Implementing Logic in a Variable Displacement Pump to Maintain Resonance to Maximize Flow Gain in Unknown Time-Variant Systems

Leonardo Piñero-Pérez
Undergraduate Aerospace Engineering Student
University of Texas at Arlington Honors College
Department of Mechanical and Aerospace Engineering
University of Texas at Arlington
Arlington, Texas 76019
Email: leonardo.pinero-perez@mavs.uta.edu

The goal of this design was to implement logic-control on a variable displacement pump to maintain resonance to a changing system. It was designed and built around a sample static 4th-order system to test methods and was then simulated on a changing 2nd-order system. The logic implementation was successful, as the controlled pump was able to track the changing resonance frequencies and adapted its pulse train to maximize flow gains.

Nomenclature

fQa Pump Pulse Train Frequency Setting
Kw Pump Pulse Train Width Setting
Q_a Input Flow from Pump
Q_e Measured Output Flow
sys Unknown Time-variant System
t Time [s]
ϕ Swash Plate Angle
ω Axial Rotation of Pump

1 Introduction

The purpose of this project was to design control logic for a variable displacement pump such that the peak output flow was at the greatest magnitude relative to the input flow. One practical application of such a control system can be found in hydraulic fracturing wells, where the pump forces the output fluid through rock fissures to release the shale oil trapped inside. Increasing the peak fluid flow using small input flows reduces the pump requirements and water needed to break open the rock formations. This makes hydraulic fracturing cheaper and less mechanically intensive. The best way to maximize the flow gains is to maintain the system at resonance with the system. However, like the vast majority of practical applications, the equations which model a hydraulic fracturing pipe are unknown, changing, and nonlinear. This is where logic is needed to automatically follow the changing model according to measurements made by instrumentation such as a flow meter. The variable displacement pump generates pulse train inputs. It does this by rotating a series of pistons very quickly, bringing in the fluid through an inlet section where the pistons are expanding. Half a revolution later, the pistons generate an output flow by compressing the piston, as shown in the sketch above.

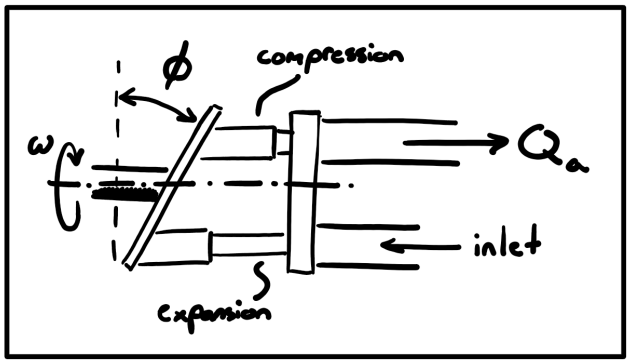

Fig. 1. Pump Schematic

2 Model

The sample model used for flow output as a function of flow input was linearized in Dr. Hullenders FPMC2017-4202 Paper [1]. This was a 26th-order linear transfer function approximating six bode peaks until a frequency of 2630 rad/s. The output of this transfer function gives the output flow normalized by the input flow. The flow gains of this model would be defined by the maximum output flow divided by the input flow of the pulses.

To make the system response easier to simulate, a reduced-order was generated using the balred() MATLAB

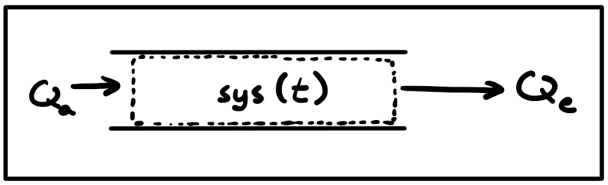

Fig. 2. Pipe Schematic

function, where a reduced, 4th-order approximation model was created from the 26th-order model. The 4th-order model is a good approximation only if the user specifies the frequency range of interest. To demonstrate the logic of the system, only the first two largest peaks were of interest. It is a better demonstration to include two bode peaks as an example, so that there is an opportunity for the logic to select the better of the two bode peaks. In this case, a range of 0 to 800 rad/s was selected to approximate. Again, it is important to note that the human or non-human operator of this pump does not know the bode response of the system on that particular day or site. The model is only described here to show an example of such an unknown system. The bode response of the high-order and 4th-order sample models are shown below.

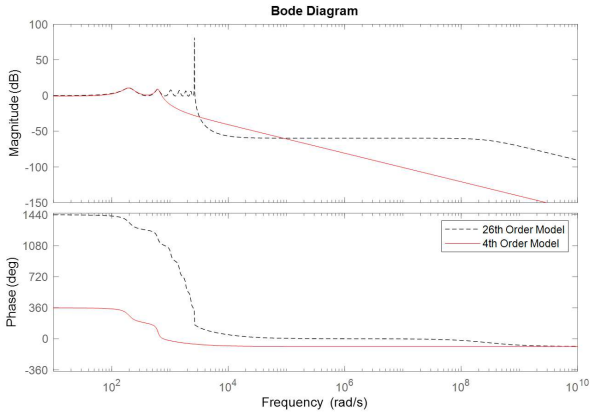

Fig. 3. Bode Plot of Reduced-Order Sample Model

3 Logic Description

The goal of this script is to simulate the logic-controlled pump to follow the resonance peak of a changing system. To do this, there first needs to be a function which simulates the arbitrarily-placed systems response to a specified pulse train input. The function will also simulate the instrument finding the normalized peak flow for each pulse after a steady state pulse train pattern has been reached. The MATLAB function lsim() was utilized here, as well as some data reduction. To use an example, say an operator working on the 4th-order system ran the pump at a fixed pulse train setting, such as 30% pulse width and 30 Hz. The input would look like the following (this is done within the function, calling MATLAB function pulstran()):

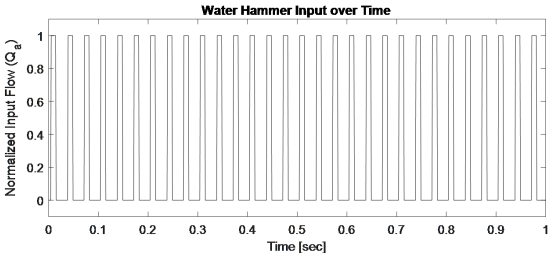

Fig. 4. Pump Input (30% Pulse at 30 Hz)

And running this input across our system yields the response:

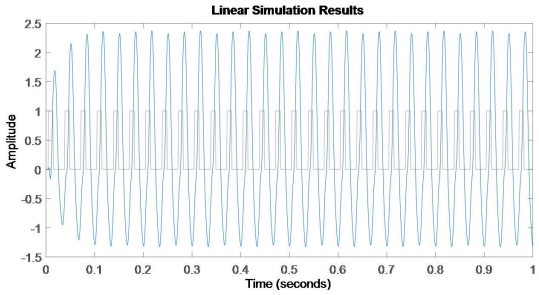

Fig. 5. Sample Model Response to Pump Input (30% Pulse at 30 Hz)

At this flow gain setting, it is apparent that after about a tenth of a second, the peak flow gains stabilize to around 2.4. Lets try a different setting with equal flow (10% pulse width at 90 Hz):

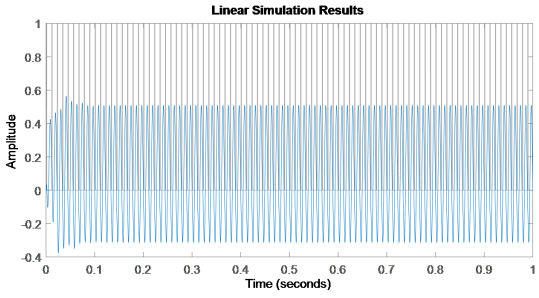

Fig. 6. Sample Model Response to Pump Input (10% Pulse at 90 Hz)

The flow settings have an effect on flow gains. In fact, here we have a flow gain of less than one, which is not optimal. As mentioned, after data from lsim() is taken, a data reduction script determines the peak flow gain. In this way, the function takes the unknown system response to pulse train

settings as an input, and outputs the peak response. This process is the main function used in the logic script, called as ControlledPulse(fQa, Kw, sys). Clearly, it would be best to experimentally determine the steady-state flow gain for a range of pulse train settings. This is the first step for the proposed logic. This is analogous to an operator learning about the unknown system by checking a variety of pump-generated pulse train widths and frequencies. The data for such a test is shown below, quickly utilizing the simulation function described above. A frequency range of 0 200 Hz is selected, with up to 50% pulse width in steps of 10%.

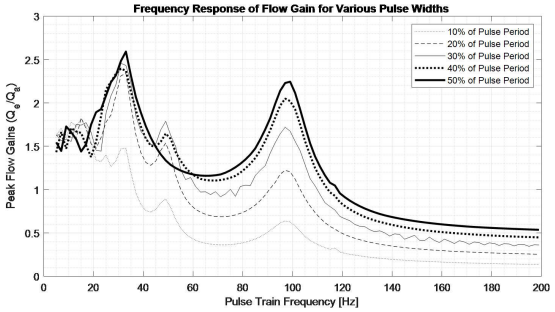

Fig. 7. System Response to Different Pulse Train Settings

These results are validated by the comments in Dr. Hullenders paper, which mentioned that increasing pulse width to 50% will increase flow gains by 25% to about 2.5. Of course, an automated control system will run through all these pump settings quickly and determine the global maxima. This is an important first step so that the logic knows where to go from the collected data. The settings with the largest flow gains are selected, and the response is measured at this optimal setting with two other values slightly above and below the optimal frequency. Measuring these three values is necessary so that the logic can adapt to a changing system. Since the peak value was selected, at first, the two surrounding frequency flow gain values should be below that of the middle optimal frequency. However, if a changing system were to make one of the outer frequencies larger, that frequency becomes the new optimal frequency setting, with its own two surrounding probing frequencies. This physically translates to the pump oscillating slightly around the optimal frequency, with the ability to have a sense of which direction it needs to go because of this consistent probing. In this way, the pump is automatically able to follow a moving resonant peak.

4 Simulating a Changing System

The logic will be demonstrated by introducing a 2nd-order system with a clear resonant peak changing with time. The unknown frequency response plot of the system as it changes is plotted.

For simplicity of visualization, the pulse width setting will be fixed at 50% based on an assumption from previous testing that this will yield the highest flow gains. To show

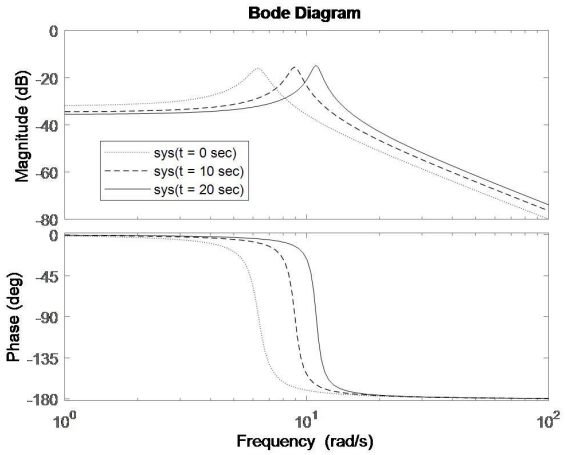

Fig. 8. Bode Plot of Unknown Time-Variant System

how each system will respond to different pulse frequencies, the response plots for each system state are shown.

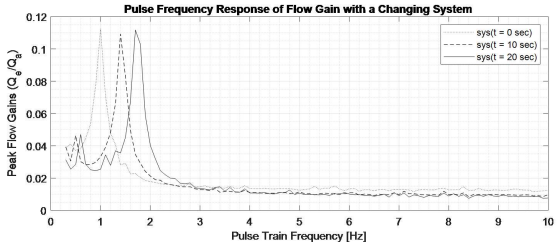

Fig. 9. Unknown Time-Variant System Response to Different Pulse Train Settings

From this plot, it is shown that the resonant frequency for this system is increasing. Now the logic should follow this change. For the simulation, a history vector of the optimal frequency will be taken.

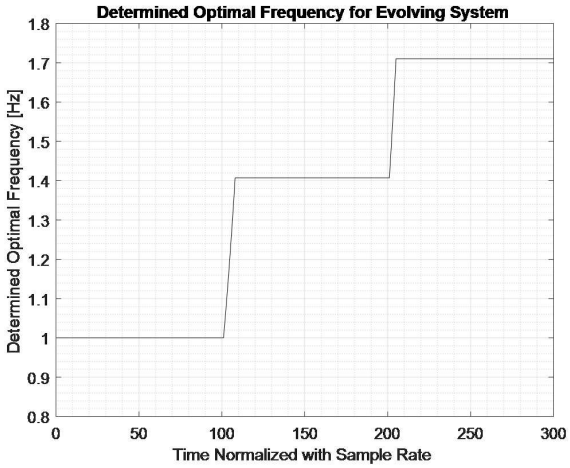

Fig. 10. Pump Frequencies Adapting to Match Resonant Peak of Time-Variant System

As expected, the change in determined optimal frequencies match the unknown resonant frequencies of the changing system. While the system is constant, the best frequency holds steady because both of the probes cannot find a better flow gain response, so the best frequency holds its status. Notice how there is a slight time delay for the logic to reach this steady state. This is dependent on how aggressive the probes are set to be. That is, how far away from the current frequency are they willing to check? This time should be assumed negligible if the changes are gradual instead of abrupt, as shown in this simulation.

5 Conclusion

The goal of this design was to implement logic-control on a variable displacement pump to maintain resonance to a changing system. It was designed and built around a sample static 4th-order system to test methods and was then simulated on a changing 2nd-order system. The logic implementation was successful, as the controlled pump was able to track the changing resonance frequencies and adapted its pulse train to maximize flow gains. Performance of this system can be improved by updating the scanning range as a function of the magnitude of the difference between the optimal and probe frequency. In this way, the logic-controlled pump can adapt more cautiously or aggressively depending on the severity of the temporary error of input frequency and true optimal frequency.

6 References

[1] Hullender, David A.; Snyder, Natalie N.; Gans, Jan C. Application of a Linear Model for Simulating Hydraulic Systems Containing Internal Lines with Turbulent Flow. 2017, Oct. 16 -17. ASME Symposium on Fluid Power and Motion Control, FPMC2017 4202.