State observers and Kalman filtering for high performance vibration isolation systems

M. G. Beker, **A. Bertolini**, **J. F. J. van den Brand**, **H. J. Bulten**, **E. Hennes** and **D. S. Rabeling**

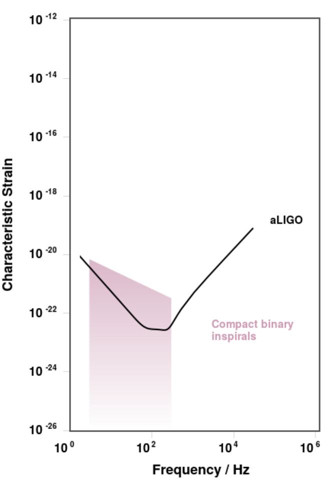
Sean Leavey

Low Frequency Gravitational Wave Astronomy

(a.k.a. *why do we care?*)

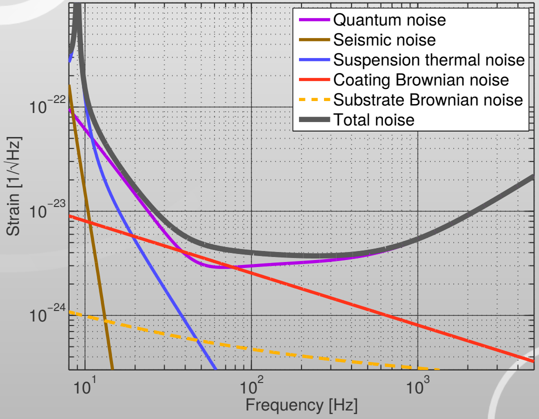
Intro

- Scientific benefit to studying low frequencies below 10 Hz
- Spinning neutron stars, black holes, CBCs
- CBCs spend more time at low frequencies, so extra observation time can be gained with better sensitivity

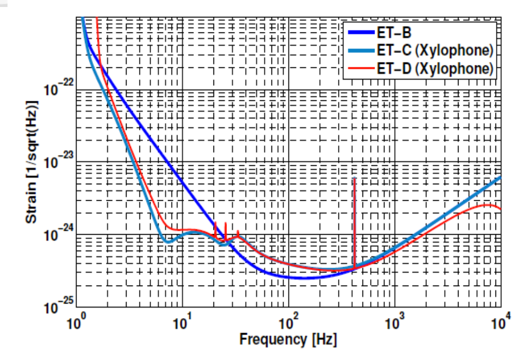


2G and 3G at Low Frequencies

- Advanced LIGO and Advanced Virgo will push low frequency sensitivity down to 10 Hz.
- This is already good, but we want to do better...

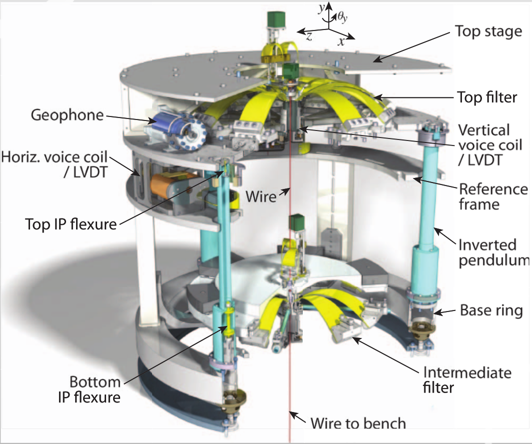


- ET-LF will go to 2 Hz to access this astronomy
- We hit a wall with seismic noise in this region



Seismic attenuation systems in gravitational wave observatories

In Advanced Virgo, the vibration isolation system for the arms, recycling mirrors and beam splitter suppress external motion by up to **15 orders of magnitude** above 10 Hz.



- Existing classical control techniques are already pushed to their limit
- Non-optimal control introduces additional noise into detection band
- This will be more of a problem in 2G and 3G detectors, leading to lack of sensitivity and lock loss during bad weather
- **Need a new approach**

Modern control

(Disclaimer: I am not an expert.)

(Second disclaimer: get ready for slides without images)

- We already know something about the dynamics of the suspension system
- Simple control systems are somewhat ignorant of known system dynamics
- Can we use this information somehow to help control the system?
- **This is the basis of state observation and Kalman filtering**

States

It is useful to think of the ways in which a system can exist as *states*.

The number of states in a system is determined by the system.

Examples:

- Single pendulum
- Rigidly coupled pendulum (not the same as two single pendulums)
- Gases, magnetic materials, etc...

State variables

State variables fully describe the system in such a way that its future behaviour can be determined (in the absence of external forces).

For a mechanical system, the **position** and **velocity** of individual components are the state variables.

It's therefore possible to take the position and velocity of the system at time t and calculate it at time $t + \delta t$.

Hidden states

However, in complex systems there are always hidden states present which can influence your cost function, and thus your ability to control the system.

Examples:

- Aircraft: air density (turbulence), wind speed and direction, etc.
- Oil refinery: temperature variations, flow rate of liquids
- Interferometry: **seismic noise coupling into the suspended optics in the SAS**

These hidden states can impact a classical system enough that the controller would lose control.

Control using states

Control of a system is the process of taking the current system state and feeding back to the system some combination of these signals, with appropriate gain (filtering). Mathematically:

$$\mathbf{u} = -K\mathbf{x}, \tag{1}$$

where $\mathbf{u}$ and $\mathbf{x}$ are actuator signals and states, respectively, and K is a matrix of gains.

Cost functions

What is K for?

You use it to minimise a *cost function*.

Example: a suspended mirror's set-point. Usually we want to keep mirrors at their 'zero' position.

Feedback in modern control

For suspended optics using modern control, feedback with K is typically performed by a *linear quadratic regulator* (LQR).

Skipping the detail, an LQR is a way of controlling a linear system with a quadratic cost function. The key constraint is that **all of the states in the system need to be known by the LQR to be able to feed back appropriately**.

How can we possibly know every state of a multi-stage seismic attenuation system?

LQR and Kalman filtering

Help comes from the *Kalman state estimator*, which is able to estimate the states on behalf of the LQR.

Put simply...

State estimator + linear quadratic regulator = optimal control

State space representation

(sorry)

State space equations

A linear, first order, time-invariant system can be described with the state space equations:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}, \tag{2}$$

and

$$\mathbf{y} = C\mathbf{x} + D\mathbf{u}. \tag{3}$$

State space equations

State space equations

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$$

$$\mathbf{y} = C\mathbf{x} + D\mathbf{u}$$

$\mathbf{x}$ is the vector containing the system's states e.g. the position or velocity of an object within the system.

If there are n states, then $\mathbf{x}$ is size $n \times 1$.

State space equations

State space equations

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$$

$$\mathbf{y} = C\mathbf{x} + D\mathbf{u}$$

A is the matrix characterising the dynamics of system - the *model*. It is of size $n \times n$.

For example, one entry of this matrix might map the velocity of one state to the position of another.

State space equations

State space equations

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$$

$$\mathbf{y} = C\mathbf{x} + D\mathbf{u}$$

u is the vector representing the system's inputs, i.e. the actuators, of size $p \times 1$. The input matrix B maps these actuators to the state dimensions they operate on.

For example, an actuator might actuate on multiple degrees of freedom, so B maps one actuator to many dimensions of the state space.

State space equations

State space equations

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$$

$$\mathbf{y} = C\mathbf{x} + D\mathbf{u}$$

y is the vector containing the measurements of the system by its sensors, of size $q \times 1$. The sensing matrix C is the opposite of B, mapping the states to the sensors.

State space equations

State space equations

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$$

$$\mathbf{y} = C\mathbf{x} + D\mathbf{u}$$

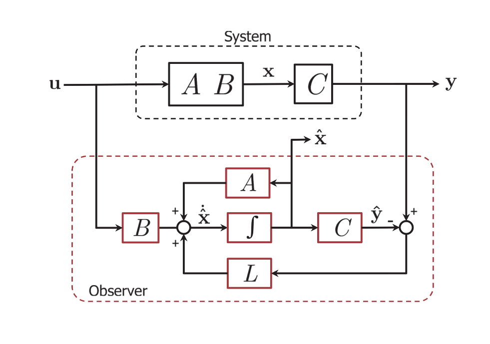
D simply maps any direct coupling between the actuators and sensors. For now, we will ignore it ($D = 0$).

How do we use the state space representation to model our system?

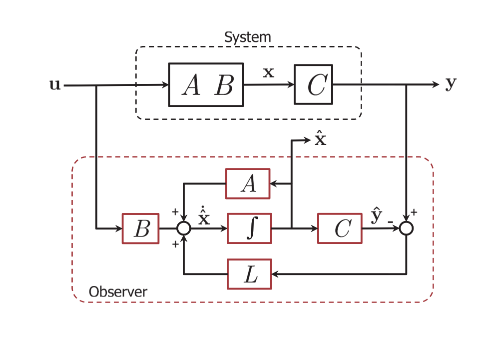
State Estimators

As mentioned before, we use a state estimator.

A state estimator runs in parallel to the system under control. It receives the same inputs from the controller $\mathbf{u}$ as the real system and produces an *estimate* of the corresponding state vector given these inputs, $\dot{\hat{\mathbf{x}}}$.



State Estimators



Since it predicts the state of the system given its inputs, it can also produce a corresponding sensor vector $\hat{\mathbf{y}}$, similar to the real sensor measurements contained in $\mathbf{y}$.

State Estimators

The dynamics of the state observer are the same as the real system (we just add a few hats):

$$\dot{\hat{\mathbf{x}}} = A\hat{\mathbf{x}} + B\mathbf{u} \tag{4}$$

$$\hat{\mathbf{y}} = C\hat{\mathbf{x}} \tag{5}$$

(remember $D = 0$ here)

State Estimators

Estimator

$$\dot{\hat{\mathbf{x}}} = A\hat{\mathbf{x}} + B\mathbf{u}$$

$$\hat{\mathbf{y}} = C\hat{\mathbf{x}}$$

We can then calculate the error between the real system and the state observer:

$$\begin{aligned} \mathbf{e} &= \mathbf{y} - \hat{\mathbf{y}} \\ &= \mathbf{y} - C\hat{\mathbf{x}} \end{aligned} \tag{6}$$

State Observers

Estimator

$$\dot{\hat{\mathbf{x}}} = A\hat{\mathbf{x}} + B\mathbf{u}$$
$$\hat{\mathbf{y}} = C\hat{\mathbf{x}}$$

Error between estimator and system

$$\begin{aligned} \mathbf{e} &= \mathbf{y} - \hat{\mathbf{y}} \\ &= \mathbf{y} - C\hat{\mathbf{x}} \end{aligned}$$

The state estimator's state space equation can be modified to include a term proportional to the error:

$$\begin{aligned} \dot{\hat{\mathbf{x}}} &= A\hat{\mathbf{x}} + B\mathbf{u} + L\mathbf{e} \\ &= A\hat{\mathbf{x}} + B\mathbf{u} + L\left(\mathbf{y} - C\hat{\mathbf{x}}\right) \end{aligned} \tag{7}$$

Observer gain matrix

Observer equation

$$\begin{aligned}\dot{\hat{\mathbf{x}}} &= A\hat{\mathbf{x}} + B\mathbf{u} + L\mathbf{e} \\ &= A\hat{\mathbf{x}} + B\mathbf{u} + L\left(\mathbf{y} - C\hat{\mathbf{x}}\right)\end{aligned}$$

L is the *observer gain matrix*. It has n rows and q columns (number of states $\times$ number of actuators).

Kalman Filtering

L is where a modern controller can select whether to listen to the sensors or the model in different situations.

Kalman filtering removes the need for the systems engineers to create the observer gain matrix manually. A Kalman filter will use the complete history of measurements $\mathbf{y}(t)$ to create an *optimal* estimator at any given moment.

This leads to an optimal state estimator that minimises the difference between the measured states and the state estimator's estimates.

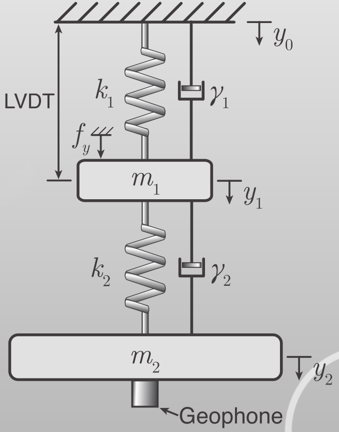
Effective state estimation for the Advanced Virgo seismic attenuation system

Modelling the seismic attenuation system

An accurate model of the seismic attenuation system is required for the state estimator.

This is achieved using the Euler-Lagrange equation to model the double mass-spring system.

Note: we only care about vertical motion.



Modelling the seismic attenuation system

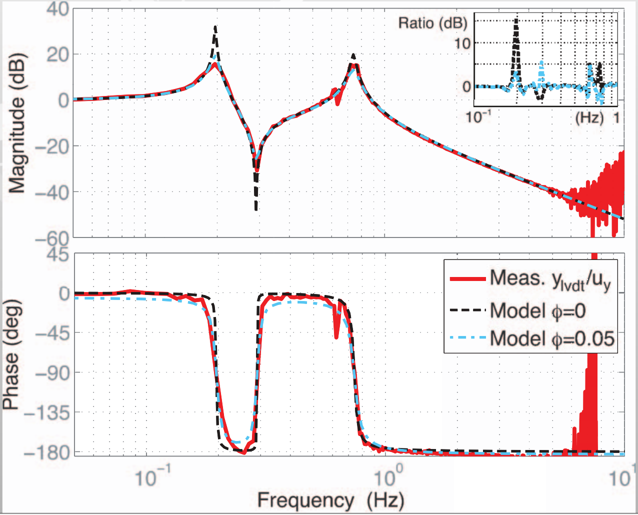
$$\underbrace{\begin{bmatrix}\dot{y}_1\\ \dot{y}_2\\ \dot{v}_1\\ \dot{v}_2\end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ \frac{-(k_1+k_2)}{m_1} & \frac{k_2}{m_1} & \frac{-(\mathcal{Y}_1+\mathcal{Y}_2)}{m_1} & \frac{\mathcal{Y}+2}{m_1}\\ \frac{-k_2}{m_2} & \frac{-k_2}{m_2} & \frac{-(\mathcal{Y}_2)}{m_2} & \frac{-\mathcal{Y}}{m_2}\end{bmatrix}}_{A} \underbrace{\begin{bmatrix}y_1\\ y_2\\ v_1\\ v_2\end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix}0\\ 0\\ \frac{k_1}{m_1}\\ 0\end{bmatrix}}_{B} \mathbf{u} + B_{pn}\omega_d$$

$$\underbrace{\begin{bmatrix}y_{lvdt}\\ y_{geo}\end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}}_{C} \underbrace{\begin{bmatrix}y_1\\ y_2\\ v_1\\ v_2\end{bmatrix}}_{\mathbf{x}}$$

Subscripts 1 and 2 are the two stages, and $v_i = \dot{y}_i$. $B_{pn}\omega_d$ is the external disturbance from seismic motion.

Modelling the seismic attenuation system

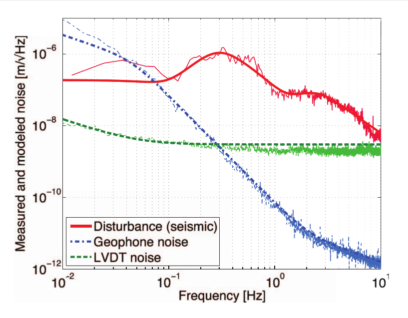
The model fits the measurements well:



Sensor noise

There are two primary sensors in the SAS:

- Linear variable displacement transducers (LVDTs)
- Inertial sensors (geophones)



The LVDTs and geophones have different SNRs at different frequencies:

LVDT

DC to 0.1 Hz: **excellent**
greater than 0.1 Hz: **not so good**

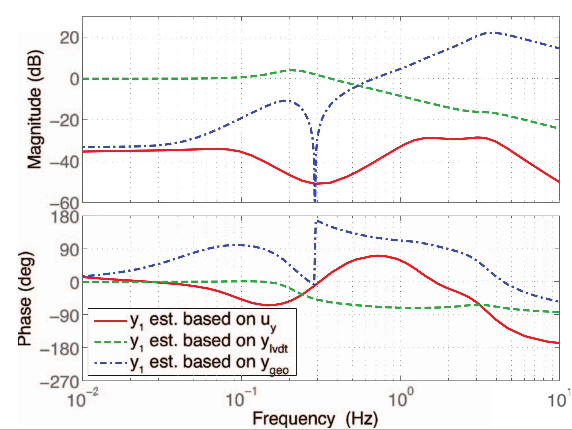
Geophones

Below 0.1 Hz: **bad**
Above 0.5 Hz: **excellent**

Which sensor do I trust more?

Feedback from Kalman

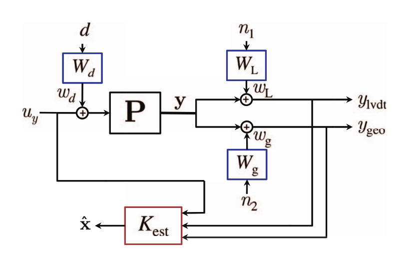
The frequency response of each of the estimators for the intermediate stage is shown below:



This shows the Kalman filter's relative trust in each of the sensors as a function of frequency.

The cross-over frequency here roughly corresponds to the cross-over of the SNRs shown previously.

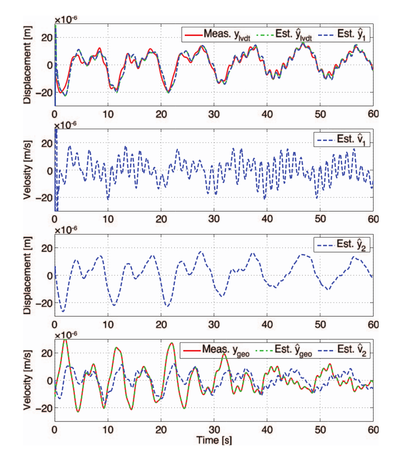
Feedback from Kalman



A key feature is that it **it can choose to listen more to the sensors than the model, or vice versa, based on how well each of them previously matched the actual state.**

How good is the state estimator?

How good is the state estimator?

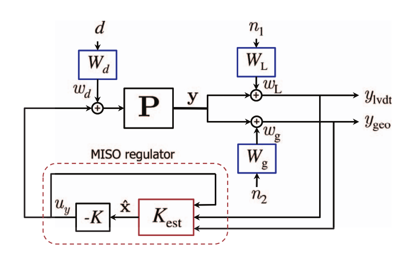


After 30 s, injected sinusoidal motion becomes visible.

The estimators based on the LVDT and geophone signals begin to accurately measure this sinusoidal injection.

The Kalman filter is accurately modelling all states of the system.

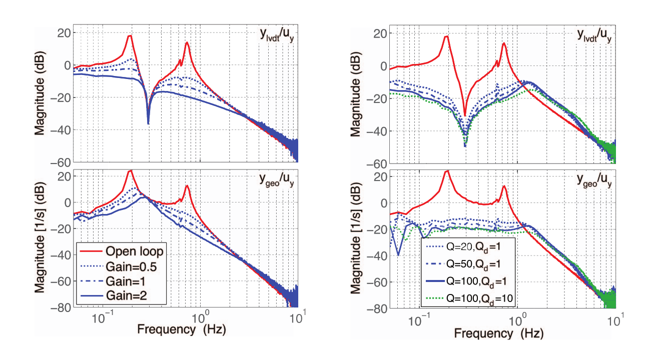
Vertical control using the LQR



We refer back to the LQR. Using the state estimator, an effective gain matrix K can be produced using knowledge of the different noise performances of each of the sensors.

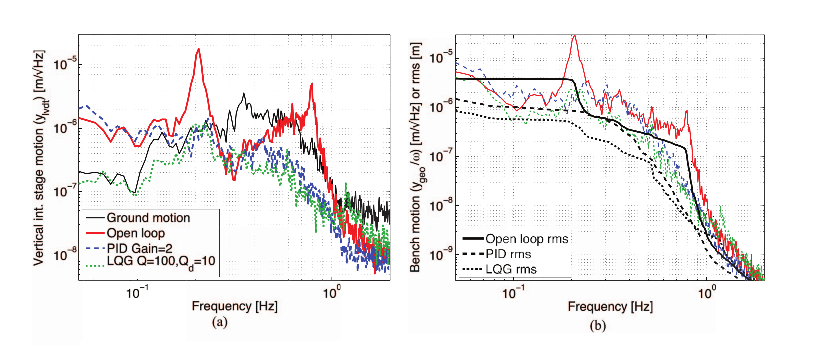
The state estimator's estimate is attached to K and fed back to the system.

Improvement over "classical" control



Observe input noise (present on red curve) being coupled to the LVDT and geophone measurements on the left, and being suppressed on the right.

Improvement over "classical" control



Conclusions

- Better results using modern control in this case
- Ability to handle multiple sensors and actuators in one loop ("MIMO")
- However, the "optimal" part is still determined by the (human-designed) cost function
- Added complexity, harder to understand when things go wrong
- **Overall, probably a required technology going forward as SASes get more complex**

What else?

https://www.youtube.com/watch?v=LaDd0gqwCCs