Monthly Report (Yamamoto Lab.)

date: 2018.Apr.27
Author: R. Oechslin (M2)

Research theme: **Haptic Feedback Controller with Palm Pressurization**

— Research Plan —

Term \Month	2	3	4	5	6	7	8	9	10	11	12	1
Literature review												
Design PlayStation Controller												
Test PlayStation Controller												
Frequency Response Analysis												
Design Pilot Controller												
Test Pilot Controller												
Theoretical Analysis												
Analyze data and compare												
Write Thesis												

— Work Contents —

1 Introduction

This report is the continuation of the first report about the project "Haptic Feedback Controller with Palm Pressurization". The last report has left off with the conclusion that the Arduino had a limited operating frequency and that the gathered data was not reliable enough, since not the whole region of interest in frequency could be covered. First, the idea was to use an mbed and program it to be able to replace the Arduino. However, with a few tricks it was possible to reduce the time of some commands to a minimum to stay at an operating frequency of 1kHz.
This report explains the setup and states the result for the adapted controller and provides a somewhat short explanation and discussion of the gathered data.

2 Experiment and Data Gathering

The setup can be seen in figure 1. For this experiment, a total of four springs, arranged

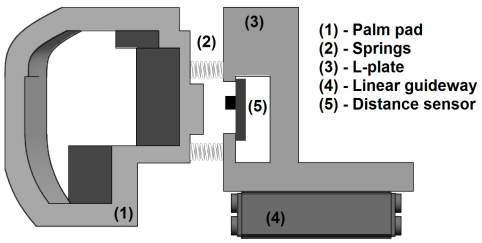

Figure 1: Schematic of the spring system setup.

symmetrical on the palmpad have been used. The springs have a spring constant of $k_s = 1.5\text{N/mm}$ which corresponds to an equivalent spring constant of $k_{eq} = 6\text{N/mm}$. They are distributed around

the palm pad, where in their middle a distance sensor (called photoreceptor) has been attached, to measure the distance between the L-plate and the palm pad. It therefore measures the compression of the springs, which can be related to the output force by Hooke's law as:

$$F = k_{eq}x \tag{1}$$

Setup

In this experiment a thorough frequency response analysis shall be done on the controller. On the left-hand side the motor with a reduction ratio of 33 : 1 has been used, whereas for the right-hand side, the motor has a reduction ratio of 112 : 1.
A reference signal is fed into the Arduino, which then controls the motors to match the compression of the springs with the reference. The operational distance of the photoreceptor to the palm pads is 2 to 4mm which lies within the more sensitive region of the sensor.

Control Scheme

The reference signal is given by the Function Generator SG-4115. This generator has an intrinsic output impedance of 50Ohm. This means, that it expects to have a device connected to it with the same value as input impedance. If this is the case, these two elements form a simple voltage divider and only half of the voltage is applied to the target device. However, this is not the case for the Arduino, since it has a considerably higher input impedance. Therefore, the settings made on the function generator result in double the voltage on the Arduino. From this point on, this issue shall be neglected and all future voltage indications refer to the voltage level as seen by the Arduino.
The function generator produces a sine wave between 0 and 5V with a frequency ranging from 1 to 100Hz. The Arduino reads this voltage and controls the motor to have a proportional spring compression accordingly. In this case 0V as reference signal is 0% compression and 5V corresponds to 100% compression. The control scheme of this setup can be seen in figure 2. In this case the

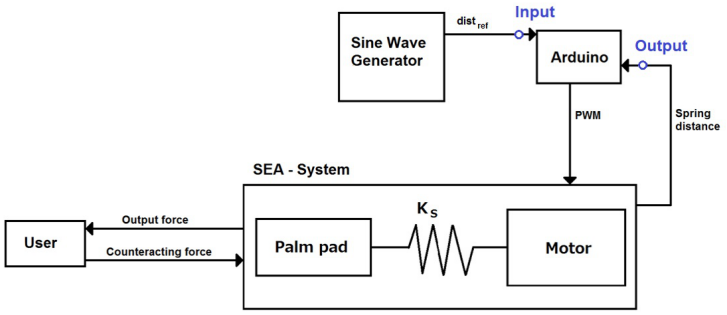

Figure 2: Control scheme for the frequency response analysis.

counteracting force from the user is infinite, since the palm pads have been blocked by a wall.

Photoreceptor Circuit

The circuit of the photoreceptor can be found in figure 3. The components chosen are the TPR-105 for the sensor, $R_1 = 330\Omega$ and $R_2 = 27k\Omega$.

The photoreceptors working principle is based on detecting the amount of reflected light. This controls the base current i_B of the transistor in the schematic. This base current then determines

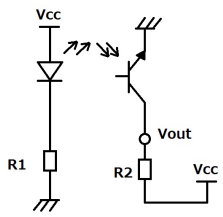

Figure 3: Photoreceptor circuit.

the collector current i_C. Specific to this setup, the distance and therefore the output force controls the amount of light that is reflected on the wall of the palm pad. Therefore, we have:

$$F_{output} = k_{eq}\Delta x \propto i_B \tag{2}$$

$$i_C = h_{FE} i_B \tag{3}$$

$$V_{out} = V_{CC} - h_{FE} R_2 i_B = V_{CC} - K R_2 \Delta x \tag{4}$$

Where h_{FE} is the forward current gain and K is a constant given by $h_{FE} k_{eq}$.

The two resistor values have been empirically found to have the highest sensitivity but not saturating the measurement. The sensitivity decreases with a smaller resistor, since at a certain point, the Arduino cannot detect a change in voltage anymore. Saturation occurs, when the value of R_2 is too high.

Identification of Operational Range (Photoreceptor)

To find the receptor values at maximum compression of the springs, a simple test has been made, where one time 0V has been applied to the motors, and another time 20V has been applied. This 20V value is within a safety margin of the maximum applicable voltage given by the datasheet of the motors. The result can be seen in figure 4.

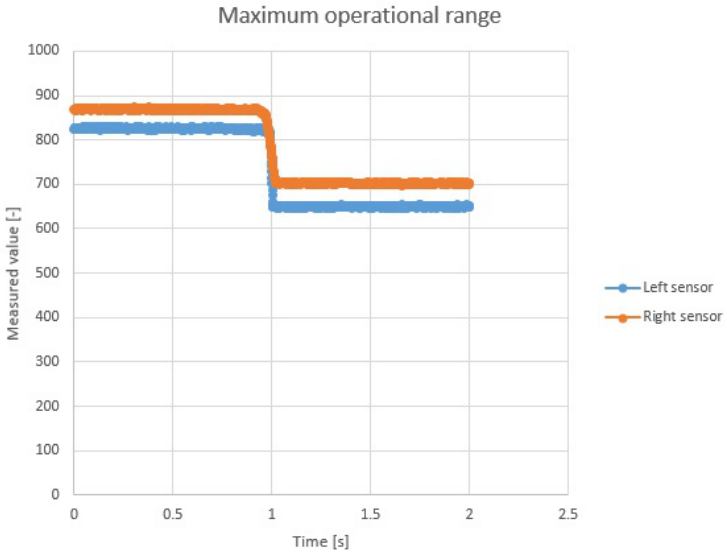

Figure 4: Operational range of photoreceptor with springs at rest and in full compression.

The values that have been found to be the limiting values are resumed in table 5.

Using these values, the photoreceptor measurements can be mapped to this range with an 8-bit value. This means that 0 is no compression and 255 is fully achievable compression.

Author: R. Oechslin

	Sensor reading MAX (rest)	Distance (rest)	Sensor reading MIN (compression)	Distance (compression)	Max output force
Left side	830	2.35mm	650	4mm	9.9N
Right side	870	2.2mm	700	4mm	10.8N

Figure 5: Identified operational range

3 Frequency Response Analysis

To conduct the frequency response analysis in a meaningful environment, the controller has to be tested under conditions close to the operational ones. To be able to control the distance between the palm pads and the L-plates, and therefore the compression ratio of the springs, the palm pads have been blocked mechanically. With the wave generator producing the reference signal, the Arduino controlled the motors to match the compression with the reference. Figures 6 to 8 show some testcases.

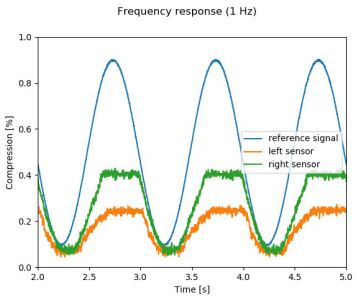

Figure 6: 1 Herz

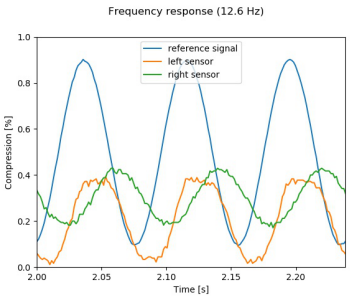

Figure 7: 12.6 Herz

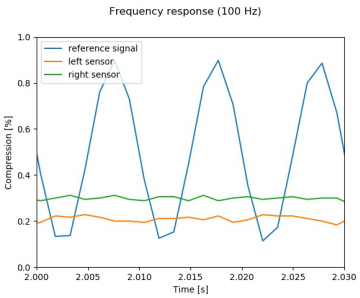

Figure 8: 100 Herz

With these signals one can calculate the amplification factor between the reference signal and the output signal, as well as the phase shift.

Sinewave Fitting

As described in this example[exnumerus, 2010], one can fit a sinewave with the linear least square method to the samples. This function returns the phase, amplitude and the bias. In this case, the difference in phase between input and output, as well as the amplitude ratio is needed in order to plot a Bode diagram.

Bode Diagram

The results of the frequency response analysis are shown in figures 9 and 10.

For the left-hand side, a resonance top at around 13Hz can be found, with a gain of roughly -12dB for lower frequencies. At higher frequencies a slope of roughly -26dB/dec has been calculated. For doing so the values in the range of 50Hz and 100Hz have been used. The phase shifts from $-25°$ to $-222°$, a total shift of roughly $200°$.

For the right-hand side, no resonance top is visible, and a constant gain of roughly -6dB for lower frequencies can be identified. At higher frequencies a slope of roughly -30dB/dec has been calculated. Again using the values in the range of 50Hz and 100Hz. The phase shifts from $-25°$ to $-223°$, a total shift of roughly $200°$.

4 Discussion

As it can be seen in figues 6 to 8 the signal is not followed very smoothly. For one, the magnitude ratio even at low frequencies is at roughly 0.5. This shows that the proportional gain is

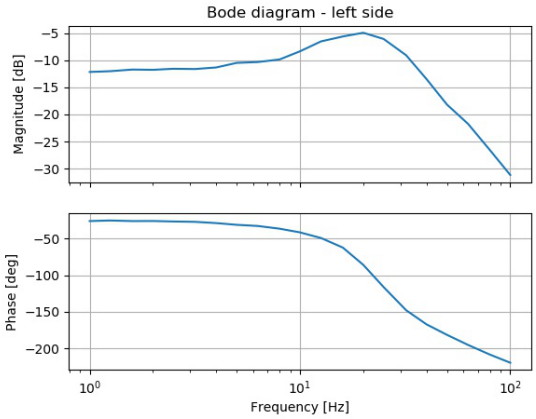

Figure 9: Left-hand side, Motor reduction ratio 33 : 1

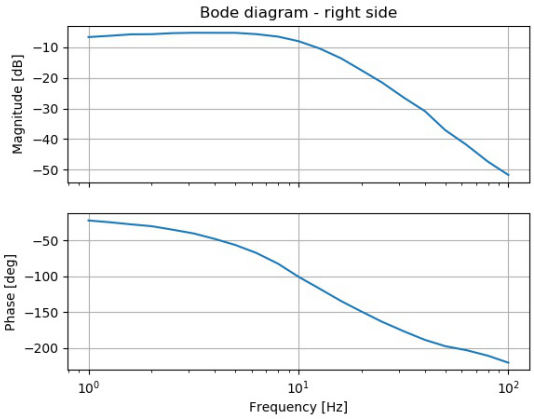

Figure 10: Right-hand side, Motor reduction ratio 112 : 1

too low. This would not only lift up the curve indicated in the Bode diagram and shift it towards higher frequencies (to the right). Another important aspect is the saturation behavior of the right motor. The origin of this effect needs to be investigated in more detail.

From the Bode diagrams one can conclude several things. First of all, the gain magnitude at lower frequencies should be around 0dB to have perfect following of the reference signal. This can be influenced by tuning the proportional value of the controller.

Second, the two motors show different characteristics. While the weaker motor (left-hand side) shows a resonance top around 13 Herz, the stronger motor has no such behavior. In order to have a more linear system, it is therefore favorable to use the stronger motor for this spring system.

Furthermore, the first pole of the system can be found around 13 Herz. The communication frequency between robot and graphical user interface is suggested (according to the datasheet of the robot) to be between 1Hz and 5Hz. This shows that the series elastic actuator system as it is implemented in this experiment, is not the limiting factor in the robotic system.

When one wants to determine the order of the system, one can look at the slope at the tail of the magnitude curve in the Bode plot. In our case they are -26dB/dec and -30dB/dec for left and right respectively. Since the slopes can only be a multiple of 20dB/dec, it can be concluded, that the $2\xi\omega_0$ term from equation 5 has a non-negligible influence at the tail of the Bode plot. The equation of a second order transfer function is as follows:

$$H(s) = \frac{\omega_0^2}{s^2 + 2\xi\omega_0 s + \omega_0^2} \tag{5}$$

Therefore one should also have a look at the phase lag diagram. The phase shift of roughly 180° suggests a second order transfer function.

The only thing that might risk the reliability of the Bode diagrams is the fact that the output signal has shown to be electrically interfering with the input signal. This most probably comes from the fact that the output impedance is too high.

5 Conclusion

The Bode diagrams show promising results. This is a first hint that the implementation of the series elastic actuators is not slowing down the system, since the bottleneck is given by the robot GUI communication link.

6 Outlook

The input signal is interfering with the output signal on the Arduino. Therefore it is necessary to investigate its impact by implementing a voltage follower to reduce the output impedance.
A theoretical model shall be calculated to take into account the different aspects and parameters of the setup. The goal is to come up with a precise model to simulate different parameter settings, such as equivalent spring constant, motor parameters or operating frequency.

References

[exnumerus, 2010] exnumerus (2010). How to fit a sinewave, an example in python. http://exnumerus.blogspot.com/2010/04/how-to-fit-sine-wave-example-in-python.html.