An Application of the Fast Gradient Method to Model Predictive Control of an Atomic Force Microscope X-Y Stage

Roger A. Braker and Lucy Y. Pao

Abstract— **Random sub-sampling imaging methods in Atomic Force Microscopy require the piezo X-Y stage to track a sequence of step inputs. Control slew-rate limits combined with linear feedback methods have been shown to limit achievable performance in this scenario. Due to its natural ability to account for actuation constraints, we consider the application of Model Predictive Control. By recasting the problem in an incremental form, the arising quadratic program takes a form that can be solved efficiently. Specifically, we solve an input constrained Model Predictive Control problem with 50 states, a control horizon of 12 samples, and a sample frequency of 25 kHz using the Fast Gradient Method. We compare simulation and experimental results using this method.**

I. Introduction

The traditional Atomic-Force-Microscope (AFM) imaging method is to raster scan the specimen with an atomically sharp probe. Since the performance of an AFM depends on the overall closed-loop dynamics, efforts to increase imaging speed have focused both on increasing the raster scan rate via advanced control algorithms [1], [2], [3] and on increasing the mechanical bandwidth through improved mechanism design [4], [5].

Nonetheless, raster scanning is still not only a very time consuming process, but can also lead to damaging either the specimen or the AFM probe tip or both. This has led researchers to consider alternatives to the traditional raster scan. Specifically, there is recent motivation and interest in sub-sampling (acquiring, e.g., approximately 10-30% of the pixels) using, e.g., a subset of the scan lines [6], or other alternative scan patterns [7], [8]. Another approach is to randomly sample the specimen [9], [10] and reconstruct the topology of the specimen via compressed sensing (CS) [11]. Sub-sampling reduces the tip-specimen interaction and can improve the integrity of both the specimen and AFM tip. In general, the best reconstruction for a given sub-sampling percentage occurs when using a random sampling of the pixels. When sub-sampling in this point-to-point manner, minimizing the imaging time requires that the rest-to-rest maneuver times between point measurements be minimized.

Thus, while the goal of control laws designed for raster scanning is to accurately track a triangle wave, the control goal in the random sub-sampling approach is to track a sequence of step inputs with minimum settling times. While the minimum-time regulation problem has been successfully addressed for a variety of lower order systems [12], [13], [14], these techniques become intractable to generalize to the higher order models needed for many AFM stages.

Moreover, AFM controllers which yield excellent results in raster scanning applications do not necessarily give adequate performance when tracking step commands, as we showed in [15]. We also showed in that paper that control input constraints, and in particular slew-rate constraints, present a significant limitation on achievable performance when standard linear feedback laws are used and most of the existing AFM control methods do not explicitly account for such constraints.

Model Predictive Control (MPC) is a method with a natural ability to handle general input and state constraints [16]. In linear MPC, a constrained, finite-horizon Linear Quadratic optimal control problem is solved at each sample instant. The solution of this problem results in a sequence of optimal control inputs. The first element of the sequence is applied to the actual system, the rest are discarded and the process repeats at the next sample instant. Our interest in MPC lies in the ability to naturally incorporate control input constraints in the optimization.

In linear MPC problems, the optimization can be recast as a Quadratic Program (QP) and there has been significant research into developing algorithms specifically tailored to the QPs arising in MPC. These include active set methods [17], interior point methods [18], and gradient methods [19], [20]. For MPC problems with only input constraints and which are implemented in a Field Programmable Gate Array (FPGA), the Fast Gradient Method (FGM) formulation described in [21] is particularly attractive and is the algorithm we use in this paper.

MPC has been applied before to AFMs but with only a 1 kHz sample rate [22], which is considerably slower than typical AFM sample rates. In [23], the authors use MPC to suppress vibration in an active structure, achieving a 25 kHz sampling rate with a control horizon of 4 samples. In [24], the authors apply MPC to a piezo-actuator with a 10 kHz sample rate. We note that in their work, the absence of a mechanical stage leads to a considerably simpler set of plant dynamics (second-order model) than what we consider here.

In this paper, we develop and apply an input constrained MPC control formulation that satisfies actuator slew-rate constraints and is implemented at 25 kHz. We discuss some numerical conditioning issues and how this relates to tuning the controller. The main contributions of this paper are (1)

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.

to show that MPC can be an effective control method for piezo nano-positioning stages tracking setpoints using standard AFM sample rates, (2) to demonstrate an application of the FGM on a real system with 50 states, a control horizon of 12 samples, and a sample frequency of 25 kHz, and (3) to explore numerical conditioning issues and their implications for performance. In Section II, we outline our experimental setup as well as develop a system model. In Section III, we derive the particular formulation of MPC that we have adopted. In Section IV, we give details about our implementation of the FGM QP solver, and we outline our method for tuning the MPC controller in Section V. Section VI compares simulation results with experiment, and we provide concluding remarks in Section VII.

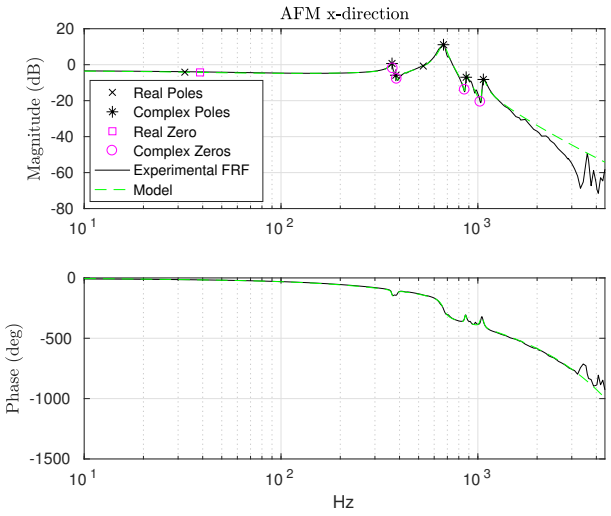
II. Experimental Setup and System Modeling

The experimental setup in our lab centers around an nPoint NPXY100A piezo x-y stage with a range of 100 μm in both directions. The stage is driven by an nPoint C300 signal conditioner. The C300 has the ability to run in open-loop mode as well as provide basic PID control, though the PID functionality is always turned off for the work described in this paper. Unfortunately, whether in Open-Loop or PID mode, signals into and out of the C300 always run through an internal DSP, which introduces about 400 μs of total delay from input to output. Inputs to the C300 saturate at $\pm$10v and the actual output to the NPXY100A is current limited to 100 mA.

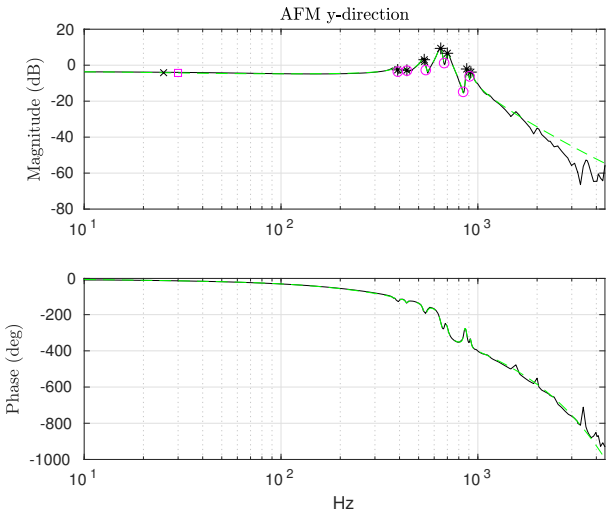
All control algorithms discussed in this paper use a sample frequency of 25 kHz and are programmed into a National Instruments Compact Rio 9082. The main features of this hardware are a Xilinx Spartan LX150 FPGA and 16-bit Analog-to-Digital (ADC) and Digital-to-Analog (DAC) modules.

It is common in the AFM literature to consider the x and y axes as decoupled, single-input-single-output (SISO) systems, and that is the point of view we adopt here. However, rather than implementing two separate control loops for the two axes, we implement a single MIMO control loop where the model has a block diagonal structure. We emphasize this fact because it means that in future work, compensation of the cross axis coupling is essentially reduced to a modeling and control tuning problem–the *structure* for MIMO control is already in place.

To obtain a parametric model of the NPXY100A stage for control purposes, we first obtain the frequency response function (FRF). We generate the FRF for both axes 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. Taking the ratio of these coefficients yields the FRF. More details on this method can be found in the appendix of [25], which our implementation follows closely. The black curves in Figs. 1a and 1b are the resulting FRFs for the x and y axes, respectively.



(a)



(b)

Fig. 1: Frequency response of NPXY100A stage: (a) x-direction and (b) y-direction. In both figures, the physical system is represented by the solid black curve and the state-space model is dashed 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.

From previous modeling efforts as well as communication with nPoint Inc., we know that at a sample frequency of 25 kHz, the C300 introduces $n_d = 10$ units of delay. This amount of delay is first removed from the experimental frequency responses. To obtain a parametric model, we employ an eigenspace realization algorithm (ERA) [26], [27]. The impulse responses required by the ERA are obtained via an inverse Fourier transform on the FRF data. We then apply the ERA outlined in [26] to the impulse data for the x and y directions, independently. This yields two SISO state-space

systems

$$G_x = \{A_x,\, B_x,\, C_x,\, 0\}$$
$$G_y = \{A_y,\, B_y,\, C_y,\, 0\}.$$

We truncate the model order for G_x at $n_x = 12$ and for G_y at $n_y = 16$. The resulting magnitude fits are shown in Fig. 1. We then add the 10 units of delay into the model, which recovers the phase fits shown in Fig. 1. Modeling all delay as input delay and combining the two systems as a decoupled MIMO system, the dynamics for both axes are given by

$$\begin{aligned} \xi(k+1) &= A\xi(k) + Bu(k) \\ \nu(k) &= C\xi(k) \end{aligned} \tag{1}$$

where

$$A = \begin{bmatrix} \begin{bmatrix} A_x & \Psi_x \\ 0 & S \end{bmatrix} & 0 \\ 0 & \begin{bmatrix} A_y & \Psi_y \\ 0 & S \end{bmatrix} \end{bmatrix}, \quad B = \left[\begin{array}{c|c} \begin{matrix} 0_{n_x} \\ e_{n_d}^{n_d} \end{matrix} & 0 \\ \hline 0 & \begin{matrix} 0_{n_y} \\ e_{n_d}^{n_d} \end{matrix} \end{array}\right]$$

$$\Psi_x = \begin{bmatrix} B_x & 0 \end{bmatrix}, \Psi_y = \begin{bmatrix} B_y & 0 \end{bmatrix},$$

$$C = \left[\begin{array}{cc|cc} C_x & 0_{n_d}^T & 0 & \\ \hline 0 & & C_y & 0_{n_d}^T \end{array}\right]$$

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

where $e_m^k \in \mathbb{R}^m$ is the k^{th} unit vector. When it is easily inferred, 0 is the appropriately sized zero-vector (matrix), otherwise its size is indicated by 0_m $(0_{m\times k})$. Note that $S \in \mathbb{R}^{n_d \times n_d}$ is the left-shift matrix and serves to delay the input into A_x and A_y by 10 samples. The total model order is $n_s = n_x + n_d + n_y + n_d = 48$ and the system has $n_u = 2$ inputs and $n_o = 2$ outputs. We refer to this system as

$$G_p = \{A, B, C, 0\}. \tag{2}$$

III. Model Predictive Control

In previous work, we saw that in trying to develop a high-bandwidth linear feedback law for tracking step inputs for our piezo stage, the slew-rate constraint presented by the current limit of the power amplifier is of primary importance [15]. In this paper, we develop an MPC formulation that only accounts for the slew-rate constraint. This allows us to formulate the problem in a way such that an efficient QP solver can be utilized, and allows us to achieve a 25 kHz sample rate.

To develop the MPC formulation, we convert the system G_p in (2) into an incremental form on $\Delta u(k) \coloneqq u(k) - u(k-1)$, by augmenting G_p with n_u extra states $\xi_u(k) \in \mathbb{R}^{n_u}$ such that

$$\xi_u(k) = u(k-1).$$

It follows that

$$\tilde{\xi}(k+1) = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \tilde{\xi}(k) + \begin{bmatrix} B \\ I \end{bmatrix} \Delta u(k) \tag{3a}$$

$$\nu(k) = \begin{bmatrix} C & 0 \end{bmatrix} \tilde{\xi}(k) \tag{3b}$$

$$\tilde{\xi}(k) \coloneqq \begin{bmatrix} \xi(k) \\ \xi_u(k) \end{bmatrix} \tag{3c}$$

$$\tilde{\xi}(0) = \begin{bmatrix} \xi(0) \\ u(-1) \end{bmatrix}. \tag{3d}$$

We call this system $\tilde{G} = \{\tilde{A}, \tilde{B}, \tilde{C}, 0\}$, which has $\tilde{n}_s = 50$ states. To solve the setpoint tracking problem, we work in the error coordinates of $\tilde{G}$. For an arbitrary reference ν_r, in steady state we have $\Delta u_{ss} = 0$ and $\tilde{\xi}_{ss} = N_\xi \nu_r$ where $N_\xi \in \mathbb{R}^{\tilde{n}_s \times n_u}$ is found by solving

$$\begin{bmatrix} N_\xi \\ N_u \end{bmatrix} = \begin{bmatrix} I - \tilde{A} & -\tilde{B} \\ \tilde{C} & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ I \end{bmatrix}, \tag{4}$$

which will give $N_u \equiv 0$. The error state, $\tilde{\xi}_e(k) = \tilde{\xi}(k) - \tilde{\xi}_{ss}$ has dynamics

$$\begin{aligned} \tilde{\xi}_e(k+1) &= \tilde{A}\tilde{\xi}(k) + \tilde{B}\Delta u(k) - \tilde{\xi}_{ss} \\ &= \tilde{A}\tilde{\xi}_e(k) + \tilde{B}\Delta u(k). \end{aligned}$$

Given the current state $\tilde{\xi}_e(k)$, the MPC control law sets $\Delta u(k) = \mu(0)$ where $\mu(0)$ is obtained by solving the finite horizon linear-quadratic optimal control problem

$$\begin{aligned} V^*(\tilde{\xi}_e(k)) = \min_{\mu(i)} & \sum_{i=0}^{N-1} z(i)^T \tilde{Q} z(i) + \mu(i)^T \tilde{R} \mu(i) && \text{(5a)} \\ & + z(N)^T \tilde{Q}_p z(N) \\ \text{s.t.} \quad & z(i+1) = \tilde{A} z(i) + \tilde{B}\mu(i) && \text{(5b)} \\ & z(0) = \tilde{\xi}_e(k) && \text{(5c)} \\ & ||\mu(i)||_\infty \leq (\Delta u)_{\max} && \text{(5d)} \end{aligned}$$

where N is the horizon length, $\tilde{Q}_p = \tilde{Q}_p^T \geq 0$ is the solution of the discrete algebraic Ricatti equation, $\tilde{Q} = \tilde{Q}^T \geq 0$, and $\tilde{R} = \tilde{R}^T > 0$.

By defining a vector of stacked decision variables, $\Delta\mathcal{U}^T = \begin{bmatrix} \mu^T(0) & \mu^T(1) \dots \mu^T(N-1) \end{bmatrix}$, and writing the states in terms of the decision variables and current state $\tilde{\xi}_e(k)$, the equality constraint (5b) can be eliminated and the minimization problem (5) can be condensed into a quadratic program [16]

$$\begin{aligned} V^*(\tilde{\xi}_e(k)) = \min_{\Delta\mathcal{U}} \; & \frac{1}{2}\Delta\mathcal{U}^T H \Delta\mathcal{U} + (\Phi\tilde{\xi}_e(k))^T \Delta\mathcal{U} && \text{(6a)} \\ \text{s.t.} \; & \begin{bmatrix} I \\ -I \end{bmatrix} \Delta\mathcal{U} \leq \mathbb{1}\Delta u_{\max} && \text{(6b)} \end{aligned}$$

where $\mathbb{1} \in \mathbb{R}^{2Nn_u}$ is a vector of ones. The inequality constraints (5d) appear as the linear inequality (6b). The system dynamics matrices as well as the penalty matrices $\tilde{Q}$, $\tilde{Q}_p$, and $\tilde{R}$ are folded into the matrix $\Phi \in \mathbb{R}^{Nn_u \times \tilde{n}_s}$ and Hessian $H \in \mathbb{R}^{Nn_u \times Nn_u}$. Solving the quadratic program (6) comprises the core computational difficulty in applying the MPC control law. We discuss our hardware implementation of a QP solver able to handle (6) in Section IV-A.

Finally, note that if (6) is *unconstrained*, then it has a closed-form solution, $\Delta\mathcal{U} = -H^{-1}\Phi\tilde{\xi}_{eo}$. Taking only the first $n_u = 2$ elements of $\Delta\mathcal{U}$, we arrive at the linear feedback law

$$\begin{aligned} \Delta u(k) &= -K\tilde{\xi}_{eo} \\ K &= \begin{bmatrix} I_{n_u} & 0 & \dots & 0 \end{bmatrix} (H^{-1}\Phi). \end{aligned} \tag{7}$$

In Section VI, we will compare the system performance using the constrained MPC of (6) to the unconstrained solution (7).

A. Zero-offset Tracking

To achieve zero-offset tracking of a step input and to mitigate the effects of plant-model mismatch, we employ disturbance estimation [28], [29], [30]. Using a constant output disturbance model, we augment the original system G_p in (2) with a disturbance state, $\hat{\xi}_o(k) \in \mathbb{R}^{n_o}$ such that the estimator dynamics are

$$\begin{aligned} \begin{bmatrix} \hat{\xi}(k+1) \\ \hat{\xi}_o(k+1) \end{bmatrix} &= \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{\xi}(k) \\ \hat{\xi}_o(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(k) \\ &\quad + L_o \left(\nu(k) - \hat{\nu}(k)\right) \\ \hat{\nu}(k) &= \begin{bmatrix} C & I \end{bmatrix} \begin{bmatrix} \hat{\xi}(k) \\ \hat{\xi}_o(k) \end{bmatrix} \end{aligned}$$

where L_o is the estimator gain designed such that the estimator system, which we refer to as $G_o = \{A_o, B_o, C_o, 0\}$, is stable. The additional disturbance state $\hat{\xi}_o$ is not used in the control itself. Rather, it is used to modify the setpoint so that the setpoint driving the MPC control law is given by

$$\nu_e = \nu_r - \hat{\xi}_o(k).$$

Then the MPC with disturbance estimation control law is

$$\begin{aligned} \tilde{\xi}_e(k) &= \begin{bmatrix} \hat{\xi}(k) \\ \xi_u(k) \end{bmatrix} - N_\xi \nu_e \\ \Delta u(k) &= \begin{bmatrix} I_{n_u} & 0 & \dots & 0 \end{bmatrix} \arg\min_{\Delta\mathcal{U}} V^*(\tilde{\xi}_e(k)) \\ u(k) &= \Delta u(k) + \xi_u(k). \end{aligned}$$

which is illustrated in the block diagram in Fig. 2.

IV. MPC Implement-ability analysis

A. The FGM algorithm

One of the challenges in applying MPC to systems with fast sample rates is solving the QP in (6) within the sample period. In our case, this means the QP must be solved *faster* than 40 μs, to allow for the computation of the state estimate etc. as well. In this paper, we use a formulation of the FGM which has been optimized to take advantage of the parallelism inherent in FPGAs [21]. Although the FGM can solve the QP very efficiently, it is limited to problems where the constraint set $\mathbb{K}$ is simple, such as a box constraint. Working with the incremental form in (3) has allowed us to cast the slew-rate constraint $||\Delta u(k)||_\infty \leq (\Delta u)_{\max}$ as such a box constraint.

Our hardware implementation of the FGM differs somewhat from the one described in [21]. To highlight those differences, we repeat their algorithm here. This particular form of the FGM uses a fixed step size β and is given in Algorithm 1:

Algorithm 1 Fast Gradient Method

Require: Initial iterates $v_0 = \Delta\mathcal{U}_0 \in \mathbb{K}$; upper (lower) bound L ($\alpha > 0$) on maximum (minimum) eigenvalues of H; step size $\beta = (\sqrt{L} - \sqrt{\alpha})/(\sqrt{L} + \sqrt{\alpha})$.

1: $f = (1/L)\Phi\tilde{\xi}_e(k)$
2: **for** $j = 0 \dots J_{\max} - 1$ **do**
3: $\quad t_j = (I - H/L)v_j - f$
4: $\quad \Delta\mathcal{U}_{j+1} = \pi_{\mathbb{K}}(t_j)$
5: $\quad v_{j+1} = (1+\beta)\Delta\mathcal{U}_{j+1} - \beta\Delta\mathcal{U}_j$
6: **end for**

Note that $\mathbb{K}$ is the set of feasible control inputs, i.e., (6b) and Line 4 is a projection of the iterate onto this set. In our case with box constraints, this is a simple, parallelize-able saturation of each element of the iterate. The matrices $I - H/L$ and Φ as well as the constants β and $1 + \beta$ are all computed offline. The bulk of the computational burden of the algorithm is in the matrix-vector multiplications $f = \Phi\tilde{\xi}_e(k)$ and $(I - H/L)v_j$.

In [21], they describe a matrix-vector multiplication hardware implementation which computes dot products sequentially (e.g., one at a time) but where each scalar multiplication in a single dot product is done in parallel and the results are summed in an adder reduction tree. This method is problematic for our implementation, since there need to be as many multiply blocks as there are columns in the wider of H or Φ. In our case, Φ has 50 columns. Our later analysis indicates that between 22 and 32 bits are needed, so that each multiply block requires four DSP slices. Yet, our FPGA only has 180 DSP slices. We therefore modify the implementation so that each dot product is computed in parallel, but with no parallelism in the individual dot products. Each dot product is implemented as a sequential Multiply Accumulate (MAC) operation. Thus, taking $N = 12$ as we do in Section V, we only need $4Nn_u = 96$ DSP slices. Our implementation is illustrated in Fig. 3.

By multiplexing the inputs to each MAC and controlling how many clock cycles each MAC operation runs, we can compute dot products for arbitrarily sized vectors. This allows us to use the same hardware resources to compute f in line 1, *all* of line 3, and line 5. More specifically, we compute line 3 as a single matrix-vector multiplication

$$t_j = \begin{bmatrix} I - H/L & f \end{bmatrix} \begin{bmatrix} v_j \\ -1 \end{bmatrix}.$$

Similarly, we compute line 5 also as a matrix-vector multiplication

$$v_{j+1} = \begin{bmatrix} \Delta\mathcal{U}_{j+1} & \Delta\mathcal{U}_j \end{bmatrix} \begin{bmatrix} 1+\beta \\ -\beta \end{bmatrix}.$$

V. MPC Tuning

The convergence rate of the FGM depends on both the condition number κ of the Hessian H and the horizon

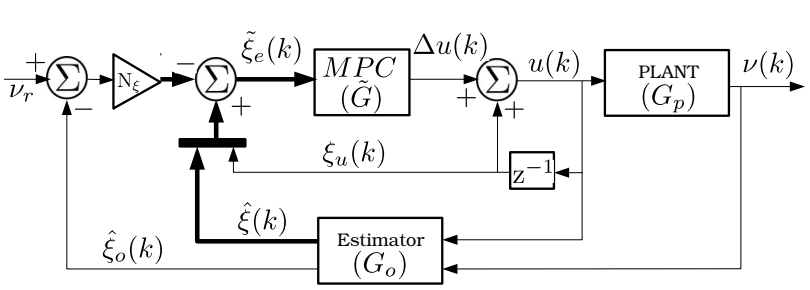


Fig. 2: Block diagram of the implementation of the MPC control law with disturbance estimation. Thin signal lines are vectors in $\mathbb{R}^2$, while thick signal lines are vectors in $\mathbb{R}^{48}$ (for $\hat{\xi}(k)$) or $\mathbb{R}^{50}$ (for $\tilde{\xi}_e(k)$).

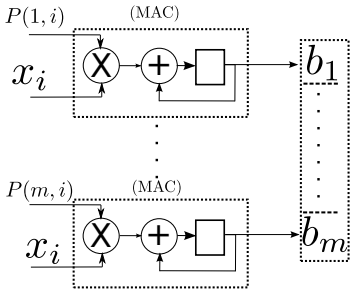


Fig. 3: Matrix multiplication architecture for a general matrix-vector multiplication $Px = b$, where $P \in \mathbb{R}^{m\times n}$. Each MAC block computes a single dot product. In our implementation, there are 24 MACs ($m = 24$) and n varies between 2 and 50.

length N [21]. In turn, κ depends on the horizon length, N, the model, and the selection of $(\tilde{Q}, \tilde{R})$. Furthermore, the achievable precision is influenced by the wordsize used in the implementation as well as κ. Effectively, this means that tuning the MPC controller is intimately tied to its implementability. Indeed, [19] notes that an aggressively designed $\tilde{Q}$ (relative to $\tilde{R}$) or a large horizon N can lead to poor numeric conditioning in condensed MPC formulations.

In this section, we outline a basic design strategy that attempts to get the best performance from these constraints. Our overall goal is to minimize the settling time to a step command while also making κ as small as possible. The strategy can summarized as a three-step process:

1) Select an initial weighting matrix $\tilde{Q}$.
2) Iterate over $\tilde{R}$ and N to determine the best trade-off between condition number and settling time.
3) Run fixed-point simulations to guide wordsize selection.

We now expound on each point.

(1) Initial $\tilde{Q}$ selection: One common method to choose $\tilde{Q}$ and $\tilde{R}$ is to set $\tilde{R} = 1$ and let $\tilde{Q} = \gamma \tilde{C}^T\tilde{C}$ for some $\gamma > 0$. As γ becomes large, this approach essentially incites pole-zero cancellation [31]. For our system, the poles and zeros are very lightly damped, are imperfectly known, and can move depending where we are on the stage [1], [25]. Thus, we do not consider this a viable approach.

Instead we first design feedback gains K_x and K_y which damp the resonant modes, increase the natural frequency of the most dominant modes, and cancel the low frequency real pole-zero pair (recall Fig. 1). Using these gains K_x and K_y, we then constrain $R_y = R_x = 1$ and solve an inverse optimal control problem [32]. That is, considering only the x-axis, solve for the decision variables $Q_x = Q_x^T \geq 0$, $P = P^T \geq 0$, and $P_1 = P_1^T > 0$, from

$$\begin{aligned} 0 =& (A_x - B_xK_x)^T P(A_x - B_xK_x) - P \\ &- K_x^T R_x K_x + Q_x \\ 0 =& B_x^T P A_x - (R_x + B_x^T P B_x)K_x \\ Q_x \geq& A_x^T P_1 A_x - P_1. \end{aligned}$$

We solve this problem using YALMIP [33]. 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. A solution is similarly obtained for the y direction.

For the complete MPC system, $\tilde{G}$, we let

$$\tilde{Q} = \text{diag}(Q_x, 0_{n_d}, Q_y, 0_{n_d}, 0, 0). \tag{8}$$

Note that the zero entries in this matrix are the states associated with the input, which we do not wish to penalize here. For a control horizon of $N = 24$ and $\tilde{R} = I$, this results in a condition number of H of nearly 4×10^6. We observe similar condition numbers if we use the same Q_x and Q_y for a system without delay and without the augmented state ξ_u such that $Q = \text{diag}(Q_x, Q_y)$, i.e., the zeros in (8) are not responsible for the poor conditioning.

(2) Iterate over $\tilde{R}$ and N: We then iterate over various values of the control weight $\tilde{R} = \gamma I$, and horizon N to explore the trade-off between stability, settling time, and the condition number κ of H. To do this, we simulate the complete, closed-loop system of Fig. 2 with floating-point arithmetic. We solve the quadratic program with Matlab's `quadprog`, since at this point we are interested in the best performance that can be expected from a given N and $\tilde{R}$, laying aside the convergence of the FGM. These results are compiled into Table I. From this table, we select $N = 12$, and $\tilde{R} = 5000I$. We note that, counter intuitively, increasing $\tilde{R}$ *decreases* the settle time up to a certain threshold.

(3) Select a fixed-point wordsize: Note that each row of Table I corresponds to a simulated, closed-loop trajectory. For a single time k on these trajectories, we can compare the effect of the wordsize used to represent the QP problem data and the maximum number of iterations $J_{\max}$ by considering the norm of the error between the $\Delta\mathcal{U}^*$ (taken to be the solution given by `quadprog`) and the solution $\Delta\hat{\mathcal{U}}_{J_{\max}}$ given by the fast gradient method for both floating point data and fixed point data using $J_{\max}$ iterations. Several results of this procedure are shown in Fig. 4, which also illustrates the importance of considering the condition κ of the Hessian H. At present, our hardware implementation can execute at

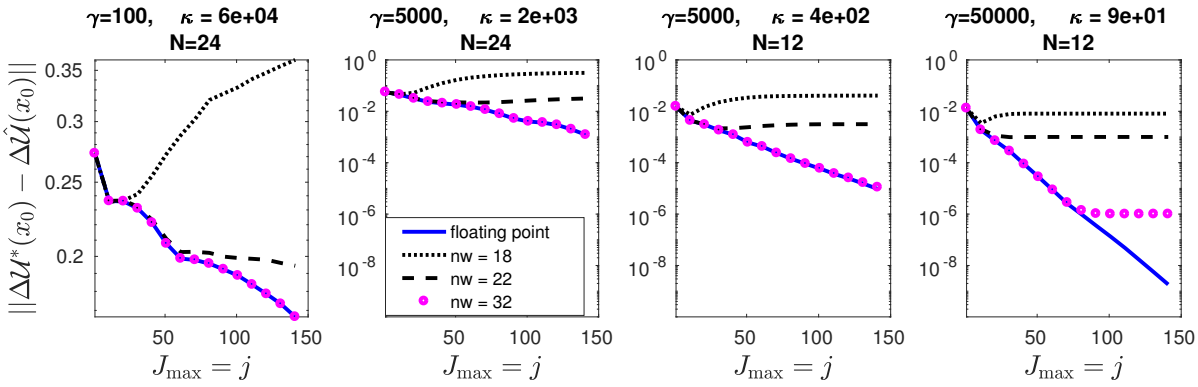


Fig. 4: This figure illustrates the effect of the relative size of $\tilde{R} = \gamma I$ and the condition number κ of H on the convergence rate of the FGM. The x-axis in each plot is the maximum number of FGM iterations, while the y-axis is the norm of the error between Matlab's `quadprog` and the FGM solution for a floating point implementation (solid blue) and fixed-point implementations with different word sizes, indicated by "nw" in the legend.

most $J_{\max} = 20$ iterations, though there is still significant room for optimization here. For $J_{\max} = 20$, the results in Fig. 4 show that little precision is gained by moving from 22 bits to 32 bits when $\gamma = 5000$ and $N = 12$. However, in terms of the number of DSP slices used, there is no difference in implementing operations between 19 and 32 bits. We have therefore chosen at this point to implement the algorithm with 32 bits since this decreases the brittleness of the implementation (e.g., when changing problem data).

			settle time [ms]		
N	R_{11}	R_{22}	x-dir	y-dir	κ
12	1.0	1.0	–	–	102943
12	100.0	100.0	–	–	4314
12	1000.0	1000.0	4.68	–	1030
12	2500.0	2500.0	4.64	5.76	585
12	5000.0	5000.0	4.68	5.08	381
12	10000.0	10000.0	4.80	5.32	248
18	1.0	1.0	–	–	1035201
18	100.0	100.0	4.76	–	25604
18	1000.0	1000.0	4.64	5.88	4344
18	2500.0	2500.0	4.64	5.76	2131
18	5000.0	5000.0	4.68	5.08	1238
18	10000.0	10000.0	4.80	5.32	717
24	1.0	1.0	4.84	–	3995883
24	100.0	100.0	4.76	–	64371
24	1000.0	1000.0	4.64	6.08	8213
24	2500.0	2500.0	4.64	5.80	3584
24	5000.0	5000.0	4.68	5.08	1909
24	10000.0	10000.0	4.80	5.32	1018

TABLE I: Settling times and condition numbers κ for different choices of $\tilde{R}$ and N. We use a 1% settling-time criterion and all times are in milliseconds. These results are obtained using Matlab's `quadprog` with full precision arithmetic.

A. State Scaling

Although the Xilinx LX150 FPGA is capable of performing floating-point math operations, fixed-point arithmetic consumes fewer hardware resources and has lower latency. Of course, one of the hurdles with fixed-point implementations is its limited dynamic range. During a unit step command to the state-space representation G_p, the state with the largest absolute value is on the order of 65000 while the smallest is on the order of 1. We therefore scale the system states so that the magnitude of every state is comparable by applying a state transformation $z = T\tilde{\xi}$, where we take T as a diagonal matrix such that $T_{jj} = 1/\tilde{\xi}_{ss,j}$. Although more complex strategies are possible [34], our method has so far proved sufficient and simplifies many implementation details of the control law.

VI. Experimental Results

We will first consider comparing the constrained MPC, where the control action is taken from (6) to that of the linear feedback law (7), where constraints are not explicitly accounted for but where the result is simply saturated at $\pm\Delta u_{\max}$. In the case of this linear feedback law, we use the same block diagram structure of Fig. 2, except the MPC block is replaced with $-K$ and a saturator. For the MPC controller, the FGM solver uses a 32 bit word size for the QP data and $J_{\max} = 20$ solver iterations. All simulation results were done in Simulink using fixed-point arithmetic and a FGM implementation for the constrained MPC case. Other simulations using floating point math and `quadprog` to solve the QP are indistinguishable and so are not shown.

Fig. 5 shows experimental results only of the input and output for both axes for a 5.0 μm step command. Here, the setpoint is small enough that the constraints are inactive and the two controllers perform comparably as expected. No simulated trajectories are shown since they do not differ significantly at the scale of the figure. On the other hand, Fig. 6 shows both experimental and simulated trajectories where the setpoint is increased to 7.465 μm (only the y-axis is shown since it best illustrates the point). This larger setpoint activates the constraints. In simulation, the quality of the linear feedback response (dashed-magenta) has decayed considerably, while the experimental response (dotted-blue) becomes unstable and the experiment is terminated early. However, the trajectory under the constrained MPC

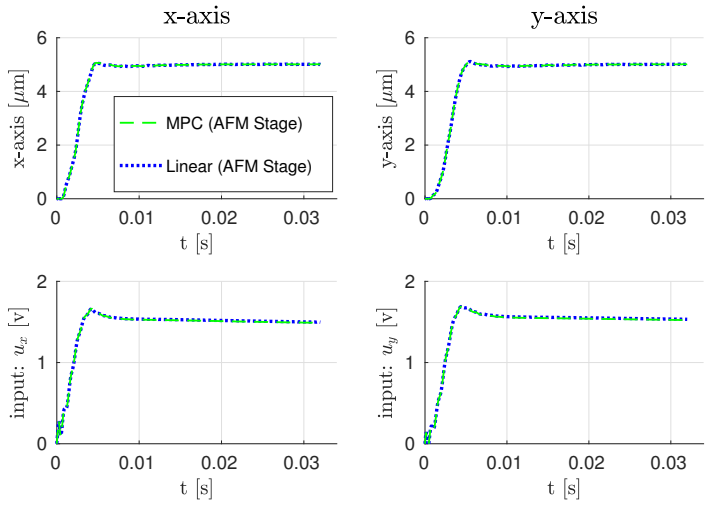


Fig. 5: Step response of both axes to a 5 μm step command which is small enough that the constraints are inactive and both controllers perform comparably. Simulated trajectories are not shown since they are not significantly different at the scale of the figure.

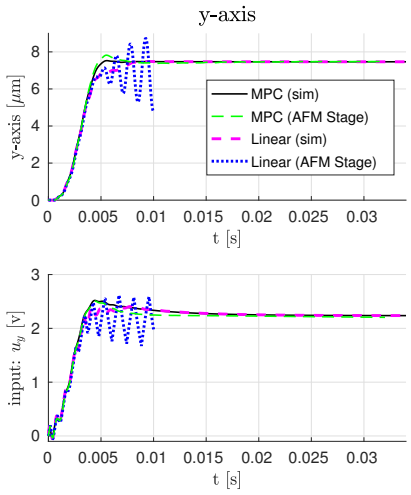


Fig. 6: Step response of the y axis when both axes are subjected to a 7.465 μm step command. The simulated response has decayed considerably and the experimental response is unstable.

control law still performs well for this size step command in both simulation (solid-black) and experiment (dashed-green), illustrating the advantage of MPC's natural constraint handling.

Indeed, we can push the stage farther using MPC, and Fig. 7 shows the response to a step command of 8.0 μm. In that figure, the middle row of plots is a zoomed in version of the top row, to better show the steady-state behavior. The black dotted line indicates the 1% settle boundary. Note that no linear controller trajectory is shown here because this size of setpoint results in instability in both simulation and experiment.

VII. Conclusions and Future Directions

In this paper, we have demonstrated an application of the FGM to a two-axis piezo stage using a sample frequency of 25 kHz. The controller has a wider maneuvering range while maintaining stability than a comparable linear feedback controller, highlighting the benefit of MPC's constraint handling capability.

Future work will look for ways to achieve better steady-state behavior. The slow convergence to the setpoint seen in Fig. 7 is likely an artifact of using a *constant* disturbance estimator to reject non-constant drift. Indeed, preliminary simulations using a higher-order drift model show this same type of behavior in simulation as well. We will therefore consider compensating drift with either a more comprehensive disturbance estimate or via inverting a drift model [35].

Alternative QP formulations could also be explored. For instance, [36] develops a sparse and condensed formulation that could lead to better numeric conditioning. It remains to be seen however if that formulation will admit a solution at a high enough sample rate.

Finally, future work will build on the methods developed here to integrate z-axis control to acquire AFM images and will also consider using the current control architecture with a MIMO model which accounts for cross-axis coupling.

References

[1] 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. Control Applications*, Sep. 2011, pp. 1081–1087.

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

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

[4] 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. Control Systems Tech.*, vol. 15, no. 5, pp. 906–915, Sep. 2007.

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

[6] A. Chen, A. L. Bertozzi, P. D. Ashby, P. Getreuer, and Y. Lou, "Enhancement and recovery in atomic force microscopy images," in *Excursions in Harmonic Analysis, Vol. 2*. Boston: Birkhäuser Boston, Nov. 2012, pp. 311–332.

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

[8] I. A. Mahmood and S. O. R. Moheimani, "Fast spiral-scan atomic force microscopy," *Nanotech.*, vol. 20, no. 36, p. 365503, 2009.

[9] 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.

[10] B. D. Maxwell and S. B. Andersson, "A compressed sensing measurement matrix for atomic force microscopy," in *Proc. American Control Conf.*, June 2014, pp. 1631–1636.

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

[12] M. L. Workman, R. L. Kosut, and G. F. Franklin, "Adaptive proximate time-optimal servomechanisms: Continuous time case," in *Proc. American Control Conf.*, June 1987, pp. 589–594.

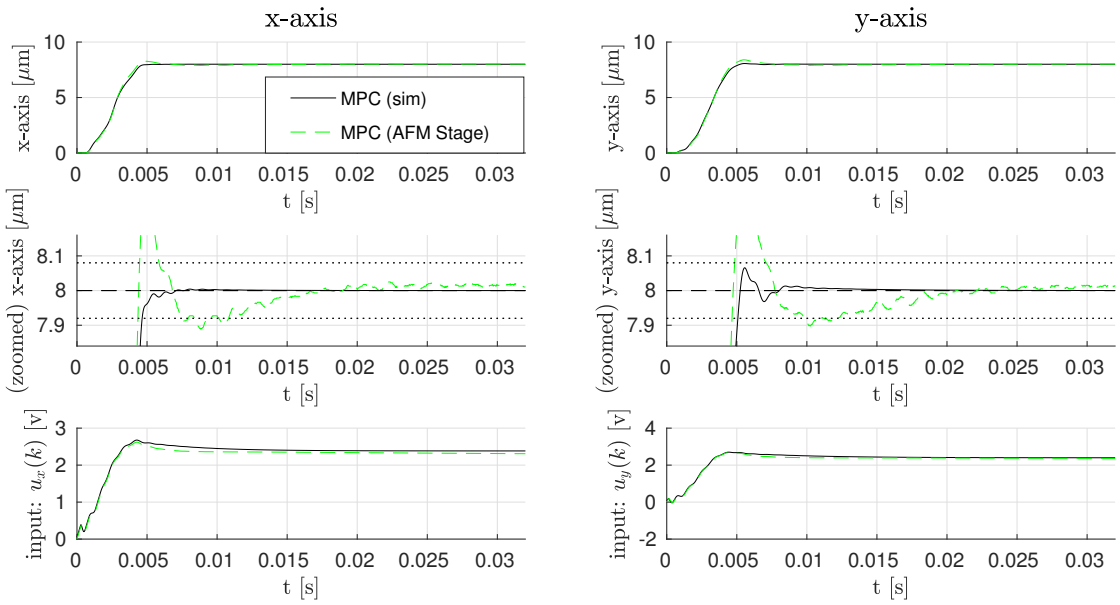


Fig. 7: Comparison of simulation and experimental results using the constrained MPC controller. The solid black curve is a simulated trajectory solved using 20 iterations of the FGM with a fixed-point word size of 32 bits. The dashed-green curve is the experimental response of the AFM stage, where the QP is also solved with the FGM using the same parameters. The dotted-black lines in the middle two plots represent the 1% settle boundary. An equivalent floating-point simulation using Matlab's `quadprog` is indistinguishable from the black curve and is therefore not shown.

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

[14] R. A. Braker and L. Y. Pao, "Proximate time-optimal reference tracking of an undamped harmonic oscillator," in *Proc. American Control Conf.*, July 2016, pp. 6221–6226.

[15] ——, "Fast setpoint tracking of an atomic force microscope x-y stage via optimal trajectory tracking," in *Proc. American Control Conf.*, 2017, to appear, pre-print available at http://ecee.colorado.edu/~pao/ conferences.

[16] K. Muske and J. Rawlings, "Model predictive control with linear models," *AIChE J.*, vol. 39, no. 2, pp. 262–287, 1993.

[17] H. J. Ferreau, H. G. Bock, and M. Diehl, "An online active set strategy to overcome the limitations of explicit MPC," *Int. J. Robust and Nonlinear Control*, vol. 18, no. 8, pp. 816–830, 2008.

[18] J. L. Jerez, G. A. Constantinides, and E. C. Kerrigan, "An FPGA implementation of a sparse quadratic programming solver for constrained predictive control," in *Proc. ACM/SIGDA Int. Symp. Field Programmable Gate Arrays*, 2011, pp. 209–218.

[19] S. Richter, C. N. Jones, and M. Morari, "Computational complexity certification for real-time MPC with input constraints based on the fast gradient method," *IEEE Trans. Automatic Control*, vol. 57, no. 6, pp. 1391–1403, June 2012.

[20] M. A. Bochat, J. Liu, H. Peyrl, A. Zanarini, and T. Besselmann, "An architecture for solving quadratic programs with the fast gradient method on a field programmable gate array," in *Proc. Mediterranean Conf. Control and Automation*, June 2013, pp. 1557–1562.

[21] J. L. Jerez, P. J. Goulart, S. Richter, G. A. Constantinides, E. C. Kerrigan, and M. Morari, "Embedded online optimization for model predictive control at megahertz rates," *IEEE Trans. Automatic Control*, vol. 59, no. 12, pp. 3238–3251, Dec. 2014.

[22] M. S. Rana, H. R. Pota, and I. R. Petersen, "High performance control of atomic force microscope for high-speed image scanning," in *Proc. Int. Conf. Control Automation Robotics Vision*, Dec. 2012, pp. 1187–1192.

[23] A. Wills, D. Bates, A. Fleming, B. Ninness, and R. Moheimani, "Application of MPC to an active structure using sampling rates up to 25kHz," in *Proc. IEEE Conf. Decision and Control*, Dec. 2005, pp. 3176–3181.

[24] C. Y. Lin and Y. C. Liu, "Precision tracking control and constraint handling of mechatronic servo systems using model predictive control," *IEEE/ASME Trans. Mechatronics*, vol. 17, no. 4, pp. 593–605, Aug. 2012.

[25] 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.

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

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

[28] K. R. Muske and T. A. Badgwell, "Disturbance modeling for offset-free linear model predictive control," *J. Process Control*, vol. 12, no. 5, pp. 617 – 632, 2002.

[29] G. Pannocchia and J. B. Rawlings, "Disturbance models for offset-free model-predictive control," *AIChE J.*, vol. 49, no. 2, pp. 426–437, 2003.

[30] U. Maeder and M. Morari, "Offset-free reference tracking with model predictive control," *Automatica*, vol. 46, no. 9, pp. 1469 – 1476, 2010.

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

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

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

[34] M. Steinbuch, G. Schootstra, and H.-T. Goh, "Closed-loop scaling in fixed-point digital control," *IEEE Trans. Control Systems Tech.*, vol. 2, no. 4, pp. 312–317, Dec. 1994.

[35] K. K. Leang, Q. Zou, and S. Devasia, "Feedforward control of piezoactuators in atomic force microscope systems," *IEEE Control Systems Magazine*, vol. 29, no. 1, pp. 70–82, Feb. 2009.

[36] J. L. Jerez, E. C. Kerrigan, and G. A. Constantinides, "A condensed and sparse QP formulation for predictive control," in *Proc. IEEE Conf. Decision and Control and European Control Conf.*, Dec. 2011, pp. 5217–5222.