ANALYSIS OF MEDICAL ROBOTIC BED CONTROL SYSTEM TO PREVENT PRESSURE SORES

Nathan M Daniel[1]

[1]Purdue University, West Lafayette, IN

ABSTRACT

Pressure ulcers, commonly referred to as pressure sores, are injuries on the skin and underlying tissue caused by prolonged pressure on skin. Patients who stay in bed for long durations can be prone to pressure sores, even in hospitals. Three engineers conceptualized a robotic bed with multiple segments that are driven by brushless DC motors that are controlled by the patient to alleviate the amount of pressure on their body. The bed also contains force pressure sensors that constantly monitor the amount of pressure on the patient. Using closed-loop control, the patient adjusts the position of the bed with a pendant to move to a more comfortable position.

Keywords: medical robotics, pressure sores, hospital bed

1. INTRODUCTION

It is estimated that 70% of individuals turning 65 will at some point require long-term hospital care in their lifetime, for an average of 3 years, likely due to physical impairment (The Need for Long-Term Care Continues to Grow, 2016). Patients who stay in hospital beds for prolonged periods of time are likely to develop some form of pressure sores at some point, which in mild cases cause patient discomfort and in severe cases hinder or even completely hinder their recovery. Common hospital beds are designed with features such as elevation at certain positions and side rails to aid patient comfort, but with extended hospital stays, the possibility of pressure sores remains.

A robotic bed (shown in **Figure 1.1**) created by three engineers (Seon, 2021) consists of 17 adjustable segments that can reorient the patient to reduce the possibility of a pressure sore. Each segment is actuated by a brushless DC motor to achieve its orientation. The bed also contains a pendant that is used by the patient to move the bed to a comfortable position using open-loop control, though this is an intrinsically flawed method of preventing pressure sores. This is because the patient will typically only be moving the bed to different positions after feeling sores on the body, as opposed to the bed moving on its own to anticipate when pressure sores could occur. Therefore, the control objective of this paper is to develop a system in which the bed will (subtly, without the patient feeling it) change orientations to prevent pressure sores over long periods of time in the bed.

FIGURE 1.1: Developed Robotic Bed

This theoretical system contains 17 inputs and 1 output: the pressure distribution of the patient across the bed (from each of the 17 segments), and the necessary orientation of the bed that would alleviate pressure best after a given amount of time, respectively. The physical system can be modeled using a sinusoidal equation, with each segment representing a "link" and the connection between each link being a joint. Therefore with 17 segments in the system, there will be 16 joints that are part of the sinusoid, shown in **Figure 1.2**. Force pressure sensors (FSRs) are placed throughout the bed (10 on each of the seventeen segments) to monitor the current pressure on the patient. Thus, given the pressure on the patient at each segment, the control system should adjust the orientation of the segments to disperse the patient's weight more evenly.

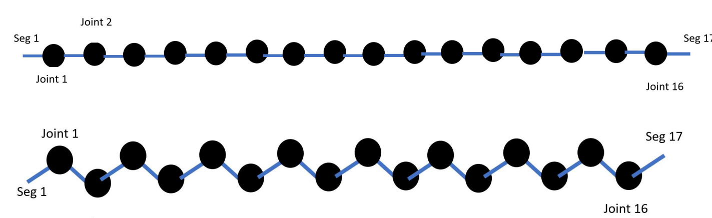

FIGURE 1.2: Sinusoidal Schematic of Physical System

One constraint on the system would be the maximum height (h) that each segment can reach, as certain orientations would make the patient uncomfortable, despite the position potentially reducing body pressure. Similarly, because there are 17 individual segments that are each adjusted using a four-bar link mechanism (**Figure 1.3**), the number of unique complete bed orientations is limited. Additionally, another constraint would be the maximum allowed patient weight, which is 300 lbs. in most hospital beds.

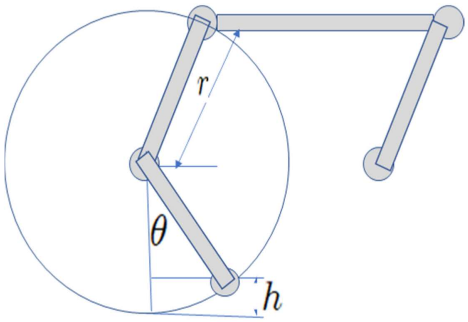

FIGURE 1.3: Kinematics of four-bar link

2. SYSTEM DYNAMICS WITH UNCERTAINTIES

NOMENCLATURE

$l_{segment}$	length of one segment (4.17 inch)
l_{total}	total length of bed
h_i	height of segment *i*
θ_i	rotation of segment *i*
A_{seg}	area of one bed segment
$\sum P_{odd}$	sum of average pressure in odd segments
$\sum P_{even}$	sum of average pressure in even segments
m	mass of patient
b	damping constant of patient
k	spring constant of
F_{body}	net vertical force of body on bed

The governing equations for the system are determined by the physical components and reference the above. For this system, we will assume that the length of the bed (l_{total}) is 80 inches (which is the standard size of a hospital bed in the United States [Different Hospital Bed Sizes and Why They Matter," SonderCare. (3)]), so each of the equal 17 segments will be 4.71 inches in length ($l_{segment}$). Additionally, the maximum each segment will rotate is 30° (θ), to account for the comfort of the patient.

If there is more pressure across the odd segments (i.e., starting from the superior end of the bed, a segment is numbered 1, 2, 3, …, 17 to the inferior end of the bed), then the bed will attempt to shift towards the sinusoid shown in **Figure 2.1.** However, if there is more pressure across the even segments, then the bed will attempt to shift towards the sinusoid shown in **Figure 2.2.** The positions were calculated using the known dimensions of the bed and basic geometry to create the two graphs.

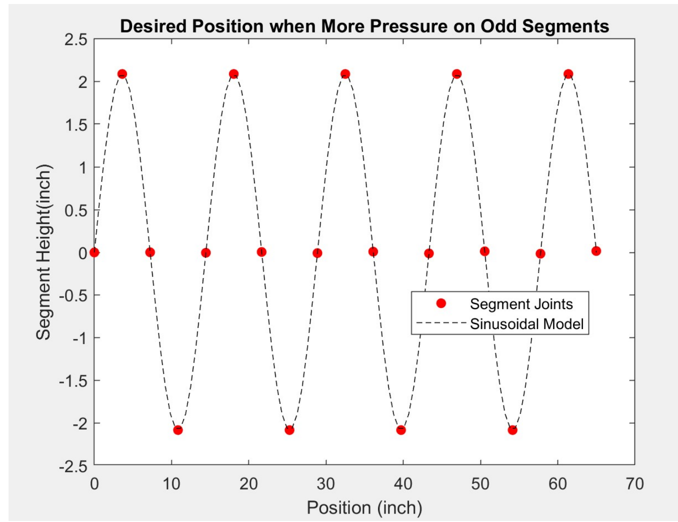

FIGURE 2.1: Desired position of the bed when there is more pressure on odd numbered segments

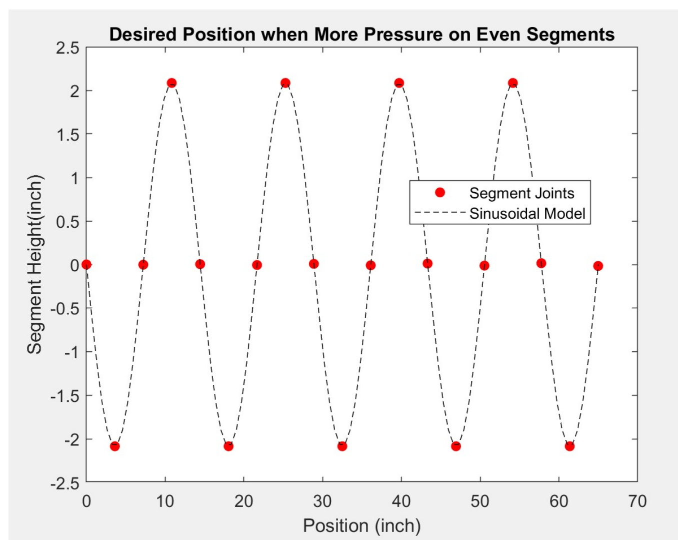

FIGURE 2.2: Desired position of the bed when there is more pressure on even numbered segments

The pressure sensors across the bed will be taking measurements as inputs, and the force at each segment (F_i) is calculated **(Eq 1).** In the system, the person's body will be represented by a mass-spring damper system shown in **Figure 2.3.**

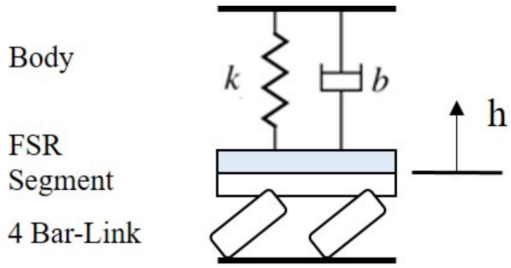

FIGURE 2.3: System Model for Control

The physical governing equations for connecting the input and output of the system are shown below. If there is more pressure on the odd segments, then we will use the sum of the odd pressures, whereas if there is more pressure on the even segments, we will use the even pressures.

$$F_{body} = A_{seg} \sum P_{odd/even} \tag{1}$$

$$F_{body} = \frac{bm}{k} y''' + my'' + ky + by \tag{2}$$

A source of potential uncertainty could be from the erratic movement of the patient. Because patients who are in hospital beds for long periods of time commonly suffer from conditions such as dementia and other debilitating disorders which can induce irregular or spasmodic motion that may interfere with the system response. This is why we are including the jerk motion s part of the system. The following block diagram shows the components involved in the control system.

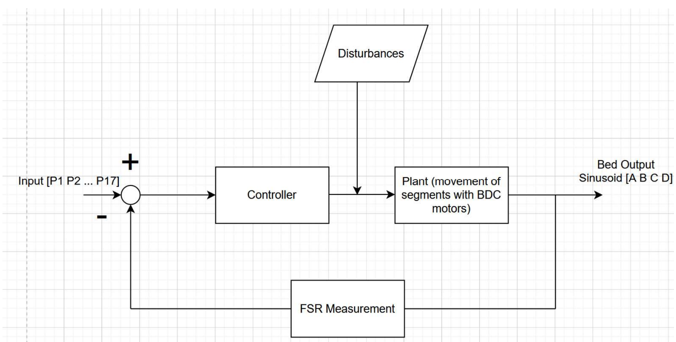

FIGURE 2.4: Block Diagram of System

Noise, disturbance, and uncertainty enters the system in equation (2), as it is where the movement of the patient enters the system. No control system could ever 100% effectively be tuned to work with all the possible combinations of movements that the patient could make, but given that most hospital patients remain in relatively similar poses during their stay, it can be ignored for now. Measurements that are feasible in the system are the pressure distributions across all of the FSR's in the bed, which could also be used to calculate the patient's weight. Disturbances would be measurable by considering very large changes in the pressure measurements (potentially by using a high pass filter).

3. LINEAR STATE SPACE AND TRANSFER FUNCTION REPRESENTATIONS

In order to linearize our model, we need a desired operating condition. For the starting condition, the patient will be an average male (around 62 kg, spring constant of 2,500 N/m [Measuring Body Weight, (4)], and damping constant of 0.7). Thus, the transfer function we have is:

$$G(s) = \frac{F_{body}}{\frac{bm}{k}s^3 + ms^2 + ks + b} \tag{3}$$

$$G(s) = \frac{608.2}{0.01736s^3 + 62s^2 + 2500s + 0.7} \tag{4}$$

The dominant pole is at approximately s=-0.00028, as shown in the root locus model in **Figure 3.1,** while the settling time is approximately 13,972 seconds. Additionally, the system is stable as all of the poles are negative.

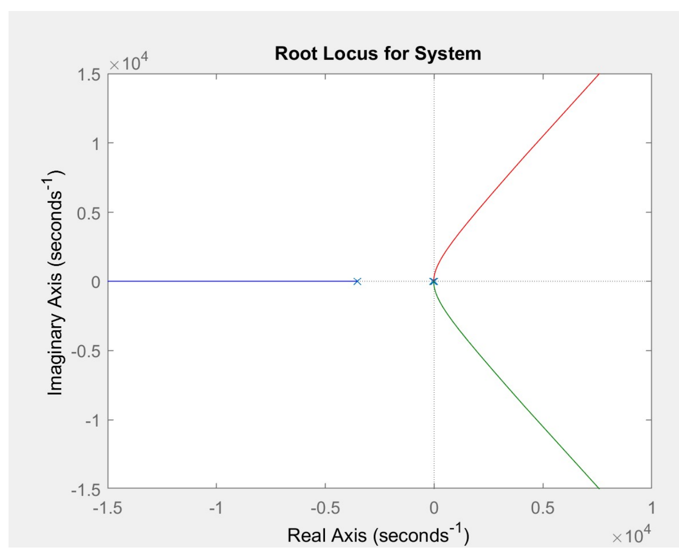

FIGURE 3.1: Root locus for linearized system

After running a simulation and analysis in MATLAB on the system, the following Bode diagrams and margin plots for the nominal plant were plotted.

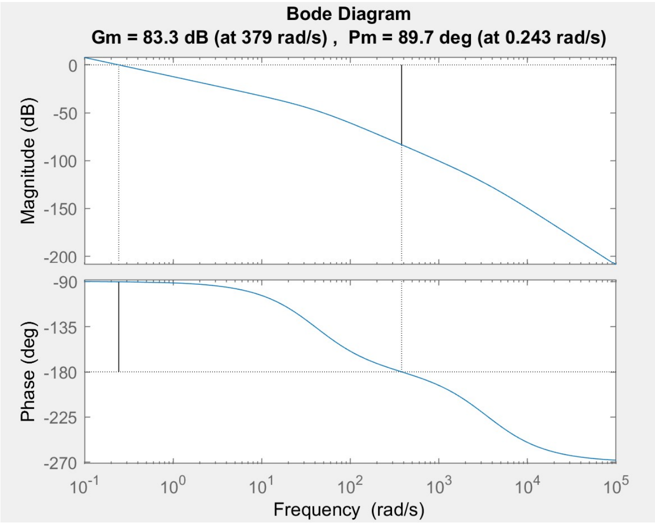

FIGURE 3.2: Bode diagram & margin plots for linearized system

Next, the trajectories for step responses with inputs (without feedback) were plotted using MATLAB. With these models, uncertainty was ignored, as the engineers creating the original controller used in the literature for this robotic bed project (Seon, 2021) were not aware of any uncertainty and chose to ignore it in their model.

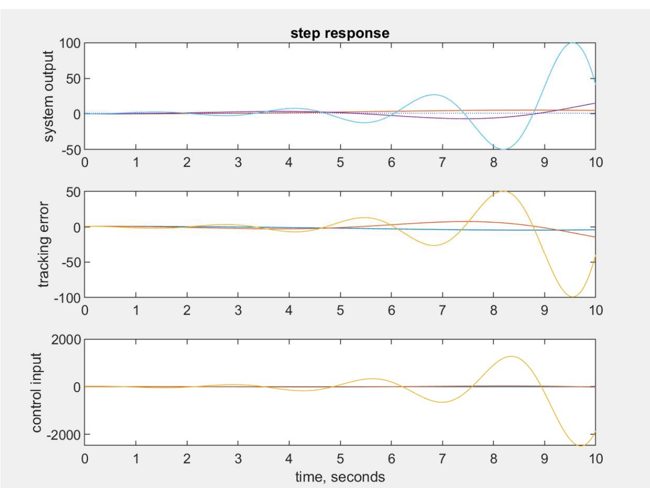

FIGURE 3.3: Trajectories from step response (u(t)=1)

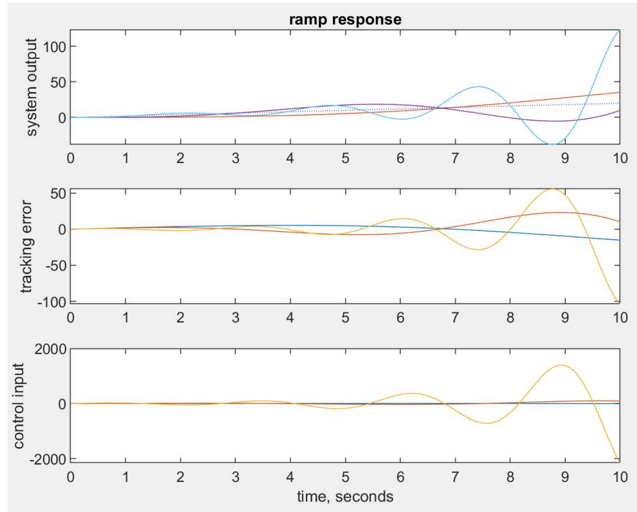

FIGURE 3.4: Trajectories from step response (u(t)=2t)

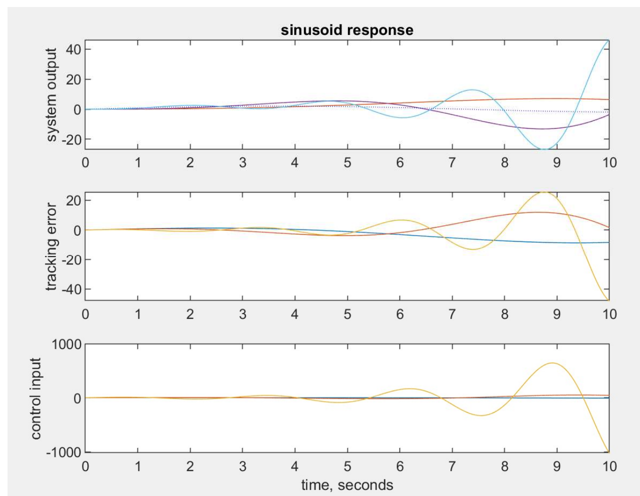

FIGURE 3.5: Trajectories from sinusoidal response (u(t)=2.085sin(0.435t))

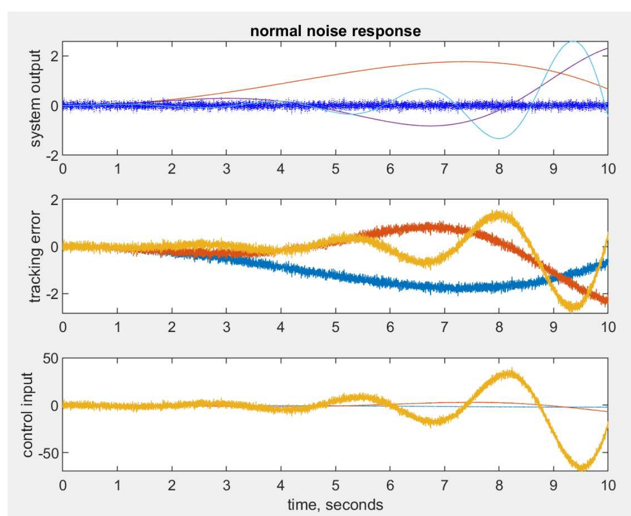

FIGURE 3.6: Normal Noise Response

Overall, the frequency response was stable which is a good sign for our model and system. However, the time response was very high (the settling time was nearly 4 hours). This is somewhat acceptable in the context of our problem, as the bed does move very slowly so that the patient does not feel discomfort from the segments constantly shifting. If anything, it would make sense for our model to reach its target at this time slowly.

4. STABILITY, CONTROLLABILITY, & OBSERVABILITY ANALYSIS

With the given transfer function, we will test the system for stability, controllability, and observability. We first begin by converting our transfer function to state space using MATLAB, giving us:

$$A = \begin{bmatrix} -3571.43 & -144009.22 & -40.3226 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$C = [0 \quad 0 \quad 35034.56\,]$$

$$D = [0]$$

Next, we will test for stability by solving for the eigenvalues of the state matrix A. Using MATLAB, the eigenvalues for the system are:

$$\lambda_1 = -310.81$$
$$\lambda_2 = -46.33$$
$$\lambda_3 = -0.0003$$

All three of the eigenvalues are negative and real, indicating that they have no oscillatory behaviors and are stable nodes in 2D. The next step was to apply Nyquist stability criterion to our system. The following Nyquist plot was obtained for our system using MATLAB:

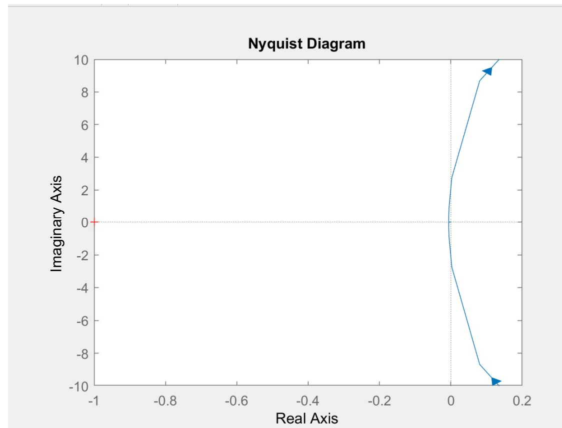

FIGURE 4.1: Nyquist Plot of System

As shown, the number of encirclements of (-1, 0j) is equal to 0 (N=0). Additionally, the number of unstable poles from the transfer function was found to be 0 from the root locus of the system in **Figure 3.1**. Therefore, the number of unstable closed-loop poles Z is 0, thus ensuring stability.

In order to test for controllability and observability, we first tested our system using the following MATLAB code to see whether or not the controllability and observability matrices had a rank equal to that of n, which was 3 in this case due to A being a 3x3 matrix:

```
%% Controllability & Observability

num = [608.2];
den = [0.01736 62 2500 0.7];
[A, B, C, D] = tf2ss(num, den);
sys = ss(A, B, C, D);

Co = ctrb(sys);
Ob = obsv(sys);

disp(rank(Co));
disp(rank(Ob));
```

FIGURE 4.2: MATLAB Code to Determine Controllability & Observability

From the MATLAB code, both the controllability and observability matrices had a rank of 3, meaning they were controllable and observable. The controllable canonical form and observable canonical forms are shown below, respectively (CCF denoted with C, OCF denoted with O):

$$A_c = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -40.32 & -144009.22 & -3571.43 \end{bmatrix}$$

$$B_c = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$C_c = [10.56 \quad 0 \quad 0\,]$$

$$D_c = [0]$$

$$A_o = \begin{bmatrix} 0 & 0 & -40.32 \\ 1 & 0 & -144009.22 \\ 0 & 1 & -3571.43 \end{bmatrix}$$

$$B_o = \begin{bmatrix} 10.56 \\ 0 \\ 0 \end{bmatrix}$$

$$C_o = [0 \quad 0 \quad 1\,]$$

$$D_o = [0]$$

Using our controllability and observability matrices, we can compute the Kalman decomposition of the system. The following MATLAB code was run in order to attain the Kalman decomposition:

```
kal_sys = minreal(sys);
[knum, kden] = tfdata(kal_sys);


kal_tf = tf(knum, kden);
[kA, kB, kC, kD] = tf2ss(knum{1}, kden{1});
```

FIGURE 4.3: MATLAB Code to Determine Kalman Decomposition

The results from the Kalman decomposition leads to the conclusion that the following subsystem is both reachable and observable:

$$\hat{A} = \begin{bmatrix} -3571.43 & -144009.22 & -40.3226 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\hat{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\hat{C} = [0 \quad 0 \quad 35034.56\,]$$

$$\hat{D} = [0]$$

In conclusion, the results from our system analysis show that our system is stable, controllable, and observable, which is very important when it comes to medical robotics. If the system was not, it could result in patient injury or even death due to an unchecked system.

5. CONTROL DESIGN VIA STATE SPACE METHODS

With the given state space model of our system from Section 4, we can apply state feedback laws.

$$A = \begin{bmatrix} -3571.43 & -144009.22 & -40.3226 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$C = [0 \quad 0 \quad 35034.56\,]$$

$$D = [0]$$

$$u = [k_1 \quad k_2 \quad k_3]$$

Solving for the characteristic equation $|sI - (A - BK)|$, we get:

$$s^3 + (k_1 + 3571.43)s^2 + (k_2 + 144009.22)s^2 + (k_3 + 0.3226)$$

If we desire to set our closed-loop poles to certain values, we can solve for the values of k_1, k_2, and k_3, though the closed-loop poles are already negative and stable. However, just for instance if our desired characteristic were of the form: $0.01439s^3 + 52s^2 + 300s + 0.3$, then our resulting input vector would be:

$$u = \begin{bmatrix} -576.04 \\ -126730 \\ -23.05 \end{bmatrix}$$

The corresponding value for the proportional input scaling to adjust output levels, k_r, is given by the equation shown below:

$$k_r = [C(BK - A)^{-1}B)]^{-1} = 0.001722$$

Next, we can design our state observers to obtain the full state estimate using separation property, since A, B, and C are known, A and B are controllable, and A and C are observable. Using the formula $|sI-(A-LC)|$, we can solve for l_1, l_2, and l_3 using a system of equations with our desired characteristic equation:

$$L=\begin{bmatrix}-194430\\55.1041\\-0.0164\end{bmatrix}$$

Now, combining everything together with our observer-based compensator, we get the following full state estimate:

$$A=\begin{bmatrix}-2995.4 & -17280 & 6.8119E9\\1 & 0 & 1.9305E6\\0 & 1 & 576.0376\end{bmatrix}$$

$$B=\begin{bmatrix}1\\0\\0\end{bmatrix}$$

$$C=[576.0376 \quad 1.2673E5 \quad 23.0456]$$

$$D=[-1]$$

$$u=[k_1 \quad k_2 \quad k_3]$$

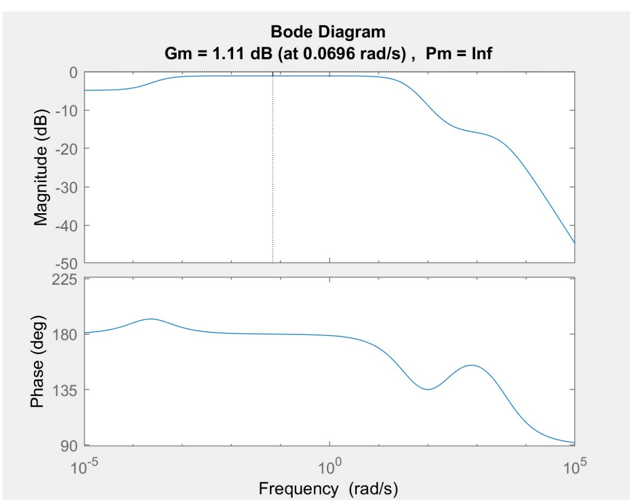

FIGURE 5.1: System Response with Respect to State Feedback Regulator

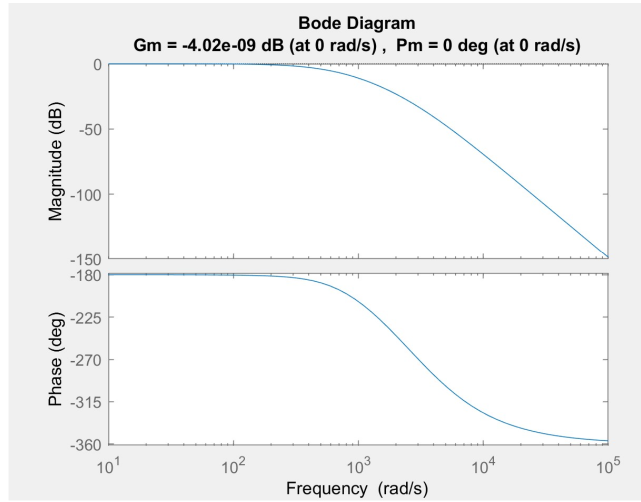

FIGURE 5.2: System Response with Respect to State Feedback with Observer

There are some potential practical issues with our design. Firstly, the method of taking the average of the odd pressure sensors and even pressure sensors is not always going to drive our system to the best bed "sinusoid" position that will prevent pressure sores at time *t*. However, as $t \rightarrow \infty$, we will hopefully have reached a point where it does not matter what the optimal position to move to is; instead, the changing of the bed positions will hopefully have prevented pressure sores from beginning to develop. Secondly, our full state estimate tells us that it takes a long period of time to switch states, and since we do not want our bed to be moving so fast that the patient feels the motion, so the bed must stay in each state for a certain period (around 2x$t_{settling}$). This ensures that the bed is achieving the full sinusoid position (based on even or odd) and not switching between the two states too quickly. This means that our pressure sensors must be taken within the right intervals, as clearly when the bed is moving towards a position, the even or odd segments that are "positive" in **Figures 2.1 & 2.2** will most likely take the brunt of the body pressure close to the settled position. Thus, the sensor reading intervals can affect the system performance, making it different than the linearization.

6. CONTROL DESIGN VIA ALGEBRAIC METHODS

In this section, we will attempt to place poles via polynomial pole placement to attain our control objectives. Preferably, we would decrease the settling time to something between 30 minutes – 1 hour, which is what we will attempt to do with the inclusion of additional left hand poles at s = -30 ±j20, s = -20 ±j10, and s = -60. We do this using the Diophantine Equation, getting our desired closed loop equation D_{CL} and solving for our unknown controller coefficients. As our plant is a third order system, we needed to add at the least 5 additional poles. Thus, we have the following equation:

$$D_{CL} = (s + 60)(s + 30 - j20)(s + 30 + j20)(s + 20 - j10)(s + 20 + j10)$$

$$D_{CL} = s^5 + 160s^4 + 10200s^3 + 334000s^2 + 5570000s + 39000000$$

Next, we need to decide the form of our controller, *C(s)*. As there are 5 new desired poles, we need 6 coefficients to be in our controller. Thus, our controller is of the form below:

$$\frac{as^2 + bs + c}{ds^2 + es + f}$$

Using our plant equation *P(s)* and setting everything equivalent to D_{CL}, we now have our controller:

$$C(s) = \frac{-69950041.16s^2 - 2853149735s - 734760.91}{57.604s^2 - 196510.86 + 694116550.69}$$

Our math can be checked using MATLAB, checking the pole zero map:

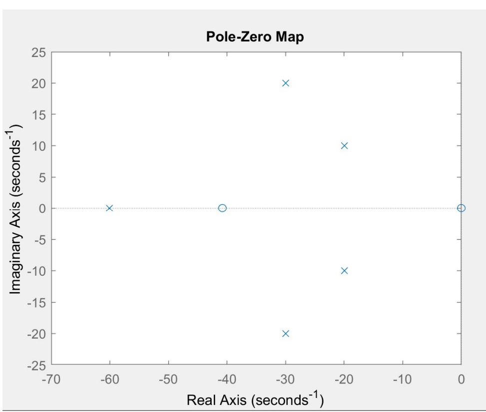

FIGURE 6.1: Pole-Zero Map of Transfer Function with Added Poles

Now, checking the step response information on MATLAB, we can see that our settling time is 2,947 seconds, or around 49.12 minutes, which is exactly within the range we set out to achieve of 30 minutes – 1 hour.

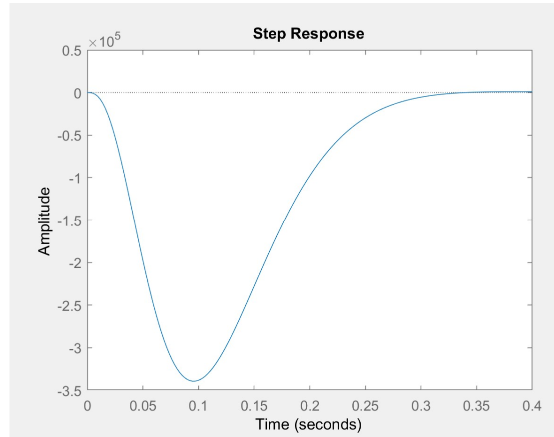

FIGURE 6.2: Step Response of Full Transfer Function

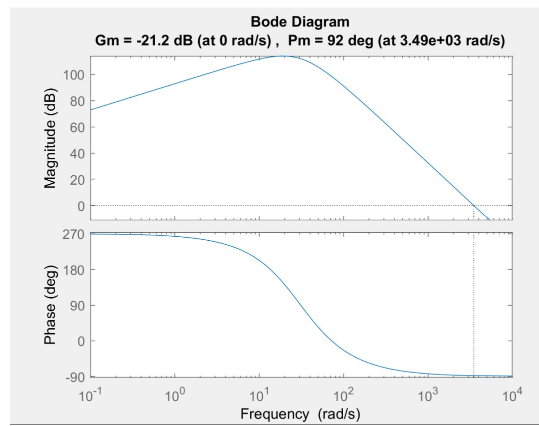

FIGURE 6.3: Frequency Response of Transfer Function

As compared to the full state feedback model, we have a lot more overshoot in our polynomial design. In practice, this does not have much of an effect, as there will be limits on the rotational angle that each bed segment will be able to attain. Thus in actuality, we will simply be achieving our desired bed positions from **Figures 2.1 & 2.2** faster with the overshoot, though the issue with that will be between the overshoot and our desired target in which the bed moves away from the desired bed positions. In comparison to the full state feedback model however, the settling time is much faster for our polynomial design (approximately 4 hours vs. 50 minutes, respectively). However, the lowered settling time does not affect the observability nor controllability, but it does change the system performance compared to the linearization. Thus, sensor readings should take place before the controller but after the plant, as it should technically take a little big longer to attain the desired bed position this way (and with the settling time of 50 minutes, we can afford this).

7. CONTROL DESIGN VIA FREQUENCY DOMAIN

In this section, we will assume that we have access to measuremetns of the tfull state of the system, and will attempt to attain our design objectives with compensators designed through frequency domain methods. For our system, we will attempt to do this with the addition of simple lead-lag compensators, using the same desired poles as the previous section. For our system, we have a desired settling time of ~50 minutes (3000 seconds), and using our damping frequency we can calculate the desired poles to be at $s = -0.0013 \pm j0.00064$. Now, we must run a root locus analysis using the angle criterion to determine our controller. First, a pole-zero map is generated with our desired poles inserted:

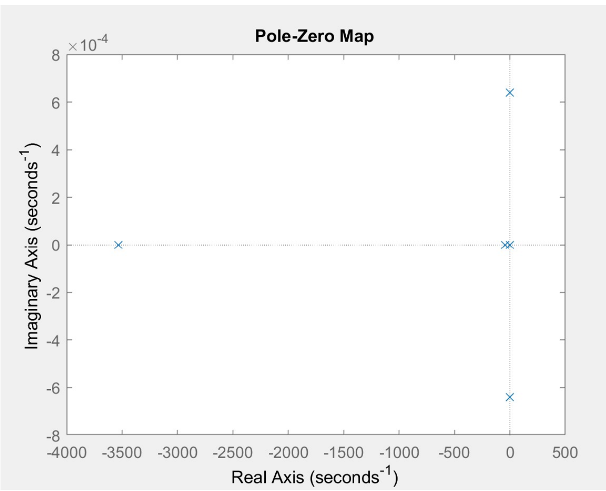

FIGURE 7.1: Pole-Zero Map with Desired Poles

Next, we must use the angle criterion to determine our compensator. The desired zero is calculated below:

$$-180 = \phi_D - \angle(s - p_1) - \angle(s - p_2) - \angle(s - p_3)$$

$$-180 = \phi_D - \angle(s + 3.53E3) - \angle(s + 40.8) - \angle(s + 0.00028)$$

$$\phi_D = 2.813$$

With our known value for the defect angle ϕ_D, we can calculate the desired zero to use in our controller using basic trigonometry. This zero comes out to be at $s = -0.01433$, meaning our compensator will be of the form:

$$C(s) = K(s + 0.01433)$$

In order to calculate gain K, we use the magnitude criterion, shown below:

$$K = |L(s)|^{-1} = \frac{|s - p_1||s - p_2|s - p_3||}{608.2|s - z|}$$

$$K = |L(s)|^{-1} = \frac{|s + 3.53E3||s + 40.8|s \pm 0.00028||}{608.2|s + 0.01433|}$$

$$K = 236.79$$

Therefore, our compensator is $C(s) = 236.79(s + 0.01433)$.

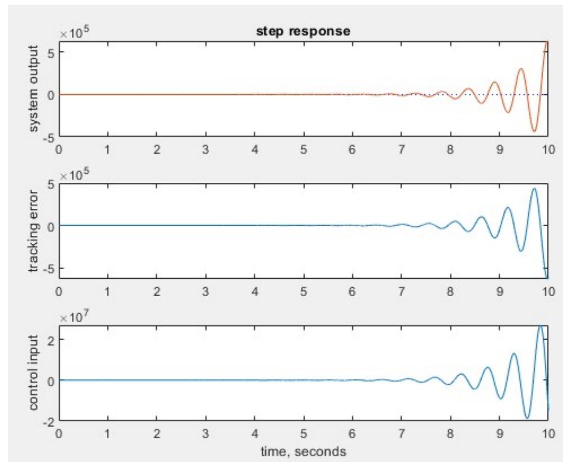

FIGURE 7.2: Plot of Full System with Step Response

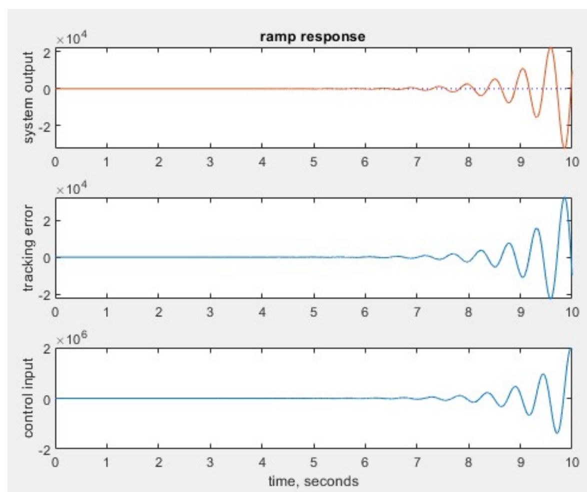

FIGURE 7.3: Plot of Full System with Ramp Response

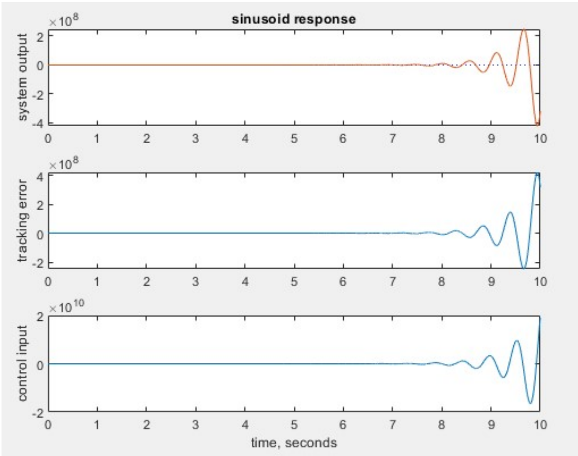

FIGURE 7.4: Plot of Full System with Sinusoidal Response

There are some issues with the frequency design method, notably that it is an "ideal" controller, or it is improper with a relative order of -1. The real-world implementation would require a low-pass filter to be added to the compensator, meaning that the system performs different in actuality than the linearization.

8. OPTIMAL CONTROL DESIGN

In this section, still assuming that we have access to measurements of the full state of the system, we will attempt to achieve our desired tracking and disturbance rejection objectives with linear quadratic control. For our purposes, we will be using LQR with optimal observers. LQR works as we know that our system is completely controllable, meaning we can drive the system from one state to another in a finite amount of time. With our given system, a suitable performance index to be minimized is of the form:

$$J = \int_0^\infty x^T(t)Qx(t) + u^T(t)Ru(t)dt$$

This performance index is chosen, as it requires the minimization of the square input u and minimization of the square of x, meaning that we are minimizing the energy required to control the actual system. The chosen weighted matrices are shown below:

$$Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$R = 1.0E - 8$$

After plugging the weighted matrices and state response into MATLAB, the following optimal gain is calculated along with the requested control law:

$$K = [0.0971 \quad 346.7825 \quad 0]$$

$$u^*(t) = -[0.0971 \quad 346.7825 \quad 0]x(t)$$

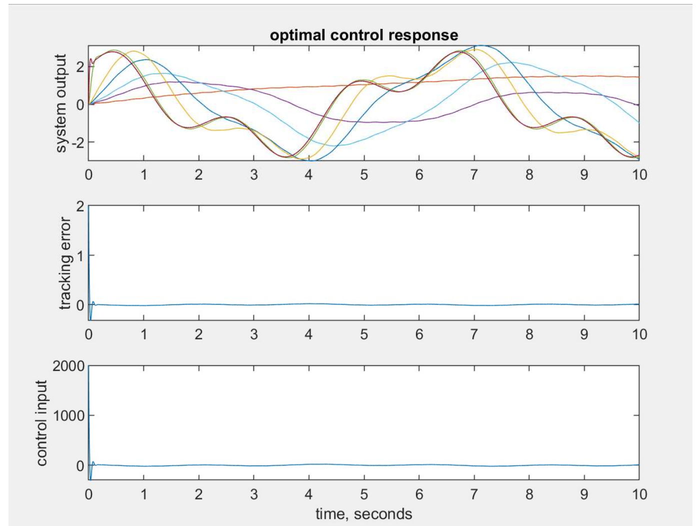

FIGURE 8.1: Plot of Optimal Control System Response, Tracking Error, & Control Input

The main issue with this optimal control design method is that the optimization criterion are not fully understood (though we assumed so), so in reality we are not really applying the most suitable optimization technique possible. This is seen in the optimal control system response, which is quite unstable, despite our optimal poles being -40.9, 0, and -353.06, respectively. This essentially makes it impractical to be used realistically in our final design, as the system performance was much better in the linearization.

9. TESTING ROBUSTNESS OF DESIGNS UNDER UNCERTAINTY

One possible source of uncertainty in our system could come from erratic or sudden forces on the bed, which could be a result of something falling on the bed surface or from spasmodic motion of a patient if he or she potentially suffers from a debilitating disorder such as dementia. Other sources of uncertainty include noise and disturbance, which would enter the system through the governing equations (specifically Equation 2). These uncertainties would be measurable to the system as well, though we must first test to see the robustness of our control designs under uncertainty. After simulating noise as a source of uncertainty, the following system output, tracking error, and control input responses were found, as well as the frequency response and Nyquist plot, using the controller design via frequency domain methods shown below:

$$C(s) = 236.79(s + 0.01433).$$

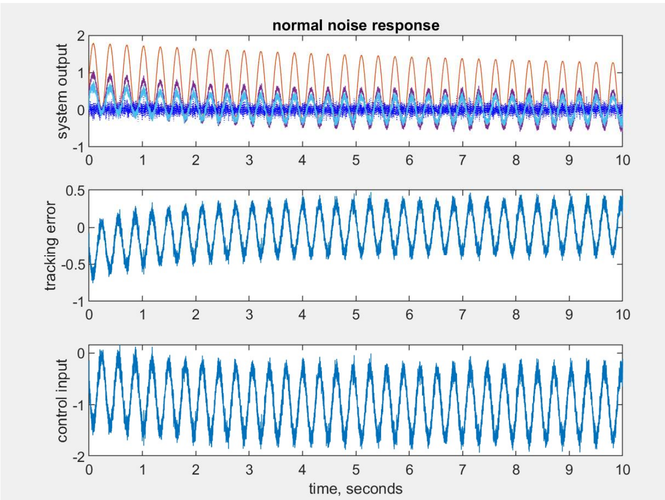

FIGURE 9.1: System Response, Tracking Error, & Control Input with Uncertainty

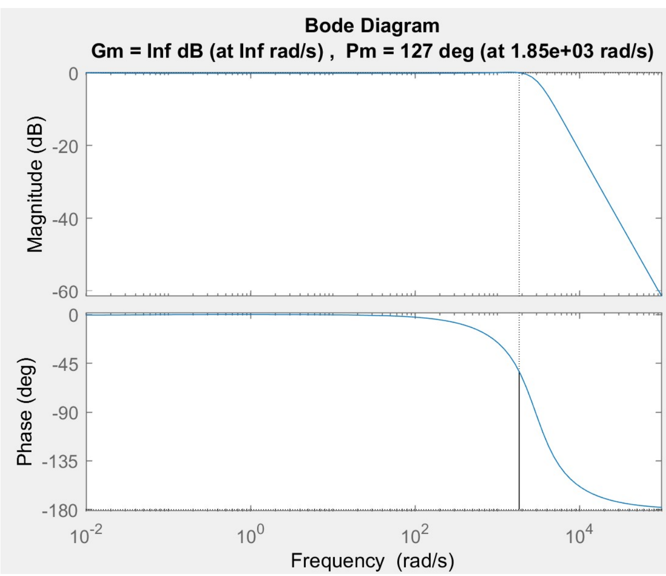

FIGURE 9.2: Bode Diagram for System Under Uncertainty

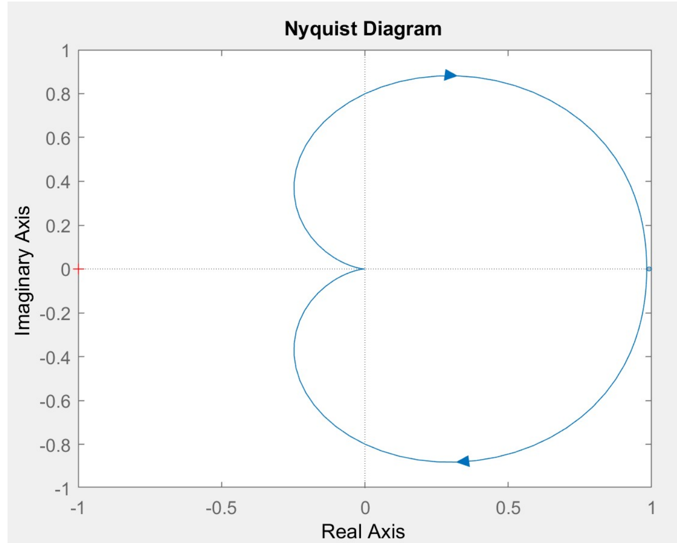

FIGURE 9.3: Nyquist Plot for System Under Uncertainty

The open loop transfer function *K(s)G(s)* of our system has zero right half plane poles, and the Nyquist criterion would state that in order for our system to be robust, the Nyquist plot must circle the point -1 that number of times for our closed loop system to remain stable. However, as seen in **Figure 9.3**, the Nyquist plot encircles the point -1 zero times, which leads us to believe that our controller is robust.

In conclusion, the performance of our control system designed is robust enough to the point where the bed fixture could be created for testing as per the rules outlined in this report (obviously, there must be a lot more testing conducted before it can actually be used in hospitals with real patients, or before human testing is even considered). The robustness tests do show promising results, though I would argue that even if it did not, the effect of uncertainty in the system would not have a devastating effect on the patient – given that there would definitely be a constraint on the maximum acceleration of the bed movements, it would be physically impossible for the bed to harm the patient through its oscillations.

10. IMPLEMENTATION ON HIGH FIDELITY MODELS

In this last section of our report, we will apply actual constraints on the control inputs and states to obtain a realistic simulation of the system using the actual dynamics.

The input constraints were on the maximum and minimum positions (as per **Figure 2.1** and **Figure 2.2**, in which the maximum and minimum heights of an individual segment was ±2.355 inches, so that the maximum θ is 30°. Additionally, the maximum velocity of the bed segments was set to 0.0785 inch/sec, or 0.003323 cm/sec, in order to ensure patient safety, and so that the quickest time to settle is below the actual settling time of the system. The results are shown below:

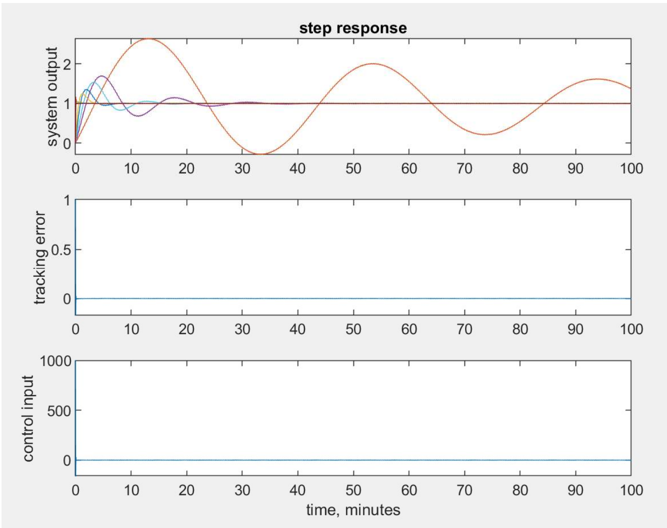

FIGURE 10.1: System Response, Tracking Error, & Control Input with Step Input and Input Constraints Applied

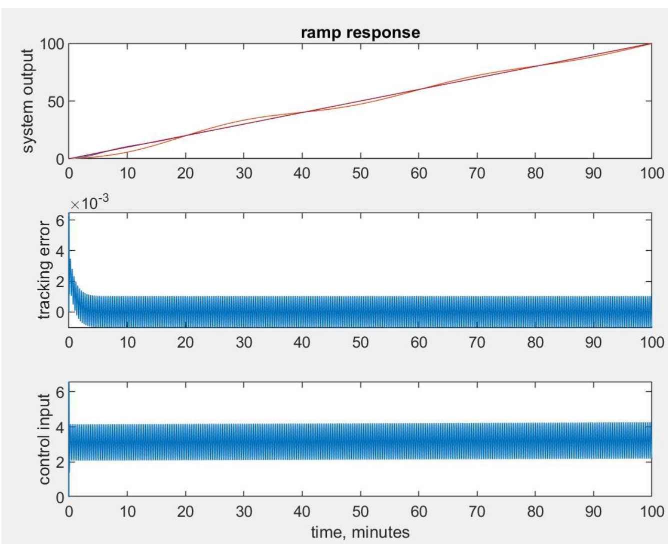

FIGURE 10.2: System Response, Tracking Error, & Control Input with Ramp Input and Input Constraints Applied

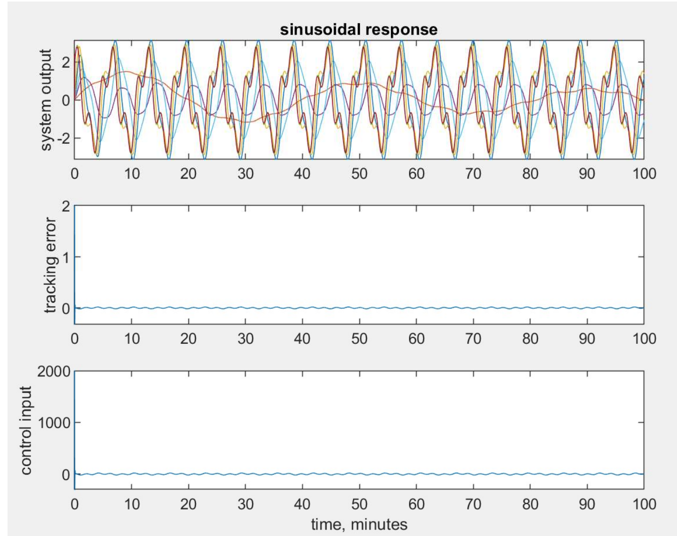

FIGURE 10.3: System Response, Tracking Error, & Control Input with Sinusoidal Input and Input Constraints Applied

In conclusion, our control design manages to fulfill the requirements even with the constraints applied to the system. The only real issue is that there is a small dip in the system output with a sinusoidal input when the system is trying to settle. Though this dip exists, it is not large enough (and the system response is not fast enough) for it to really cause any issues, though it is something that is not ideal in the real world application. This issue could be solved by studying this response and potentially adjusting the location of the open loop zeros.

However, despite this small dip, our system still performs well, settling within the right amount of time (between 30 minutes to 1 hour), and the velocity does not go over the maximum value of 0.003323 cm/sec. Additionally, it seems to be correctly alternating between the ‘even’ and ‘odd’ configurations shown in **Figures 2.1 & 2.2**, meaning that theoretically as the patient is shifting his or her weight, the bed should be driving towards the configuration that distributes the weight best, which should in turn hopefully prevent pressure sores. Obviously, there is much more testing that can be done before we can put these beds in hospitals, and there could be a benefit of including the open-loop controller from the first prototype so that the patient can manually adjust the bed orientation as he or she sees fit. Despite this, the system outlined in this report would still be immensely helpful as the patient would not have to always control the orientation of the bed, such as when he or she is sleeping.

REFERENCES

[1] Seon, M., Lee, Y., and Moon, C., 2021, "Medical Robotic Bed to Prevent Pressure Sores," *Applied Sciences Journal.*

[2] "The Need for Long-Term Care Continues to Grow," *Pennsylvania Health Care Association.*

[3] "Different Hospital Bed Sizes and Why They Matter," *SonderCare*.

[4] "Measuring Body Weight: Chapter 30," *Open Oregon Educational Resources.*

[5] Simon, D. "Analyzing Control System Robustness," *Cleveland State University Department of Electrical Engineering.*