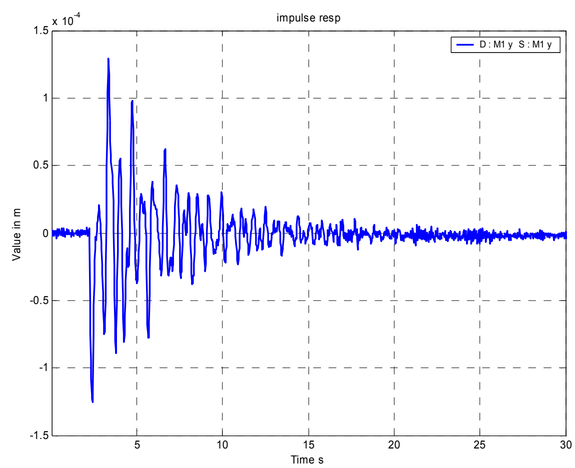
The loop were all checked individually for stability and then checked with all running at the same time. We then characterize the performances using the impulse response of the controlled pendulum.

8.1 Damping

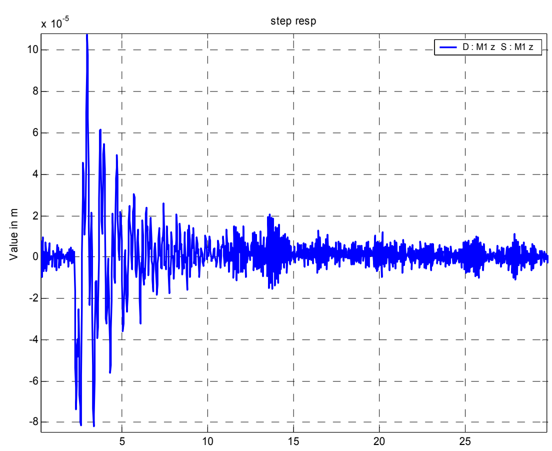
Note that we can only plot the response of the first mass since we can't measure the 6dofs on the bottom mass, so it is normal that some directions might damp a bit faster or slower than the 10sec.



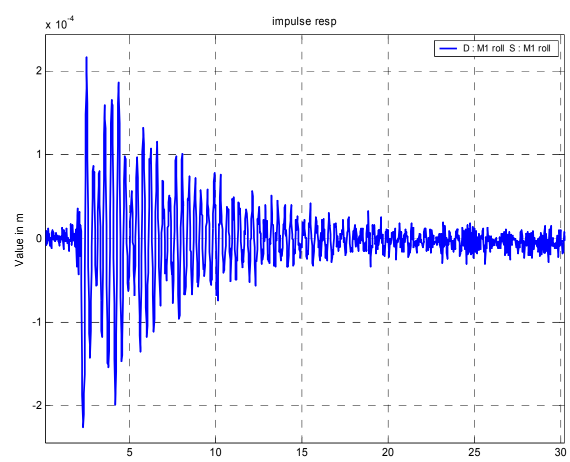
X direction : Impulse response (measurement)



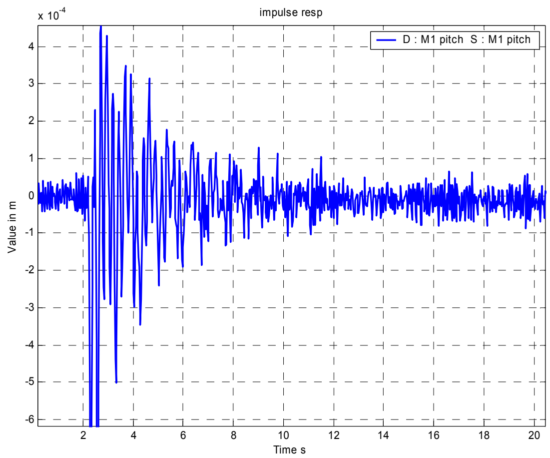
Y direction : Impulse response (measurement)



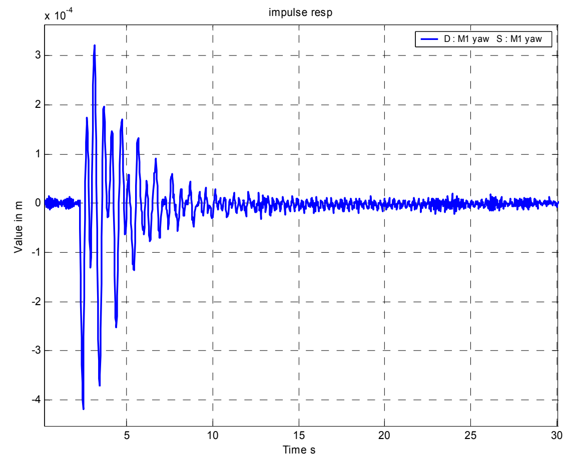
Z direction : Impulse response (measurement)



Roll direction : Impulse response (measurement)



Pitch direction : Impulse response (measurement)



Yaw direction : Impulse response (measurement)

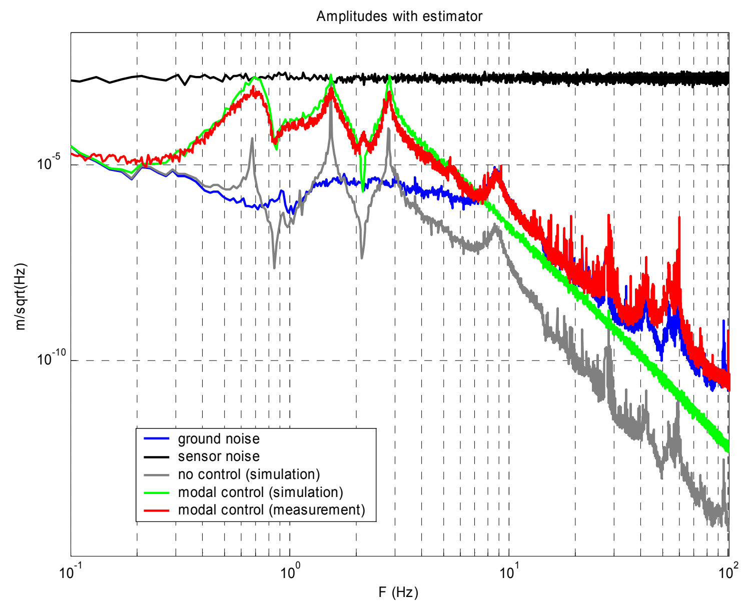
8.2 Noise

It is difficult to check the sensor noise injection with the triple pendulum in LASTI. First, our optical table is not as quiet (by more than 4 orders of magnitude) as the advanced LIGO one will be, and thus our sensor noise is negligible. We use a differential sensor, which means that when the pendulum's isolation becomes too high, we are mainly measuring the frame's motion.

The first problem can be solved by making our sensors noisier, we simply inject digital artificial noise in the Simulink diagram, this noise is a white noise and we try to have its amplitude as high as the mechanical limitations of the pendulum allow.

The second problem can't be solved. The higher the artificial noise will be the better you will see its effect on the measurement, but at one at some point, the filtering of the pendulum will become too high to distinguish its motion from the large frame's motion.

Below is the experiment in the X direction, we measure the optical table (called "ground") noise, we inject a very high sensor noise and measure the amplitude of the first mass. We can then use the noises as input in our simulation and compare the 2 results.



We see that the experiment matches the simulation pretty well. Unfortunately, above 8 hertz, it seems that the pendulum is not moving enough and that we are only measuring the frame's motion, but it gives us a good idea of the result and confirms that our simulations work.

9 CONCLUSION

We have seen how to design a new kind of loop that will make the design of the active control simpler and more efficient in sensor noise filtering. This new method uses modal control and estimator and has many advantages:

- It is simple to design
- It is flexible; we can change the damping of one resonance very easily
- It has better performance in reducing the sensor noise re-injection

We have tested it on the triple pendulum in LASTI and it worked perfectly.

The method has a few drawbacks however; the requirements on the model accuracy are still not really well-known. Even though we know which parameters of the model need to be good and how to minimize the risk by choosing good filters and gains, we can't really quantify this uncertainty. It is important to remember 3 things:

- We can develop techniques to improve the model by matching it with the plant
- Every experiment we did in LASTI (with good or less good models) worked perfectly
- Once the control has been designed and tested, it will work "forever" if the plant doesn't change, like a classic feedback method

This method opens a new way to create active control for pendulums (structure that are easy to model), with many new possibilities and we can without any doubt expect to get even better performances in the future:

Designing more complex filters for each mode can be tried and will probably improve the filtering at high frequency.

Using a MIMO estimator also opens a lot of possibilities like

- Using sensors on the optical table (or elsewhere) to add a small part of feed-forward control
- Using more sensors (for the quadruple pendulum for example)
- Improve the accuracy of the estimator
- ...

Appendix I   Matching the model with gradient minimization

A very important thing for this control is to have a good model that matches the plant as well as possible (~5%). Differences in the model, especially for the resonance frequencies, can make the system less stable and if those differences are too large, can turn the loop unstable. It is especially true if several resonances are really close to each other (for example the last 2 pitch resonances).

Our experience with the triple pendulum tells us that the model is good and complete enough for our purposes. We are also able to improve the match between the model and the plant by tweaking its parameters by few percents.

The solution we found to make the model more accurate was to use the characterization of the real plant to get the resonances frequencies, and then use a mathematical method to adjust the parameters of the model so that it gives us the same frequencies.

(See chart on next page)

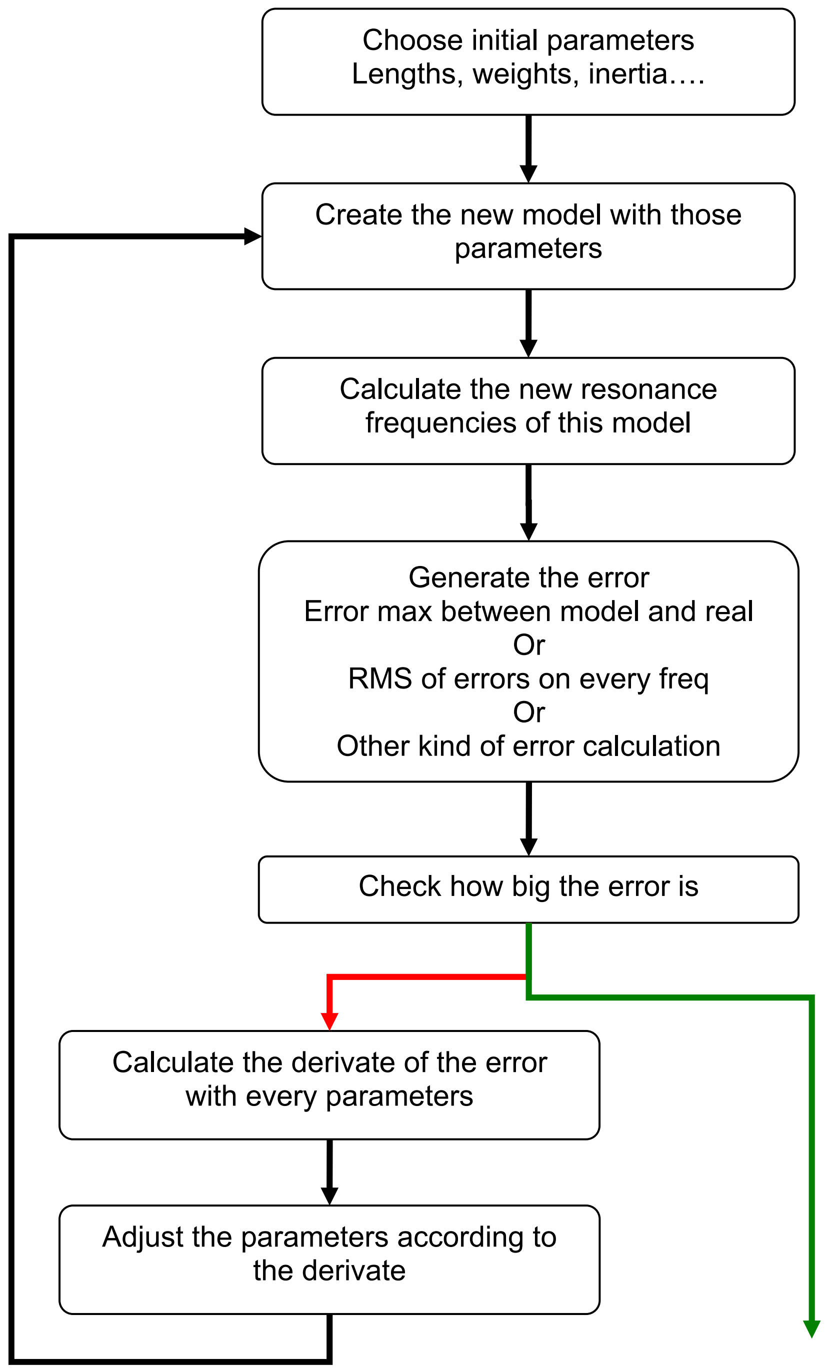
The Matlab function used for the gradient minimization is called fminimax.

We use a gradient minimization to find the best inputs (parameters) that minimize the error. We limit the parameter's change to be +/-5% of the initial parameter.

At the end, we have a slightly different new model; we check the new resonances and the new modal basis (which shouldn't change too much).

This model can be used in the estimator loop to improve performances.

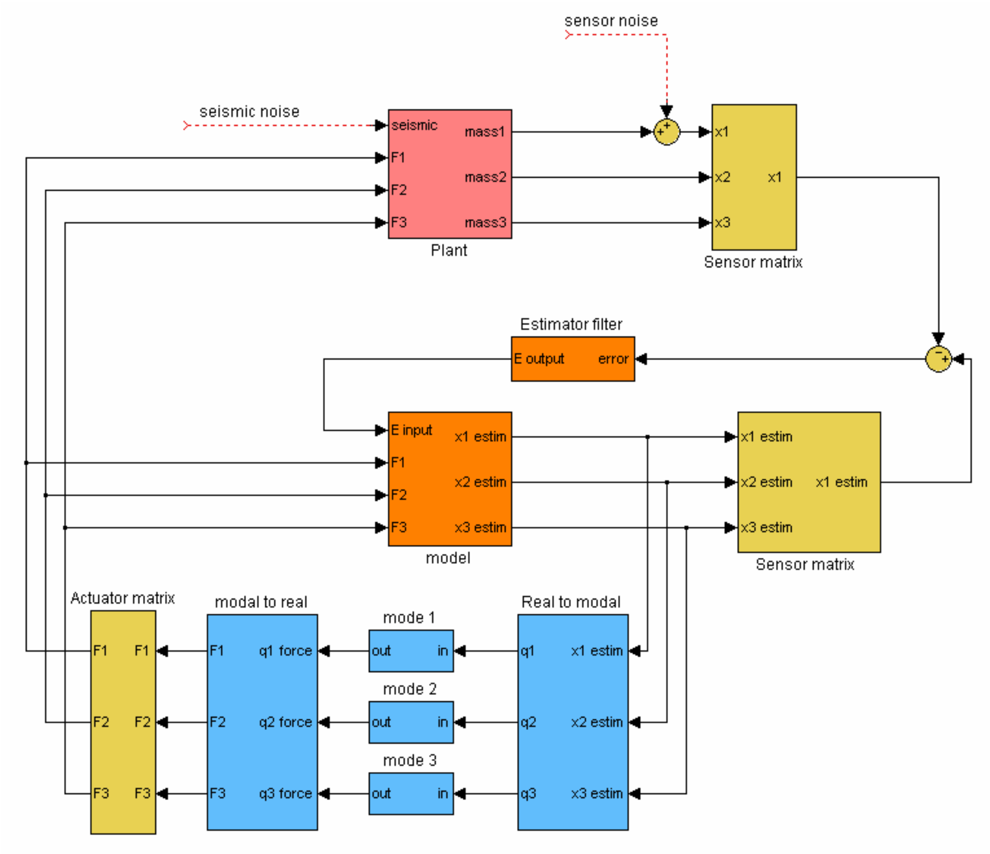
Chart of the model matching method using error gradient minimization



Appendix II   Loop model in state space

In order to build a Matlab model for the loop, we had to use a state space representation for the filters, plant, model… (Matlab has trouble dividing transfer function in tf or zpk representation).

We won't explain in detail how to create a state space model in this document. Below is the diagram of the state space model for a 3dof, with every blocks and every connections:

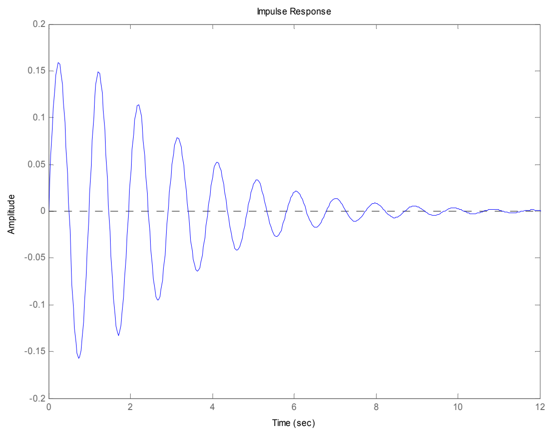


Instead of creating software that would only works for 3dof and another for 6dof, we have created a software that would work for n dof. While it has been a bit more complex at the beginning, it will help in the future to create new loop.

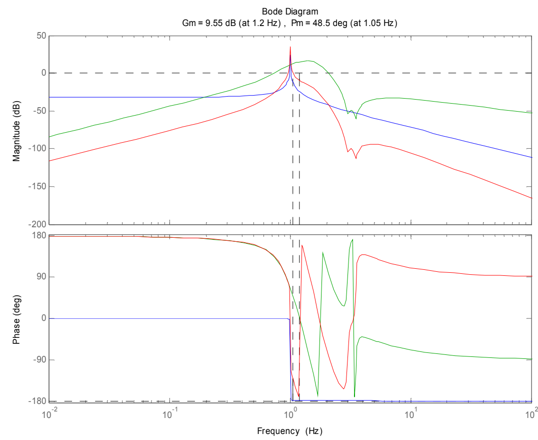
Appendix III More complicated filter and better performances

We have used very simple filters so far, we have seen that the results are already really good. In this appendix, we will show that more complicated filters could be used for the modal control to improve the performances.

The filter we choose to improve is the parameterized 1dof filter (the filter we use for each mode that is frequency dependant). Instead of using a simple filter with few poles, we design a more aggressive filter you can see below:



Impulse response for a 1dof system

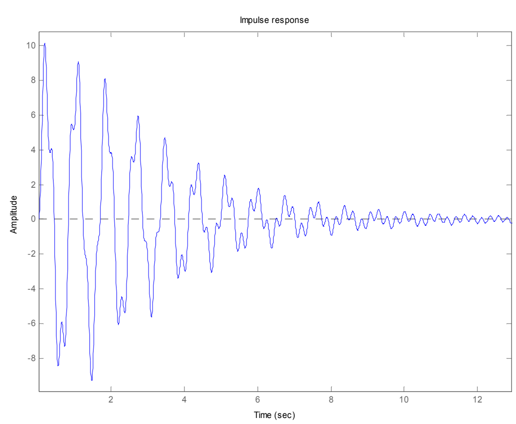


Plant, filter and open loop for a 1dof system damping

We see that the filtering is more aggressive after the resonance, while we keep our gain and phase good at the resonance.

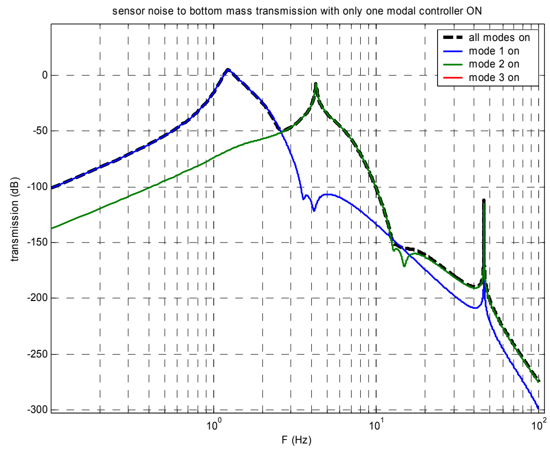
We will use this filter with the vertical model, and see how it can change the performances

We then apply the same methods we have seen above to get the 10sec damping and reduce the sensor noise transmission as much as possible. We choose the modal gains, the estimator gain (its shape remains the same) and check the results:

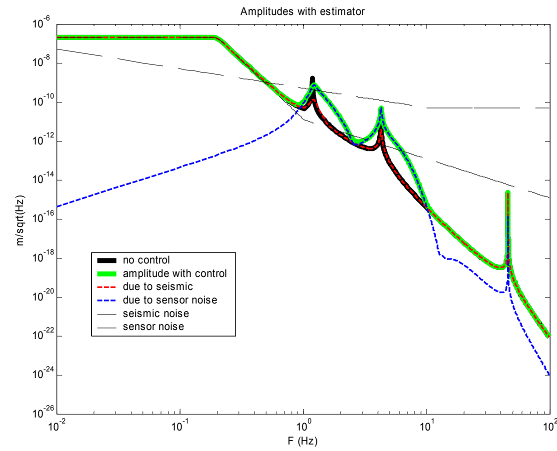


Impulse response of the controlled vertical pendulum

The impulse response gives the 10sec damping we want.

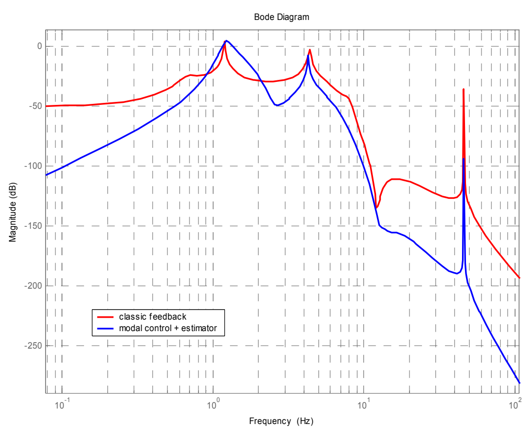


Z direction, participation of each mode in the sensor noise transmission



Z direction, amplitudes with seismic noise in Z direction

The amplitude plot show us that the sensor-noise injection becomes negligible above 10Hz. It is completely filtered by our new system.



Z direction, comparison of sensor noise to bottom mass transmission

The sensor noise transmission to the bottom mass is excellent if you compare it to a classic feedback.

This example shows us that we can easily improve our system by modifying the filter used in the modal control. This is a very easy change that immediately gives excellent results.

Appendix IV Monte Carlo method for multi-dof stability

Like we saw in the document, studying the stability of a multi-dof system is easy if you consider that the model is perfectly matching the plant. However, it is very difficult to know if your system will still be stable if the model is not perfect for 2 reasons:

- It might modify the model decomposition
- It is difficult to quantify a model mismatch for a multi-dof system

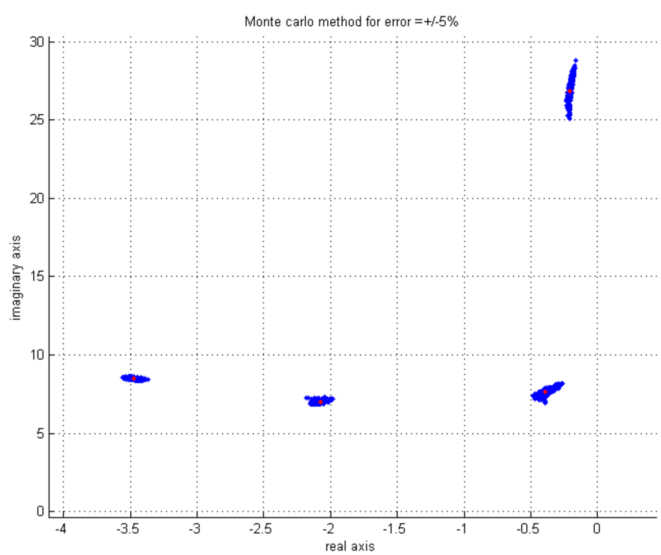
A solution that we have used recently and we will continue to study is to use a Monte Carlo method to “statistically” try many random mismatches of the model.

The method consists in randomly changing the parameters of the plant (we design the control with the model so what is actually changing is the plant), we plot the poles of the closed loop for every new set of random parameters. We can repeat the calculation as many times as we want.

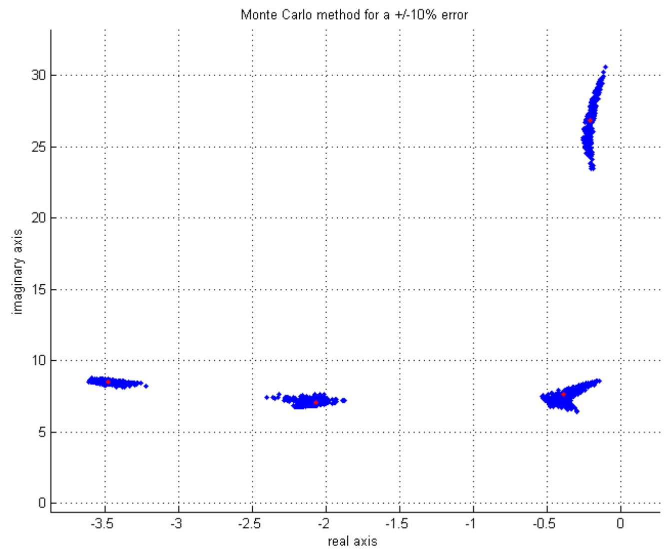
The example below shows this method on the vertical triple pendulum system.

1. We use the model and loop we have seen in the document (see 7.6).
2. We create a new plant using random parameters like the weight, the lengths (initial parameter + error)
3. We plot the poles of the new closed loop using this plant
4. We repeat the same 2 steps 300 times with random parameters every time

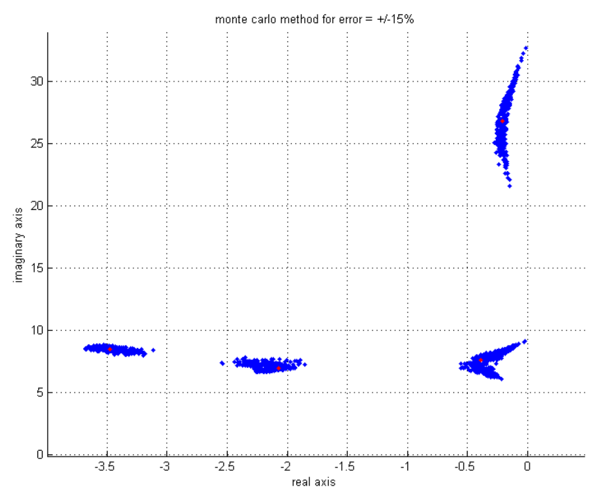
The case where plant = model (no mismatch) is the red dot, all the blues dots correspond to a randomly generated error.



Monte Carlo method on the vertical pendulum, error on parameters = +/-5 %



Monte Carlo method on the vertical pendulum, error on parameters = +/-10 %



Monte Carlo method on the vertical pendulum, error on parameters = +/-15 %

As we can see, as long as the error is lower than 15%, we didn't get any unstable point, which means that the probability the loop will be stable even with a wrong model is very high. At 15%, we notice that few points are unstable, which means that the error starts to be too big too keep the system stable if the model is bad.

Since we don't expect any error bigger than 5%, that means the system we designed is safe even with some model mismatches.

This method will be studied a bit more in the future but the first results are very promising.