

Discrete Control of an XY nano-positioning flexure stage

Gabriel Antoniak

Daniel Schulman

Taylor Sun

I. INTRODUCTION

Market demand for higher performance, increased functionality, smaller size, and lower prices in electronic chips has continued to push the industry towards smaller linewidths (currently at 7nm node) and more complex chip designs. Recent developments in scanning probe microscopy, scanning probe nanolithography, the semiconductor industry, and biological imaging require high resolution (< 10 nm) and large range (> 1 mm) nanopositioning stages to expand substrate area size for efficient testing, imaging, and manufacturing. The low throughput inspection of defects on silicon wafers is currently a bottleneck for semiconductor manufacturing yield. This impacts the cost and reliability of electronics used in computers, mobile devices, automobiles, robotics and automation.

Currently, the market relies mostly on XY stages that are comprised of magnetic bearings. Flexure bearings are not only a cheaper option than magnetic bearings, but also allow for larger range of motion, while preserving nanometric positioning capabilities. Gaurav [2] previously designed a flexure stage in order to provide a better option for semiconductor manufacturing companies. Gaurav developed a 2-input, 2-output state space model for the nanopositioning system with voltage applied to the two current drivers as inputs and the resulting displacement of the motion stage in X and Y directions as the outputs. The frequency response was characterized via chirp analysis and used to develop a plant transfer function. To control the motion stage, the thesis applies classical control loop shaping to achieve desired closed loop performance. The resulting continuous time controller was then directly digitized. The performance of this digitized controller was found to be unsatisfactory, so Gaurav then chose to apply an iterative learning controller to minimize tracking error in the next iteration of the controller. Gaurav found the transfer function from voltage to position (nm) for the flexure stage shown in Fig. 1 and is given by Eq 1.

$$G_{xx} = \frac{-8.38 \cdot 10^6}{(s^2 + 5.77s + 1.164 \cdot 10^4)} \frac{(s^2 + 21.12s + 2.28 \cdot 10^4)}{(s^2 + 24.43s + 3.14 \cdot 10^4)} \\ \times \frac{(s^2 + 11.27s + 1.58 \cdot 10^6)}{(s^2 + 17.50s + 1.54 \cdot 10^6)} \frac{(s^2 + 3.25s + 4.72 \cdot 10^6)}{(s^2 + 6.99s + 4.49 \cdot 10^6)} \\ \times \frac{(s^2 + 20.72s + 6.71 \cdot 10^6)}{(s^2 + 14.76s + 7.04 \cdot 10^6)} \frac{(s - 7500)}{(s + 7500)} \quad (1)$$

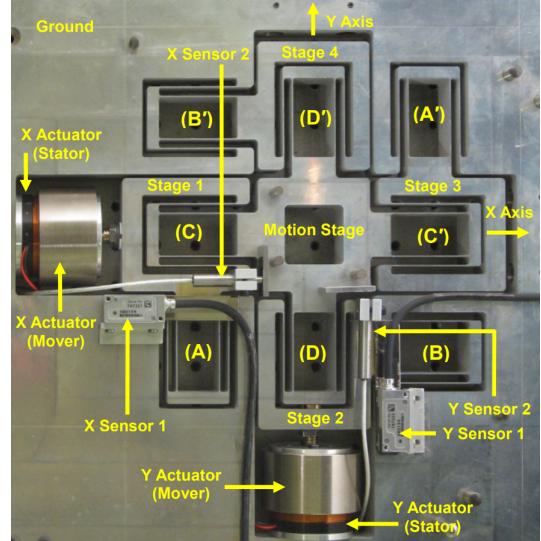


Fig. 1: Photo of the XY flexure stage for nanopositioning

The presence of secondary modes in the plant lead to non-minimum phase zeros at high frequencies of 300-400Hz, which cause significant challenges for controller performance. Lumkes [3] says that sampling at values greater than 40 times the open loop plant bandwidth ensures correct approximation of continuous controller in discrete domain. However, given the nanopositioning accuracy required, the high order modes in the plant cause significant dynamics at frequency values that are only 25 times slower than the sampling frequency of 10kHz. Further, increasing the bandwidth of the closed loop transfer function can benefit from the design of the feedback controller in the digital domain in order to account for the phase loss due to the finite loop-rate. In this study we redesigned the controller for the XY flexure stage directly in the discrete domain by using different approaches, which includes classical root locus, state feedback, observer based compensator and mixed sensitivity analysis.

Similar to [2], we attempted to utilize a 10 kHz sampling rate to compare, with the goal of outperforming the controller in the thesis by exceeding the bandwidth by 100%, limiting the peak of the sensitivity transfer function to 6.75dB (50% more than in [2]), and limiting the control effort to the 22V limit of the motors. The gain and phase margins should exceed 10 dB and 30°, respectively.

II. METHODS

A. Classical control methods

The previous designed controller in the continuous domain was discretized in order to compare to the modified design. The Controller transfer function and its discretized version with a sampling time of 0.0001s are shown below.

$$C(s) = \frac{260(s + 100)^2}{s(s + 2000)(s + 3000)} \quad (2)$$

$$C(z) = \frac{0.0248z^2 - 0.0455z + 0.02007}{z^3 - 2.56z^2 + 2.166z - 0.6065} \quad (3)$$

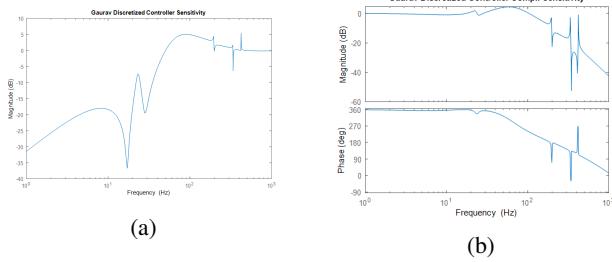


Fig. 2: Sensitivity (a) and complimentary sensitivity (b) bode plots for the discretized controller and plant with a sampling time of 0.0001s.

The original controller had a 44Hz bandwidth, defined as the frequency at which the sensitivity plot first crosses -3dB from below, as shown in Fig 2. The peak sensitivity transfer function, which is correlated to the loss of stability robustness, was 5.5dB. There is an inherit tradeoff between bandwidth and peak sensitivity peak due to the Bode waterbed effect. Thus, it is impossible to increase bandwidth while also reducing peak sensitivity. Our initial goal was to double the bandwidth while limiting the peak sensitivity to an increase of 50% and preserving gain and stability margins of 10dB and 30°respectively. In order to do so, a root locus approach was used for the discretized plant model using MATLAB's sisotool application.

B. NMP Zeros from Sampling

Non-minimum phase (NMP) zeros are unstable zeros, that is, zeros that are on the right-hand side of the s-plane. These zeros can lead to unexpected system dynamics, and though the system may still be stabilizable, it is likely that the system's capabilities are significantly more limited. Many controls design techniques such as bode and root locus plots were originally designed for continuous time systems, but control strategies that are implemented in the real world are often not truly continuous. The plant, or physical system, may be continuous in time, but control inputs and sensor measurements are often commanded by a computer, which is inherently a digital system. For the digital computer to approximate the state of a continuous time system, it will perform measurements at some fixed rate, determined by the

computer. This sampling rate cannot be too low, as it will then introduce phase lag and potentially miss key dynamics of the system. One initial thought might be to increase the sampling rate as high as possible, as this may enable the response of the digital system to approach that of the continuous system. In actuality, sampling rates that are too high or too low can potentially both introduce non-minimum-phase (NMP) zeros to the system, which can in turn limit the capabilities of the closed loop system to achieve all of the desired control parameters. This phenomenon is described in further detail in [4].

The open-loop response of the continuous time model from Gaurav's thesis [2] was compared to the response of the system discretized under a range of sampling rates to understand when non-minimum-phase zeros might be introduced. In particular, the phase portion of the bode plots were closely observed to determine when phase drops, one of the major signs of an NMP zero, are introduced.

C. State space model

Section 5.2 of [2] gives open-loop continuous-time transfer function of one of the axes of the nanopositioning system, shown in Eq. (1).

In order to use more modern control techniques like LQR or mixed sensitivity loop-shaping, it is necessary to construct a state-space representation of the system. To achieve this, we utilized two different methods: the MATLAB function `tf2ss`, and the MATLAB function `fitfrd`. The function `tf2ss` directly takes the transfer function and constructs a state-space representation in controller canonical form. The function `fitfrd`, on the other hand, fits a state space representation of dimension n , where $n \in \mathbb{N} - \{0\}$. Because the state space dimension is a parameter of the fitting process, dimensions from 1 to 15 were utilized for fitting. However, due to the implementation of `fitfrd`, MATLAB will reduce the dimension of the state space from the passed parameter if the excess dimensions would not significantly improve the fit. Thus, in actuality, state space representations of $n \in [1, 10]$ were found. To compare the goodness-of-fit of each state space representation and choose the best one, the L2 norm of the fitted transfer function magnitude and the actual transfer function magnitude were used.

D. LQR Controller

The previously discussed state-space representation, in discrete form, would result in the following linear difference equation:

$$\mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) \quad (4)$$

$$\mathbf{y}(k) = \mathbf{C}(k)\mathbf{x}(k) + \mathbf{D}(k)\mathbf{u}(k) \quad (5)$$

When designing a controller, not all of the states in $\mathbf{x}(k)$ are equally important, and not all inputs in $\mathbf{u}(k)$ are equally bounded. With classical control techniques, it is not intuitive to understand how these different priorities might be expressed in the final controller design. With modern control techniques

like LQR, however, this desire to weigh these priorities can be realized with a cost function. In this case, rather than observing the frequency response of a system via bode plots and other classical control tools, a cost function is designed to be of the form:

$$J_N = \sum_{k=0}^N \mathbf{x}^T(k) \mathbf{Q}(k) \mathbf{x}(k) + \mathbf{u}^T(k) \mathbf{R}(k) \mathbf{u}(k) \quad (6)$$

In the LQR controller, the control law is of the form:

$\mathbf{u}(k) = -\mathbf{K}(k)\mathbf{x}(k)$. Thus, there is a gain matrix, $\mathbf{K}(k)$, that acts on the system states, $\mathbf{x}(k)$, to return the desired control command. This would change the $\mathbf{A}(k)$ matrix described in 4 to be of the form $\mathbf{A}(k) - \mathbf{B}(k)\mathbf{K}(k)$. The LQR problem develops this gain matrix, $\mathbf{K}(k)$ according to the previously mentioned cost function.

Thus, the different weightings in the $\mathbf{Q}(k)$ and $\mathbf{R}(k)$ matrices will be actualized as different prioritizations for the states and inputs of the system. The LQR controller design approach will be applied to the discretized version of Gaurav's plant to better understand how modern control techniques compare to classical control techniques like bode plot loop shaping. The LQR problem is described in more detail in Chapter 11 of [5].

E. Kalman Observer

The physical constraints of the system only allow us to measure the output position and not the states directly. Hence, an observer was designed in order to estimate the states of the system to apply state feedback. Gaurav [2] was able to assess the measurement noise for the linear optical encoder that measures output position, which showed to be 4nm RMS. The design of an observer can help filter this measurement noise and estimate the states of the system with the downside of slowing down the dynamics of the disturbance response and decreasing stability robustness. Figure 3 shows the block diagram of the implementation of the observer.

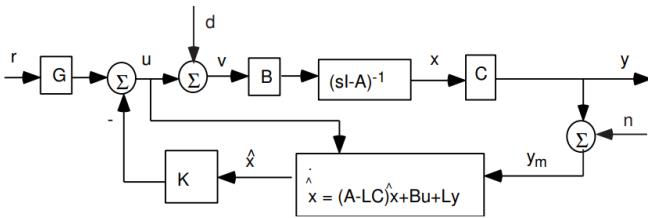


Fig. 3: Block diagram showing the implementation of state feedback and observer [6]

With this implementation and assuming small plant model uncertainty, the separation principle holds between state feedback eigenvalues and observer eigenvalues. The state space description of the closed loop response is given by (7) and (8) [6], where \tilde{x} is the difference between the actual states and the estimated states.

$$\begin{bmatrix} \dot{x} \\ \dot{\tilde{x}} \end{bmatrix} = \begin{bmatrix} A - BK & B \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} + \begin{bmatrix} BG \\ 0 \end{bmatrix} r \quad (7)$$

$$y = [C \ 0] \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} \quad (8)$$

The separation principle allows us to choose the observer eigenvalues independently of the LQR design. In a similar approach to the LQR problem, we decided to design a linear quadratic optimal estimator with infinite time horizon. Eq. (9) gives the state space representation of the system with process and measurement noise.

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w \\ y_k &= Cx_k + v \\ E(vv^T) &= R \\ E(ww^T) &= Q \end{aligned} \quad (9)$$

Where v and w are zero-mean white noise with covariance given by R and Q, respectively. The optimal discrete Kalman gain is given by Eq. (10).

$$L = APC^T(CPC^T + R)^{-1} \quad (10)$$

Where P is the solution to the discrete algebraic Riccati Equation (11)

$$BQB^T + APA^T - APC^T(CPC^T + R)^{-1}CPA^T - P = 0 \quad (11)$$

Such that the cost function (12) is minimized.

$$J = E((x_k - \tilde{x}_k)(x_k - \tilde{x}_k)^T) \quad (12)$$

In order to design the observer, we used the measurement noise of 16 nm RMS² as the R value and tuned Q to obtain desired disturbance response and stability robustness. The design was made such that the observer eigenvalues would have faster dynamics than the feedback eigenvalues. This ensures that the lag in state estimation does not drastically affect the compensator. The Kalman gain was calculated using the MATLAB function lqrd for a discrete linear quadratic optimal estimator.

F. Mixed sensitivity loop-shaping

Unlike LQR, where the weights R and Q scale the states and inputs, respectively, mixed sensitivity loop shaping in the frequency domain allows for H_∞ optimal control given frequency dependent weights on the sensitivity (weighting function W_1), complementary sensitivity (W_3), and the controller effort of the system (W_2). This method is implemented in MATLAB using the function mixsyn. Effectively, the function attempts to set the weighting function multiplied by the system properties to equal 1 (0 dB). Thus the weighting function and the sensitivity, for example, are inversely related. If mixsyn cannot simultaneously meet all requirements W_1 , W_2 , and W_3 , γ that the function returns will be greater than 1, which represents the greatest multiplier of the weighting function shape and the actual system response, i.e. the worst-case multiplier difference between the wanted response and the optimally shaped response.

For this method, multiple different weighting functions were used and tested. All W_1 weighting functions shared a low-frequency gain of 120 dB (which would drive the sensitivity low for good tracking performance at low frequencies) and a high-frequency gain of -6 dB (which would allow for the peak in the sensitivity function to be driven to below 6 dB). $W_3 = W_1^{-1}$. For W_2 , which modifies the controller effort, it decreases with an increase in frequency from 20 dB low-frequency gain to -20 dB high-frequency gain. The main difference between the iterations is the frequency location of the 0 dB gain. The final controller will be the one chosen that maximizes the bandwidth of the system and meets the wanted properties: peak sensitivity of less than 6 dB, a gain margin that exceeds 10 dB, and a phase margin that exceeds 30°.

The system that was used for the fits was a discretized version of the 11 degree of freedom G_{yy} system using 10 kHz sampling, rather than the fitted method in Section II-C, in order to optimize as much as possible the frequency response of the system. Additionally, there were no numerical issues or computation time constraints that would have necessitated using a simpler system for this part.

III. RESULTS AND DISCUSSION

A. Classical control methods

An issue encountered was that the increase in the peak sensitivity associated with secondary modes is significantly larger than the gain in bandwidth after the latter reaches a value of 52Hz. This means that even for a small increase in bandwidth, there is a large increase in peak sensitivity. While developing the initial constraints for the design problem, the detrimental increase in sensitivity for secondary modes was underestimated.

An initial attempt was made to drive the sensitivity peak to regions of higher frequency such that they would be out of the operational frequency of the system. However, because the area under the tail of the sensitivity integral is bounded, it follows that most of the trade-off between bandwidth increase and sensitivity increase must take place at low and intermediate frequencies. This means that trying to push the secondary mode peaks far away from the bandwidth range is not feasible either. As a result, we changed the initial specifications of the system such that the highest bandwidth possible was achieved for a peak in sensitivity of 10dB. Consequently, the stability margins were worse than the expected, yielding a gain margin of 4dB, which is 6dB less than the anticipated. The phase margin target of 30° was met. The sensitivity plot of the resulting controller and plant is shown in Fig 4. The resulting bandwidth was 54Hz, representing a 23% increase in the bandwidth compared to the original controller. The designed controller is shown below.

$$C(z) = \frac{0.14(z - 0.994)(z - 0.976)}{(z - 1)(z + 0.2)(z - 0.8)} \quad (13)$$

As it is possible to see in the step response in Fig 5, the new design was able to achieve faster rise time and settling time with a small increase in overshoot. Hence, the new

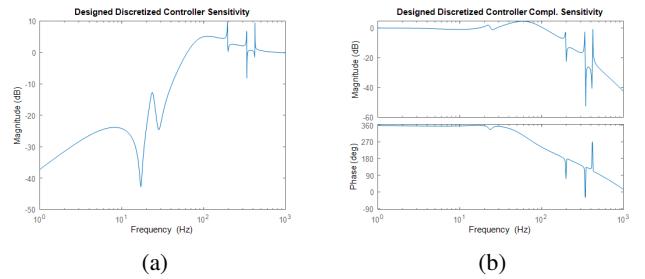


Fig. 4: Sensitivity (a) and complimentary sensitivity (b) bode plots for the designed controller and plant with a sampling time of 0.0001s.

design yielded better time response parameters. One of the main desired characteristics for semiconductor inspection XY stages is the ability to scan fast such that the long time taken to process each wafer is overall decreased. The new design presented better results in that sense, which could contribute to higher yield of the inspection process and a consequent decrease in cost for the production of each wafer without compromising accuracy.

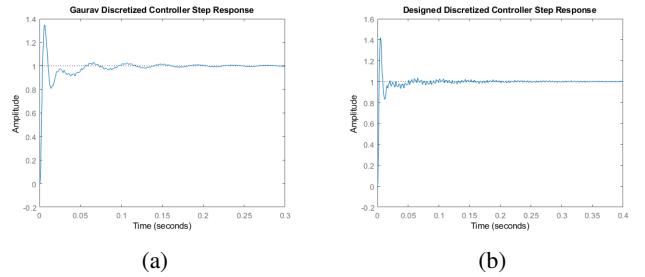


Fig. 5: Step response for the discretized controller (a) and designed controller (b).

B. NMP Zeros from Sampling

The continuous-time plant modeled in 1 was discretized under a range of sampling rates, from as low as 100 Hz to as high as 1 GHz. The frequency response of the continuous plant, observed in a bode plot in 6a, can be compared to the frequency response of the discretized models to understand when the sampling rate has introduced an NMP zero. The 10 kHz bode plot, shown in 6b demonstrates the effect of a NMP zero, as there is a significant loss of phase at around 300 Hz. Ideally the sampling rate should be adjusted so that this NMP zero is not present, which could be done by increasing the sampling rate to 100 kHz, as shown by the lack of phase drop in 6c.

Despite the introduction of a NMP zero, all subsequent control design was performed with a 10 kHz sampling rate to enable proper comparisons between the performance of Gaurav's controller and the controllers proposed in this paper. A further iteration of the controller design could increase the sampling rate to 100 kHz to prevent the introduction of a

NMP zero due to sampling, and thus prevent the limitations to control performance that NMP zeros bring.

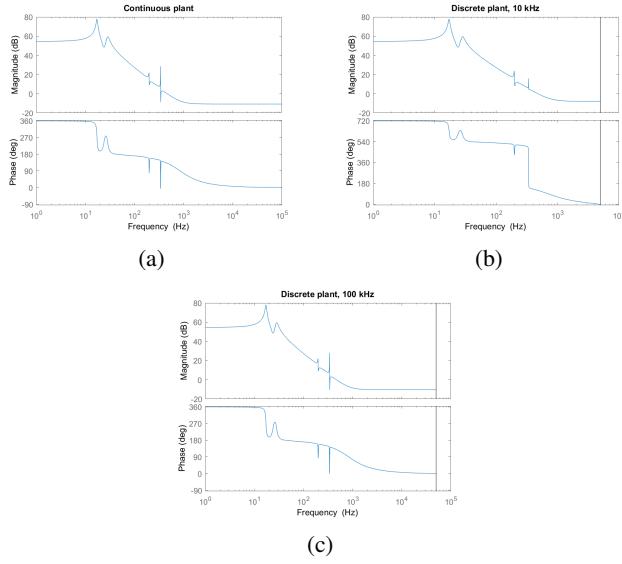


Fig. 6: Bode plots for the continuous plant (a), 10 kHz sampled plant (b), and 100 kHz sampled plant (c).

C. State space model

Using the MATLAB function `tf2ss` resulted in a very badly conditioned matrix, due to the many rows containing only a single 1, and the first row containing terms with very large magnitude due to the multiplication of the terms on the order of 10⁴ and 10⁶ giving coefficients on the order of 10⁶ to 10³⁴. As such, using such a matrix for numerical calculations is not recommended due to issues with numerical precision. For the actual matrix, please see the Appendix, section A1.

Using `fitfrd`, multiple state space models of varying dimension were fit to the transfer function (1). The average L2 norm of the error between the magnitude of the state space representation and the magnitude of the transfer function. This was plotted in Figure 7.

The minimum average L2 norm error occurred when the dimension of the state space was 8 and was equal to 2.4534 · 10⁻³. The second-lowest L2 norm error of 5.6142 · 10⁻³ was with a six-dimensional state space model. Taking a look at the Bode plot the eight-dimensional state space model (Figure 8b), one can see that it tracks the magnitude very well up until about 350 Hz, though the phase is off after 200 Hz. The peaks in the transfer function around 200 Hz are fairly well captured by the state space model, with the lower peak not quite as well captured and the gain at 200 Hz is overestimated. The additional peaks around 350 Hz are captured well initially but then the gain is smoothed out.

Meanwhile, the six-dimensional state-space representation is shown in Figure 8a. It tracks the magnitude well until the multiple peaks in the gains around 350 Hz, where it instead acts functionally like a low-pass system and filters out the

peaks in the magnitude gain. Like the eight degree of freedom system, it is unable to track the phase after 200 Hz.

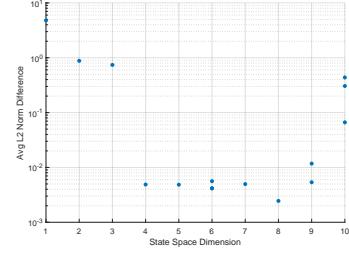


Fig. 7: The average L2 norm error between the system transfer function and the state space model fit

For the purposes of this project, we have utilized the 8 degree of freedom state space representation for our analysis. This state space representation is written out below:

$$A = 10^6 \times \begin{bmatrix} -2.01 & 8.05 & -16.10 & 32.20 & -32.20 & 32.20 & -16.10 & 4.02 \\ -2.43 & 9.71 & -19.43 & 38.86 & -38.86 & 38.86 & -19.43 & 4.86 \\ -2.79 & 11.14 & -22.29 & 44.57 & -44.57 & 44.57 & -22.29 & 5.57 \\ -2.01 & 8.02 & -16.05 & 32.09 & -32.09 & 32.09 & -16.05 & 4.01 \\ -1.96 & 7.83 & -15.66 & 31.32 & -31.32 & 31.32 & -15.66 & 3.92 \\ -1.29 & 5.18 & -10.35 & 20.70 & -20.70 & 20.70 & -10.35 & 2.59 \\ -1.08 & 4.30 & -8.61 & 17.22 & -17.22 & 17.22 & -8.61 & 2.15 \\ -0.86 & 3.44 & -6.89 & 13.78 & -13.78 & 13.78 & -6.89 & 1.72 \end{bmatrix}$$

$$B = [69.993 \quad 106.65 \quad 143.38 \quad 111.62 \quad 109.77 \quad 68.457 \quad 49.922 \quad 32.504]^\top$$

$$C = 10^7 \times [1.02 \quad -4.09 \quad 8.18 \quad -16.37 \quad 16.37 \quad -16.37 \quad 8.18 \quad -2.05]$$

$$D = [0.28375] \quad (14)$$

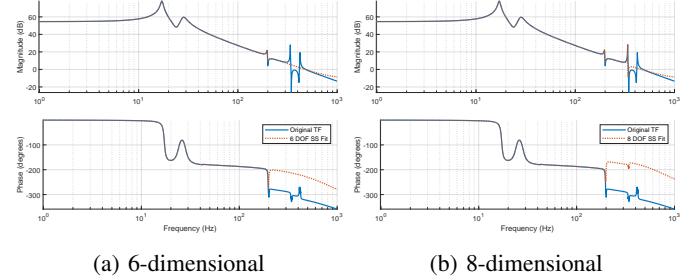


Fig. 8: The Bode plots for the system transfer function fits of dimensions 6 and 8. 8a Note the smoothing of the peaks around 300 Hz. 8b Note the capturing of the peaks around 300 Hz

D. LQR Controller

The LQR controller was implemented in MATLAB using the `lqr` function. The following Q and R matrices were used as weightings for cost function described in (6):

$$Q = \begin{bmatrix} 1000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 750 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$R = [200]$$

Note that the system has a single input, thus the R-matrix is a single scalar. The system also has a pre-compensator gain of 316.6143, which was tuned such that the closed loop system would have a DC-gain of 1.

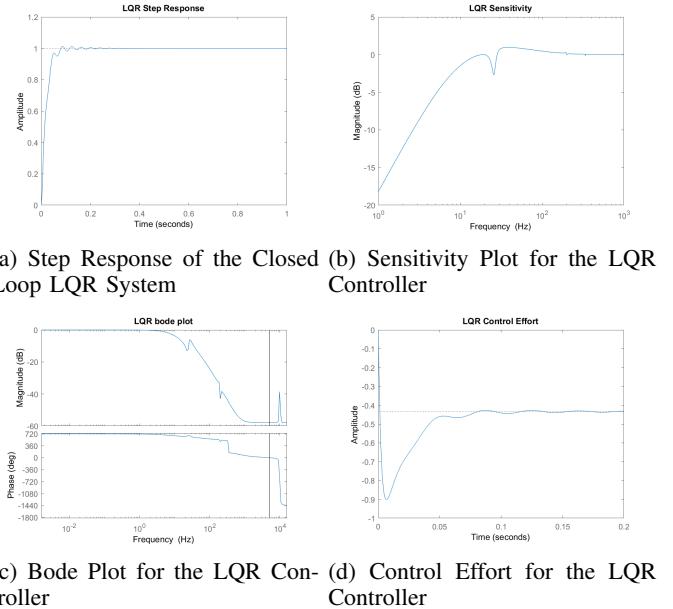
The resulting LQR controller has a peak sensitivity of 0.9516 dB, which is significantly lower than the maximum allowed design goal of 6.5 dB. The controller's gain margin of 41.5907 dB exceeds the design goal of 10 dB. Its phase margin of 172.69° meets the goal of at least 30°. Thus the LQR controller demonstrates robust stability. The bandwidth of the LQR controller was limited, however, to 8.84 Hz. This does not meet the design requirement of at least 44 Hz. The performance of the LQR controller is demonstrated in the bode plots and stability magnitude plots below, in 9b and 9c.

The step response of the closed loop system, shown in 9a. The system demonstrates zero steady state error, due to the pre-compensator gain of 316.6143. The step response features a settling time of 0.0751 seconds, rise time of 0.0366 seconds, overshoot of 1.27%. In comparison to the classical control and original controller from Gaurav, both shown in 5, the controller response features significantly better stability robustness, as can be observed by the lack of overshoot. However, the response time is much slower with the LQR controller.

During the control design process, it appeared that there was a fundamental limit to the closed loop system's bandwidth capability. The tuning process began by independently tuning each of the Q-matrix and R-matrix values to an arbitrarily large constant, 1000, to understand how each factor influenced the final closed loop system performance. However even if the other design goals were disregarded, the bandwidth appeared to only reach a maximum of 22.3 Hz. This limitation could be due to limitations of MATLAB's `dlqr` function to design a full-state-feedback gain for the 8 states, particularly when the system representation was poorly conditioned to begin with. It is possible that the NMP zero that was introduced due to too low of a sampling rate may also have limited the system's capabilities, thus it is unable to simultaneously meet the desired bandwidth and stability requirements.

The phase portion of the bode plot was unusual in that it began at 720°, and thus the MATLAB `margin` function appeared to be unable to properly determine a phase margin. The phase margin was manually determined by correcting the 720° position to 0°. The corresponding phase margin was then calculated by the usual procedure of determining the frequency at which the magnitude bode plot crossed 0 dB, and the finding that frequency's corresponding difference with -180°.

The step response of the control effort for the LQR controller was also modeled, shown in 9d. The magnitude of the effort does not appear to directly correlate to voltage applied to the actuator, which would be limited to 22V. Since we were unable to directly correlate the magnitude to an actuator input in this step response, it instead would be relevant to observe the overshoot of the control effort. Since the overshoot is around 2X the magnitude of the steady-state value of



(a) Step Response of the Closed Loop LQR System (b) Sensitivity Plot for the LQR Controller

(c) Bode Plot for the LQR Controller (d) Control Effort for the LQR Controller

Fig. 9: Plots describing the LQR method

control effort, it could be understood that the range of the nanopositioning stage would be limited to the displacement correlating to around half of the actuator's voltage limit, or 11 V.

E. Kalman Observer

The separation principle ensured that the observer does not affect the step response of the system. However, the observer will influence disturbance response and stability robustness. In order to design the observer, we ensured all estimator eigenvalues were faster than the feedback eigenvalues and looked at the peak in sensitivity function as an indicator of loss of stability robustness.

The design parameters for the Kalman filter were chosen such as shown below:

$$W = \begin{bmatrix} 0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.15 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 \end{bmatrix}$$

$$V = [15000]$$

The resulting open loop transfer function is shown in Fig 10a and yielded gain and phase margins of 35 dB and 30°, respectively. As it is possible to see, these margins are worse than the system with only state feedback. The Kalman filter design parameters were chosen such that the eigenvalues associated with $A - LC$ would provide faster dynamics than the eigenvalues of $A - BK$ as this helps mitigate the side

effects of adding an observer, while preserving a bandwidth of 8.8Hz. However, it was still necessary to ensure that (A, W) is realizable, which posed a constraint on the design process. The associated closed loop transfer function for the observer based compensator is shown in Fig 10b.

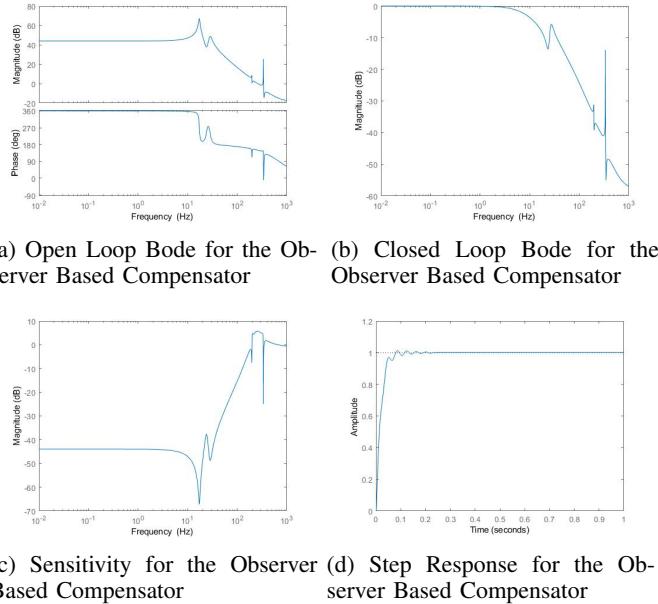


Fig. 10: Observer-based controller design

Another indicator of stability robustness is the peak sensitivity. As seen in Fig 10c, the peak in sensitivity with the addition of the observer is higher than with pure state feedback. Thus, the observer compensator has worse stability robustness, which is also indicated by the before-mentioned loss in phase and gain margins. It was still possible to keep the peak in sensitivity to 5.7dB, which over-performs the classical controls approach in terms of stability robustness.

The step response of the observer based compensator is shown in Fig 10d and is similar to the step response with pure feedback. This is due to the separation principle, which dictates that the addition of an observer does not affect the system step response. As seen, the separation principle also holds for the discretized system. There is a small difference between the state feedback and observer compensator step responses, associated with the discretization of the plant. This discretization can be treated as a form of plant uncertainty, which violates the separation principle. However, the high sampling frequency of 10kHz ensures that this uncertainty is minimal and can be considered negligible, yielding very similar results for both state feedback and observer compensator.

F. Mixed sensitivity analysis

For various combinations of the W_1 , W_2 , and W_3 weighting functions (each shown in Figure 11a, 11b, and 11c), the system performance was analyzed and calculated. $|S|_\infty$ was found to increase with an increase in the cross-over frequency of the weighting functions (XOF), but across the range tested was

found to be below the maximum cut-off of 6 dB. The gain margin of the system decreased with an increase in the XOF, as did the phase margins. The phase margins always exceeded the wanted cut-off, while the gain margins eventually settled to a value under the wanted bound of 10 dB. The bandwidth meanwhile first decreased and then increased.

In the end, the optimal XOF of the weighting functions was 668 rad/sec. This resulted in a gain margin of 10 dB, a phase margin of 76.8° , a final bandwidth of 65.7 Hz. The peak in the sensitivity was 3.34 dB. However, this method necessitates very good agreement between the actual system dynamics and the transfer function. Errors and deviance from the discrete system dynamics used in this method around frequencies between 200 and 300 Hz could greatly influence the system, due to the still slightly peaked nature of the sensitivity function and its crossings of unity gain at those frequencies.

For the controller and system responses, please refer to Appendix B for the code due to the size of the matrices involved.

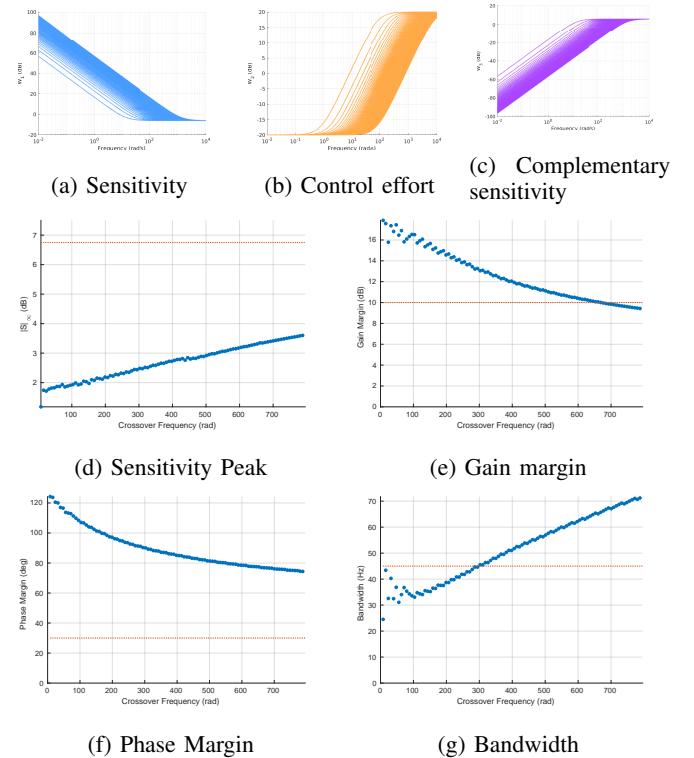


Fig. 11: Weighting functions for the Matlab function `mixsyn` used for loop shaping (11a, 11b, 11c) and the results for the sensitivity peak (11d), gain (11e) and phase (11f) margins, and bandwidth (11g). Note that for 11d through 11g, the blue dots represent the numerical results from iterating through triplets of the loop shaping functions in 11a through 11c and the orange lines the wanted parameters (the yellow line represents a smaller bound should it have been necessary).

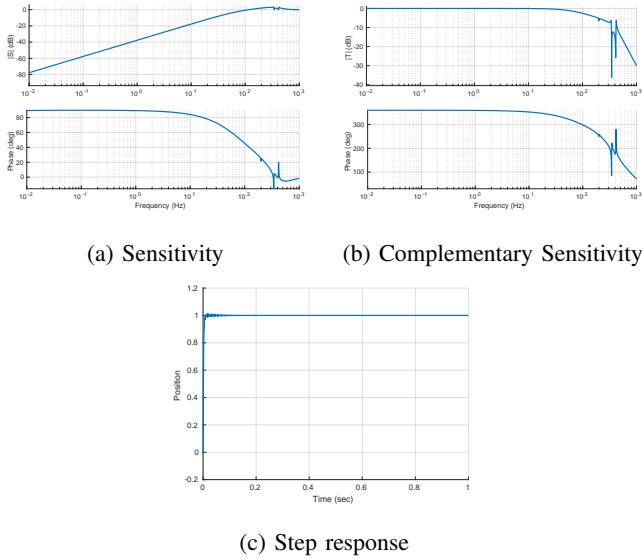


Fig. 12: System response with optimal controller and its step response

IV. CONCLUSION

The root locus method for controller design yielded a bandwidth of 54Hz, which represents a 23% increase compared to the original continuous controller. It was possible to limit the peak sensitivity to 10dB, ensure a phase margin of 30° and a gain margin of 4dB. Increasing the bandwidth past this point would result in substantial increases in peak sensitivity associated with secondary modes of the plant. With a higher order controller, it might have been possible to achieve better bandwidth/sensitivity balance. However, in the classical domain controllers with order higher than 4 become abstract. Thus, a state space approach followed in order to implement a higher order controller.

The continuous plant in 1 was discretized under a range of settling times to better understand the sampling rates at which NMP zeros might be introduced to the plant model. It was found that a 10 kHz sampling rate did introduce an NMP zero to the model, as there was an instance of significant phase drop in the bode plot, shown in 6b. If the sampling rate were increased to 100 kHz, as shown in 6c, this NMP zero was no longer present. Despite this NMP zero, all subsequent controls design proceeded with the 10 kHz, to ensure that the performance of our controller designs could be directly compared with Gaurav's controller in 13.

The LQR approach provided an optimized control strategy, according to the cost function defined in 6. Stability robustness was one of the key highlights of the LQR approach, as it was able to achieve a significantly lower peak on the sensitivity plot, shown in 9b, and also featured significantly less overshoot, shown in 9a, than was found with the classical controls approach. However, this stability robustness came at the cost of response time, as the LQR controller featured significantly greater rise and settling times.

In order to apply the state feedback, the states of the system had to be estimated and filtered since the only measurement output is position. Thus, an observer was designed applying the optimal linear estimator to find the Kalman gain for the infinite horizon problem. This design was informed by both the state feedback eigenvalues and the sensor measurement noise. The observer design approach used the separation principle in order to choose an observer with faster dynamics eigenvalues than the state feedback in order to ensure that the lag in measurement did not compromise stability. The observer was implemented along with the LQR controller to create a linear quadratic Gaussian controller. This implementation yielded a peak in sensitivity of 5.7dB, which represents an increase compared with the state feedback controller as an observer decreases stability of the system invariably. It was still possible to ensure gain and phase margins of 35dB and 30°, respectively.

The loop sensitivity function trades the robustness of the LQR method as represented by the peak in the sensitivity function for greatly expanding the bandwidth of the system to exceed Gaurav's system by about 49%, and the bandwidth of the mixed-sensitivity loop-shaping design outperforms the bandwidth of the LQR or the observer-based method by a factor of 7.4. The step response of the system has a faster rise-time than either the LQR or observer-based controller (but slower than the classical controller); however, the LQR based controller oscillates with a lower frequency when settling.

Method	$\ S\ _\infty$ (dB)	G_M (dB)	P_M (deg)	Bandwidth (Hz)
Original	5.5	10.0	30.0	44
Classical	10.0	4.0	30.0	54
LQR	0.95	41.59	172.69	8.84
Observer	5.7	35	30.0	8.84
H_∞	3.34	10.01	76.84	65.70

TABLE I: Performance overview of the several methods used to control the system

Overall, the initial approach to use a classical controller was limited by the level of abstraction with creating a controller with order higher than 4. The introduction of NMP zeros due to sampling was investigated, which showed to not happen for the discretization of the plant with 10kHz. A state space approach was then used by fitting an 8 order model to the plant and further applying LQR state feedback, a Kalman filter observer and a mixed sensitivity loop shaping. Exploring the tradeoff between bandwidth and stability robustness in the discrete domain was the major focus of this project. In the future, other advanced controller techniques could be applied along with the designed controller in order to further improve performance, such as with an iterative learning controller or a finite horizon optimal controller with model predictive control. Further, It would be beneficial to compare the effectiveness of the designed controllers on the physical plant and compare it to the results yielded with the continuous design approach.

V. CONTRIBUTIONS

Classical Controls: D.S.

NMP Zeros from Sampling: T.S.

State-space model: G.A.

LQR Controller: T.S.

Kalman Observer Design: D.S.

Mixed Sensitivity Loop-shaping: G.A.

Writing: G.A., D.S., T.S.

REFERENCES

- [1] S. Awtar and G. Parmar. "Design of a large range XY nanopositioning system," *J. Mechanisms Robotics*, vol 5., no. 2, May 2013.
- [2] G. Parmar. "Dynamics and Control of Flexure-based Large Range Nanopositioning System," PhD Dissertation, Dept. of Mech. Eng., University of Michigan, Ann Arbor, 2014.
- [3] Lumkes John H. *Control Strategies for Dynamic Systems: Design and Implementation*, 2002
- [4] Y. Fu and A. Dumont. "Choice of Sampling to Ensure Minimum-Phase Behavior," *IEEE Transactions on Automatic Control*, vol. 34, no. 5, May 1989.
- [5] C. L. Phillips, H. T. Nagle, and A. Chakrabortty, *Digital Control System Analysis & Design*. Boston: Pearson, 2015.
- [6] J. S. Freudenberg, *A first Graduate Course in Feedback Control*, 2020.

APPENDIX

A. State space model

1) Transfer function directly to state space – continuous:

$$A = \begin{bmatrix} -7.6e3 & -1.4e7 & -9.9e10 & -5.6e13 & -3.8e17 & -6.8e19 & -3.8e23 & -1.4e25 & -1.6e28 & -1.9e29 & -1.3e32 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$B = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^\top$$

$$C = [0 \ -8.38e6 \ 6.23e10 \ -1.06e14 \ 8.14e17 \ -3.84e20 \ 3.13e24 \ -3.28e26 \ 3.21e30 \ 5.76e31 \ 7.17e34]$$

$$D = [0]$$

B. Code

The code can be found [here](#).

Each folder contains the scripts needed for running the files. Note that MATLAB 2019b or later is required to run the mixed sensitivity loop-shaping script. Furthermore, a @umich.edu account is required.