Frequency Dependent Diagonalization Matrices of Payload Actuation for Local Feedback Control

Tsang Terrence Tak Lun
The Chinese University of Hong Kong

April 17, 2020

The following is a proposal to modify the actuation output diagonalization scheme in hope to reduce actuation cross-coupling from the local damping control. The implementation of this scheme may be trivial but this will help eliminating one source of resonance mode ring up, which, in turn, makes debugging easier when a mode is suddenly excited. Although the system is functional currently, the actuators are not really diagonalized, as in, there are plenty of room for improvement.

I am sure that most people realize the importance of diagonalized systems. Here, I briefly explain the idea just in case. The reason for a diagonalized system is simple. The techniques we have been using, Bode plots, Nyquist criterions, etc, are optimized for a single degree of freedom (DoF). This means that, before doing anything related to feedback control, we should configure our system so that our systems are as close to multiple Single-Input-Single-Output (SISO) systems as possible so we can treat the each separated system one by one.

In the multiple SISO picture, if there is actuation coupling between different degrees of freedom, then the actuation/feedback of one DoF will become external disturbance into another which could cause undesirable excitation. Of course, we can patch the system with additional loops suppressing those unwanted excitation (such as the band-pass comp filters). But, the additional loop will cause additional excitation because of coupling, making this process very tedious and complicated. However, if we started off with systems that were already diagonalized, everything will be much simpler as cross-DoF excitation will be minimal. And, my opinion is to build our control systems on an already diagonalized system.

Ideally, a perfectly diagonalized system has one-to-one input output relationship. For example, if we start with a general system with N degrees of freedom x_1, x_2... x_N with actuation x_1^{act}, x_2^{act}... x_N^{act}, then diagonalization will give N separated subsystems that only has one input and one output as shown in Fig. 1. Then we can individually make control loops for each subsystems. In such a case, the diagonalization utilizes cancellation filters (feedforward filters basically) to cancel the coupling effects. For example, in a 2 DoFs system shown in Fig. 2, the first system output is a combination of the two inputs, i.e. $x_1 = P_{11}x_1^{\mathrm{act}} + P_{12}x_2^{\mathrm{act}}$. If we add a cancellation filter such that $x_1 = P_{11}x_1^{\mathrm{act}} + P_{12}x_2^{\mathrm{act}} - C_{12}P_{11}x_2^{\mathrm{act}}$, then the system is diagonalized if we set $C_{12} = P_{12}/P_{11}$.

Without loss of generality, if we consider a system with N degrees of freedom, then the system output reads

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} P_{11} & P_{12} & P_{13} & \dots & P_{1N} \\ P_{21} & \ddots & & & \\ P_{31} & & \ddots & & \\ \vdots & & & \ddots & \\ P_{N1} & & & & P_{NN} \end{pmatrix} \begin{pmatrix} x_1^{\mathrm{act}} \\ x_2^{\mathrm{act}} \\ x_3^{\mathrm{act}} \\ \vdots \\ x_N^{\mathrm{act}} \end{pmatrix} - \begin{pmatrix} 0 & C_{12} & C_{13} & \dots & C_{1N} \\ C_{21} & \ddots & & & \\ C_{31} & & \ddots & & \\ \vdots & & & \ddots & \\ C_{N1} & & & & 0 \end{pmatrix} \begin{pmatrix} P_{11}x_1^{\mathrm{act}} \\ P_{22}x_2^{\mathrm{act}} \\ P_{33}x_3^{\mathrm{act}} \\ \vdots \\ P_{NN}x_N^{\mathrm{act}} \end{pmatrix}, \tag{1}$$

where P_{ij} are the physical plants/paths and C_{ij} are the cancellation filters/diagonalization matrix. If we set the

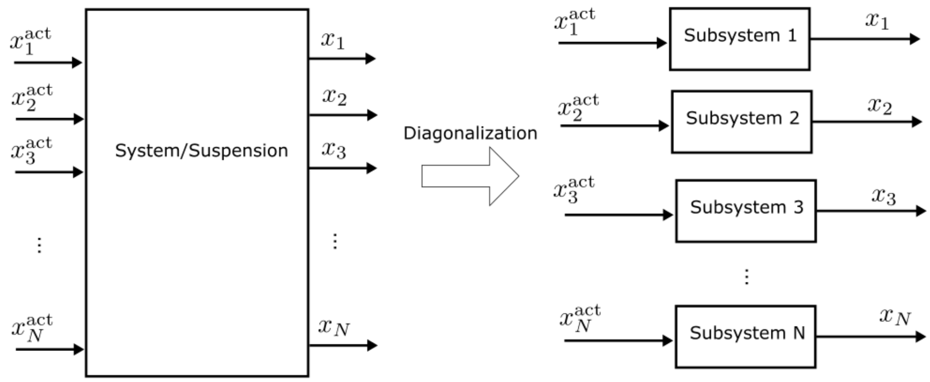

Figure 1: Fully diagonalized systems.

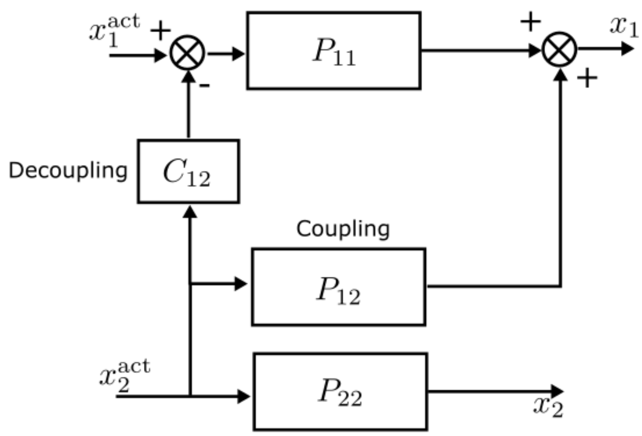

Figure 2: Cancellation filter for decoupling one degree of freedom from actuation of another degree of freedom

cancellation filters correctly, i.e. $C_{ij} = P_{ij}/P_{ii}$, then the input output relationship is completely diagonal, i.e.

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} P_{11} & 0 & 0 & \dots & 0 \\ 0 & \ddots & & & \\ 0 & & \ddots & & \\ \vdots & & & \ddots & \\ 0 & & & & P_{NN} \end{pmatrix} \begin{pmatrix} x_1^{\mathrm{act}} \\ x_2^{\mathrm{act}} \\ x_3^{\mathrm{act}} \\ \vdots \\ x_N^{\mathrm{act}}. \end{pmatrix} \tag{2}$$

And, from here, we can treat each degree of freedom separately and construct feedback loops separately.

Coming back to our case, there are at least 2 levels of actuation "diagonalization" I have seen in KAGRA and but none of those truly diagonalize the actuators. In all cases, we start with an output matrix "EUL2COIL" which has the geometry information of the actuators. This matrix coarsely diagonalizes the actuators but is usually not enough. The first level of such is the so-called "coil balancing". This procedure equalizes the strength of the coils. In the free-mass limit, this level of "diagonalization" would be enough. However, in reality, the masses are suspended rather than free floating, and there are more mechanical complications involved. For example, equal longitudinal offsets acting on a pendulum will tilt the pendulum as well. So, balancing the coils are actually not directly related to any sort of diagonalization in reality. The second level of diagonalization was used at the inverted pendulum stages to diagonalize the longitudinal, transverse and yaw actuation degrees of freedom, given a reliable sensor readout. Each of the coils were used to push the inverted pendulum at a particular frequency and the responses is observed. From the coupling ratios measured, a correction matrix can be applied directly to the EUL2COIL matrix to diagonalize the actuators. This is almost equivalent to the aforementioned cancellation filter matrix, except it only diagonalize the actuators at one frequency because the matrix has scalar entries. This approach works for the inverted pendulum because actuation couplings are assumed to have no frequency dependency. However, at the payload, the dynamics of the pendulums dictate that the actuations are intrinsically coupled and have frequency dependency. Therefore, the only way to diagonalize the actuators at the payload is via cancellation filter matrix.

Here, we can choose to keep the EUL2COIL matrix and the coil output filters, but the EUL2COIL matrix is not necessary because it can be absorbed into the diagonalization filter matrix. Because the actuation coupling goes both intra-stage (e.g. TM-L to TM-P) and inter-stage (e.g. IM-L to TM-P), in general, the diagonalization matrix should span all stages (at least all stages of the payload). So, for the payload, the real-time model should ideally look like Fig. 3. Actuation signals go through a gigantic diagonalization filter matrix and the diagonalization matrix

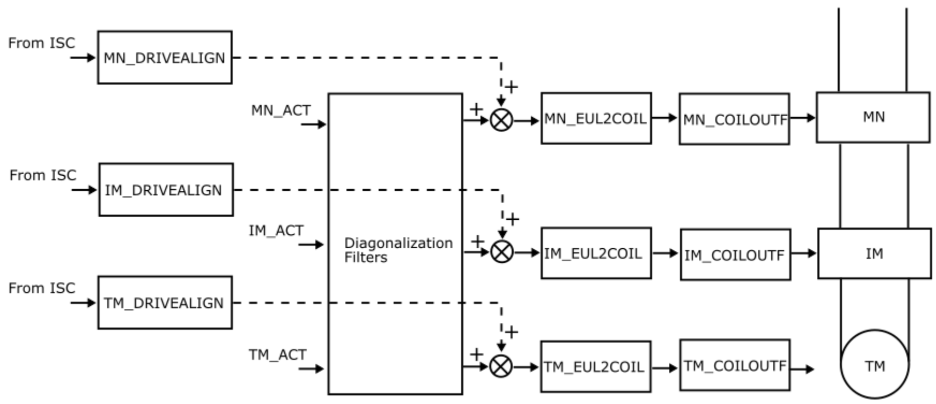

Figure 3: Diagonalization filter spanning all payload stages.

redistribute the signals to the coils. Notice that this diagonalization matrix only affects local control. Ideally, The diagonalization matrix should kill all actuation coupling, so this also means killing the possibility of hierarchical control. Therefore, the ISC path along with the DRIVEALIGN matrix should bypass the diagonalization filter so hierarchical control is possible. Of course, we can also configure the diagonalization matrix such that actuation from higher stage to lower stage is not decoupled. In this case, the ISC path can go before the diagonalization filters. If we choose to not decouple inter-stage coupling, then we can have three separate diagonalization filter matrix for each stage as well. It is very hard to decide an actual configuration right now because it depends on the final implementation and the person who is going to deal with this IFO-to-suspension interface. But, the configuration in Fig. 3 should be general enough to cover all cases. The diagonalization matrix should be in the form of

$$\text{Diagonalization matrix} = \begin{pmatrix} 1 & C_{12} & \dots & C_{1N} \\ C_{21} & 1 & & \\ \vdots & & \ddots & \\ C_{N1} & & & 1 \end{pmatrix}, \tag{3}$$

where C_{ij} are the aforementioned cancellation filters.

The difference between DRIVEALIGN and the diagoanlization matrix should be clear although they might share similar filters. The purpose of the diagonalization matrix is to separate the degrees of freedom and turn the suspension system, which is a MIMO, into a multiple SISO where we can construct damping loops separately for each degree of freedom. On the other hand, the purpose of the DRIVEALIGN matrix is to decouple the test mass's degrees of freedom (the inteferometer to be exact), and in this case, the actuation can come from all three stages. To illustrate their difference, let's consider the length to pitch coupling from MN to TM. In the case of the local diagonalization matrix, the cancellation of such coupling is via TM pitch actuation. So, if there is MN length to TM pitch coupling, the diagonalization matrix will be configured such that the coupling will be cancelled by TM pitch actuation. In contrast, the DRIVEALIGN matrix will be configured such that the coupling will be cancelled by MN pitch actuation via the MN pitch to TM pitch path.