Fast Setpoint Tracking of an Atomic Force Microscope X-Y Stage via Optimal Trajectory Tracking

Roger A. Braker and Lucy Y. Pao

***Abstract*— Acquiring Atomic Force Microscope images using compressed sensing requires the piezo X-Y stage to track a sequence of step commands. To achieve fast tracking of such commands requires a precise system model. We show that once such a model is obtained, standard linear feedback can be used to achieve excellent tracking of step inputs. The system under consideration has a significant amount of time delay, and we develop a computationally efficient state estimator for this scenario. We demonstrate that beyond a certain step size threshold, the control law attempts to violate actuator slew-rate limits, resulting in a severely deteriorated response. We then show how this can be overcome by tracking an optimal trajectory obtained by solving a constrained, finite horizon Linear Quadratic Regulator problem and demonstrate the feasibility of this approach experimentally.**

I. Introduction

The traditional Atomic-Force-Microscope (AFM) imaging method is to raster scan the sample with an atomically sharp probe located at the end of a cantilever. Interaction with the sample surface causes the cantilever to deflect. By recording this vertical deflection, the sample surface topology can be reconstructed. Although this method yields excellent spatial resolution, its temporal resolution is poor by comparison and has been the subject of intensive research. See [1] for a brief history.

These research efforts can roughly be divided into two camps. Some efforts to increase imaging speed have focused on increasing the raster scan rate via advanced control algorithms [2], [3], [4], [5]. The achievable speed of these methods is limited by the bandwidth of both the piezo stage and the power amplifier, which has led other researchers to build custom hardware with extremely high-bandwidths [6], [7]. Excellent results have been achieved by combining both approaches [8].

Other researchers have begun to investigate using alternative scan patterns. Noting that the high frequency content of the raster pattern contributes to the difficulty in following it at high speeds, the researchers in [9] used a smooth Lissajous pattern which has a lower frequency content to avoid exciting the stage resonances. Others have developed scanning patterns to minimize the time spent over uninteresting regions [10]. An advantage of these ideas is that they can be implemented independent of custom hardware fabrication. In a similar vein, [11] suggested a method of AFM imaging whereby a random selection of point-to-point measurements are taken. The sample topology is then reconstructed using the theory of compressed sensing [12]. Taking this random selection of point-to-point measurements reduces the tip-sample interaction and can improve the integrity of both the sample and AFM tip.

Minimizing the imaging time in this scheme requires that the rest-to-rest maneuver times between point measurements be minimized. While the goal of control laws designed for raster scanning is to accurately track a triangle wave, the goal in compressed sensing is to track a step input with minimum settling time. This problem has been successfully addressed for many second-order and some third-order systems by the Proximate Time-Optimal Servomechanism, which is a robust approximation of a time-optimal controller [13], [14], [15]. Unfortunately, the development of these control laws relies on geometric phase-plane arguments, making them intractable for high-order systems.

Since most AFM control studies focus on accurately tracking a raster pattern [16], [17], [2], it is unclear how well these same controllers can be applied to step inputs. Indeed, as we argue in Section II-A, good performance is not guaranteed. Moreover, much of the current literature does not consider saturation or slew-rate constraints. In particular, the slew-rate constraint imposes a very real limitation and is not hard to violate in a setpoint tracking application. We demonstrate this in Section III-C and show that such a violation yields a severely deteriorated system response.

The key contribution of this paper is to experimentally demonstrate the feasibility of the following approach:

1) Design a state feedback gain, K, which would give the desired response if no constraints were present.
2) Find matrices (Q, R) which yield (approximately) K when used to solve the infinite horizon LQR problem.
3) Using the obtained (Q, R), solve a *finite* horizon, *constrained* LQR problem.
4) Track the resulting trajectory with linear feedback.

In a compressed sensing scheme, we know the collection of setpoints to visit in advance and hence the optimal trajectory for each setpoint can be computed offline.

This paper is organized as follows. In Section II, we describe the experimental setup and the modeling of the system. In Section II-A, we show that a controller which achieved excellent performance tracking a raster pattern fails to provide good settling time to a step input. Moving beyond this, we develop a linear state feedback controller in Section III.

This work was supported in part by the US National Science Foundation (NSF Grant CMMI-1234980), Agilent Technologies, Inc., a 2016 CU-Boulder ECEE Graduate Student Summer Fellowship, and a Fellowship from the Hanse-Wissenschaftskolleg, Delmenhorst, Germany.

R.A. Braker is a graduate student and L.Y. Pao is the Richard & Joy Dorf Professor; both are with the Dept. of Electrical, Computer, and Energy Engineering at the University of Colorado, Boulder, CO 80309, roger.braker@colorado.edu, pao@colorado.edu.

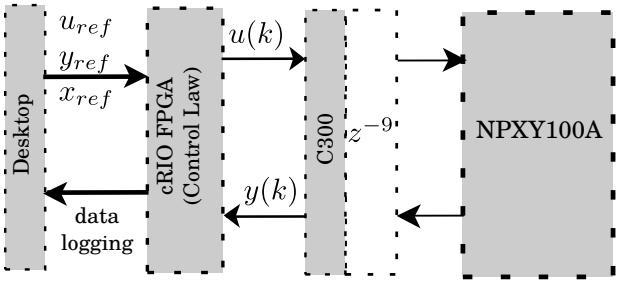

Fig. 1: Schematic diagram of the experimental setup.

The system model developed in Section II includes 9 samples of delay and in Section III-A we show a way to reduce the computational burden of the state estimator given these extra states. Section III-C demonstrates how the linear feedback approach can fail because it cannot account for the system constraints. To address this, we consider the constrained, finite-horizon LQR problem in Section IV. We review how to cast the problem as a quadratic program, which we solve offline to produce a desired trajectory. We show that tracking this trajectory yields excellent performance in Section V and provide concluding remarks in Section VI.

II. Experimental Setup

The experimental setup is shown schematically in Fig. 1. The X-Y piezo scanner is an nPoint NPXY100A which has a range of 100 μm. For control, we employ a National Instruments Compact Rio 9082. This hardware includes an FPGA into which all control logic is programmed and also includes the capacity for data logging and interfacing with a development desktop computer. Signals to and from the Compact Rio run through an nPoint C300 signal conditioner and controller. We use the C300 in open-loop mode to provide power amplification to the piezo scanner. The inputs to the C300 saturate at ± 10 V and is current limited to 100 mA. Even though the C300 is operated in open-loop mode, the signal always runs through a DSP in the C300, introducing around 360 μs of delay. Throughout this paper, we use a sampling frequency of 25 kHz.

A. A Motivating Example

The frequency response of the NPXY100A in the x-direction is shown in Fig. 2 (discussed further in the next section). Prior control designs, which were intended to track a raster pattern, were derived from lower-order models which ignored certain low-frequency dynamics as well as the resonances at approximately 350 Hz and 1000 Hz [2], [18].

Shown in Fig. 3 is the step response of the system (red curve) using the H_∞ controller designed in [18]:

$$D(z) = 0.74736 \frac{(z+1)(z^2 - 1.816z + 0.8284)}{(z+0.7072)(z-0.7899)(z-1)} \times \frac{(z^2 - 1.959z + 0.9864)}{(z^2 - 1.882z + 0.9019)}. \quad (1)$$

Also shown is the step response in simulation (dashed-black curve), where the plant is the model given in [2]. Although

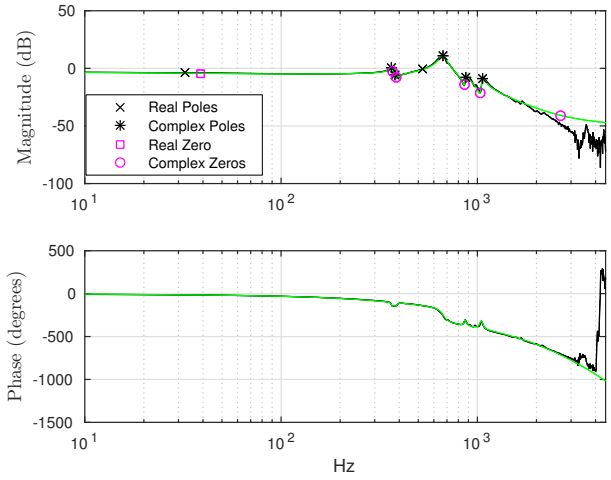

Fig. 2: Frequency response of the hardware system (solid black) and of the state-space model (solid green). Each of the poles and zeros (excluding the poles associated with delay) of the state-space model are marked as shown in the legend.

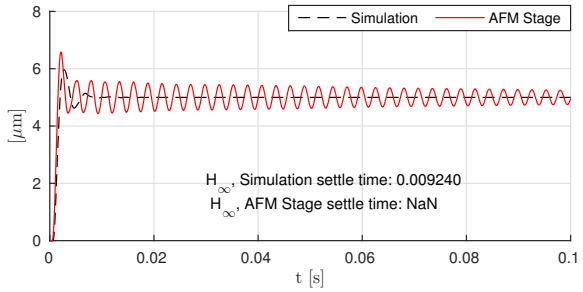

Fig. 3: Step input with the system in closed loop, using the H_∞ controller in (1).

the response has a good rise time, the experimental response is severely affected by undamped resonances. It is notable that the controller in (1), combined with feedforward model inverse methods, was able to achieve excellent raster tracking performance [18] and we emphasize that (1) was designed for this *individual* stage. Although adding such feedforward control here would likely reduce the overshoot seen in the simulation response, it would not eliminate the residual vibrations seen in the experimental response in Fig. 3, since the modes responsible for those vibrations are un-modeled. In this work therefore, we attempt to fit, and hence control, every feature until the system has rolled-off sufficiently.

B. System Modeling

To obtain a parametric model for control purposes, we first obtain the frequency response function (FRF) of the system using the method of swept sines. That is, at frequencies between 10 Hz and Nyquist, we perturb the system with a sine wave. After a settling period, we record both the input and output. Numerically integrating each signal in the Fourier Integral yields the first Fourier coefficients for both the input and output. The ratio of these coefficients yields the FRF.

Prior work with this particular X-Y stage fit a low-order order model using a weighted least squares parameter fitting technique [2], [18]. Successfully using the weighted least squares parameter fit depends on both correctly choosing the model-order as well as the weighting matrix used. As model order increases this method, in our experience, becomes unwieldy and suffers from a lack of repeatability. For example, changing the model order often means adjusting the weighting matrix. As an alternative, we employ an eigenspace realization algorithm (ERA) [19], [20].

From previous modeling efforts as well as communication with nPoint Inc., we know that at a sample frequency of 25 kHz, the C300 introduces $n_d = 9$ units of delay. This amount of delay is first removed from the experimental frequency response. The ERA requires an impulse response which we obtain via an inverse Fourier transform on the FRF data. Applying the ERA outlined in [19] and truncating the model order at $n_s = 12$ yields a state-space realization[1] $G_1 = \{A,\ B,\ C,\ 0\}$ with the magnitude fit shown in Fig. 2. We then add the 9 units of delay into the model, yielding the phase fit of Fig. 2. To do this, denote by $x(k)$ the states associated with G_1 and by x_d the states associated with delay. Then, modeling all delay as input delay yields the complete dynamics as

$$\xi(k+1) = \Phi\xi(k) + \Gamma u(k) \tag{2}$$
$$y(k) = H\xi(k) \tag{3}$$

where

$$\xi(k) = \begin{bmatrix} x(k) \\ x_d(k) \end{bmatrix}, \quad \Phi = \begin{bmatrix} A & \Psi \\ 0 & S \end{bmatrix}, \tag{4}$$

$$\Psi = \begin{bmatrix} B & 0 \end{bmatrix}, \quad H = \begin{bmatrix} C & 0 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} 0 \\ e_{n_d}^{n_d} \end{bmatrix}, \tag{5}$$

$$S = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & & 0 \\ & \vdots & & \ddots & 0 \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \tag{6}$$

where $e_m^k \in \mathbb{R}^m$ is the k^{th} unit vector, 0 is the appropriately sized zero matrix and $S \in \mathbb{R}^{n_d \times n_d}$ is the left-shift matrix. Thus, the total model order is $n = n_s + n_d = 21$. We will refer to this system as

$$G_2 = \{\Phi, \Gamma, H, 0\}. \tag{7}$$

III. Linear State Feedback Control

We consider the general case of tracking a trajectory, $(\xi_{ref}(k),\ y_{ref}(k),\ u_{ref}(k))$ where $\xi_{ref}(k)$ is the desired reference trajectory, $u_{ref}(k)$ is the associated open-loop control, and $y_{ref}(k) = H\xi_{ref}(k)$. In the simplest case, $(\xi_{ref}(k),\ y_{ref}(k),\ u_{ref}(k))$ are all constants such that y_{ref} is a desired setpoint, ξ_{ref} is the associated steady-state state, and u_{ref} is the associated feedforward steady-state control.

[1]Due to the size of these matrices, we do not include them here. The interested reader may contact the author at roger.braker@colorado.edu to obtain this or other data used in this paper.

To achieve zero steady-state error to a step input, we augment the system with integral action. These augmented dynamics can be written as

$$\begin{bmatrix} \xi_i(k+1) \\ \xi(k+1) \end{bmatrix} = \Phi_{aug} \begin{bmatrix} \xi_i(k) \\ \xi(k) \end{bmatrix} + \Gamma_{aug} u(k) - \begin{bmatrix} 1 \\ 0 \end{bmatrix} y_{ref}(k) \tag{8}$$

$$u(k) = -K_{aug} \begin{bmatrix} \xi_i(k) \\ \hat{\xi}(k) - \xi_{ref}(k) \end{bmatrix} + u_{ref}(k) \tag{9}$$

$$y(k) = H_{aug} \begin{bmatrix} \xi_i(k) \\ \xi(k) \end{bmatrix} \tag{10}$$

where

$$\Phi_{aug} = \begin{bmatrix} 1 & H \\ 0 & \Phi \end{bmatrix}, \quad \Gamma_{aug} = \begin{bmatrix} 0 \\ \Gamma \end{bmatrix} \tag{11}$$

$$H_{aug} = \begin{bmatrix} 0 & H \end{bmatrix}, \quad K_{aug} = \begin{bmatrix} k_i & K \end{bmatrix}, \tag{12}$$

and $\hat{\xi}(k)$ is the estimate of $\xi(k)$ which is explained next.

A. Efficient State Estimation for Systems with Large Delay

The inclusion of the 9 units of delay significantly increases the dimension of the state space and the total *units* of delay will increase if we increase the sample frequency. This has the potential to make the computational cost of the state estimator quite high, either in terms of FPGA fabric or time or both. With an eye toward future expansion of the control architecture, we seek a method to minimize this cost. If we denote by $\hat{\xi}$ the estimate of ξ, then typically, the estimator update would be given by

$$\hat{\xi}(k+1) = (\Phi - LH)\hat{\xi}(k) + \Gamma u(k) + Ly(k), \tag{13}$$

where $L \in \mathbb{R}^n$ is the estimator gain. The bulk of the computation cost lies in the the matrix-vector multiplication of $(\Phi - LH)\hat{\xi}(k)$. Since Φ is block upper triangular, its poles are the union of the poles of A and S. If we agree that L should only move the poles of A but leave the poles of S (i.e., those associated with delay) unchanged, the computational cost can be significantly reduced. Partition the state $\hat{\xi}$ and estimator gain L as

$$L = \begin{bmatrix} L_1^T & L_2^T \end{bmatrix}^T \tag{14}$$

$$\hat{\xi}(k) = \begin{bmatrix} \hat{x}^T(k) & \hat{x}_d^T(k) \end{bmatrix}^T \tag{15}$$

$$L_1, \hat{x}(k) \in \mathbb{R}^{n_s}, \ L_2, \hat{x}_d(k) \in \mathbb{R}^{n_d}. \tag{16}$$

By letting $L_2 = 0$, then

$$\Phi - LH = \begin{bmatrix} A - L_1 C & \Psi \\ -L_2 C & S \end{bmatrix} \tag{17}$$

$$= \begin{bmatrix} A - L_1 C & \Psi \\ 0 & S \end{bmatrix}. \tag{18}$$

Thus, when $L_2 = 0$, the poles of $\Phi - LH$ are the union of the poles of $A - L_1 C$ and the poles of S, as desired. We can thus exploit the structure of (18) by expressing the estimator update equation as

$$\hat{x}(k+1) = (A - L_1 C)\hat{x}(k) + \Psi\hat{x}_d(k) + L_1 y(k) \tag{19}$$

$$\hat{x}_d(k+1) = S\hat{x}_d(k) + e_{n_d}^{n_d} u(k). \tag{20}$$

The only job of the matrix S is to shift $\hat{x}_d(k)$ to the left. This can be efficiently accomplished in the FPGA via a register type structure, obviating the need for another matrix-vector multiplication. Additionally, we can reduce the matrix-vector multiplication of $\Psi\hat{x}_d(k)$ to the vector-scalar multiplication $Bu(k-n_d)$.

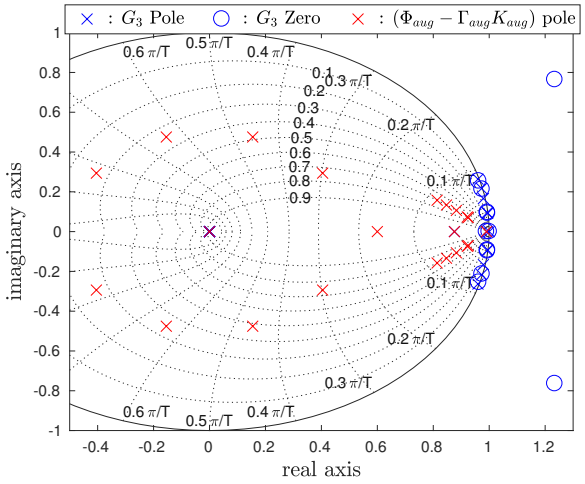

Fig. 4: Pole-zero plot of the open-loop system (blue) and the closed-loop poles (red).

B. Controller Design

In designing the linear control law, we maintain two systems

$$G_1 = \{A, B, C, 0\} \tag{21}$$

$$G_3 = \{\Phi_{aug}, \Gamma_{aug}, H_{aug}, 0\} \tag{22}$$

where G_3 is used to design the feedback gain K_{aug}, and G_1 is used to design the estimator gain L_1. We design the observer gain as the solution to the steady-state LQR problem for the dual of G_1 for weighting matrices $Q = 55.5C^TC$ and $R = 1$. In designing the feedback gain K_{aug}, we employ a pole-placement algorithm. All complex poles are assigned to have a damping of $\zeta = 0.7$. The natural frequency of the slowest two is increased by a factor of 1.16. We move the slowest real pole to cancel its neighboring zero and place 8 of the delay poles symmetrically about the origin with a radius of $r = 0.5$, leaving one pole at $z = 0$. A pole-zero plot of G_3 as well as the eigenvalues of $(\Phi_{aug} - \Gamma_{aug}K_{aug})$ can be seen in Fig. 4.

C. The Trouble with Linear Feedback

Aside from the obvious magnitude constraints on the input, the nPoint C300 is also current limited to 100 mA. Effectively, this translates to a maximum slew-rate limit on the input. Consider visiting the sequence of points

$$\{y^1_{ref} = \ 7.0\ \mu\text{m},\ y^2_{ref} = -7.0\ \mu\text{m}\} \tag{23}$$

with the stage beginning at rest.

Fig. 5 shows the NPXY100A response to following the step input sequence (23) (red curve) under the linear feedback control law designed in Section III-B. There are 800 samples between each command. The response to the first command y^1_{ref} is well damped with good rise time. However, in moving from y^1_{ref} to y^2_{ref}, the computed control changes too rapidly and the slew-rate threshold is reached, resulting in a severely deteriorated control action, as evidenced by the oscillations induced in the second part of Fig. 5. This same type of behavior will also be seen if, e.g., the magnitude of y^1_{ref} were increased beyond about 7.5 μm, as shown in Fig. 5 by the black-dotted trajectory. Indeed, in other trials not shown here, the difference between these two types of trajectories essentially bifurcates once a certain step size threshold is crossed (for a given gain K_{aug}).

IV. Constrained LQR

The main failure of designing a linear feedback law to track a step change in reference is its failure to account for both actuator saturation and input slew-rate limits. One way to deal with these constraints is to solve a finite horizon constrained LQR problem and track the resulting trajectory with linear feedback. If our model of the system is close to reality, the error between trajectories will be small, meaning that the perturbation from the designed input (which satisfies the constraints) will also be small.

To solve the finite horizon constrained LQR problem, we work in the error coordinates of G_1. Given a desired setpoint, y_{ref}, let u_{ss} be the associated steady-state control input and $x_{ss} = (I - A)^{-1}Bu_{ss}$ be the associated steady-state state vector. Define the error coordinates as $x_e = x - x_{ss}$ with control $\bar{u}(k)$ given by

$$\bar{u}(k) = (u(k) - u_{ss}) \in [\bar{u}^-,\ \bar{u}^+] \tag{24}$$

$$\bar{u}^- = -u_{\max} - u_{ss} \tag{25}$$

$$\bar{u}^+ = u_{\max} - u_{ss}. \tag{26}$$

Also, define the input slew rate as

$$\Delta u(k) \coloneqq u(k) - u(k-1) = \Delta\bar{u}(k). \tag{27}$$

Now we seek a solution to

$$\min_u \sum_{k=0}^{N-1} x_e(k)^T Q x_e(k) + \bar{u}(k)^T R\bar{u}(k) + x_e(N)^T Q_p x_e(N) \tag{28a}$$

$$\text{s.t.} \quad x_e(k+1) = Ax_e(k) + B\bar{u}(k) \tag{28b}$$

$$x_e(0) = x(0) - x_{ss} \tag{28c}$$

$$\bar{u}^- \le \bar{u}(k) \le \bar{u}^+,\ k = 0 \dots N-1 \tag{28d}$$

$$|\Delta\bar{u}(k)| \le (\Delta u)_{\max},\ k = 1 \dots N-1 \tag{28e}$$

where $(\Delta u)_{\max}$ is the slew-rate limit, $Q_p = Q_p^T \ge 0$ is the solution of the discrete algebraic Ricatti equation, $Q = Q^T \ge 0$, and $R > 0$. By writing $x_e(k)$ for $k = 1 \dots N$, in terms of $x_e(0)$ and the decision variables $\bar{u}(k)$, the equality constraints (28b) and (28c) can be eliminated and (28a) can be condensed into a quadratic program [21]

$$\min_{\bar{\mathcal{U}}} \ \frac{1}{2}\bar{\mathcal{U}}^T\mathcal{H}\bar{\mathcal{U}} + (\mathcal{M}x_e(0))^T\bar{\mathcal{U}} \tag{29}$$

$$\text{s.t.} \quad \mathcal{F}\bar{\mathcal{U}} \le b \tag{30}$$

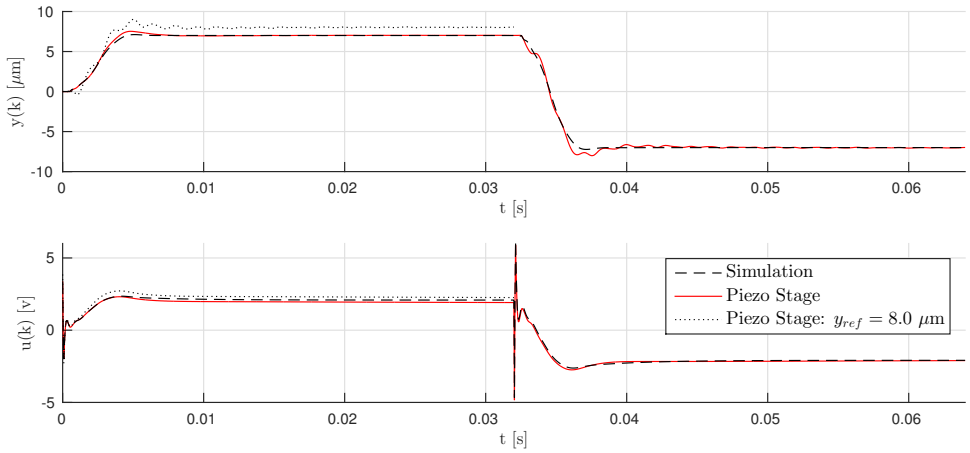

Fig. 5: Response of the NPXY100A to a sequence of step commands under the linear feedback law designed in Section III-B. The first step in the sequence, from $0 \rightarrow y^1_{ref} = 7.0$ μm shows good performance. The effect of canceling the real pole-zero pair manifests as the control signal decaying after the output has settled. However, in the second part, in the move $y^1_{ref} \rightarrow y^2_{ref}$, the control law requests a faster change in input than the C300 can provide and the slew-rate limit is reached. The result is a severely deteriorated system response. The black-dotted trajectory shows that this phenomenon will also occur by simply increasing the first step size.

where $\bar{\mathcal{U}} = \begin{bmatrix}\bar{u}(0) & \bar{u}(1) & \dots & \bar{u}(N-1)\end{bmatrix}^T$ is a stacked column vector of decision variables. The inequality constraints (28d)-(28e) appear as the linear inequality (30). The system dynamics as well as the penalty matrices Q and R are folded into the matrix $\mathcal{M}$ and Hessian $\mathcal{H}$.

Solving this problem offline using, for instance, Matlab's `quadprog` yields the optimal vector of stacked control inputs in the error coordinates, $\bar{\mathcal{U}}^*$. At time k, the optimal input in the non-error coordinates is $u^*(k) = (e_N^{k+1})^T\bar{\mathcal{U}}^* + u_{ss}$, which we compute along with the resulting optimal state trajectory, $\xi^*(k)$, and optimal output $y^*(k)$ when the $u^*(k)$ are applied in open-loop to the system G_2 (i.e., the system with delay included). Together, this results in the triple $(\Xi, \mathcal{Y}, \mathcal{U})$, where [2]

$$\Xi = \{\xi^*(k)\}_{k=0}^{N-1} \tag{31}$$

$$\mathcal{Y} = \{y^*(k)\}_{k=0}^{N-1} \tag{32}$$

$$\mathcal{U} = \left\{(e_N^{k+1})^T\bar{\mathcal{U}} + u_{ss}\right\}_{k=0}^{N-1}. \tag{33}$$

We adopt the notation that, e.g., Ξ_k is the k^{th} element in the sequence of optimal states, Ξ. We then track $(\Xi, \mathcal{Y}, \mathcal{U})$, i.e., at each sample instant, apply the feedforward control $\mathcal{U}_k$ and track the desired output $\mathcal{Y}_k$ and state trajectory Ξ_k. Thus, the control input to the piezo stage is

$$u(k) = \mathcal{U}_k - K(\hat{\xi}(k) - \Xi_k) - K_i\xi_i(k) \tag{34}$$

$$= \mathcal{U}_{ff,k} - K\hat{\xi}(k) - K_i\xi_i(k) \tag{35}$$

where

$$\mathcal{U}_{ff} = \{\mathcal{U}_k + K\Xi_k\}_{k=0}^{N-1}. \tag{36}$$

[2]We emphasize for clarity that (31)-(33) are *sequences*, not stacked vectors.

A. The Inverse LQR Problem

One key difficulty in applying the constrained LQR approach is the selection of weighting matrices, (Q, R). One common method is to set $R = 1$ and let $Q = \gamma C^TC$ for some $\gamma > 0$. As γ becomes large, this approach essentially incites pole-zero cancellation [22]. For our system, the poles and zeros are not only imperfectly known, but are liable to move depending where we are on the stage [2], [18]. Thus, we do not consider this a viable approach.

Ultimately, we would like a trajectory that looks like the first step command in Fig. 5 but which respects the saturation and slew-rate limits for larger step sizes. In other words, we would like to solve the constrained LQR problem using (Q, R) matrices which, in the infinite horizon, unconstrained case, would yield K. One way to do this is to try to solve the inverse optimal control problem [23]. That is, solve for decision variables $Q = Q^T \geq 0$, $R > 0$, $P = P^T \geq 0$, and $P_1 = P_1^T > 0$, from

$$(A - BK)^TP(A - BK) - P - K^TRL + Q = 0 \tag{37}$$

$$B^TPA - (R + B^TPB)K = 0 \tag{38}$$

$$A^TP_1A - P_1 \leq Q. \tag{39}$$

We solve this problem using YALMIP [24]. Note that this problem does not always have a solution, since not all possible eigenvalues are reachable via LQR. Thus, as with most methods of choosing LQR weights, some iteration is required.

B. Practical Considerations

Implementing the control law as (35) rather than (34) offers a significant advantage in a real-time control loop. Equation (34) implies that at each sample instant, the control

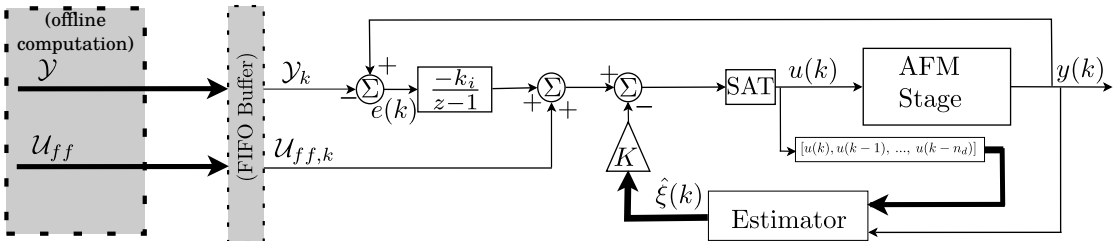

Fig. 6: Block diagram of the trajectory tracking controller implementation.

loop needs to acquire 23 distinct pieces of data from the FIFO buffer: the optimal input $\mathcal{U}_k$, the optimal state trajectory Ξ_k, and the optimal output $\mathcal{Y}_k$ (or compute $H\Xi_k$ online). By instead implementing (35), we only need to read two distinct values from the FIFO buffer, since the sequence $\mathcal{U}_{ff}$ from (36) is a sequence of *scalars* which can be computed offline.

Now, consider the problem of visiting (for simplicity) a collection of two points, $\{y^1_{ref}, y^2_{ref}\}$. In the entire scheme of compressed sensing, between visiting y^1_{ref} and y^2_{ref}, we must lower the tip, take a measurement, and retract the tip. Since this process will take a non-deterministic amount of time, another important question is how to choose both the horizon length, N, and the initial condition $x^2(0)$ to solve (28a) for setpoint y^2_{ref}. Although the tip is not lowered until $y(k)$ has settled, the issue is complicated by the fact we are designing (Q, R) to cancel the low-frequency real pole-zero pair. Thus, although we might declare the position $y(k)$ to have settled, there will be an internal state which has not yet reached its steady-state value. However, once the position has settled, we begin the tip-down-measure-tip-up process, which is relatively time consuming. During this time, we expect the remaining states to have settled. Thus, we use $x^2(0) = x^1_{ss}$. We select the horizon length to be long enough so that once $k = N$, the internal states are sufficiently close to their steady-state values. Thus, for samples $k > N$, i.e., we have exhausted the data in the FIFO buffer, we use $u_{ff}(k) = u_{ss} + K\xi_{ss}$ and $y_{ref}(k) = y_{ss}$.

V. Constrained LQR Results

In designing the gains for the Constrained LQR trajectory tracking scheme, we use the same observer design as before. In addition, we design two separate control gains, K_{trk} and K_{lqr}. The gain K_{lqr} is chosen to generate the (Q, R) matrices and was chosen so that the closed-loop eigenvalues of $(A - BK_{lqr})$ are as shown in Fig. 7. As in the standard linear feedback case, it is designed such that the low-frequency real pole-zero pair cancel. The gain K_{trk} is chosen and implemented to track the optimal trajectory.

We consider visiting the same sequence of setpoints as in (23). We solved the constrained LQR problem (28a) for each move (i.e., $0 \to y^1_{ref}$ and $y^1_{ref} \to y^2_{ref}$) using a horizon of $N = 400$ samples but track each setpoint for 800 samples using the strategy outlined at the end of Section IV-B. This results in two trajectories which we label as

$$(\Xi^1, \mathcal{Y}^1, \mathcal{U}^1) \tag{40}$$
$$(\Xi^2, \mathcal{Y}^2, \mathcal{U}^2). \tag{41}$$

The results of tracking the optimal trajectory are shown in Fig. 8, which shows a drastic improvement over the results of the standard linear feedback controller in Fig 5. The settling times of all experiments are collected in Table I. Although the linear controller yields a better settle time in simulation, the constrained LQR trajectory tracking controller performs significantly better in real application because actuator constraints are not violated.

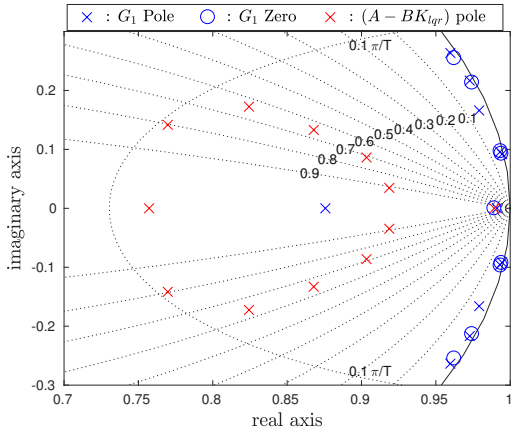

Fig. 7: Pole-zero map of the system G_1 (blue) and the requested eigenvalues of $(A - BK_{lqr})$ (red).

	Lin Fdbk		Constrained LQR	
	$0 \to y^1_{ref}$	$y^1_{ref} \to y^2_{ref}$	$0 \to y^1_{ref}$	$y^1_{ref} \to y^2_{ref}$
sim.	0.00564	0.00592	0.00492	0.0074
exp.	0.00724	0.025	0.00684	0.0126

TABLE I: Settling times for the linear feedback controller and constrained LQR trajectory tracking controller for the sequence of setpoints (23). We use a 1% settling-time criterion and all times are in seconds.

VI. Conclusions and Future Directions

In this paper we have argued that a high-fidelity model of a piezo X-Y stage is critical to obtain good setpoint

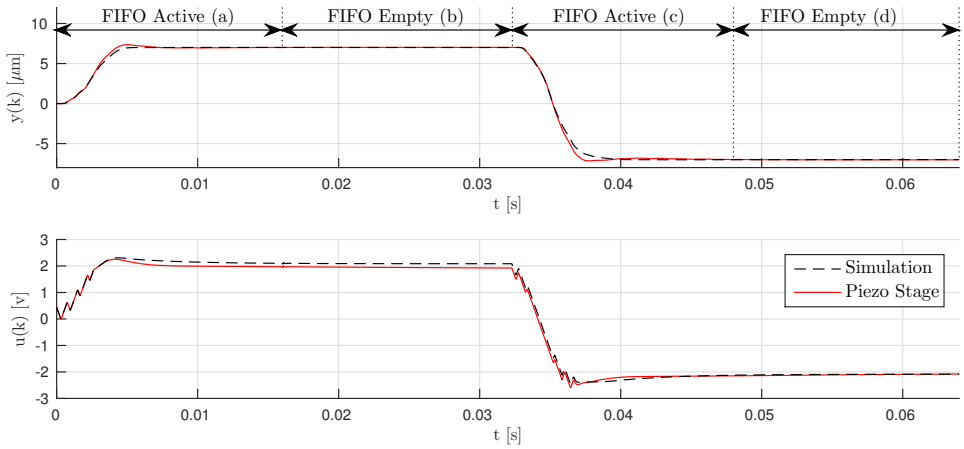

Fig. 8: System response tracking the constrained LQR optimal trajectories for the sequence of setpoints in (23). In part (a): The FIFO buffer is active, feeding values $(\mathcal{Y}^1(k), \mathcal{U}^1 + K\Xi^1(k))$. In part (b), the FIFO buffer is empty, and the values $(y^1_{ref}, u^1_{ss} + K\xi^1_{ss})$ are used. The process repeats for parts (c) and (d).

tracking performance. Moreover, we demonstrated that actuator constraints, and in particular slew-rate constraints, pose a significant challenge when applying linear feedback to track a step input in a highly resonant piezo stage. We then showed that this can be overcome by tracking a sequence of optimal trajectories obtained by solving a finite horizon, constrained LQR problem. We also addressed many of the related practical concerns and showed that our approach can yield good experimental performance.

Future work will investigate whether the small overshoot seen in Fig. 5 and Fig. 8 can be explained with a better linear model or if it is due to hysteresis and drift. We will also compare the trajectory tracking method presented here to a model predictive control approach, where the optimization is solved *online* but over a much shorter horizon.

REFERENCES

[1] T. Ando, "High-speed atomic force microscopy coming of age," *Nanotech.*, vol. 23, no. 6, p. 062001, 2012.

[2] J. A. Butterworth, L. Y. Pao, and D. Y. Abramovitch, "Dual-adaptive feedforward control for raster tracking with applications to AFMs," in *Proc. IEEE Conf. on Control Applications*, Sep. 2011, pp. 1081–1087.

[3] S. Salapaka, A. Sebastian, J. P. Cleveland, and M. V. Salapaka, "High bandwidth nano-positioner: A robust control approach," *Rev. of Scientific Instruments*, vol. 73, no. 9, pp. 3232–3241, 2002.

[4] B. Bhikkaji, M. Ratnam, A. J. Fleming, and S. O. R. Moheimani, "High-performance control of piezoelectric tube scanners," *IEEE Trans. on Control Systems Tech.*, vol. 15, no. 5, pp. 853–866, Sep. 2007.

[5] M. S. Rana, H. R. Pota, I. R. Petersen, and H. Habibullah, "Effect of improved tracking for atomic force microscope on piezo nonlinear behavior," *Asian J. of Control*, vol. 17, no. 3, pp. 747–761, 2015.

[6] Y. K. Yong and S. O. R. Moheimani, "Collocated z-axis control of a high-speed nanopositioner for video-rate atomic force microscopy," *IEEE Trans. on Nanotech.*, vol. 14, no. 2, pp. 338–345, Mar. 2015.

[7] G. Schitter, K. J. Astrom, B. E. DeMartini, P. J. Thurner, K. L. Turner, and P. K. Hansma, "Design and modeling of a high-speed AFM-scanner," *IEEE Trans. on Control Systems Tech.*, vol. 15, no. 5, pp. 906–915, Sep. 2007.

[8] B. J. Kenton and K. K. Leang, "Design and control of a three-axis serial-kinematic high-bandwidth nanopositioner," *IEEE/ASME Trans. on Mechatronics*, vol. 17, no. 2, pp. 356–369, Apr. 2012.

[9] Y. K. Yong, A. Bazaei, and S. O. R. Moheimani, "Video-rate Lissajous-scan atomic force microscopy," *IEEE Trans. on Nanotech.*, vol. 13, no. 1, pp. 85–93, Jan. 2014.

[10] J. Worthey and S. Andersson, "Local circular scanning for autonomous feature tracking in AFM," in *Proc. American Control Conf.*, July 2015, pp. 3490–3495.

[11] S. Andersson and L. Pao, "Non-raster sampling in atomic force microscopy: A compressed sensing approach," in *Proc. American Control Conf.*, June 2012, pp. 2485–2490.

[12] E. Candes and M. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[13] M. Workman, "Adaptive proximate time-optimal servomechanisms," Ph.D. dissertation, Stanford University, Stanford, CA, Mar. 1987.

[14] L. Pao and G. Franklin, "Proximate time-optimal control of third-order servomechanisms," *IEEE Trans. on Automatic Control*, vol. 38, no. 4, pp. 560–580, Apr. 1993.

[15] R. A. Braker and L. Y. Pao, "Proximate time-optimal control of a second-order flexible structure," in *Proc. IEEE Conf. on Control Applications*, Sep. 2015, pp. 840–845.

[16] B. Bhikkaji and S. O. R. Moheimani, "Integral resonant control of a piezoelectric tube actuator for fast nanoscale positioning," *IEEE/ASME Trans. on Mechatronics*, vol. 13, no. 5, pp. 530–537, Oct. 2008.

[17] I. A. Mahmood and S. O. R. Moheimani, "Improvement of accuracy and speed of a commercial AFM using positive position feedback control," in *Proc. American Control Conf.*, June 2009, pp. 973–978.

[18] J. A. Butterworth, "Combined feedback and adaptive feedforward control for tracking applications in atomic force microscopes," Ph.D. dissertation, University of Colorado, Boulder, CO, Apr. 2011.

[19] R. N. Jacques and D. W. Miller, "Multivariable model identification from frequency response data," in *Proc. IEEE Conf. on Decision and Control*, Dec. 1993, pp. 3046–3051.

[20] J.-N. Juang and R. S. Pappa, "An eigensystem realization algorithm for modal parameter identification and model reduction," *J. of Guidance, Control, and Dynamics*, vol. 8, no. 5, pp. 620–627, 1985.

[21] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3 – 20, 2002.

[22] B. D. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods*. Englewood Cliffs, N.J.: Prentice-Hall, 1989.

[23] E. Ostertag, *Mono- and Multivariable Control and Estimation: Linear Quadratic and LMI Methods*. New York;Heidelberg: Springer, 2011.

[24] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *CCA/ISIC/CACSD*, Sep. 2004. [Online]. Available: http://control.ee.ethz.ch/index.cgi?action=details;id=2088;page=publications