Title

Adaptive and Iterative Learning Control for Robot Trajectory Tracking

Author

LEE, YU-HSIU

Publication Date

2019

UNIVERSITY OF CALIFORNIA

Los Angeles

Adaptive and Iterative Learning Control for Robot Trajectory Tracking

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Mechanical and Aerospace Engineering

by

Yu-Hsiu Lee

2019

ABSTRACT OF THE DISSERTATION

Adaptive and Iterative Learning Control for Robot Trajectory Tracking

by

Yu-Hsiu Lee
Doctor of Philosophy in Mechanical and Aerospace Engineering
University of California, Los Angeles, 2019
Professor Tsu-Chin Tsao, Chair

This thesis develops adaptive and iterative learning control methods for robot trajectory tracking applications. Specifically, iterative learning control is applied when a desired reference trajectory is known beforehand, and adaptive control is designed to cope with unknown patient motion (disturbance) in Magnetic Resonance Imaging (MRI)-guided robot-assisted intervention.

For robot manipulator tracking, a nested-loop Iterative Learning Control (ILC) is proposed. This method requires only nominal kinematic parameters from factory setting, gives fast convergence, and can be added on top of existing servo loop. The ILC learning architecture includes an inner loop that accounts for motor dynamics, and an outer loop that addresses the static bias from the payload or imprecise kinematics. A data-based learning filter design is extended to cope with motion constraint and multivariate systems. It is experimentally verified on a 6-DOF serial robot that the proposed method mitigates the maximum dynamic tracking error by an order of magnitude, and is applicable to different payloads due to small system variation from torque shielding of gear reduction.

For tracking of general nonlinear dynamic systems, an efficient data-driven ILC algorithm is proposed. As opposed to the model-based methods, for which nonlinear identification and learning law design can be cumbersome, this method uses adaptive filter to implicitly (and automatically) construct linearized system inverse for effective learning. An existing adjoint-based ILC for LTI system is also extended to cope with nonlinear dynamics, and for comparative study. The SISO algorithms are simulated and experimentally validated on a fully-actuated 2-DOF laboratory pendulum system. Algorithms are also developed to circumvent the difficulty when adapting a right inverse for MIMO systems.

The automated MRI-guided intervention is motivated by the current procedural inefficiency from constraints posed by MR environment. As lots of researchers focus on either MR-safe/conditional robot to augment the reach of the physician, or MR image tracking for motion estimation of tissue/instrument, this work aims at addressing a more flexible setting: use real-time MRI for instrument control when a target is in motion. It is enabled via the integration of robot hardware, MRI sensing, and control techniques. On the control aspect, we characterize the MR imaging process and the robot dynamics, then propose adaptive control schemes to overcome the system delay and high noise variance from MRI measurement. The study is conducted on a hydrostatically actuated platform, which consists of a target motion module that emulates respiratory motion, and an instrument manipulation module regulating the instrument-target distance.

The dissertation of Yu-Hsiu Lee is approved.

James S Gibson

Lieven Vandenberghe

Holden H Wu

Tsu-Chin Tsao, Committee Chair

University of California, Los Angeles

2019

To my parents

who support and believe me no matter what.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to acknowledge my advisor, Prof. Tsu-Chin Tsao, who has infinite patience for guiding students and infinite passion in doing research. His unique mentoring style: giving just the right amount of nudge without explicitly telling the answer, helps a lot to develop the logical reasoning skills and creativity for an independent researcher. I'd also like to thank all my committee members, Prof. Steve Gibson, Prof. Lieven Vandenberghe, and Prof. Holden Wu for their acute insights and helpful comments.

In this thesis, the nested-loop ILC for robot manipulators [LHD19] was the first project I took lead on. At the time Prof. Tsao had this concept about doing several inverse identification when system varies, and this later inspired the idea of the data-driven ILC using adaptive filtering [LT19]. The MR robotics project was inherited from James Simonelli, who laid significant groundwork for the realization of MR-compatible research platform [SLM17, SLCed]. None of the work is possible without the efforts and inspiration from all the co-authors.

To Cheng-Wei, who I had issues with at the beginning but truly appreciate in the end. You were a strict mentor but fair enough to give me both the harsh truth and the deserved compliments. To James, who taught me all the valuable technical skills ranging from machining to programming. To Sandeep, who always willingly shares his knowledge and research ideas. To Chris and Clyde, who looked out for me when I was still green. To Matt, thank you for teaching me directly or indirectly about academic presentation, technical communication, and research mentalities/methodologies. To Xinzhou, thank you for making the MR feedback control possible. Thanks to the members in Mechatronics and Control Laboratory: Jack, Han, Shahin, Paul, Yao, Aziz, Edwin, and Omar.

To my dearest friends in Taiwan, who always remember to stop by LA and say hi. To Sarah, for the memorable graduation trip and your warm regards in the past two years. Lastly, I would like to thank my family, especially my parents, for seeing the best in me and bare with my willfulness.

Vita

2007	B.S. (Mechanical Engineering), National Taiwan University, Taipei, Taiwan.
2009	M.S. (Electrical Engineering), National Taiwan University, Taipei, Taiwan.
2010-2011	Teaching Assistant (Mechanical Engineering), National Taiwan University, Taipei, Taiwan.
2011-2012	Mechanical Designer, Jabil Design Services, Hsinchu, Taiwan.
2012-2014	Mechanical Designer (Health Care Products), Qisda Corporation, Taoyuan, Taiwan.

Publications

Lee, Y. H. and Tsao, T. C. (2019). Data-driven ILC for Trajectory Tracking in Nonlinear Dynamic Systems. Proceedings of Proceedings of the ASME 2019 & Dynamic Systems and Control Conference (DSCC 2019). Oct. 9-11, 2019, Park City, UT, USA.

Lee, Y. H. and Hsu, S. C. and Du, Y. Y. and Hu, J. C. and Tsao, T. C. (2019). A Nested-Loop Iterative Learning Control for Robot Manipulators. Joint Conference 8th IFAC Symposium on Mechatronic Systems (MECHATRONICS 2019), and 11th IFAC Symposium on Nonlinear Control Systems (NOLCOS 2019). Vienna, Austria, Sept. 4-6, 2019: Elsevler.

S., Simonelli, J., Lee, Y. H., Chen, C. W., Li, X., ..., and Tsao, T. C. (2019). Hydrostatic

Actuation for Remote Operations in MR Environment. IEEE transactions on mechatronics. Accepted.

Li, X., Lee, Y. H., Mikaiel, S., Simonelli, J., Tsao, T. C., and Wu, H. H. (2019). Respiratory Motion Prediction using Fusion-Based Multi-Rate Kalman Filtering and Real-Time Golden-Angle Radial MRI. IEEE transactions on bio-medical engineering.

Mikaiel, S., Simonelli, J., Li, X., Lee, Y. H., ..., and Wu, H. H. (2019). MRI-Guided Targeted Needle Placement During Motion Using Hydrostatic Actuators. The International Journal of Medical Robotics and Computer Assisted Surgery, Accepted.

Prince, S. W., Kang, C., Simonelli, J., Lee, Y. H., Gerber, M. J., Lim, C., ..., and Tsao, T. C. (2019). A Robotic System for Telementoring and Training in Laparoscopic Surgery. The International Journal of Medical Robotics and Computer Assisted Surgery.

Chen, C. W., Francone, A. A., Gerber, M. J., Lee, Y. H., Govetto, A., Tsao, T. C., and Hubschman, J. P. (2019). A Study on Semi-Automated OCT-Guided Robotic Surgery for Porcine Lens Removal. Journal of Cataract & Refractive Surgery.

Chen, C. W., Lee, Y. H., Gerber, M. J., Cheng, H., Yang, Y. C., Govetto, A., ..., and Tsao, T. C. (2018). Intraocular robotic interventional surgical system (IRISS): Semi-automated OCT-guided cataract removal. The International Journal of Medical Robotics and Computer Assisted Surgery, 14(6), e1949.

Simonelli, J., Lee, Y. H., Mikaiel, S., Chen, C. W., Li, X., Sung, K., ..., and Tsao, T. C. (2017). An MR-compatible stage for respiratory motion emulation. IFAC-PapersOnLine, 50(1), 6073-6078.

CHAPTER 1

Introduction

Robot tracking is a relevant application in various manufacturing processes, such as part assembly, welding, polishing, laser cutting, and so forth. As robots are known for their manufacturing flexibility, there are also factors that could affect the tracking performance, and thus finished workpiece quality. For example, the calibration accuracy, joint deflection, and robot dynamics.

In Chapter 2, we consider the trajectory tracking of robot manipulator with geared motors. Under the assumption that the actuator dynamics are decoupled and linear time-invariant due to gear shielding, we develop a nested-loop ILC wherein the inner loop (using motor encoder feedback) is for motor dynamic compensation, and the outer loop (using laser tracker feedback) accounts for static bias due to calibration and joint deflection. For inner loop, a data-based dynamic inversion method is proposed, featuring in simple implementation, and capabilities to incorporate motion constraints.

As real-world systems are often nonlinear and time/parameter-varying, the nonlinear model identification, learning design, and convergence analysis can be a tedious task. Therefore, in Chapter 3 we develop a novel data-driven ILC utilizing adaptive filtering. This method addresses the nonlinearity via linearization around the trajectory, uses adaptive filter to automatically construct learning filter, and gives fast convergence by implicitly doing system inverse identification. As a comparative study, a state-of-the-art data-driven ILC using adjoint system is also extended for nonlinear dynamic systems.

Aside from the robot tracking where a desired trajectory is known beforehand, we will also investigate image-guided robot-assisted intervention, specifically MRI-guided intervention. MRI-guided intervention is of great interest for several reasons: first, some abnormalities only appear on MRI; secondly, it provides excellent soft tissue contrast; thirdly, it is not subject to ionized radiation exposure as X-ray and thus can be run continuously. Conventionally, physician needs to move patient in and out of the scanner repeatedly to gain patient access. This procedure requires several confirmation scans for incremental instrument advancement, and it relies on breath holding of the patient to keep the target tissue still. Clearly the real-time capability of MRI is not fully utilized, the image guidance is sub-optimal, and the procedure can be time-consuming. In the past, most research groups either focus on augmenting physician's reach by MR-safe/conditional robots, or developing tissue/instrument tracking algorithms for real-time diagnosis. Recently, some groups try to perform closed-loop MRI-guided intervention for stationary target. However, a more flexible setting will be conducting closed-loop intervention when the target is in motion.

In Chapter 4, we study the MRI-guided intervention with robotic system under dynamic target setting. This is carried out on an MR-compatible hydrostatically actuated platform, which consists of a target motion module that emulates human respiratory motion, and an instrument manipulation module that controls the biopsy needle motion. Via modelling of MRI and actuator dynamics, adaptive control is applied to address the unknown disturbance (patient motion), image processing delay, and noisy MRI feedback. This is then experimentally verified in MR environment with the hydrostatically actuated platform, showing great promise towards fully-automated MRI-guided intervention.

Finally, in Chapter 5 we summarize the work presented, and list ongoing tasks and possible future directions.

CHAPTER 2

A Nested-Loop Iterative Learning Control for Robot Manipulators

2.1 Trajectory Tracking of Robot Manipulators

Trajectory tracking of robot manipulators is of interest in industrial processes where high positional accuracy is essential. Applications include welding, laser cutting, part assembly, and glue dispensing. Most robot manipulators employ geared motors because they are cost effective, light-weight, and have small system variation under torque disturbance. Nonetheless, this is at the expense of additional compliance and nonlinearity introduced by the gear transmission. The end effector tracking performance is, therefore, subject to intrinsic and extrinsic factors. Intrinsic factors include the accuracy of kinematic parameters, joint flexibility, drive dynamics, and non-smoothness such as joint friction and gear backlash. Process disturbances serve as an extrinsic error source.

Due to the repeated nature of these industrial processes, ILC for robot manipulators has been investigated extensively for tracking application under repeated disturbances, see [Tay04]. ILC is also an effective tool for parameter identification, for example, the work by [ZLX15] suggests that ILC can also be used for kinematic parameter refinement along the tracked path to enhance the positioning accuracy. The friction effect on the actuator, if repeatable, can also be compensated by ILC. To address the joint flexibility, [MKM86] first proposed the dual-stage learning on motor side and subsequently load side to give better end effector tracking. Recently, [CT14] presents a simultaneous dual-stage learning of both sides that result in more effective learning. [WZW18] formulates the simultaneous dual-

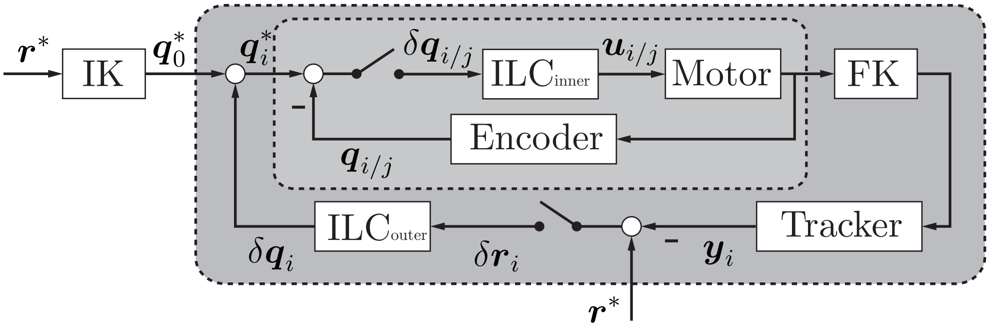


Figure 2.1: Nested-loop learning structure.

stage learning as a two-degree-of-freedom robust controller design under system uncertainty to achieve robust performance. These are classified as model-based approaches wherein extensive system identification is needed to proceed with the learning filter design.

2.2 Nested-loop Learning Architecture

In this work, we adopt the nested-loop learning concept of [MKM86], and develop a learning structure that requires only nominal kinematic parameters and some simple system responses. This approach assumes each drive unit is an independent linear single-input-single-output (SISO) system because the sensed torque disturbance is attenuated by inverse squared gear ratio, see [SHV06]. Another assumption is that as joint tracking error converges, the end effector error is mostly attributed to static bias, such as imprecise kinematic parameters, and deflection due to gravity, provided care is taken to avoid flexible mode excitation. This structure is depicted in Fig. 2.1, wherein $\boldsymbol{r}^*$ is the desired trajectory in task domain, $\boldsymbol{q}_0^*$ is the joint reference generated by nominal inverse kinematics IK. The subscript i and j denote outer and inner loop iteration indices, respectively. Note this is a multi-rate system: at i^{th} outer loop, the inner loop iterates, $(j = 1 \cdots J)$ trying to invert drive dynamics and make joint encoder values $\boldsymbol{q}_{i/j}$ follow a reference $\boldsymbol{q}_i^*$. Upon convergence of the inner loop, the end effector position error $\delta\boldsymbol{r}_i$ computed from laser tracker measurement $\boldsymbol{y}_i$ is used to generate path correction $\delta\boldsymbol{q}_i$ for the next $i + 1^{\text{th}}$ outer loop iteration. Variables $\delta\boldsymbol{q}_i = \delta\boldsymbol{q}_i(k)$ and $\delta\boldsymbol{r}_i = \delta\boldsymbol{r}_i(k)$ are time-dependent but the time index k is dropped for simplicity.

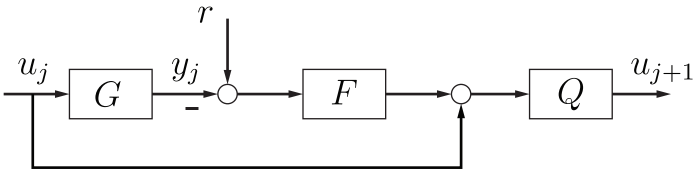


Figure 2.2: A generic ILC structure.

The proposed scheme uses a data-based inversion technique for inner loop learning filter design, which allows engineers to incorporate motion constraints, such as velocity, acceleration, and jerk limits of the system. The resultant filter in finite impulse response (FIR) representation provides good approximation of inversion, fast ILC convergence, and good joint tracking performance. The outer loop learning matrix is simply inverse Jacobian, and does not require additional identification. This scheme is command-based, which means it alters only the joint commands sent to an existing position servo loop, making it readily applicable without additional signal access.

2.3 Inner Loop: Data-based Dynamic Inversion

2.3.1 Inversion-based ILC

Consider a pre-stabilized plant G. The generic ILC update law in Fig. 3.1 can be written as follows:

$$\begin{aligned} u_{j+1} &= Q(z^{-1})\left[u_j + F(z^{-1})e_j\right] \\ &= Q\left[u_j + F(r - y_j)\right] \\ &= Q(1 - FG)u_j + QFr \end{aligned} \tag{2.1}$$

wherein r is the reference trajectory to be tracked, u_j is the the control sequence, j is the iteration index, y_j is the the output sequence, e_j is the tracking error, F is the learning filter, and Q is a zero-phase low-pass filter for robustness.

The recursion formula in Eqn. 2.1 is asymptotically stable if the convergence rate

$$\gamma \triangleq \|Q(1 - FG)\|_\infty < 1, \tag{2.2}$$

and the converged error e_∞ in z-domain can be written as:

$$e_\infty = \frac{1 - Q(z^{-1})}{1 - Q(z^{-1})[1 - F(z^{-1})G(z^{-1})]} r. \tag{2.3}$$

See [NG02] for time and frequency domain stability criteria.

It is evident from Eqn. 2.2 and 2.3 that using system inversion as the learning filter, i.e., $F = \hat{G}^{-1}$, is advantageous: it results in zero convergence rate ($\gamma = 0$), namely the learning process reaches steady state within one iteration. Also, within the bandwidth of Q ($Q \approx 1$), this gives zero tracking error ($e_\infty = 0$).

Recent studies in comparing different inversion techniques employed in ILC can be found in [TT15] and [ZO18]. They can be roughly categorized as 1) approximate inverse schemes, e.g. ZPETC and ZMETC, 2) norm-optimal formulation, 3) optimal controller synthesis, e.g. $\mathcal{H}_2$ and $\mathcal{H}_\infty$, and 4) data-based inversion, see [TT15].

The approximate inverse schemes propose ad-hoc methods to deal with nonminimum-phase zeros, but caution needs to be taken for nonminimum-phase zero locations and high-frequency amplification, see [BPA12]. The norm-optimal formulation minimizes a quadratic cost over a finite horizon N where N is the number of reference points. It is often described in the lifted notations and the computational complexity is $\mathcal{O}(N^3)$. In the re-formulation proposed by [ZBK16], it can be regarded as a classic LQ tracking problem so that Riccati recursion is made use of. The norm-optimal problem can then be solved efficiently with $\mathcal{O}(N)$ complexity. The optimal controller synthesis approach solves for the inversion by the well-developed state-space solutions to $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems.

As opposed to the model-based approaches listed above, data-based inversion techniques feature in finding the inversion without any modeling. This usually provides satisfactory inversion quality if the input data sufficiently excites system dynamics, and the output data has high signal-to-noise ratio (SNR).

2.3.2 Data-Based Inversion with Motion Constraints

In this work, we propose a method for dynamic inversion based on the technique developed by [TT15] for several considerations. First, this approach suits for robot manipulators, which have high SNR. Second, actuator limits and model uncertainty can be addressed using frequency domain weighting. Thirdly, this method uses a FIR representation of system dynamics directly from data and does not have model fitting error. Based on these, we generalize the approach by the introduction of identification input design for which motion constraints, such as joint limit and slew rate that are commonly seen in industrial servo drives, can be taken into account.

The data-based inversion method by [TT15] is briefly summarized as follows, and is visualized in Fig. 2.3:

1. Construct plant $G(z^{-1}) = \mathcal{Z}\{g[n]\} \approx \sum_{n=1}^{N_{fir}} g[n]z^{-n}$ with FIR representation by one-step subtraction of step response data.

2. Convolve $g[n]$ with a smoothing window function $w[n]$ and then pad the convolved sequence $g_w[n]$ with sufficient zeros.

3. Compute discrete Fourier transforms (DFT) of $M(e^{-j\omega_n})$ and $G_w(e^{-j\omega_n})$ wherein $M(z^{-1})$ is a desired zero-phase reference model. Then, apply element-wise frequency domain division to construct inversion $F(e^{-j\omega_n}) = M(e^{-j\omega_n}) \oslash G_w(e^{-j\omega_n})$.

4. Perform inverse DFT to get $f[n] = \mathcal{F}^{-1}\{F(e^{-j\omega_n})\}$. The inversion in FIR form is then $F(z^{-1}) = \mathcal{Z}\{f[n]\} = \sum_{n=1}^{N_{fir}} f[n]z^{-n}$.

The generalization of this method is first by observing that although the identification procedure was developed for positional step response, namely a velocity impulse, higher order step response data apply because $G = \frac{Y}{U} = \frac{Y(1-z^{-1})^l}{U(1-z^{-1})^l}$, wherein l is the order of finite difference(s). Also, by adjusting the amplitude of the command step size, higher order motion constraints can be employed because numerical derivative is always finite with sampling time T_s in the denominator.

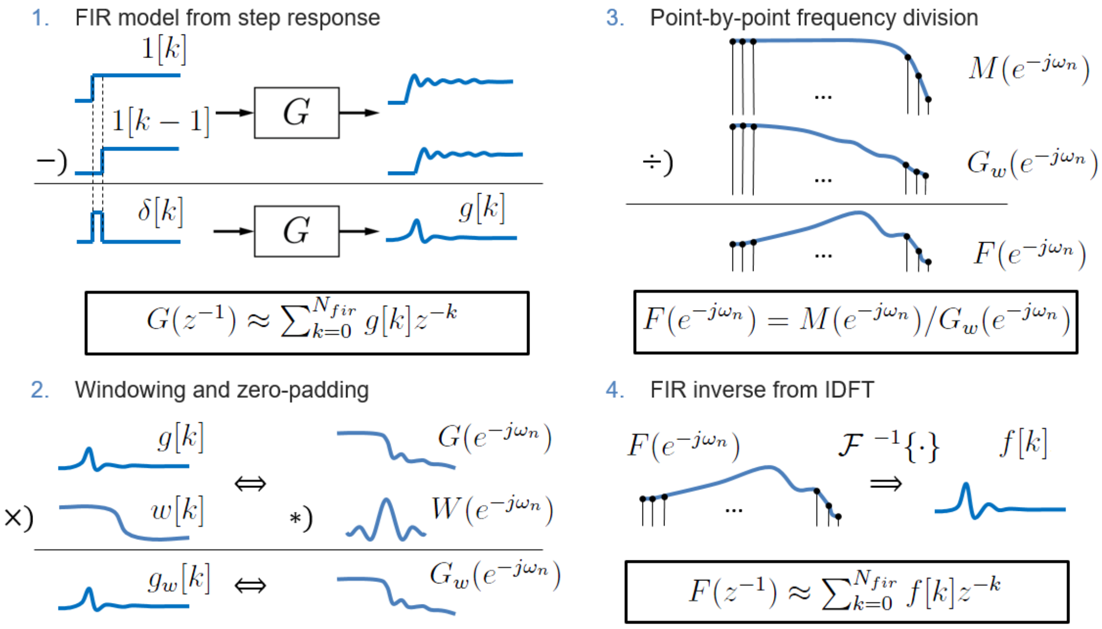

1. FIR model from step response
$1[k]$
$1[k-1]$
G
−)
$\delta[k]$
G
$g[k]$
$G(z^{-1}) \approx \sum_{k=0}^{N_{fir}} g[k]z^{-k}$
2. Windowing and zero-padding
$g[k]$
$G(e^{-j\omega_n})$
⇔
×)
$w[k]$
∗)
$W(e^{-j\omega_n})$
$g_w[k]$
⇔
$G_w(e^{-j\omega_n})$
3. Point-by-point frequency division
$M(e^{-j\omega_n})$
...
÷)
$G_w(e^{-j\omega_n})$
...
$F(e^{-j\omega_n})$
...
$F(e^{-j\omega_n}) = M(e^{-j\omega_n})/G_w(e^{-j\omega_n})$
4. FIR inverse from IDFT
$F(e^{-j\omega_n})$
$\mathcal{F}^{-1}\{\cdot\}$
$f[k]$
⇒
...
$F(z^{-1}) \approx \sum_{k=0}^{N_{fir}} f[k]z^{-k}$

Figure 2.3: Procedure for data-based inversion construction.

In practice, the identification input can be heuristically designed as the follows. Assume motion constraints, sensor noise statistics, and desired SNR are given. We start with a position impulse whose amplitude is determined by the prescribed SNR, and then check if all the motion constraints are met. If so, the amplitude of the impulse can be increased until one of the constraints is activated, otherwise a pulse of next derivative (velocity) is used instead, and the amplitude is adjusted in a similar fashion. Note using impulse of higher derivatives will relax lower-order motion constraints, but at the expense of noise amplification. This process repeats until a feasible impulsive reference is generated, or a compromise on SNR needs to be made.

2.3.3 Extension to MIMO Systems

The point-by-point frequency domain division can also be extended to the multivariate case. Because the transfer function matrices in general do not commute, the choice of inverse, either left or right, makes difference in ILC performance. From the recursion of tracking error:

$$\begin{aligned} e_{j+1} &= r - Gu_{j+1} \\ &= r - G(u_j + Fe_j) \\ &= (I - GF)e_j, \end{aligned} \tag{2.4}$$

it is clear that we should use the system right inverse because it minimizes $(I - GF)$. If a reference model M is used instead of identity, the frequency response of F at frequency grid point ω_k can be solved via:

$$\left[G(e^{-j\omega_k})\right]\left[F(e^{-j\omega_k})\right] = \left[M(e^{-j\omega_k})\right]. \tag{2.5}$$

For a two-input-two-output system the expanded form (argument $e^{-j\omega_k}$ omitted) is:

$$\begin{bmatrix} G_{11} & G_{12} & 0 & 0 \\ G_{21} & G_{22} & 0 & 0 \\ 0 & 0 & G_{11} & G_{12} \\ 0 & 0 & G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{21} \\ F_{12} \\ F_{22} \end{bmatrix} = \begin{bmatrix} M_1 \\ 0 \\ 0 \\ M_2 \end{bmatrix}. \tag{2.6}$$

Note the resulting system of linear equations is block diagonal, so the complex matrix $F(e^{-j\omega_k})$ can be obtained by solving 2 separate 2-by-2 sub-problems in parallel. Similarly, for an m-input-m-output square system, this results in solving m decoupled m-by-m linear systems, and each sub-problems produces a column of $F(e^{-j\omega_k})$:

$$\begin{bmatrix} G_{11} & \cdots & G_{m1} & & & \\ \vdots & \ddots & \vdots & & & \\ G_{1m} & \cdots & G_{mm} & & & \\ & & & G_{11} & \cdots & G_{m1} \\ & & & \vdots & \ddots & \vdots \\ & & & G_{1m} & \cdots & G_{mm} \\ & & & & & \end{bmatrix} \begin{bmatrix} F_{11} \\ \vdots \\ F_{m1} \\ F_{12} \\ \vdots \\ F_{m2} \\ \end{bmatrix} = \begin{bmatrix} M_1 \\ \vdots \\ 0 \\ 0 \\ M_2 \\ \vdots \\ \end{bmatrix}. \tag{2.7}$$

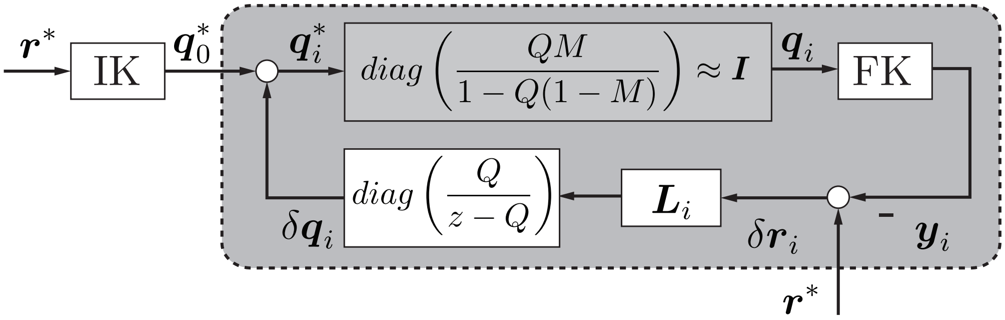


Figure 2.4: Outer-loop learning structure with a converged inner loop.

Because the independence of sub-problems, parallelization is possible and the algorithm is computationally tractable. If the system is non-square with n_i inputs and n_o outputs, then we can solve for least square solution if G is tall ($n_o > n_i$) and full column rank, or least norm solution if G is fat ($n_o < n_i$) and full row rank. After F_{ij} at each ω_k is collected, inverse Fourier transform is applied and a FIR representation $\sum f_{ij}[n]z^{-n}$ for each channel is recovered.

2.4 Outer Loop: Kinematic Inversion with Smooth Filtering

Here, we develop and analyze the outer loop ILC for static error that cannot be seen by joint encoders. Suppose the outer loop update law has the following form:

$$\delta\boldsymbol{q}_{i+1} = \boldsymbol{Q}(\delta\boldsymbol{q}_i + \boldsymbol{L}_i\delta\boldsymbol{r}_i), \tag{2.8}$$

wherein $\delta\boldsymbol{q}_i \in \mathbf{R}^{n_i}$ is the joint command correction input at i^{th} iteration, low-pass $\boldsymbol{Q}$ is for noise attenuation, $\delta\boldsymbol{r}_i \in \mathbf{R}^{n_o}$ is the end effector output, and $\boldsymbol{L}_i$ is a learning matrix yet to be determined.

In Eqn. 2.2, the converged inner loop can be regarded as a complementary sensitivity $T_{r2y} = 1 - S_{r2e} \approx \frac{QM}{1-Q(1-M)}$. This means $\boldsymbol{q}_i \approx \boldsymbol{q}_i^*$ if the spectrum of joint command $\boldsymbol{q}_i^*$ is within the bandwidth of T_{r2y}. Assume same reference model M and low-pass Q are used, the nested-loop structure can thus be visualized as in Fig. 2.4, where the integrator $\frac{Q}{z-Q}$ is with respect to the iteration index i. It is noted if $\boldsymbol{L}_i$ is selected to be the inverse Jacobian matrix, an approximation of inverse kinematics, it cancels out with the forward kinematics block and the feedback loop behaves like an integrator with unity negative feedback, which is stable.

Now we turn to the algebraic approach for the selection of $\boldsymbol{L}_i$. Suppose the output $\boldsymbol{y}$ is mapped from input $\boldsymbol{q}$ through a static map $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{q}, \boldsymbol{p})$, wherein $\boldsymbol{p}$ contains the kinematic parameters of the manipulator. Assume also the desired output is achieved by input $\boldsymbol{q}_r$ and true parameter vector $\boldsymbol{p}_r$ so that $\boldsymbol{y}_r = \boldsymbol{r}^* = \boldsymbol{f}(\boldsymbol{q}_r, \boldsymbol{p}_r)$. Then, the task space error $\delta\boldsymbol{r}_i$ in Eqn. 2.8 can be expanded as follows:

$$\begin{aligned}\delta\boldsymbol{r}_i &= \boldsymbol{y}_r - \boldsymbol{y}_i \\ &= \boldsymbol{f}(\boldsymbol{q}_r, \boldsymbol{p}_r) - \boldsymbol{f}(\boldsymbol{q}_i, \boldsymbol{p}_0) \\ &= \boldsymbol{f}(\boldsymbol{q}_i, \boldsymbol{p}_0) + \boldsymbol{F}_{\boldsymbol{q}}(\boldsymbol{q}_r - \boldsymbol{q}_i) + \boldsymbol{F}_{\boldsymbol{p}}(\boldsymbol{p}_r - \boldsymbol{p}_0) - \boldsymbol{f}(\boldsymbol{q}_i, \boldsymbol{p}_0) \\ &= \boldsymbol{F}_{\boldsymbol{q}}(\boldsymbol{q}_r - \boldsymbol{q}_0 - \delta\boldsymbol{q}_i) + \boldsymbol{F}_{\boldsymbol{p}}(\boldsymbol{p}_r - \boldsymbol{p}_0).\end{aligned} \tag{2.9}$$

where $\boldsymbol{F}_{\boldsymbol{q}} \triangleq \frac{\partial}{\partial \boldsymbol{q}}\boldsymbol{f}(\bar{\boldsymbol{q}}_i, \boldsymbol{p}_0)$ and $\boldsymbol{F}_{\boldsymbol{p}} \triangleq \frac{\partial}{\partial \boldsymbol{p}}\boldsymbol{f}(\boldsymbol{q}_i, \bar{\boldsymbol{p}}_0)$. This is from mean value theorem with $\bar{\boldsymbol{q}}_i = \boldsymbol{q}_i + \tau_1(\boldsymbol{q}_r - \boldsymbol{q}_i)$, $\bar{\boldsymbol{p}}_r = \boldsymbol{p}_0 + \tau_2(\boldsymbol{p}_r - \boldsymbol{p}_0)$, and τ_1, $\tau_2 \in [0, 1]$. From line three to line four, $\boldsymbol{q}_i = \boldsymbol{q}_0 + \delta\boldsymbol{q}_i$ is substituted.

Now substitute end effector error $\delta\boldsymbol{r}_i$ back in Eqn. 2.8:

$$\begin{aligned}\delta\boldsymbol{q}_{i+1} = \boldsymbol{Q}(\boldsymbol{I} - \boldsymbol{L}_i\boldsymbol{F}_{\boldsymbol{q}})\delta\boldsymbol{q}_i \\ + \boldsymbol{Q}\boldsymbol{L}_i\boldsymbol{F}_{\boldsymbol{q}}(\boldsymbol{q}_r - \boldsymbol{q}_0) \\ + \boldsymbol{Q}\boldsymbol{L}_i\boldsymbol{F}_{\boldsymbol{p}}(\boldsymbol{p}_r - \boldsymbol{p}_0).\end{aligned} \tag{2.10}$$

Suppose $\boldsymbol{F}_{\boldsymbol{q}}$ is square ($n_i = n_o$), $\boldsymbol{L}_i = [\frac{\partial}{\partial \boldsymbol{q}}\boldsymbol{f}(\bar{\boldsymbol{q}}_i, \boldsymbol{p}_0)]^{-1}$ exists, and there is no parametric error ($\boldsymbol{p}_r = \boldsymbol{p}_0$), then the recursion converges within one iteration with $\delta\boldsymbol{q}_1 = \boldsymbol{q}_r - \boldsymbol{q}_0$, provided both $\boldsymbol{q}_r$ and $\boldsymbol{q}_0$ are within the bandwidth of $\boldsymbol{Q}$. For non-square $\boldsymbol{F}_{\boldsymbol{q}}$, $\boldsymbol{L}_i$ can be solved via least square if $n_o > n_i$, or else other criterion needs be imposed for the case $n_o < n_i$ because multiple solutions exist.

In Eqn. 2.10, the first term can be interpreted as internal dynamics with system matrix $\boldsymbol{Q}(\boldsymbol{I} - \boldsymbol{L}_i\boldsymbol{F}_{\boldsymbol{q}})$, whereas the last two terms are forced excitation from static joint error and kinematic bias, respectively. It should be noted for implementation the learning matrix is approximated by inverse Jacobian $\boldsymbol{J}^{-1} = [\frac{\partial}{\partial \boldsymbol{q}}\boldsymbol{f}(\boldsymbol{q}_i, \boldsymbol{p}_0)]^{-1}$, which is evaluated at $\boldsymbol{q}_i$ instead of $\bar{\mathbf{q}}_i$. Therefore, the convergence rate $\|\boldsymbol{Q}(\boldsymbol{I} - \boldsymbol{L}_i\boldsymbol{F}_{\boldsymbol{q}})\|$ degrades if the deviation is large.

2.5 Experimental Results

To demonstrate the efficacy of the nested-loop algorithm, a linear path in Cartesian space which involves simultaneous motion of all six joints is chosen as in Fig. 2.5 and Fig. 2.6. The maximum tip speed is 200 mm/s, and two different payloads, 686 g and 2732 g (corresponds to 10 % and 40 % of maximum capacity) will be tested. This serial-type robot manipulator AR-607 (Industrial Technology Research Institute, Taiwan) is controlled by a C/C++ based MIO (Motion Intelligence Orchestration) unit running at 1 kHz. A laser tracker (AT901, Leica) is used for outer loop ILC execution and end effector positional error evaluation. We will first walk through the inner loop ILC design procedures, and then combine it with the outer loop to show the overall performance.

2.5.1 Inner Loop ILC Design

Herein the inner loop ILC design procedures are demonstrated using joint J2 due to the page limit. It is chosen because J2 is subject to the largest load variation with gravity as investigated by [RHB09].

First, a reference model M and a low-pass filter Q are chosen as in Fig.2.7. M serves to invert the system within limited bandwidth so that input saturation will not be triggered, whereas Q ensures stability by attenuating the learning gain when $(1 - FG) \approx (1 - M)$ becomes large. Since the validity of the inversion is limited by M, the bandwidth of Q should be smaller or equal to that of M.

Secondly, an impulsive reference in Fig. 2.8 is used to sufficiently excite system dynamics.

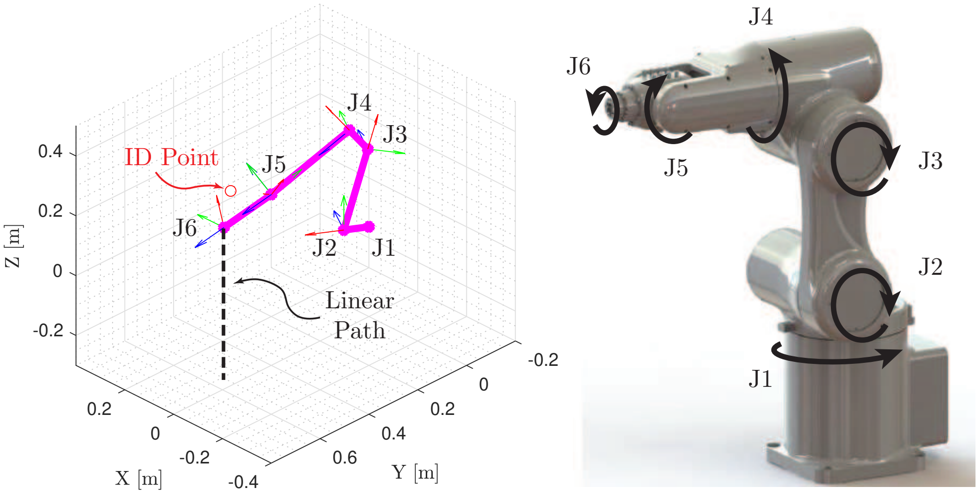


Figure 2.5: Serial-type robot under study and the linear test path in Cartesian workspace. Magenta: robot initial configuration; black dashed line: linear test path; red circle: tip position where the system is identified.

In this example, a constraint $v_{max} = 20^\circ/s$ is triggered for a velocity step, and thus system FIR is acquired from the acceleration pulse.

Thirdly, following the procedures described in Sec. 2.3.2, the inversion is constructed as in Fig. 2.9 and 2.10 wherein time- and frequency-domain responses are shown. Note that $FG \approx M(f[n] \star g[n] \approx m[n])$ since F $(f[n])$ is an inversion (deconvolution) of G $(g[n])$ with respect to M $(m[n])$. Note also the non-causality of F $(f[n])$ so that a zero-phase M $(m[n])$ can be achieved.

In Fig. 2.11, the effect of Q filter bandwidth is shown using the impulsive reference in Fig. 2.8 as target profile. The reason of choosing such reference is because the converged feedforward sequence can be used as refinement of system inversion. Though in this work the improved learning filter is not employed, interested reader may refer to [CT17] for iterative learning filter refinement. Clearly a conservative Q (5 Hz bandwidth) guarantees asymptotic stability; however, this is at the expense of performance degradation. An aggressive Q (60 Hz bandwidth) may gives better learning transient, but it could result in divergence due to the violation of stability criterion in high-frequency region. This study shows the trade-off

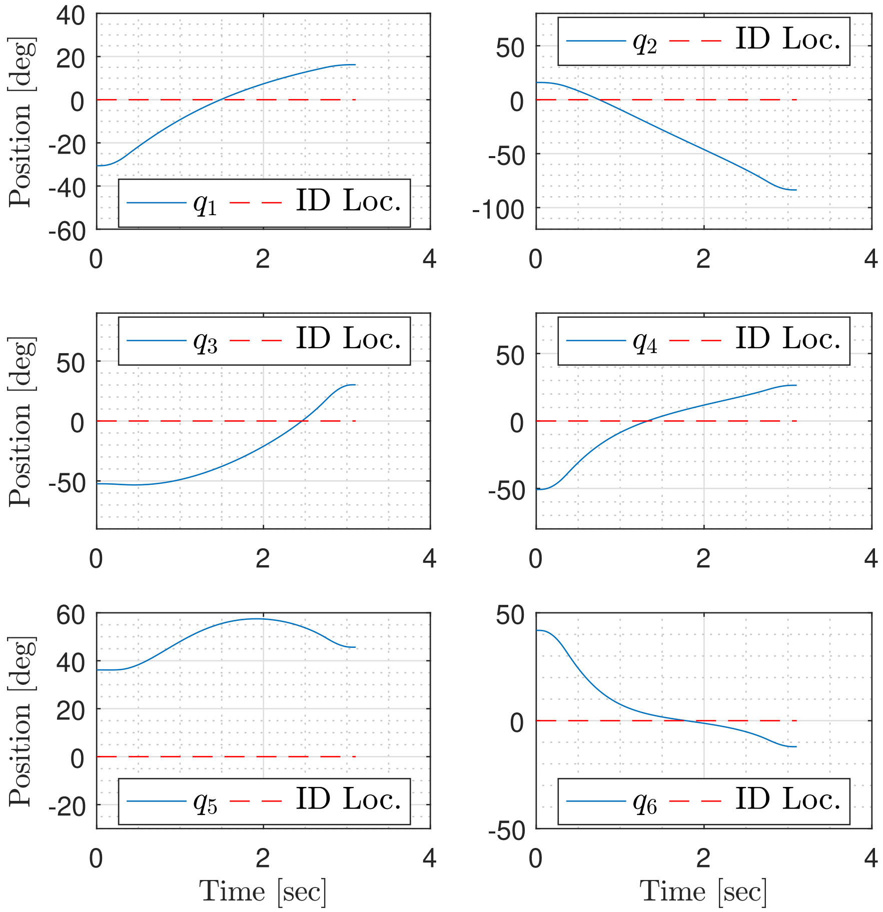


Figure 2.6: Joint reference profiles associated with test path in Fig. 2.5. Red dashed line shows the position where the joint is identified.

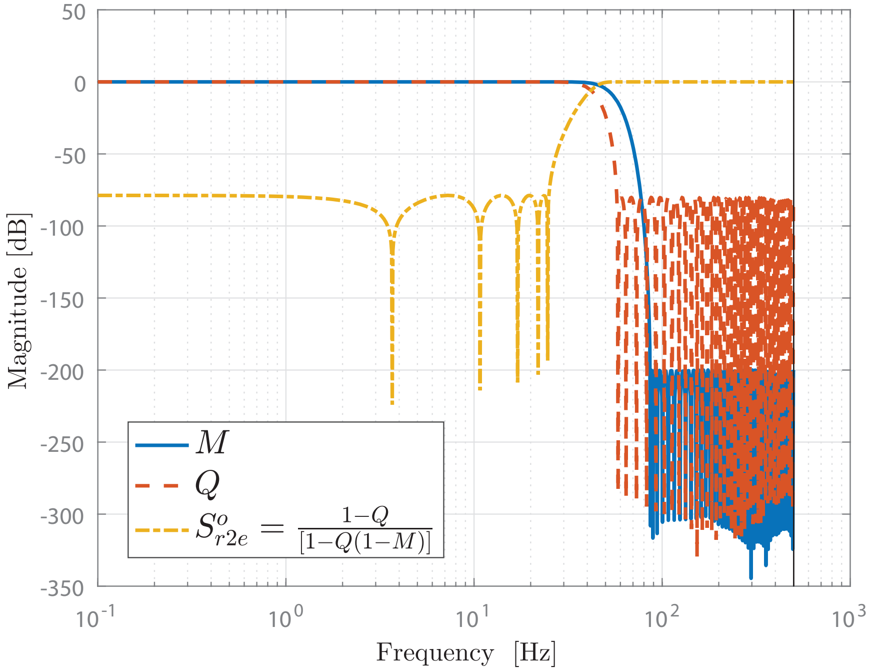


Figure 2.7: Magnitude plot of reference model M, low-pass filter Q, and the optimal sensitivity S^o_{r2e} from reference to tracking error.

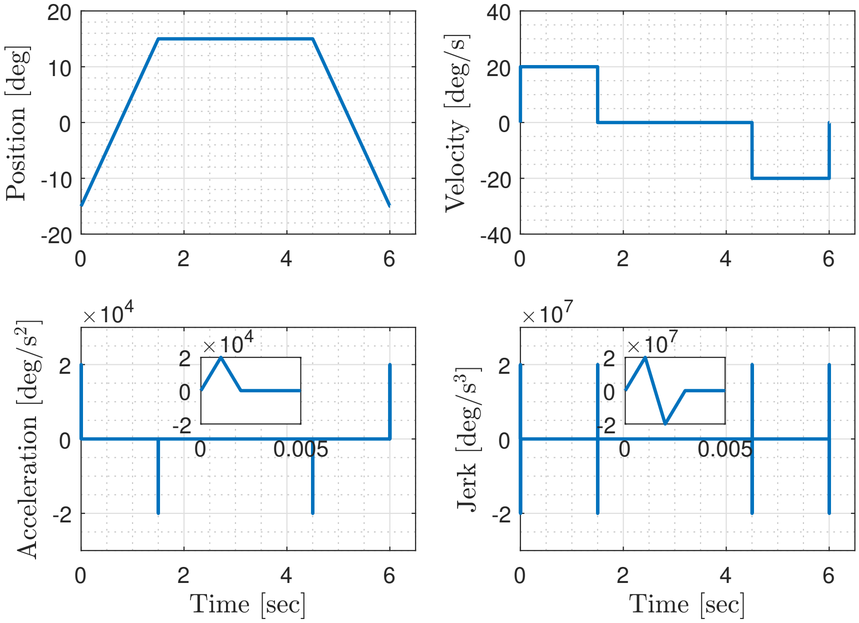


Figure 2.8: Impulsive reference (velocity step) with motion constraints.

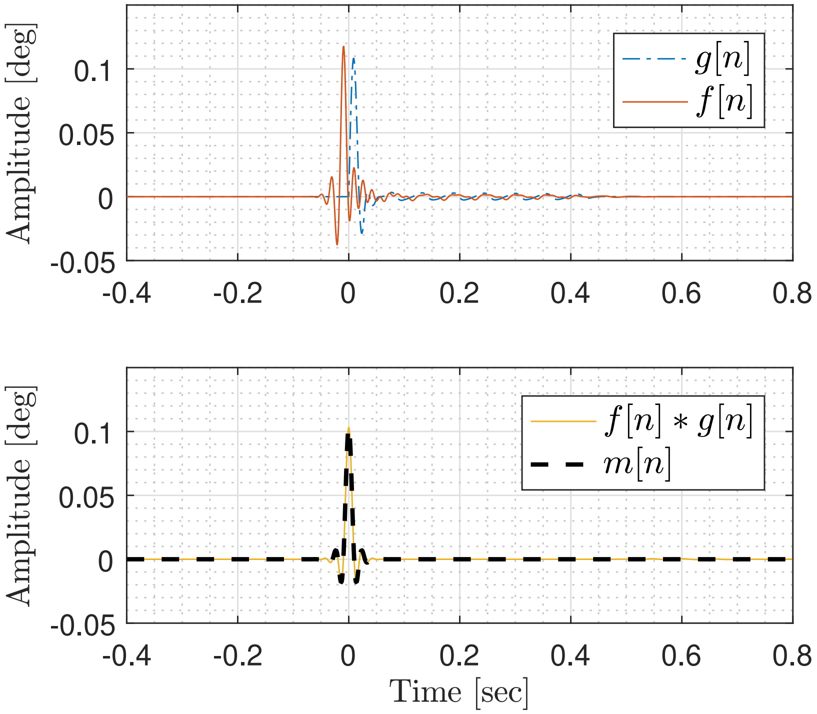


Figure 2.9: Time-domain finite impulse responses of plant $g[n]$, inversion $f[n]$, and the convolution $f[n] * g[n]$.

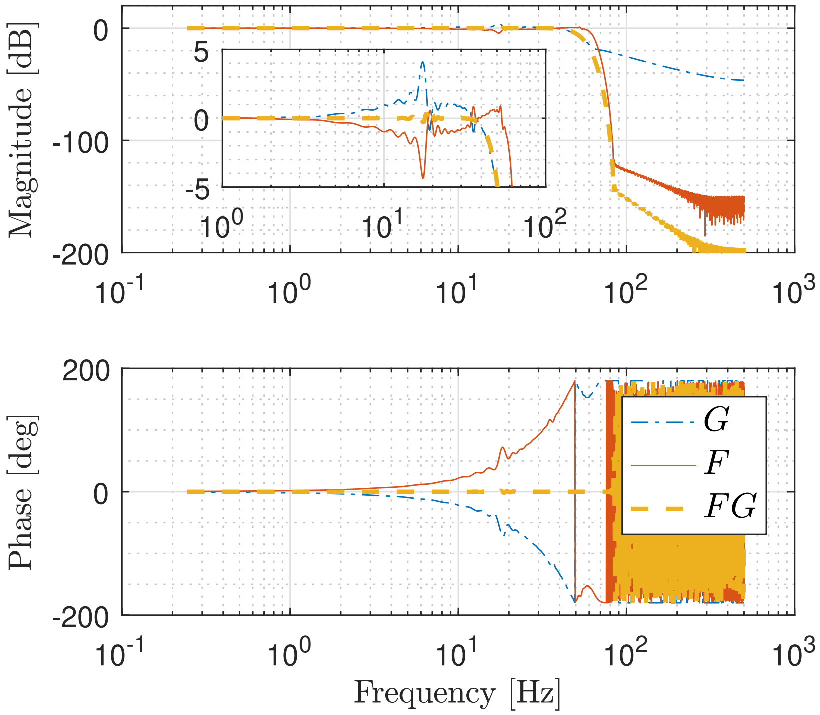


Figure 2.10: Frequency responses of plant G, inversion F, and FG.

between robustness and performance for Q selection.

To conclude this section, we conduct the inner-loop ILC using the joint profiles in Fig. 2.6. The convergence plot of experiment and simulation is shown in Fig. 2.12, where iteration 0 denotes the servo-loop performance. In this test, the experiment shows fast convergence, which is a feature of inversion-based ILC. The reference model M and low-pass filter Q for all six joints are as in Fig. 2.7, and will be employed for the proposed nested-loop ILC

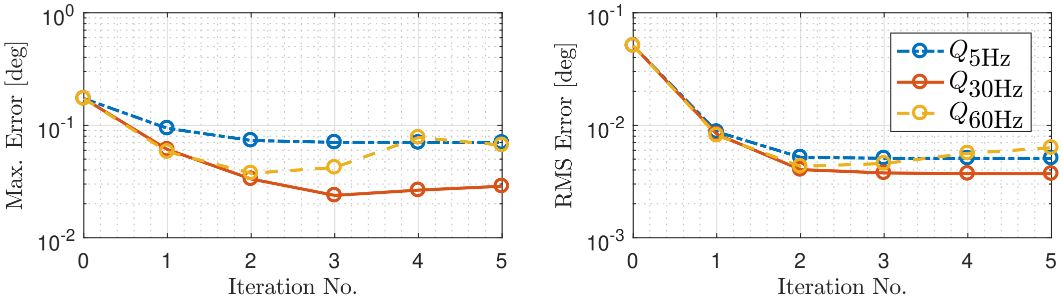
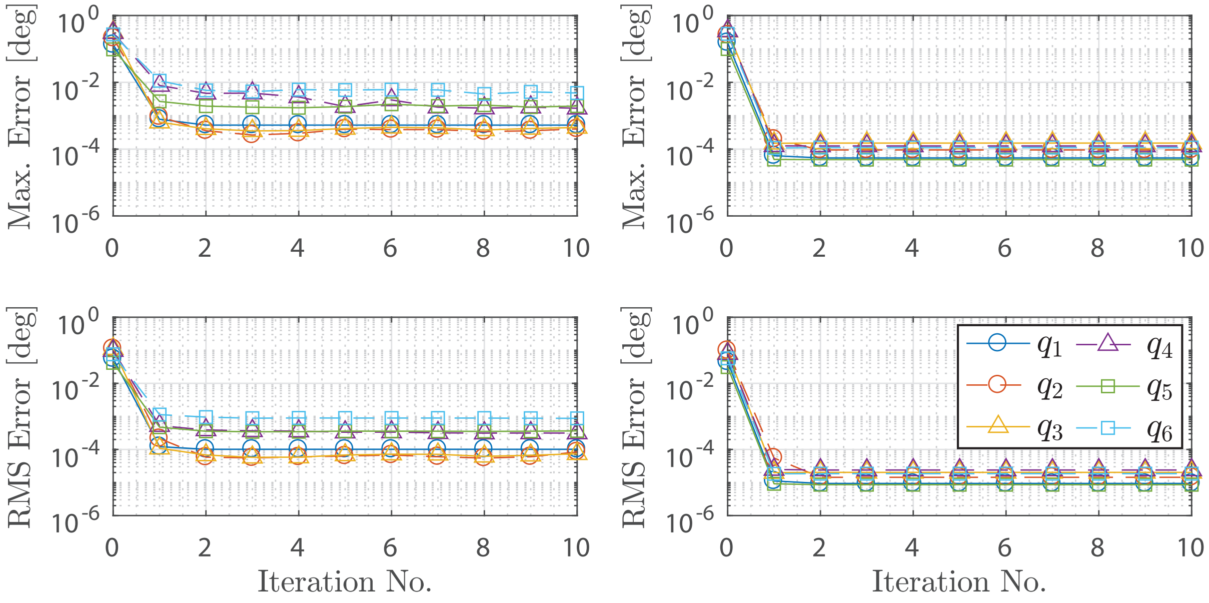


Figure 2.11: Effect of Q filter bandwidth on ILC convergence.



(a) Experiment (b) Simulation

Figure 2.12: Error convergence using the joint profiles in Fig.2.6

experiments hereafter.

2.5.2 Nested-Loop ILC Tracking Performance

For the nested-loop ILC experiment, outer and inner loop iteration numbers are chosen to be $I = 2$ and $J = 5$, respectively. In inner loop, the algorithm updates feedforward input $\boldsymbol{u}_{i/j}$ for $j = 1, \cdots, J$ and $\boldsymbol{q}_i^*$ is the target joint output. Upon inner loop convergence $(j = J)$, the outer loop computes the update for $\boldsymbol{q}_{i+1}^* = \boldsymbol{q}_0^* + \delta\boldsymbol{q}_{i+1}^*$ and $\boldsymbol{q}_{i+1}^*$ will be the target joint output for the consecutive J inner loop iterations. We denote data set $\boldsymbol{q}_{0/0}$ and $\boldsymbol{y}_{0/0}$ for servo-loop performance, which is independent of ILC recursion. The initialization of $\boldsymbol{u}_{0/1}$ is from zero-input zero-output condition, and the result $\boldsymbol{q}_{0/1}$ shows inversion quality. This can be seen by plugging $u_0 = 0, y_0 = 0$ into Eqn. 2.1. For initialization of $\boldsymbol{u}_{i+1/1}$, we select $\boldsymbol{q}_{i+1}^*$ for convenience; however, one can use $\boldsymbol{u}_{i+1/1} = \boldsymbol{Q}[\boldsymbol{u}_{i+1/J} + \boldsymbol{F}(\boldsymbol{q}_{i+1}^* - \boldsymbol{q}_{i/J})]$ to accelerate the convergence.

To avoid vibration excitation, prior to the start of inner loop iteration a notch filter is

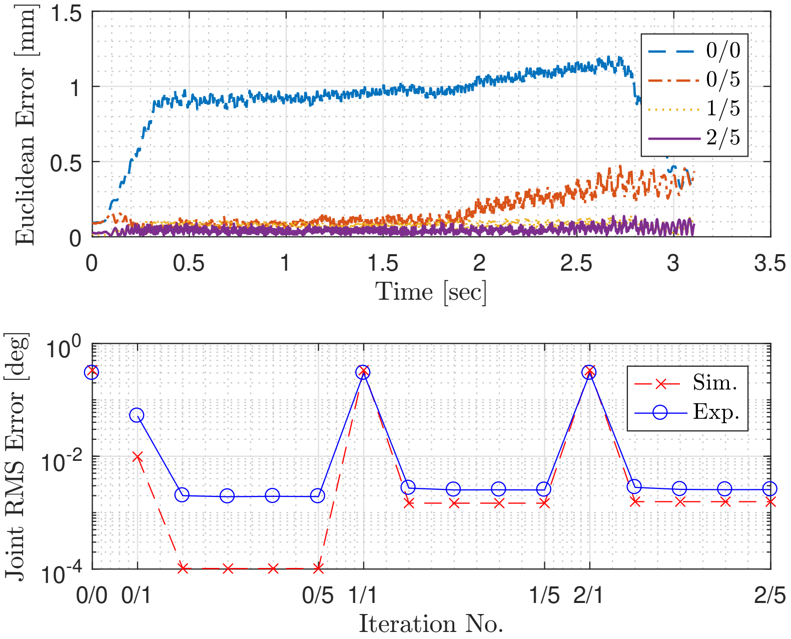


Figure 2.13: Nested-loop ILC performance with 686 g payload. Top: tip error progression. Bottom: joint RMS error convergence.

automatically constructed. The algorithm utilizes the power spectral density (PSD) of laser tracker measurement, and adjusts the corresponding notch location(s), bandwidth(s), and depth(s) based on the prominence of the mode(s). This notch filter is combined with the low-pass Q for zero-phase filtering.

The experimental results are shown in Fig. 2.13 and Fig. 2.14 for 686 g and 2732 g payload, respectively. The top plot shows the tip error progression for selected iterations. Iteration 0/0 shows the nominal servo-loop (PID) performance, iteration 0/5 shows the performance limit of inner loop ILC, and iteration 1/5 shows the converged result after one outer-loop update. The lower part shows the RMS joint error convergence. The joint tracking simulation is running on the FIR model acquired from the identification input. The RMS error at iteration 0/1 illustrates the inversion quality, and the performance at iteration 1/1 and 2/1 is comparable to that of iteration 0/0 because they correspond to a command update without any learning action. Note also the performance level from simulation is varying for different outer loops. This is due to the update of the notch filtering.

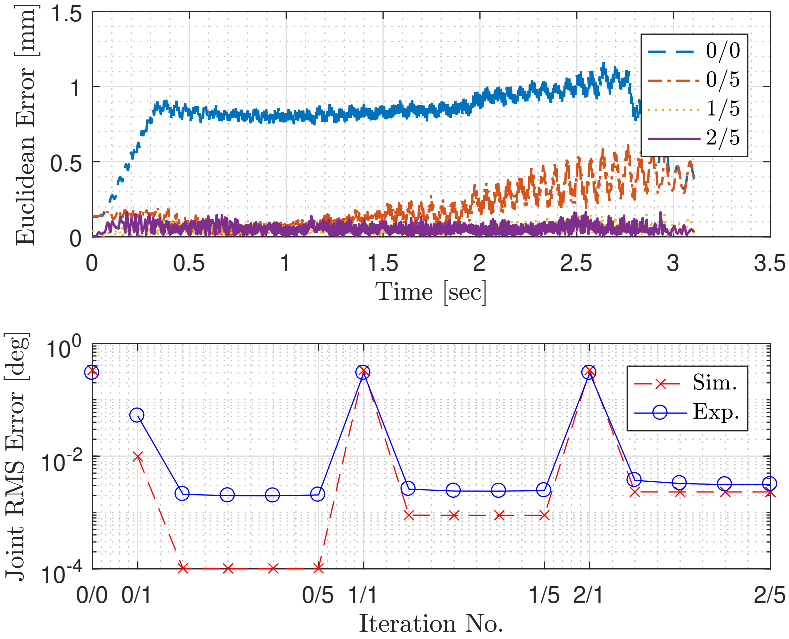


Figure 2.14: Nested-loop ILC performance with 2732 g payload. Top: tip error progression. Bottom: joint RMS error convergence.

The performance of the nested-loop ILC is summarized in Tab. 2.1, and note the outer loop converges almost within one iteration. The error increase at iteration 2/5 for 2732 g could be steady state oscillation, but requires more iterations for verification. The maximum error is mitigated by an order of magnitude, showing the effectiveness of the proposed algorithm. We should mention that the system is identified without any payload, and the configuration used is not on the test path as well. This verifies the assumption that the disturbance torque sensed is mostly shielded by the gear ratio effect. In practice, to further enhance tracking performance the dynamic inversion can be automated with some configuration when payload or trajectory changes.

2.6 Conclusion

This chapter proposes a nested-loop ILC structure for industrial robot manipulators. It attributes the dynamic error to the drives, and static error to imprecise kinematics as well as

Table 2.1: Tip tracking error statistics of nested-loop ILC

Iteration	686 g		2732 g	
No.	RMS	Max.	RMS	Max.
0/0	0.9210	1.2033	0.8267	1.1614
0/5	0.2024	0.4735	0.2458	0.6184
1/5	0.0763	0.1452	0.0598	0.1577
2/5	0.0495	0.1383	0.0643	0.1672

*All values are in units of mm.

joint bias. The inner loop uses a data-based inversion for learning filter design that takes motion constraints into account. The resultant filter does not require model reduction and thus give rise to good inverse approximation and fast ILC convergence. The outer loop ILC analysis is also given, which suggests using inverse Jacobian as learning matrix. To demonstrate the proposed scheme, simulation and experimental studies are performed on a serial six degree-of-freedom robot manipulator. This scheme requires only kinematic parameters of robot manipulators, showing effective learning and flexibility. Prospective future research includes investigating learning strategies by incorporating time-varying dynamics and coupling effect into account.

CHAPTER 3

Data-driven ILC for Nonlinear Dynamical Systems

3.1 Tracking Control for Nonlinear Systems

Iterative learning control (ILC) is well suited for industrial processes that are repeated in nature. The essence of ILC is that the error information from previous iteration is used to update the control sequence for the next iteration, which amounts to an integral action over iteration domain, and thus the tracking error is asymptotically zero in theory. Applications of ILC can be found in machine tools [KK96], robot manipulators [WZW18], flatbed printers [BWK16], wafer positioning systems [DB02], and so forth.

Many systems, however, are generally nonlinear and time-varying. For example, the dynamics of robot manipulators are nonlinearly position- and velocity-dependent. To address this, a common approach is to obtain a linearized model of the nonlinear system and apply linear ILC [Gor02]. To ensure robustness of such ILC, extensive experiments are required to quantify system variation as uncertainty [SPS14][VDB11]. It is also not trivial to justify the nominal model on which linear ILC design is based. Another possibility is to adopt ILC law for nonlinear systems such as Lyapunov-based [CHX08] and gain-scheduled [BWJ05][HXV14] approaches. Nonetheless, these approaches rely on the knowledge of a nonlinear model, which can be difficult and expensive to acquire.

In this work, we propose a data-driven approach that implicitly identifies the linear time-varying (LTV) system inverse along the trajectory tracked. The system inverse is used as learning filter for the error sequence, and so it has the fast convergence property of inversion-based ILC algorithms [ZO18]. This is achieved by exploiting the tracking capabilities of

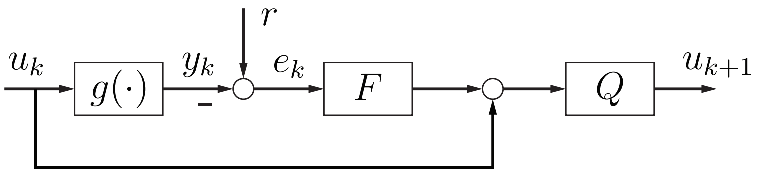


Figure 3.1: Generic SISO ILC structure. $u_k = u_k(t)$, $y_k = y_k(t)$, $r = r(t)$, and $e_k = e_k(t)$ are signals in $\mathbb{R}$. $F = F(z^{-1})$, $Q = Q(z^{-1})$ are filters with z^{-1} the delay operator.

adaptive filtering [Hay08], provided the system is linear and the variation is slow along the trajectory. The approach is thus model-free because the adaptation accounts for the system variation as time progresses. Another advantage is to avoid explicitly constructing the system inverse, which greatly reduces the computation. The time-varying tap weights of the adaptive filter also alleviates the memory usage.

3.2 Data-driven ILC for nonlinear SISO Systems

3.2.1 Preliminaries

Transfer Function Representation

Consider a single-input-single-output (SISO) stabilized plant $g(\cdot)$ that maps input vector $\boldsymbol{u} \in \mathbb{R}^N$ and initial state $\boldsymbol{x}_0$ into output sequence $\boldsymbol{y} \in \mathbb{R}^N$, namely $\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{u}, \boldsymbol{x}_0)$ or abbreviated as $\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{u})$, with N the length of command sequence. Given initial input $\boldsymbol{u}_0 = [u_0(0), \cdots, u_0(N-1)]^T$ and desired trajectory $\boldsymbol{r} = [r(0), \cdots, r(N-1)]^T$, a generic ILC process can be illustrated as in Fig. 3.1, and put into the following filter representation:

$$u_{k+1}(t) = Q(z^{-1}) \left[u_k(t) + F(z^{-1}) e_k(t) \right], \tag{3.1}$$

the subscript $k = 0, 1, \cdots$ is the iteration index, $e_k = r - y_k$ is the tracking error, F is called learning filter, Q is a low-pass filter that shuts off learning in high-frequency region. If $g(\cdot) = G(z^{-1})$ is a linear process, namely $y_k(t) = G(z^{-1}) u_k(t)$, or $y_k = G u_k$ as shorthanded, then Eqn. (3.1) can be expanded as follows:

$$\begin{aligned} u_{k+1} &= Q(u_k + Fe_k) \\ &= Q[u_k + F(r - y_k)] \\ &= Q(1 - FG)u_k + QFr, \end{aligned} \tag{3.2}$$

and thus ILC stability can be determined by the convergence rate γ:

$$\gamma \triangleq \|Q(1 - FG)\|_\infty < 1. \tag{3.3}$$

Eqn. (3.3) motivates inversion-based methods, for which $F \cong G^{-1}$ is the plant inverse. If the inversion F is exact, the convergence rate $\gamma = 0$, meaning the ILC converges within one step, and the converged error $e_\infty = 0$ [ZO18]. If the inversion F is with respect to a limited bandwidth zero-phase reference model M, namely $FG = M$, then the bandwidth of Q should be selected to be equal or smaller than that of M to ensure Eqn. (3.3) is satisfied. The reference model can incorporate delays for ILC preview action.

Lifted Representation

For LTV systems analysis, it is common to represent G with finite length N in the lifted domain:

$$\mathbf{G} = \begin{bmatrix} h_0(0) & 0 & \cdots & 0 \\ h_1(1) & h_1(1) & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ h_{N-1}(N-1) & h_{N-1}(N-1) & \cdots & h_1(N-1) \end{bmatrix}, \tag{3.4}$$

in which $h_i(t)$ is the i^{th} impulse response coefficient at instant t, and $G(t, z^{-1}) = \sum_{i=0}^{\infty} h_i(t) z^{-i}$. If $\mathbf{G}$ is a linear-time-invariant (LTI) system, each diagonal has a constant value and $\mathbf{G}$ is Toeplitz. Similarly, the stability condition can be derived in the lifted domain as [NG02]:

$$\rho\left(\mathbf{Q}(\mathbf{I} - \mathbf{FG})\right) < 1, \tag{3.5}$$

wherein $\rho(\cdot)$ is the maximum eigenvalue of a matrix. Moreover, monotonic convergence is established when:

$$\gamma = \bar{\sigma}\left(\mathbf{Q}(\mathbf{I} - \mathbf{FG})\right) < 1, \tag{3.6}$$

and $\bar{\sigma}$ is the maximum singular value.

ILC Design for Nonlinear Time-varying System

For nonlinear time-varying system, one common design approach is to treat the system variation as uncertainty and apply an LTI learning filter. As such, the stability condition in Eqn. (3.3) can be rewritten as:

$$||Q[1 - FG(1 + W\Delta)]||_\infty < 1, \tag{3.7}$$

where $g(\cdot) = G(1 + W\Delta)$ is substituted, W is a frequency weighting filter and Δ is the phase uncertainy with $||\Delta||_\infty < 1$. However, this approach is sacrificing performance for robust stability. The question of which nominal model G should ILC be designed based on also needs to be answered. And, lots of experiments are required to quantify the uncertainty so that Eqn. (3.7) can be used.

Another approach to account for system variation is build a look-up table for F so that the learning gain is scheduled in accordance with some system parameters [BWJ05][HXV14]. This approach improves the tracking performance significantly but further increases the storage requirement. Considerable amount of experiments and off-line learning filter design are needed.

As opposed to the model-based design approaches where either a model set of G's or the variation $W\Delta$ is needed, herein we introduce data-driven algorithms utilizing experimental input-output data that implicitly determine the learning filter or matrix. Our proposed approach which makes use of adaptive filtering will first be presented, showing how the

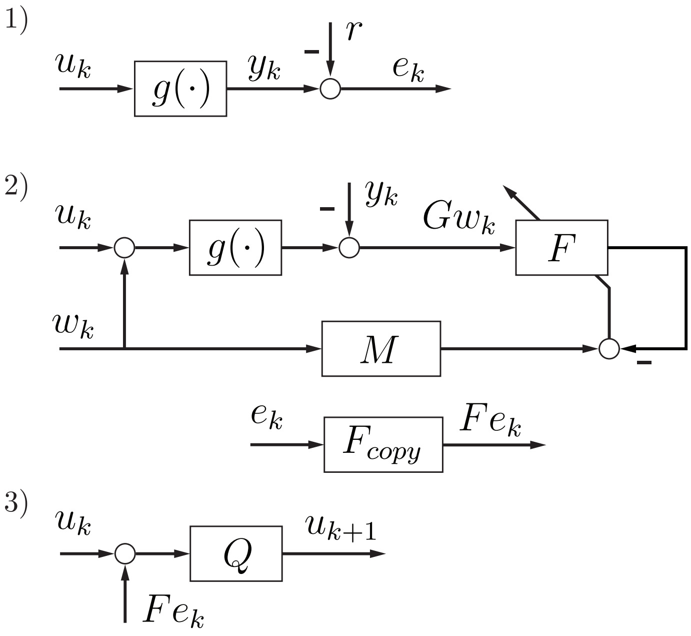


Figure 3.2: Proposed SISO adaptive ILC: 1) Inject u_k to the system and compute tracking error $e_k = r - y_k$. 2) Error filtering utilizing adaptive filter: w_k is the input to excite system dynamics around y_k, the nonlinear effect is mitigated by subtracting y_k obtained from step 1). The linearized plant $G = G(t, z^{-1})$ along y_k and the adaptive inverse $F = F(t, z^{-1})$ are both time-varying. 3) The ILC command update for next iteration.

tedious identification of model-based design can be avoided, while maintaining the desirable features as those inversion-based methods. As a comparison to state-of-the-art data-driven ILC, an extension of adjoint-based approach [BO15] to nonlinear time-varying systems will also be given and simulated.

3.2.2 SISO Adaptive ILC

A key step in inversion-based ILC algorithms is the computation of control correction, namely the filtered error Fe_k. If the system is nonlinear and time-varying, the computation of F can be intensive, and storage of look-up table for F's grows with the size of the trajectory or the region of operation.

In Fig. 3.2, an innovative data-driven ILC is presented to circumvent these difficulties. The critical error filtering process is shown in step 2): by superimposing an excitation input w_k on the nominal input u_k, the linearized time-varying system inverse $F(t, z^{-1})$ is implicitly identified along y_k via adaptive filtering that minimizes:

$$\min_F \|(M - FG)w_k\|_2, \tag{3.8}$$

wherein M is the reference model, G is the linearized plant, and F is the approximate inverse under the assumption that variation of system dynamics is slow compared to adaptation. The filtered error sequence is obtained by feeding e_k through the copy filter F_{copy} without explicitly knowing F. Notice that the nonlinear effect is accounted for by subtracting the term y_k acquired from a separate experiment with only u_k as input, as is shown in step 1) of Fig. 1.

Going back to the lifted domain ILC update $\boldsymbol{u}_{k+1} = \mathbf{Q}(\boldsymbol{u}_k + \mathbf{F}\boldsymbol{e}_k)$, adaptive filtering can be thought of as constructing the learning matrix $\mathbf{F}$ by filling in time-varying FIR coefficients row by row:

$$\mathbf{F} = \begin{bmatrix} \ddots & 0 & \cdots & 0 \\ \cdots & \boldsymbol{f}_m(t-1) & \cdots & 0 \\ \cdots & \cdots & \boldsymbol{f}_m(t) & 0 \\ & \vdots & \vdots & \ddots \end{bmatrix}, \tag{3.9}$$

where $\boldsymbol{f}_m(t) = [f_{m-1}(t), \cdots, f_0(t)]$ is the coefficient row vector. The complete algorithm can be summarized in Algorithm 1.

Algorithm 1 (SISO adaptive ILC) *Given an initial input $u_0 = r$, and iteration index $k = 0$, the algorithm is described as follows:*

1. *Conduct experiment with u_k, and compute $e_k = r - y_k$.*

2. *Superimpose excitation signal w_k on u_k, and conduct experiment as shown in step 2) of Fig. 3.2 to obtain filtered error signal Fe_k.*

3. *Compute ILC update $u_{k+1} = Q(u_k + Fe_k)$.*

4. *Terminate iteration if stop criterion is met, otherwise increment iteration index $k \leftarrow k + 1$ and go to step 1).*

In practice, a training phase should be added at the beginning of the trajectory for adaptive filter convergence. The choices of reference model M and low-pass filter Q serve as design parameters as those in inversion-based algorithms. The energy and spectral spread of perturbation/excitation sequence w_k can be inferred from nominal servo performance and an estimate of system bandwidth.

To conclude this section, we itemize the advantages enabled by the introduction of adaptive filtering:

- It is an inversion-based approach and therefore gives fast convergence and good tracking performance.

- The adaptive filtering in step 2) of Algorithm 1 can be done off-line so long as $g(u_k+w_k)$ is stored. This alleviates the hardware requirement for real-time adaptive filtering.

- The adaptive filter has finite-impulse-response (FIR) structure with time-varying tap coefficients, and so the storage is only proportional to m, the number of taps, instead of the trajectory size N.

- The time-varying inverse is entirely data-driven by minimizing the criterion $\|(M - FG)w_k\|_2$ instead of an explicit construction. The well-developed fast adaptive algorithms [Hay08], which have computation complexity $\mathcal{O}(m)$, making this approach tractable.

Improving Error Convergence Omitting the effect of smoothing matrix $\mathbf{Q}$, from control recursion $\boldsymbol{u}_{k+1} = (\mathbf{I} - \mathbf{F}_L\mathbf{G})\boldsymbol{u}_k + \mathbf{F}_L\boldsymbol{r}$, the convergence rate is $\gamma = \bar{\sigma}(\mathbf{I} - \mathbf{F}_L\mathbf{G})$, wherein the subscript L denotes left inverse. Previously we have applied adaptive filter to construct $\mathbf{F}_L$ for fast convergence. However, since our goal is to enable fast error convergence, we should instead look into error recursion:

$$\boldsymbol{e}_{k+1} = (\mathbf{I} - \mathbf{G}\mathbf{F}_R)\boldsymbol{e}_k, \tag{3.10}$$

wherein the subscript R denotes right inverse. In adaptive filtering theory, a right inverse cannot be adapted directly [Hay08]. Our workaround is instead adapting the non-causal $\mathbf{G}^T$, or its filter form $G^* = G(z, z) = \sum_{i=0}^{\infty} h_i(t)z^i$, which is often referred to adjoint system. Because $\mathbf{I} - \mathbf{F}_R^T\mathbf{G}^T$ is identical to $\mathbf{I} - \mathbf{G}\mathbf{F}_R$, adapting on $\mathbf{G}^T$ allows for having $\mathbf{F}_R^T$ on the left-hand side for adaptive filtering. From signal processing, $\mathbf{G}^T$ is equivalent to $\mathcal{T}\mathbf{G}\mathcal{T}$ where $\mathcal{T} \in \mathbb{R}^{N\times N}$ is the time-reversal operator:

$$\mathcal{T} = \begin{bmatrix} 0 & \cdots & 1 \\ \vdots & \iddots & \vdots \\ 1 & \cdots & 0 \end{bmatrix}. \tag{3.11}$$

Assume nonlinearity has been addressed, shown in Fig. 3.3 is how the non-causal filtering is used for ILC error filtering. The key idea is to first get an adjoint F_R^* version of F_R by adapting G^*, then take adjoint again and $F_R = (F_R^*)^*$ is recovered. The algorithm is summarized in 2.

Algorithm 2 (SISO adaptive ILC with improved error convergence) *Given an initial input $u_0 = r$, and iteration index $k = 0$, the algorithm is described as follows:*

1. *Conduct experiment with u_k, and compute $e_k = r - y_k$.*
2. *Superimpose excitation signal $\mathcal{T}w_k$ on u_k, remove nonlinearity by subtraction of y_k, then obtain linearized system response G^*w_k by output time-reversal.*

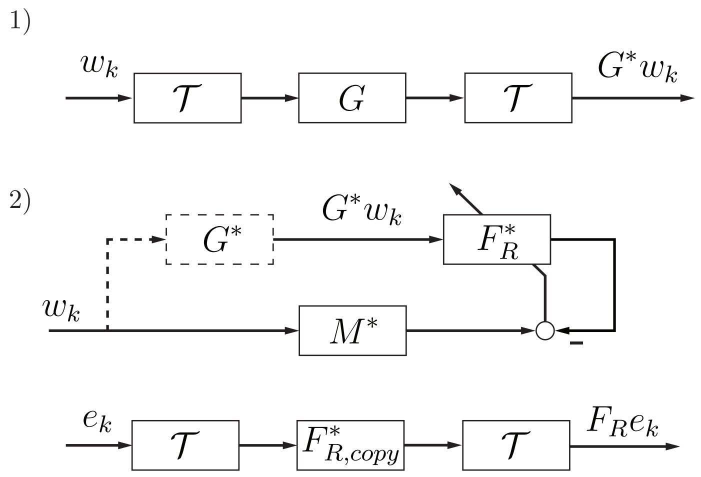


Figure 3.3: ha

3. Use adaptive filter as in Fig. 3.3 to obtain filtered error signal Fe_k.

4. Compute ILC update $u_{k+1} = Q(u_k + Fe_k)$.

5. Terminate iteration if stop criterion is met, otherwise increment iteration index $k \leftarrow k+1$ and go to step 1).

3.2.3 SISO Adjoint ILC

Herein we review the adjoint-based ILC by [BO15], and extend it to nonlinear time-varying systems. The essence of adjoint-based approach for linear systems is to utilize the adjoint system $\mathbf{G}^T$ as the learning matrix so that the ILC stability can be ensured by applying a small learning gain ε:

$$\rho(\mathbf{I} - \varepsilon \mathbf{G}^T \mathbf{G}) < 1. \tag{3.12}$$

The selection of scalar ε becomes a simpler design problem because $\mathbf{G}^T\mathbf{G}$ is symmetric, or equivalently, the filter form G^*G is zero-phase [YW05], and $G^* = \sum_{i=0}^{\infty} h_i(t) z^i$ is the non-causal system adjoint. The adjoint ILC is stabilizing so long as the learning gain ε is small enough.

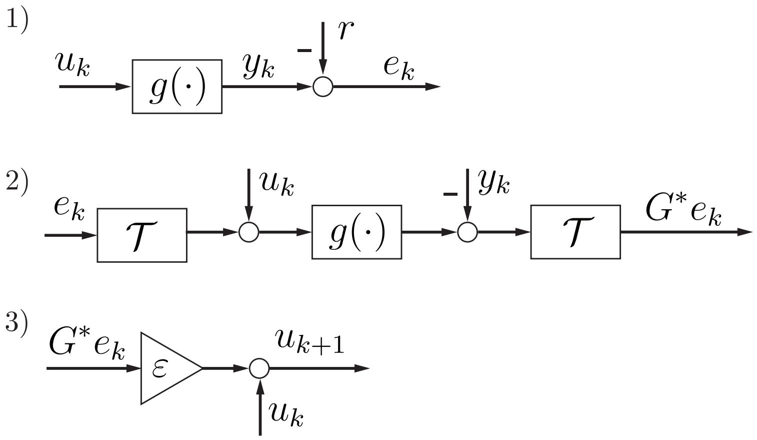


Figure 3.4: Modified SISO adjoint/gradient ILC: 1) Inject u_k to the system and compute tracking error $e_k = r - y_k$. 2) Error filtering using system adjoint G^*. The nonlinear effect is mitigated by subtracting y_k obtained from step 1). 3) The ILC command update for next iteration.

The error filtering process with adjoint operator is depicted in step 2) of Fig. 3.4, wherein $\mathcal{T}$ is a time-reversal operator, and G is the linearized plant along y_k. Note the same term $y_k = g(u_k)$ is subtracted as in Fig.3.2 to account for nonlinear effect because in general $g(u_k + \mathcal{T} e_k) - g(u_k) \neq g(\mathcal{T} e_k)$ for a nonlinear mapping $g(\cdot)$. The complete algorithm can be summarized in Algorithm 3. We also refer to it as gradient ILC because filtered error with system adjoint is the gradient with respect to $\mathcal{J} = \boldsymbol{e}_{k+1}^T \boldsymbol{e}_{k+1}$, as will be shown later in this section.

Algorithm 3 (SISO adjoint/gradient ILC) *Given an initial input $u_0 = r$, and iteration index $k = 0$, the data-driven ILC with system adjoint as gradient can be described in the following steps:*

1. *conduct experiment with u_k, and compute $e_k = r - y_k$.*

2. *superimpose time-reversed error sequence $\mathcal{T} e_k$ on u_k, and conduct experiment as shown in step 2) of Fig. 3.4 to obtain filtered error signal $G^* e_k$.*

3. *compute ILC update $u_{k+1} = u_k + \varepsilon G^* e_k$.*

4. terminate iteration if stop criterion is met, otherwise increment iteration index $k \leftarrow k+1$ and go to step 1).

The adjoint-based approach eliminates the need of learning filter F and low-pass filter Q. The only design parameter is the choice of ε. The theoretical upper bound $\bar{\varepsilon}$ can be determined by $\bar{\varepsilon} = 2||G^*G||_\infty^{-1}$ using ILC stability condition. However, the system gain $||G^*G||_\infty$ is usually not known, and so usually a conservative learning gain is used, leading to slow ILC convergence.

Quasi-Newton Method In [BKO18], an accelerated adjoint ILC for LTI systems is proposed. This is by the optimization formulation with cost function $\mathcal{J}(\boldsymbol{u}_k) = \boldsymbol{e}_k^T \boldsymbol{e}_k$: From optimality condition $\frac{\partial \mathcal{J}(\boldsymbol{u}_{k+1})}{\partial \boldsymbol{u}_{k+1}} = 0$, one can derive the famous Newton's method:

$$\arg\min \mathcal{J}(\boldsymbol{u}_{k+1}) = \boldsymbol{u}_k - \underbrace{\left(\frac{\partial^2 \mathcal{J}(\boldsymbol{u}_{k+1})}{\partial \boldsymbol{u}_{k+1}^2} \right)^{-1}}_{(\mathbf{G}^T\mathbf{G})^{-1}} \underbrace{\left(\frac{\partial \mathcal{J}(\boldsymbol{u}_{k+1})}{\partial \boldsymbol{u}_{k+1}} \bigg|_{\boldsymbol{u}_{k+1}=\boldsymbol{u}_k} \right)}_{-2\mathbf{G}^T \boldsymbol{e}_k} \tag{3.13}$$

Note when inverse Hessian is replaced with identify, we recover adjoint as a gradient descent method. Since the inverse Hessian is not directly accessible, in [BKO18] a data driven estimate $\mathbf{B}_k$ is used instead, and the corresponding control update is as follows:

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \varepsilon_k \mathbf{B}_k \mathbf{G}^T \boldsymbol{e}_k. \tag{3.14}$$

The trial-varying update for $\mathbf{B}_k$ is from the famous Broyden-Fletcher-Goldfarb-Shannon (BFGS) algorithm [Fle13]:

$$\mathbf{B}_{k+1} = \mathbf{B}_k - \frac{\boldsymbol{\Delta}_k \boldsymbol{\zeta}_k^T \mathbf{B}_k + \mathbf{B}_k \boldsymbol{\zeta}_k \boldsymbol{\Delta}_k^T}{\boldsymbol{\Delta}_k^T \boldsymbol{\zeta}_k} + \left(1 + \frac{\boldsymbol{\zeta}_k^T \mathbf{B}_k \boldsymbol{\zeta}_k}{\boldsymbol{\Delta}_k^T \boldsymbol{\zeta}_k} \right) \frac{\boldsymbol{\Delta}_k \boldsymbol{\Delta}_k^T}{\boldsymbol{\Delta}_k^T \boldsymbol{\zeta}}, \tag{3.15}$$

where $\boldsymbol{\Delta}_k = \boldsymbol{u}_k - \boldsymbol{u}_{k-1}$, $\boldsymbol{\zeta}_k = \boldsymbol{v}_k - \boldsymbol{v}_{k-1}$ with $\boldsymbol{v}_k = 2\mathbf{G}^T \boldsymbol{e}_k$. And the iteration-varying gain:

$$\varepsilon_k = \frac{||\boldsymbol{v}_k||_{\mathbf{B}_k}^2}{||\mathbf{G}\mathbf{B}_k \boldsymbol{v}_k||^2}, \tag{3.16}$$

is obtained by substituting Eqn. 3.14 into cost functional $\mathcal{J}(\boldsymbol{u}_k) = \boldsymbol{e}_k^T \boldsymbol{e}_k$. The accelerated performance is at the expense of extra $\mathcal{O}(N^2)$ computation complexity in inverse Hessian estimate and another experiment with input $\mathbf{B}_k \boldsymbol{v}_k$ for determining the learning gain ε_k. This accelerated method can be summarized in Algorithm 4. For more details, we refer readers to the original work in [BKO18].

Algorithm 4 (SISO quasi-Newton ILC) *Given an initial input $\boldsymbol{u}_0 = \boldsymbol{r}$, $\mathbf{B}_0 = I$, and iteration index $k = 0$, the adjoint-based ILC with Hessian estimate can be described in the following steps:*

1. *conduct experiment with $\boldsymbol{u}_k$, and compute $\boldsymbol{e}_k = \boldsymbol{r} - \boldsymbol{y}_k$.*
2. *superimpose time-reversed error sequence $\mathcal{T}\boldsymbol{e}_k$ on $\boldsymbol{u}_k$, and conduct experiment as shown in Fig. 3.4 to obtain filtered error vector $\mathbf{G}^T \boldsymbol{e}_k$.*
3. *compute $\mathbf{B}_k$ using Eqn. 3.15, conduct experiment as in step 2 to experimentally determine $\mathbf{GB}_k v_k$, and compute learning gain $\varepsilon_k = ||\boldsymbol{v}_k||^2_{\mathbf{B}_k} / ||\mathbf{GB}_k \boldsymbol{v}_k||^2$.*
4. *compute ILC update $\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \varepsilon_k \mathbf{B}_k \mathbf{G}^T \boldsymbol{e}_k$.*
5. *terminate iteration if stop criterion is met, otherwise increment iteration index $k \leftarrow k+1$ and go to step 1.*

3.2.4 Comparison of SISO Data-driven ILC Algorithms

Shown in Table 3.1 is the comparison of SISO data-driven ILC algorithms. The adaptive ILC is inversion-based in nature and thus gives fast convergence. This is done by doing interleaving system inverse identification between ILC iterations with efficient use of time-varying FIR coefficients, as opposed to the Quasi-Newton method whose computation in inverse Hessian update grows with the trajectory size N.

Table 3.1: Comparison of SISO data-driven ILC algorithms. The methods "Adaptive" and "Adaptive (Improved)" are original contribution of this work in Sec. 3.2.2. "Gradient" and "Quasi-Newton" are the adjoint-based methods in Sec. 3.2.3 based on the work by [BKO18].

Method	Adaptive	Adaptive (Improved)	Gradient	Quasi-Newton
Type	Recursive	Recursive	Batch	Batch
Objective	$\min_{F_L} \|(M - F_L G)w\|_2$	$\min_{F_R} \|(M^* - F_R^* G^*)w\|_2$	$\min_{\boldsymbol{u}_{k+1}} \boldsymbol{e}_{k+1}^T \boldsymbol{e}_{k+1}$	$\min_{\boldsymbol{u}_{k+1}} \boldsymbol{e}_{k+1}^T \boldsymbol{e}_{k+1}$
No. Experiments for Error Filtering	1	1	1	2
System ID	Yes	Yes	No	For peak gain
Memory Usage in System ID	m	m	N/A	$N^2 + 2N$

3.2.5 Simulation and Experimental Results

3.2.5.1 Simulation Results

Simulated System: Fully-actuated Furuta Pendulum To validate and compare data-driven ILC algorithms in Sec. 3.2, a fully-actuated 2-DOF rotary pendulum is used as simulation test bed as in Fig. 3.5. The prescribed joint motion is a limited-jerk profile [EA01] with maximum speed of $\pi/3$ [rad/s]. The system is known for its rich dynamics, which is both nonlinear and state-dependent. The equation of motion can be written as follows:

$$\boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{D}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \mathcal{G}(\boldsymbol{q}) = \boldsymbol{\tau}, \tag{3.17}$$

wherein $\boldsymbol{q}$ represents joint angle vector, $\boldsymbol{J}(\boldsymbol{q})$ is the inertia matrix, $\boldsymbol{D}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ contains the Coriolis and damping term, $\mathcal{G}(\boldsymbol{q})$ is the gravitation, and $\boldsymbol{\tau}$ is the motor torque input vector.

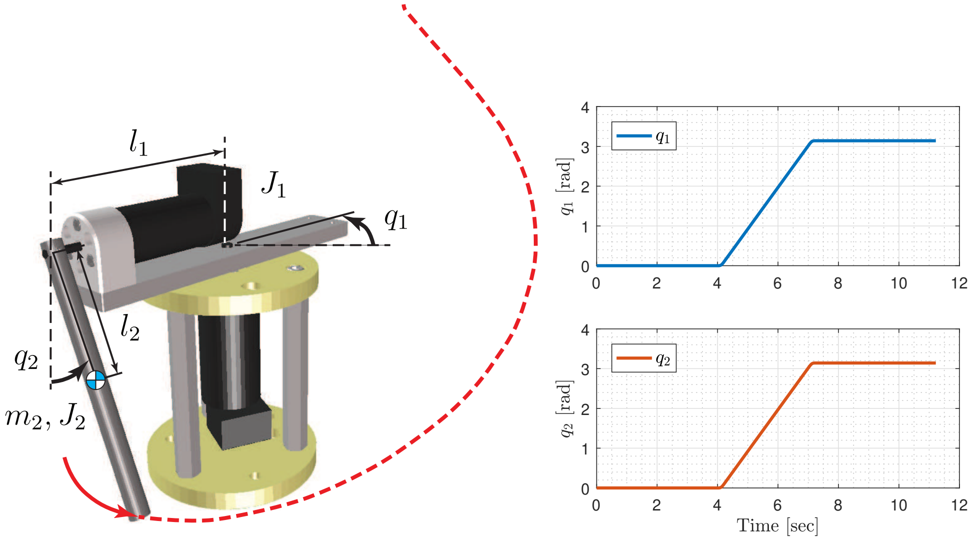


Figure 3.5: Task and joint domain trajectory of the 2 DOF robot. The joint motion is a jerk-limited profile with maximum speed of $\pi/3$ [rad/s].

These terms can be expanded as follows:

$$
\begin{aligned}
\boldsymbol{J}_{11} &= J_1 + m_2 l_1^2 + [J_2 + m_2 l_2^2]\mathrm{sin}(q_2)^2 \\
\boldsymbol{J}_{12} &= m_2 l_1 l_2 \mathrm{cos}(q_2) \\
\boldsymbol{J}_{21} &= m_2 l_1 l_2 \mathrm{cos}(q_2) \\
\boldsymbol{J}_{22} &= m_2 l_2^2 + J_2 \\
\boldsymbol{D}_{11} &= b_1 \\
\boldsymbol{D}_{12} &= m_2[l_2^2 \mathrm{sin}(2q_2)\dot{q}_1 - l_1 l_2 \mathrm{sin}(q_2)\dot{q}_2] + J_2 \mathrm{sin}(2q_2)\dot{q}_1 \\
\boldsymbol{D}_{21} &= -1/2[m_2 l_2^2 + J_2]\mathrm{sin}(2q_2)\dot{q}_1 \\
\boldsymbol{D}_{22} &= b_2 \\
\mathcal{G}_1 &= 0 \\
\mathcal{G}_2 &= m_2 l_2 g \mathrm{sin}(q_2),
\end{aligned}
$$

wherein the physical parameters are listed in Tab. 3.2. The system is actuated under 24 V by direct drive DC motors MMG 98023 (Maxon motor) and 28DT12 222E (Portescap) for base rotor and pendulum, respectively.

Table 3.2: Parameters for dynamic model construction.

Physical parameter	Notation	Value
Rotor total inertia	J_1	1.13×10^{-3} [kg$\cdot$ m^2]
Rotor arm length	l_1	0.103 [m]
Pendulum mass	m_2	2.42×10^{-2} [kg]
Pendulum inertia	J_2	3.22×10^{-5} [kg$\cdot$ m^2]
Pendulum length	l_2	0.122 [m]
Friction coefficient	b_1, b_2	10^{-4} [N$\cdot$m$\cdot$s rad^{-1}]

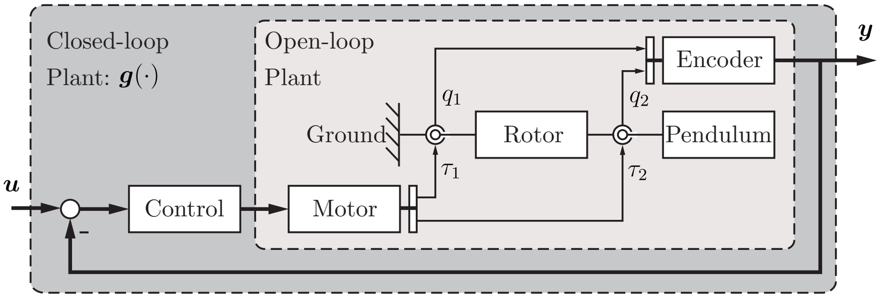


Figure 3.6: Simulated model: a continuous-time open-loop plant and a stabilizing discrete-time PID controller running at 100 Hz.

This continuous-time open-loop model is constructed in MapeSim (Maplesoft, Waterloo) and simulated in Simulink (Mathworks, MA). The signal connections between simulation blocks can be seen in Fig. 3.6: the motors apply torque (τ_1, τ_2) to two revolute joints, and the joint angle output (q_1, q_2) is fed back to two PID controllers running at 100 Hz to stabilize the plant so that the output $\boldsymbol{y} = [q_1, q_2]^T$ follows command reference $\boldsymbol{u}$. The PID gains for q_1 and q_2 are $(k_p^1, k_i^1, k_d^1) = (0.2, 0.1, 0.1)$ and $(k_p^2, k_i^2, k_d^2) = (0.6, 0.3, 0.1)$, respectively. The pre-stabilized plant $\boldsymbol{g}(\cdot)$ is used for ILC simulation.

System variation along the desired state trajectory $(\boldsymbol{q_d}, \dot{\boldsymbol{q}}_{\boldsymbol{d}})$ is shown in Fig. 3.7. It can be seen that the off-diagonal terms are 20 dB lower than the diagonal entries, and thus the SISO ILC can be used and the cross-coupling is respected as repeated disturbance. The

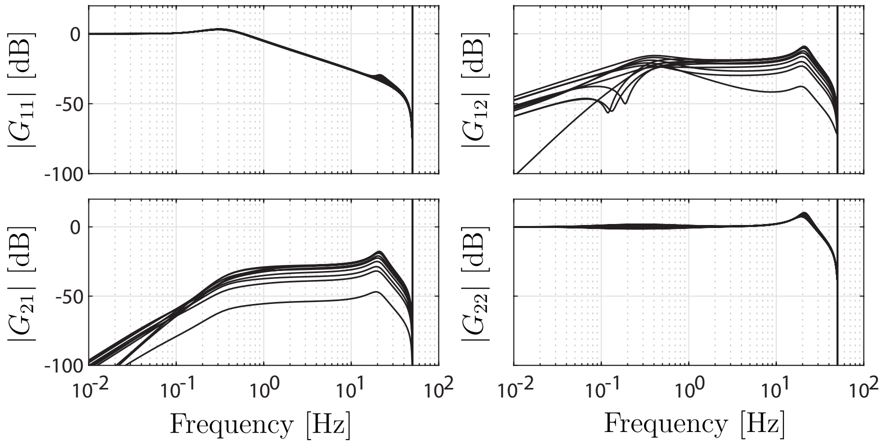


Figure 3.7: System variation along state trajectory $(\boldsymbol{q_d}, \dot{\boldsymbol{q}}_{\boldsymbol{d}})$. These magnitude plots are acquired from ten linearized models with equally spacing joint angles from 0 to π.

magnitude plot also serves to determine the largest possible learning gain $\bar{\varepsilon} = 2\|\mathbf{G}^T\mathbf{G}\|^{-1}$, or equivalently $\bar{\varepsilon} = 2\|G^*G\|_\infty^{-1}$, for the modified adjoint-based method in Algorithm 3.

ILC Algorithm Setup Herein the tracking performance for four different ILC algorithms listed in Table 3.1 are compared. One additional set of these algorithms are tested without the inclusion of u_k and $-y_k$ as those in step 2) of Fig. 3.2 and Fig. 3.4, namely the nonlinearity is not properly addressed. This serves to illustrate the nonlinear effect when filtering the error signal.

The adaptive filtering algorithm used here is a rotation-based recursive least square as in [RB91], which is known for low computational complexity, good numerical robustness, and fast convergence. The lattice filter structure of [RB91] also gives desirable order-recursive property, namely the leading FIR coefficients retain optimality when the tap length is reduced or extended. The number of FIR coefficients used is $m = 25$. The reference models M_1 and M_2 are constructed by 8th-order Butterworth low-pass filters with bandwidth 4 Hz and 8 Hz, respectively. The same delay steps $d_1 = d_2 = 3$ are used in the reference models. The low-pass filters Q_1 and Q_2 are the same as their corresponding reference models. Intuitively, the pendulum drive has less inertia and thus higher bandwidth. The choice of M (or Q)

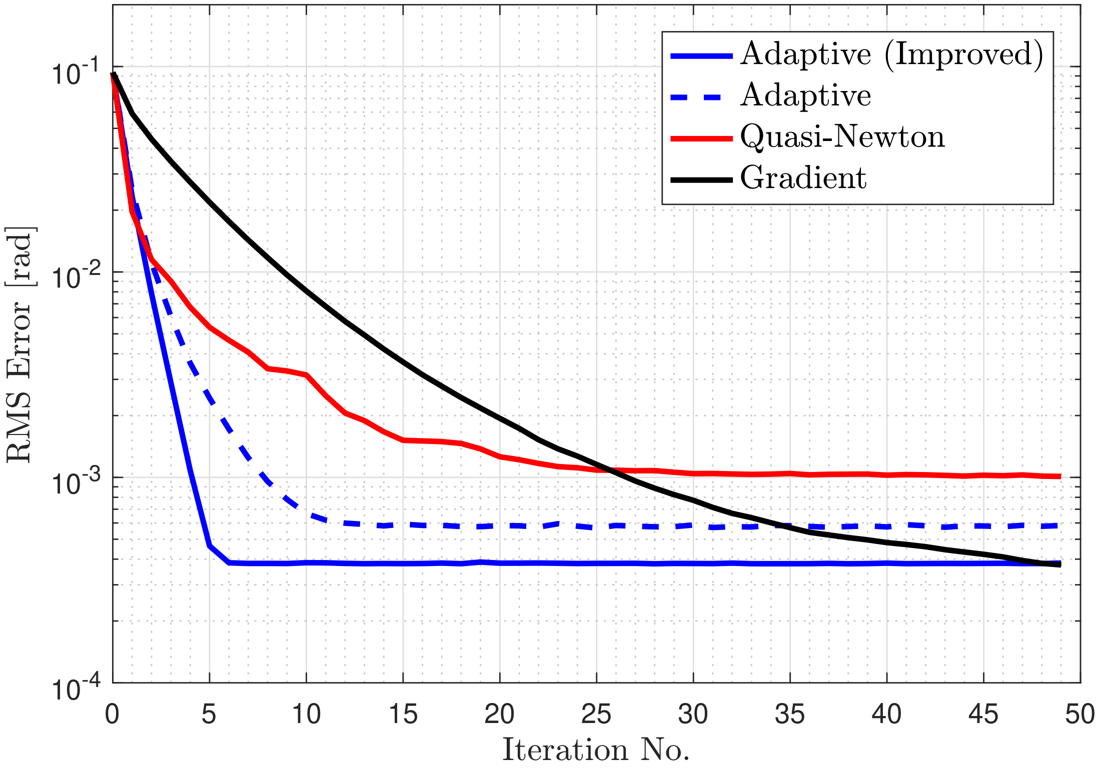


Figure 3.8: ILC Error convergence for nonlinear system in Fig. 3.5. **(i)** Adaptive (Improved): uses right inverse for improved error convergence as in Algorithm 2, **(ii)** Adaptive: proposed Algorithm 1 with adaptive filtering, **(iii)** Quasi-Newton: adjoint-based method as in Algorithm 4, **(iv)** Gradient: adjoint-based method as in Algorithm 3.

can also be experimentally estimated. For example, prior to algorithm execution step 2) in Fig. 3.2 can be run with $M = 1$, namely $||(1 - FG)w_k||_2$ is minimized. By comparing the power spectral density of w_k and $(1 - FG)w_k$, the validity of inversion, and thus bandwidth of M (or Q) may be inferred. The excitation signals w_1 and w_2 are generated from normally distributed random sequence with variance $\sigma_1 = \sigma_2 = 5 \times 10^{-2}$. As for the adjoint-based methods, the learning gains $\bar{\varepsilon}_1 = 0.95$ and $\bar{\varepsilon}_2 = 0.22$ are computed based on the maximum amplitude observed in Fig. 3.7. The quasi-Newton ILC is initialized with $\mathbf{B}_0 = \mathbf{I}$.

Fig. 3.8 shows the overall joint RMS error convergence of different data-driven algorithms

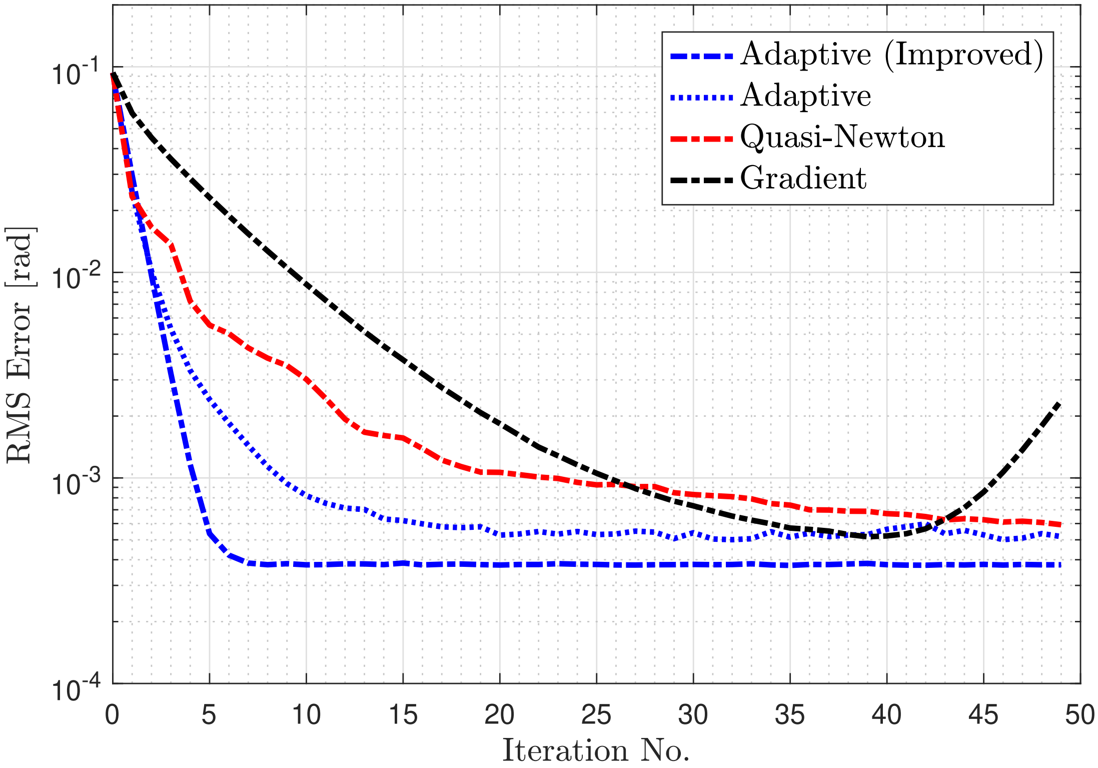


Figure 3.9: ILC error convergence when nonlinearity is not addressed, namely u_k and $-y_k$ are not included. **(i)** Adaptive (Improved): uses right inverse for improved error convergence as in Algorithm 2, **(ii)** Adaptive: proposed Algorithm 1 with adaptive filtering, **(iii)** Quasi-Newton: adjoint-based method as in Algorithm 4, **(iv)** Gradient: adjoint-based method as in Algorithm 3.

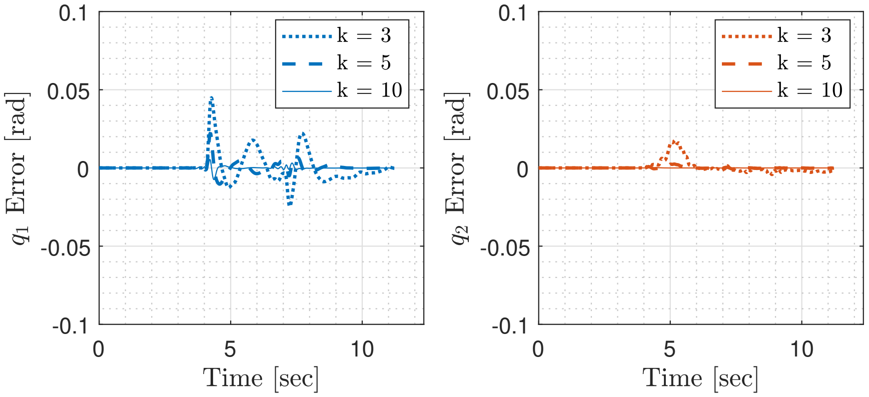


Figure 3.10: Error progression for the proposed adaptive ILC.

introduced. It is noted the proposed adaptive ILC (solid and dashed blue lines) reaches convergence much faster than the adjoint-based approaches owing to the effective inversion-based learning. For the gradient method in Algorithm 3 (black solid line), the convergence rate is much slower despite the maximum learning gain is used. The quasi-Newton method in Algorithm 4 (red solid line) takes up to 25 iterations for convergence. The convergence speed is not as fast as the adaptive ILC because the inverse Hessian estimate is constructed by signals $\boldsymbol{\Delta}_k = \boldsymbol{u}_k - \boldsymbol{u}_{k-1}$ and $\boldsymbol{\zeta}_k = \boldsymbol{v}_k - \boldsymbol{v}_{k-1}$ with $\boldsymbol{v}_k = 2\mathbf{G}^T \boldsymbol{e}_k$, and is thus limited by spectrum coverage of these signals.

In Fig. 3.9 when the nonlinear effect is not taken into account, the adjoint ILC (black dashdotted line) diverges due to the nonlinearity. Such nonlinearity is, however, not very strong because it resembles the convergence of the nonlinear case, and takes up to 40 iterations to start diverging. Similar degradation is also seen for the adaptive ILC (blue dashdotted and dotted lines) and quasi-Newton ILC (red dashdotted line), wherein the algorithm is able to maintain stability by approximating a nonlinear system with a linear filter, only with slower convergence.

In Fig. 3.11, 3.10, 3.12, and 3.13 we present the error progression of Algorithm 2, 1, 4, and 3, respectively. The proposed adaptive ILC algorithm has superior tracking performance compared to the adjoint-based algorithms. It is also noted the adjoint-based method has undesirable transient behavior at the end points. This is a consequence of the forward-

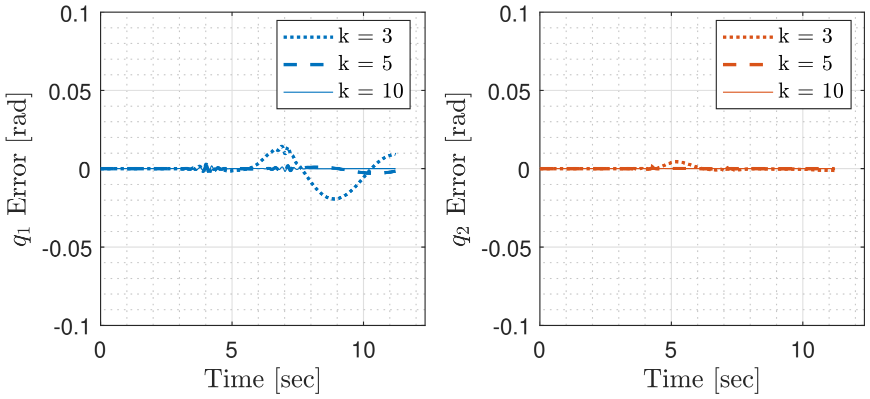


Figure 3.11: Error progression for the proposed adaptive ILC with improved error convergence.

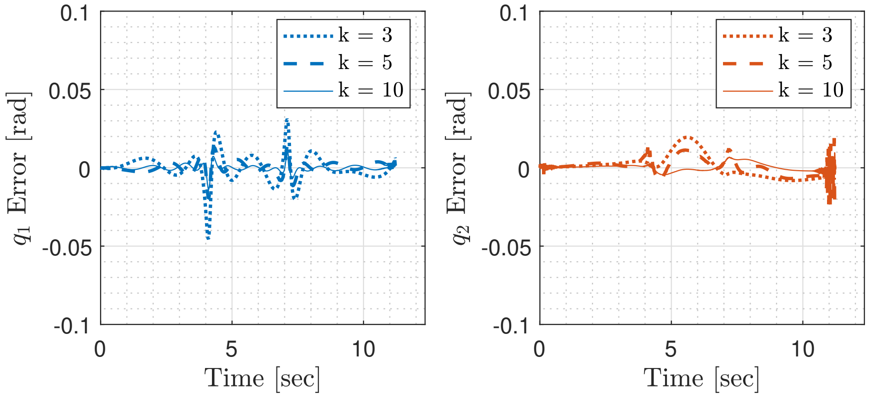


Figure 3.12: Error progression for the quasi-Newton ILC.

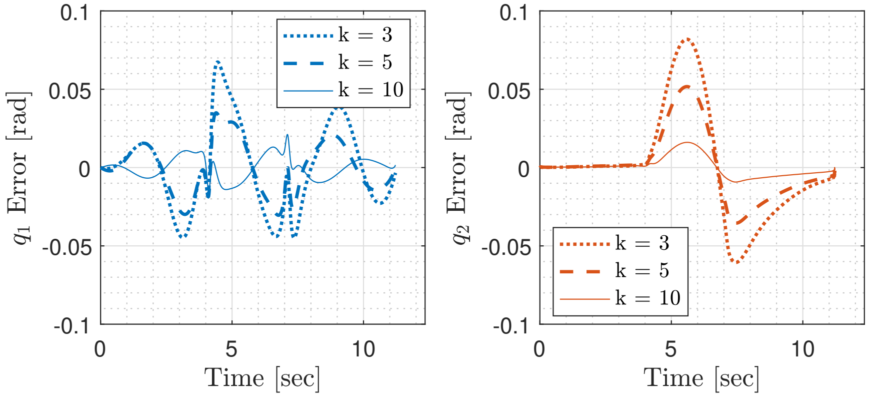


Figure 3.13: Error progression for the gradient ILC.

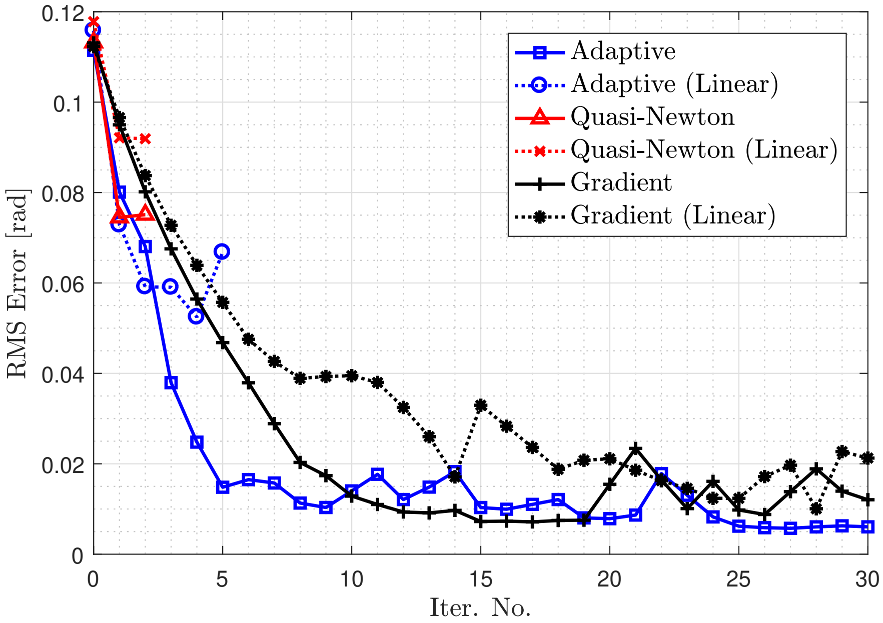


Figure 3.14: Shown is the ILC error convergence of the data-driven ILC algorithms. The experiments are run on the 2-DOF pendulum illustrated in Fig. 3.5.

backward filtering $\mathbf{G}^T\mathbf{G}$ when the padding at two ends are not sufficiently long, leading to mismatched initial conditions [Gus96], and it impedes the achievable performance level. Although this can be addressed by padding more zeros, the computation complexity increases accordingly for the accelerated adjoint-based method because the dimension of $\mathbf{B}$ grows with trajectory length N.

3.2.5.2 Experimental Results

For experimental verification, the fully-actuated 2-DOF pendulum system in Fig. 3.5 is built, and the adaptive ILC with improved error convergence is yet to be implemented. The resultant ILC convergence for different data-driven algorithms are shown in Fig. 3.14. The adaptive ILC outperforms other algorithms as expected, and the effect of nonlinearity can degrade the performance or lead to divergence as discussed before. Note also the quasi-Newton ILC triggers program stop due to excessive large command (comparable to the

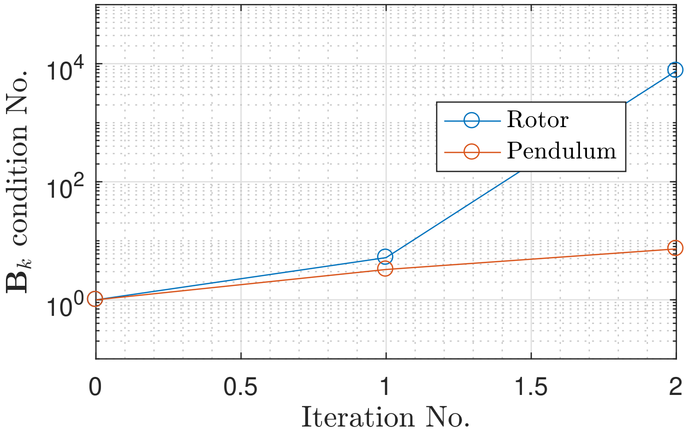


Figure 3.15: Shown is the condition number of $\mathbf{B}_k$ of accelerated adjoint ILC when static friction of q_1 is incorporated into the simulation model. Note the dramatic increase at iteration $k = 2$ for the rotor.

reference signal size) in step 3 of Algorithm 4, despite the nonlinearity has been addressed. The is caused by the large condition number of inverse Hessian estimate $\mathbf{B}_k$, thus leading to large signal $\mathbf{B}_k \boldsymbol{v}_k$.

The large condition number of $\mathbf{B}_k$ is due to the static friction in the low-velocity region. From the experiment, the smallest duty cycle to break the static friction of rotor q_1 and pendulum q_2 are approximately 0.01 and 0.005, respectively. For the sake of demonstration, we incorporate only the static friction of q_1 into the simulation model. In simulation result, similar behavior is observed: the sudden condition number increase is depicted in Fig. 3.15, and the excessive intermediate signal $\mathbf{B}_k \boldsymbol{v}_k$ is shown in Fig. 3.16. Also note the nonzero initial value of $\mathbf{B}_k \boldsymbol{v}_k$ of rotor and the subsequent active signal level.

This can also be explained from the BFGS update of Eqn. 3.15, which is rewritten as follows:

$$\mathbf{B}_k = \left(\mathbf{I} - \frac{\boldsymbol{\Delta}\boldsymbol{\zeta}^T}{\boldsymbol{\zeta}^T\boldsymbol{\Delta}}\right)\mathbf{B}_{k-1}\left(\mathbf{I} - \frac{\boldsymbol{\zeta}\boldsymbol{\Delta}^T}{\boldsymbol{\zeta}^T\boldsymbol{\Delta}}\right) + \frac{\boldsymbol{\Delta}\boldsymbol{\Delta}^T}{\boldsymbol{\zeta}^T\boldsymbol{\Delta}}. \tag{3.18}$$

The preservation of positive definiteness is under the assumption that $\boldsymbol{\zeta}^T\boldsymbol{\Delta} > 0$ (true for

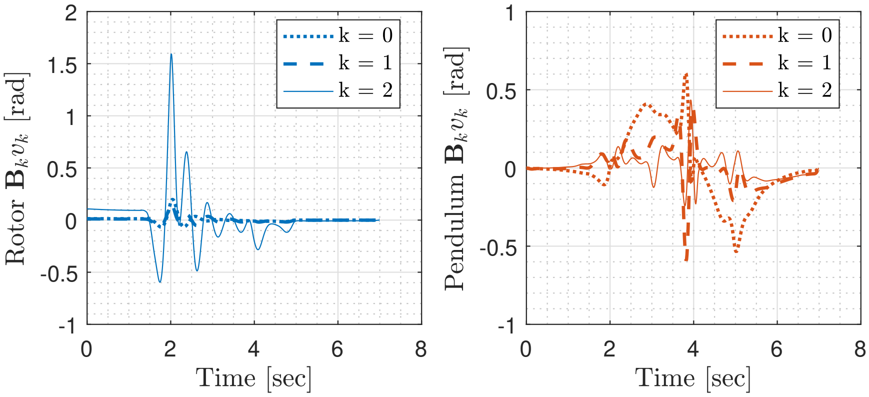


Figure 3.16: Shown is the resultant signal $\mathbf{B}_k \boldsymbol{v}_k$ from the ill-conditioned $\mathbf{B}_k$ in Fig. 3.15. Recall the reference step size is only $\pi \approx 3.1416$ [rad].

strictly convex objective). Recall the definition of $\boldsymbol{\Delta}_k = \boldsymbol{u}_k - \boldsymbol{u}_{k-1}$, $\boldsymbol{\zeta}_k = \boldsymbol{v}_k - \boldsymbol{v}_{k-1}$ with $\boldsymbol{v}_k = 2\mathbf{G}^T \boldsymbol{e}_k$: if the error does not change in accordance with the control update (due to friction), then $\boldsymbol{\zeta}^T \boldsymbol{\Delta} \approx 0$, resulting in excessive intermediate signal $\mathbf{B}_k \boldsymbol{v}_k$. One may mitigate the effect by scaling down $\mathbf{B}_k \boldsymbol{v}_k$, however, the problem of having $\boldsymbol{\zeta}^T \boldsymbol{\Delta} \approx 0$ can still cause data overflow. In [BKO18], it is recommended to superimpose the reference with a small constant speed to avoid such region. This strategy would require additional mending for a general motion profile, and may be applied to nonlinear system because superposition principle does not hold.

3.3 Data-driven ILC for nonlinear MIMO Systems

In the case when coupling effect can longer be ignored, now we investigate the data-driven ILC algorithms for MIMO systems. We will start with the MIMO adjoint ILC because the adjoint operator will later be used for MIMO adaptive ILC.

3.3.1 MIMO Adjoint ILC

The MIMO adjoint ILC for nonlinear system is based on [BKO18], and interested readers are referred to the original work developed for MIMO LTI systems. Here we will introduce the necessary materials to explain the extended algorithm. Specifically, we will show how to obtain filtered error $G^* e_k$ where G^* is conjugate transpose of the linearized system G. Note $G = G(t, z^{-1}) \in \mathbb{C}^{n_o \times n_i}$ is a transfer function matrix with n_i inputs and n_o outputs.

From [BKO18], the adjoint G^* is equivalent to

$$G^* = \mathcal{T}_{n_i} \left(\sum_{i=1}^{n_i} \sum_{j=1}^{n_o} E^{ij} G E^{ij} \right) \mathcal{T}_{n_o}, \tag{3.19}$$

where $\mathcal{T}_n$ is a time-reversal operator that reverses n signal sequences, and $E^{ij} \in \mathbb{R}^{n_i \times n_o}$ is a static system with n_i outputs and n_o inputs. All elements in E^{ij} are zero except $(i, j)^{th}$ entry, which is one:

$$E^{ij} = \begin{bmatrix} 0_{(i-1)\times(j-1)} & 0_{(i-1)\times 1} & 0_{(i-1)\times(n_o-j)} \\ 0_{1\times(j-1)} & 1 & 0_{1\times(n_o-j)} \\ 0_{(n_i-1)\times(j-1)} & 0_{(n_i-1)\times 1} & 0_{(i-1)\times(n_o-j)}. \end{bmatrix} \tag{3.20}$$

In $y = G E^{ij} u$, E^{ij} selects the j^{th} entry of u and apply it to the i^{th} input of G whereas the rest of the inputs to G are zero. The time-reversal operators $\mathcal{T}_{n_i}$ and $\mathcal{T}_{n_o}$ deals with conjugation, while the summation $\sum_{i=1}^{n_i} \sum_{j=1}^{n_o} E^{ij}(\cdot)E^{ij}$ transposes the system.

The objective is to compute $G^* e_k$ by performing experiments on G. From Eqn. 3.3.1, the indirect evaluation of G^* is recast into $n_i \times n_o$ experiments. This procedure is illustrated in step 2) of Fig. 3.17. For $(i, j)^{th}$ iteration, j^{th} output error is reversed then superimposed on i^{th} input, and the corresponding output at j^{th} output is collected after subtracting y_k. The i^{th} input update is summed over $j = 1, \cdots, n_o$ iterations. A 2-by-2 ($n_i = n_o = 2$) example is shown in Fig. 3.18 to better demonstrate the procedure.

Quasi-Newton Method For computing the inverse Hessian estimate $\mathbf{B}_k$ and the maximum learning gain $\overline{varepsilon}$, the formulas are similar to Eqn. 3.15 and Eqn. 3.16. The only difference is that $\boldsymbol{\Delta}_k = \text{vec}(\boldsymbol{u}_k) - \text{vec}(\boldsymbol{u}_{k-1})$, $\boldsymbol{\zeta}_k = \text{vec}(\boldsymbol{v}_k) - \text{vec}(\boldsymbol{v}_{k-1})$, where the vec($\cdot$) operator denotes concatenation of signals from all channels, and $\text{vec}(\boldsymbol{u}_k), \text{vec}(\boldsymbol{v}_k) \in \mathbb{R}^{n_i N \times 1}$.

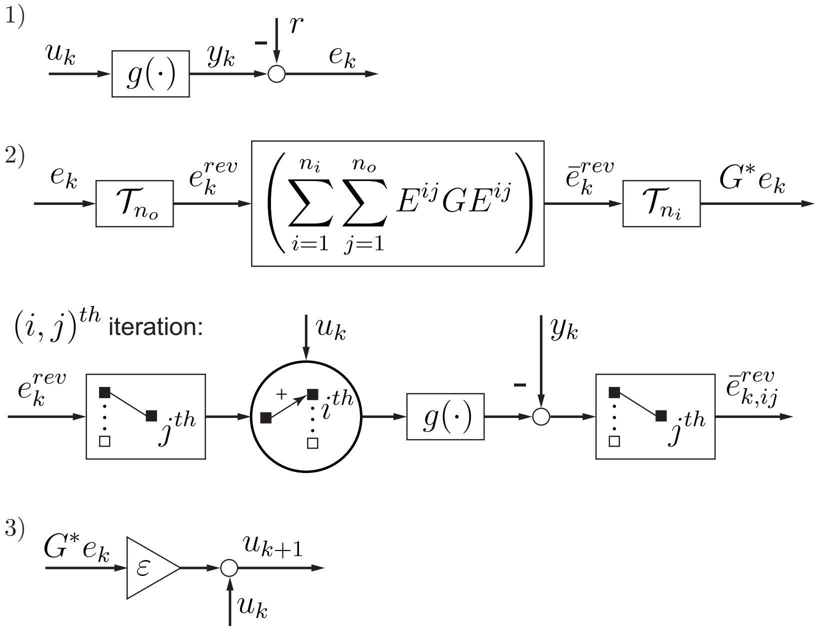


Figure 3.17: Modified MIMO adjoint-based algorithm: 1) Inject u_k to the system and compute tracking error $e_k = r - y_k$. 2) Error filtering using system adjoint G^*. Matrix $\mathcal{T}_n$ is a time-reversal operator on n channels. The filtered error for the i^{th} input channel is $(\bar{e}_k^{rev})^i = \sum_{j=1}^{n_o} \bar{e}_{k,ij}^{rev}$. In the $(i,j)^{th}$ iteration $\bar{e}_{k,ij}^{rev}$ is collected by adding the j^{th} channel reversed error e_k^{rev} to i^{th} channel input, and extracted from the j^{th} output after subtracting y_k. 3) The ILC command update for next iteration.

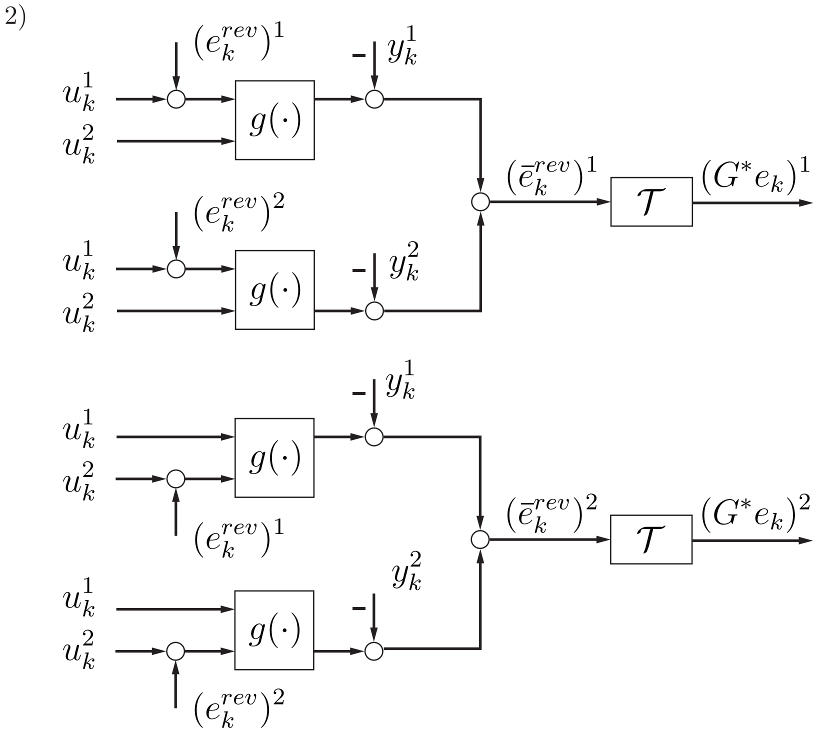


Figure 3.18: A 2-by-2 ($n_i = n_o = 2$) example illustrating step 2) of the modified MIMO adjoint-based algorithm. Notations $(\cdot)^1$ and $(\cdot)^2$ denote the first and second channel of a signal.

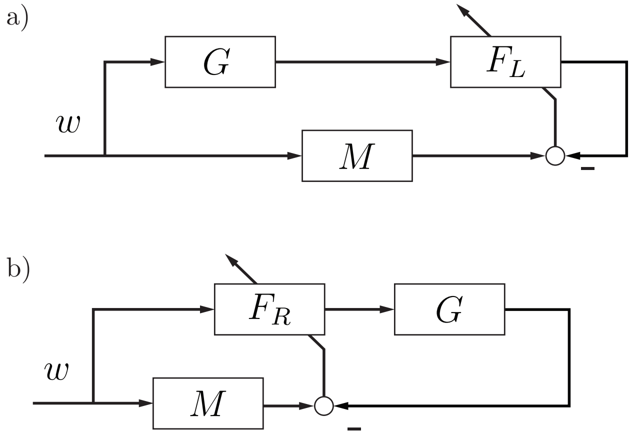


Figure 3.19: Adaptive filtering for (a) left inverse F_L by $\min\limits_{F_L} \|(M - F_L G)w\|_2$, and (b) right inverse F_R by $\min\limits_{F_R} \|(M - GF_R)w\|_2$.

3.3.2 MIMO Adaptive ILC

Technical Difficulty for MIMO Extension Non-commutativity is the main difficulty for extending adaptive ILC to MIMO systems. For systems $G(z^{-1}) \in \mathbb{C}^{n_o \times n_i}$ and $F(z^{-1}) \in \mathbb{C}^{n_i \times n_o}$, $FG \neq GF$ in general. From previous analysis in Sec. 3.2.2 and the method developed in Algorithm 2, it has already been shown using a right inverse F_R will result in better error convergence. This, however, causes a problem for the adaptive filtering because the right inverse cannot be adapted directly. As shown in Fig. 3.19, when adapting a F_R the unknown F_R is sandwiched between G and w in the objective $\|(M - GF)w_k\|_2 = \|Mw_k - GFw_k\|_2$, and cannot be solved conveniently.

In the following, we will investigate how to obtain filtered error signal $F_R e_k$ for fast error convergence. We will assume a linearized time-varying system response is available and the nonlinearity has been addressed.

Adaptation with Adjoint Operator From MIMO adjoint ILC we have learned how to obtain $G^* e_k$ from e_k, namely how to get a filtered signal through a conjugate transposed system by running $n_i \times n_o$ experiments. In adaptive ILC, we can apply the same technique to first obtain $G^* w_k$ from w_k. Then, adaptation is performed to minimize $\|(M^* - F_R^* G^*)w_k\|_2$

Table 3.3: Impulse response matrix for example system $[H(z^{-1})]$ with three inputs, two outputs, and two taps. The entry $[(\cdot_{ij})]_t$ denotes t^{th} tap coefficient from input j to output i.

$[h_{11}]_0$	$[h_{12}]_0$	$[h_{13}]_0$	$[h_{11}]_1$	$[h_{12}]_1$	$[h_{13}]_1$
$[h_{21}]_0$	$[h_{22}]_0$	$[h_{23}]_0$	$[h_{21}]_1$	$[h_{22}]_1$	$[h_{23}]_1$

with adaptive filter F_R^*. As our goal is to obtain $F_R e_k$, another $n_i \times n_o$ filtering steps is performed. This MIMO algorithm is illustrated in Fig. 3.20.

Adaptation with Fast Transpose Filtering The adaptive ILC with adjoint requires $n_i \times n_o$ experiments and $n_i \times n_o$ adaptations. Herein we make use of the "fast transpose" method proposed in [Ple01] to obtain $F_R e_k$ with single experiment and adaptation. The technique introduced here rely on a key observation: if the adaptation is run on the transposed system, namely minimizing $(M^T - F_R^T G^T)$ with respect to some signal, then the transpose of the adaptive filter will be the system right inverse because $(M^T - F_R^T G^T)^T \leftrightarrow (I - GF_R)$. This is illustrated in Fig. 3.21 with a reference model M to make it more general.

This efficient method makes a direct transposed copy of the adaptive filter instead of indirect signal manipulation. As a result, tedious experiments can be avoided. The transposed copy is through re-organization of the components of the impulse response matrices. This is demonstrated by a simple example as in [Ple01], which considers a system $[H(z^{-1})]$ with three inputs, two outputs, and two taps. The weight matrix are visualized in Tab. 3.3.

Taking the z transform, this can be written as:

$$[H(z^{-1})] = \begin{bmatrix} H_{11}(z^{-1}) & H_{12}(z^{-1}) & H_{13}(z^{-1}) \\ H_{21}(z^{-1}) & H_{22}(z^{-1}) & H_{23}(z^{-1}) \end{bmatrix}, \tag{3.21}$$

where each entry is a transfer function with two taps. The transposed system is then:

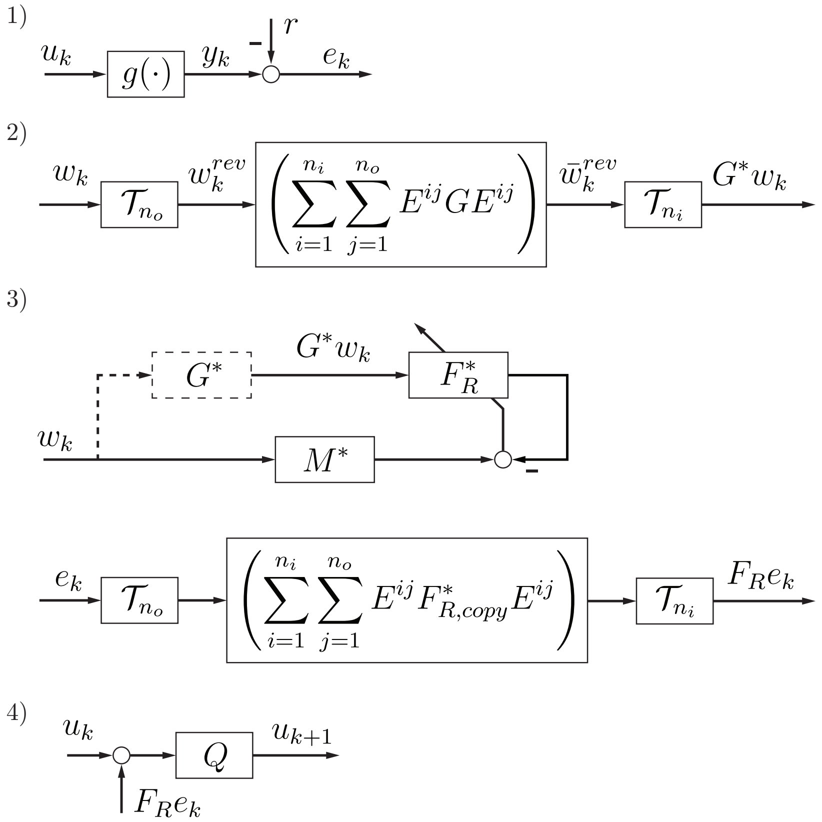


Figure 3.20: Proposed MIMO data-driven ILC with adaptive filtering: 1) Inject u_k to the system and compute tracking error $e_k = r - y_k$. 2) With w_k, make use of the same procedure described in MIMO adjoint-based algorithm (see step 2) in Fig. 3.17) obtain $G^* w_k$. Note $n_i \times n_o$ experiments are needed. 3) Error filtering utilizing adaptive filter: With w_k and $G^* w_k$, adapt F_R^*. Since the desired output is $F_R e_k$, again we make use of the adjoint by running $n_i \times n_o$ offline adaptations. 4) The ILC command update for next iteration.

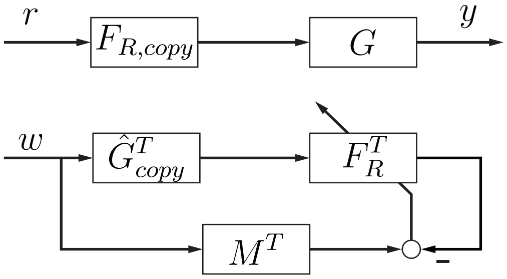


Figure 3.21: Adapting a MIMO linear controller F_R that minimizes $\|(M^T - F_R^T \hat{G}^T) w_k\|_2$, or equivalently, $\|w_k^T (M - \hat{G} F_R)\|_2$. Model $\hat{G}$ can either be offline determined or online identified.

Table 3.4: Impulse response matrix for system $[H(z^{-1})]^T$

$[h_{11}]_0$	$[h_{21}]_0$	$[h_{11}]_1$	$[h_{21}]_1$
$[h_{12}]_0$	$[h_{22}]_0$	$[h_{12}]_1$	$[h_{22}]_1$
$[h_{13}]_0$	$[h_{23}]_0$	$[h_{13}]_1$	$[h_{23}]_1$

$$[H(z^{-1})]^T = \begin{bmatrix} H_{11}(z^{-1}) & H_{21}(z^{-1}) \\ H_{12}(z^{-1}) & H_{22}(z^{-1}) \\ H_{13}(z^{-1}) & H_{23}(z^{-1}) \end{bmatrix}, \tag{3.22}$$

and the corresponding weight matrix can be visualized in Tab. 3.4.

Compared to the algorithm in Fig. 3.20, the adaption is with respect to $\hat{G} = G$ in the identification step 3) in Fig. 3.22. Therefore, performance degradation can be expected. It is also interesting to point out the fast transpose method has no effect on adapting a SISO right inverse because there is no channel permutation for SISO systems, whereas adjoint adaptation does have a SISO counterpart.

3.3.3 Comparison of MIMO Data-driven ILC Algorithms

Shown in Table 3.1 is the comparison of MIMO data-driven ILC algorithms. The fast convergence of the proposed adaptive ILC has been established from the inversion-based

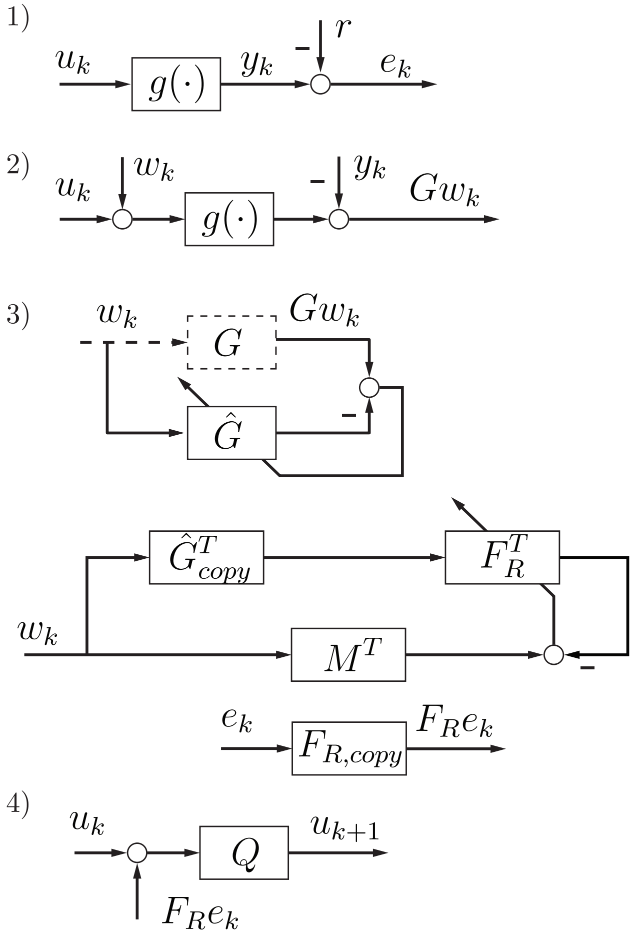


Figure 3.22: Proposed efficient MIMO data-driven ILC with adaptive filtering: 1) Inject u_k to the system and compute tracking error $e_k = r - y_k$. 2) Inject w_k to excite system dynamics around y_k, and collect linearized filtered excitation signal Gw_k by subtracting y_k. Note only single experiment is needed. 3) Error filtering utilizing adaptive filter: With w_k and $G^T w_k$, adaptively identify system estimate $\hat{G}$. Then, the transposed copy $\hat{G}^T_{copy}$ is used to adapt transposed right inverse F_R^T. The filter F_R^T is again copied and transposed to obtain $F_R e_k$. The transposed copy uses the fast transpose algorithm by I/O channel manipulation. Note only single adaptation is needed. 4) The ILC command update for next iteration.

Table 3.5: The methods “Adaptive”, “Adaptive (Improved)”, and “Adaptive (Efficient)” are original contribution of this work in Sec. 3.3.2. “Gradient” and “Quasi-Newton” are the adjoint-based methods in Sec. 3.2.3 based on the work by [BKO18].

Method	Adaptive	Adaptive (Improved)	Adaptive (Efficient)	Gradient	Quasi-Newton
Type	Recursive	Recursive	Recursive	Batch	Batch
Objective	$\min_{F_L} \|(M - F_L G)w\|_2$	$\min_{F_R} \|(M^* - F_R^* G^*)w\|_2$	$\min_{F_R} \|(M^T - F_R^T G^T)w\|_2$	$\min_{\boldsymbol{u}_{k+1}} \boldsymbol{e}_{k+1}^T \boldsymbol{e}_{k+1}$	$\min_{\boldsymbol{u}_{k+1}} \boldsymbol{e}_{k+1}^T \boldsymbol{e}_{k+1}$
No. Experiments for Error Filtering	1	$n_i \times n_o$	1	$n_i \times n_o$	$n_i \times n_o + 1$
System ID	Yes	Yes	Yes	No	For peak gain
Memory Usage in System ID	$n_i n_o m$	$n_i n_o m$	$n_i n_o m$	N/A	$(n_i N)^2 + 2n_i N$

nature. Note the number of additional experiments needed is comparable to or much less than existing adjoint-based algorithms. The memory usage stays competitive because time-varying FIR coefficients are used.

3.3.4 Simulation Results

To demonstrate the MIMO algorithms, we revisit the 2-DOF laboratory pendulum example in Fig. 3.5. The parameter settings are the same as in Sec. 3.2.5.1 except $\bar{\varepsilon} = 0.22$ from the MIMO system peak gain, and $m = 20$ is used to distinguish the performance when less taps are used. Shown in Fig. 3.23 is the ILC error convergence. First note the superior performance of the proposed MIMO adaptive ILC with right inverse, which are labelled “Adaptive (Improved)” and “Adaptive (Efficient)”. Slow convergence results when a left inverse is used, labelld “Adaptive”. The performance degradation in “Adaptive (Efficient)” is attributed to the fact the adaption is on $\hat{G}$ rather than G, as shown in Fig. 3.22 step 2). The adjoint-based methods do not have sufficient system excitation, and thus the convergence speed is limited as in the SISO case.

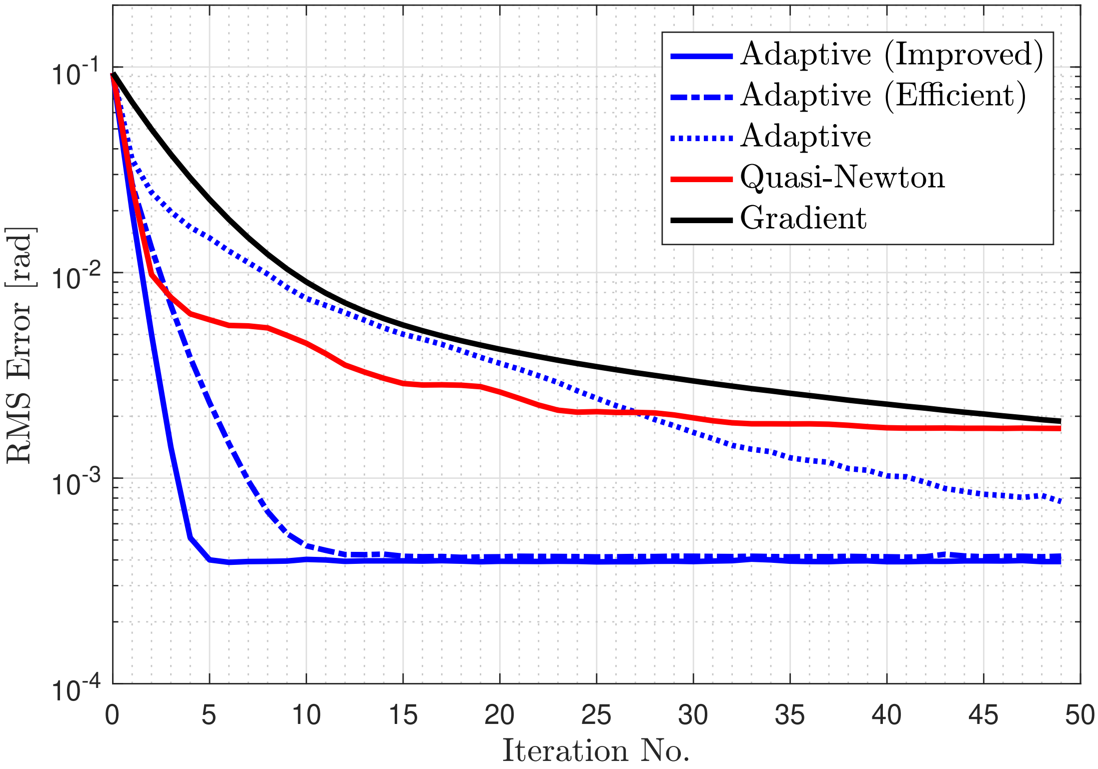


Figure 3.23: Error convergence for different MIMO data-driven ILC algorithms. **(i)** Adaptive (Improved): uses right inverse for improved error convergence with the scheme in Fig. 3.20, **(ii)** Adaptive (Efficient): uses right inverse for improved error convergence with the scheme in Fig. 3.22, **(iii)** Adaptive: use left inverse with the scheme in Fig. 3.19(a), **(iv)** Quasi-Newton: uses adjoint filtering in Fig. 3.17 with data-driven inverse Hessian, **(v)** Gradient: uses adjoint filtering in Fig. 3.17.

3.4 Conclusion

In this chapter we present a novel data-driven ILC algorithm, adaptive ILC, for nonlinear dynamical systems. Via the introduction of adaptive filtering, linearized time-varying inversion along trajectory facilitates the error filtering process in ILC without explicitly knowing the inversion. The adaptive ILC is fast converging, computationally tractable, and memory/resource efficient in contrast to gain-scheduled ILC methods. As a comparison to other data-driven algorithm, we also include an extension of the adjoint-based ILC algorithms. Through simulation and preliminary experiments on a fully actuated 2 DOF rotary pendulum, it is shown the adaptive ILC gives significant improvement in terms of error convergence. MIMO adaptive ILC algorithms are also studied, wherein two methods are proposed to circumvent the difficulty of directly adapting a right inverse.

Our ongoing tasks include the experimental validation of MIMO algorithms, design guidelines for the selecting excitation input, reference model, and low-pass filter, also a systematic study of algorithm robustness against non-smoothness, such as input deadzone and static friction.

CHAPTER 4

MRI-guided Robot-assisted Intervention

4.1 Image-guided Intervention with MRI

Image-guided intervention uses real-time tracking of target tissue and instrument to guide the surgical procedure in an accurate and timely fashion. Noninvasive imaging modalities, as opposed to direct line-of-sight visualization with cameras, can access imagery of internal structures, and enables minimally invasive procedure for fast recovery. Magnetic resonance imaging (MRI), amongst existing modalities, offers high contrast and image quality of soft tissues [TMT07]. It allows for multiplanar imaging, and can be run continuously because it is not subject to ionizing radiation exposure as X-ray computer tomography (CT).

Despite MRI has numerous advantages, in clinical practice the real-time imaging capability is not fully utilized for intervention. A major reason is the space restriction of the closed-bore scanner [KGM15], which demands the clinician to move the patient outside of the scanner for incremental instrument manipulation, and inside for confirmation several times for instrument positioning [WNL08]. The high field strength inside the scanner poses another difficulty for the development of the assisting device. Ferromagnetic materials are prohibited due to dangerous projectile motion from magnetic pull, and conductive materials could have undesirable heating due to eddy current effect. Moreover, if electric components are used, they could either inject noise into the scanner's receiver, leading to image degradation, or malfunction because of the field imposed by the scanner [GYC06, Fis15].

In response to these engineering challenges, MR safe or conditional robotic platforms are developed for accurate instrument manipulation under intraoperative MRI. Two main

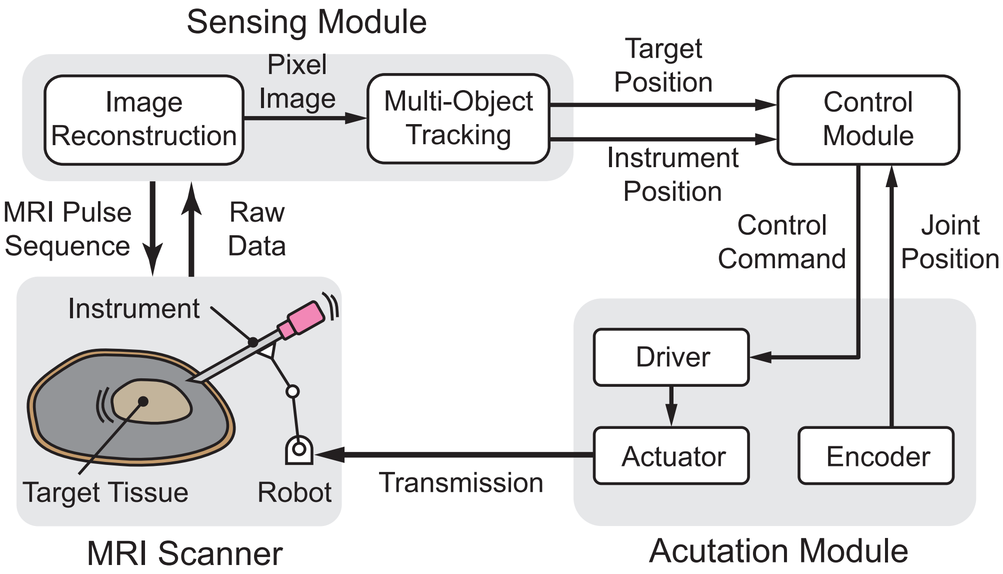


Figure 4.1: Shown is the system architecture for a general MRI-guided robotic system.

streams for the actuation of a robotic platform are ultrasonic motors (USM) and pneumatics. Benefits of USM include high torque output and precision, but comes with a price of signal degradation and shielding requirement [FKI08]. Pneumatic actuation features complete MR stealth, yet the performance is limited by the low stiffness and dependency on the line length. Recently, a promising alternative with faster system response is hydrostatic transmission [WGB14, MW19]. With the introduction of rolling diaphragm that circumvents the trade-off between sealing and friction in conventional piston-cylinder set, success has been reported for accurate needle placement [MSL16, MSL17, BFG17]. A thorough literature review of existing robots used in MR environment can be found in [AHF15, MCS18].

Although robotic platforms significantly augment the reach of a clinician, there are other practical issues. The target may be biased from the pre-procedural scan due to patient motion [Fis15], and the instrument can deviate from the desired pose due to instrument-tissue interaction and robot structural compliance. Moreover, MRI as a sensor is low frame rate (typically 5-20 Hz), subject to tracking error (variance 0.25-1.5 mm), and has delay (200-600 ms) from image reconstruction and processing, posing additional challenge for the instrument motion control. Recently, closed-loop motion control under continuous intraoperative MRI guidance has been investigated in [PKL15, MSK17, WSG18], however, it is assumed the target is almost stationary (e.g. prostate) and the dynamics only comes from the instrument control device. As involuntary and respiratory motion remain in organs such as livers and lungs, this prolongs the procedure, makes it error-prone for MRI-guided intervention. Such assumption also poses safety concern when the instrument goes to unintended region.

In this work, we aim at filling the gaps to enable real-time MRI-guided instrument motion control when target motion is present, as illustrated in Fig. 4.1. This is achieved via the following contributions:

1. The integration of real-time MRI tracking and motion control algorithm, setting a milestone for fully-automated image-guided intervention.
2. Modeling, analysis, and motion controller design for the dynamic tracking problem.
3. In MR environment, experimentally verify with an one-degree-of-freedom hydrostatically actuated platform, including a target motion module emulating respiratory motion and an instrument manipulation module following the target.

4.2 Hydrostatically-actuated Research Platform

Herein we present the one-degree-of-freedom hydrostatically actuated platform [SLCed] that allows for MRI-feedback motion control study inside the scanner. The platform, as depicted in Fig. 4.2, is composed of two modules: 1) an instrument manipulation module for intervention, and 2) a target motion module emulating tissue motion. The goal is to use MRI feedback and generate compensating signal on the instrument driver so that the relative position of instrument tip and a tracked point on the motion module end effector is maintained. The electronic components, including motors, power amplifiers, and micro-controllers are placed in the equipment room for MR safety. The remote actuation is via fluid hydrostatics as in [SLM17].

Shown in Fig. 4.3 is the instrument manipulation module. The double-acting fluid ac-

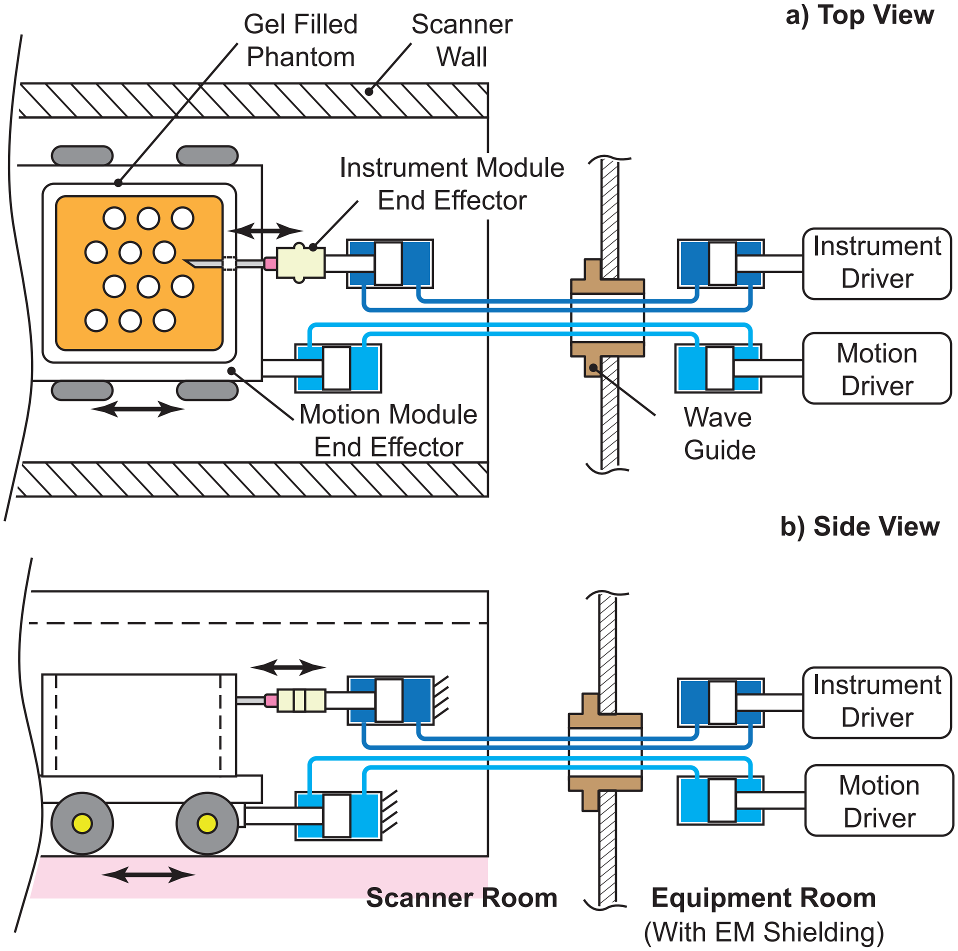


Figure 4.2: Schematic of the 1 DOF research platform comprising 1) an instrument manipulation module for intervention, and 2) a target motion module for motion emulation.

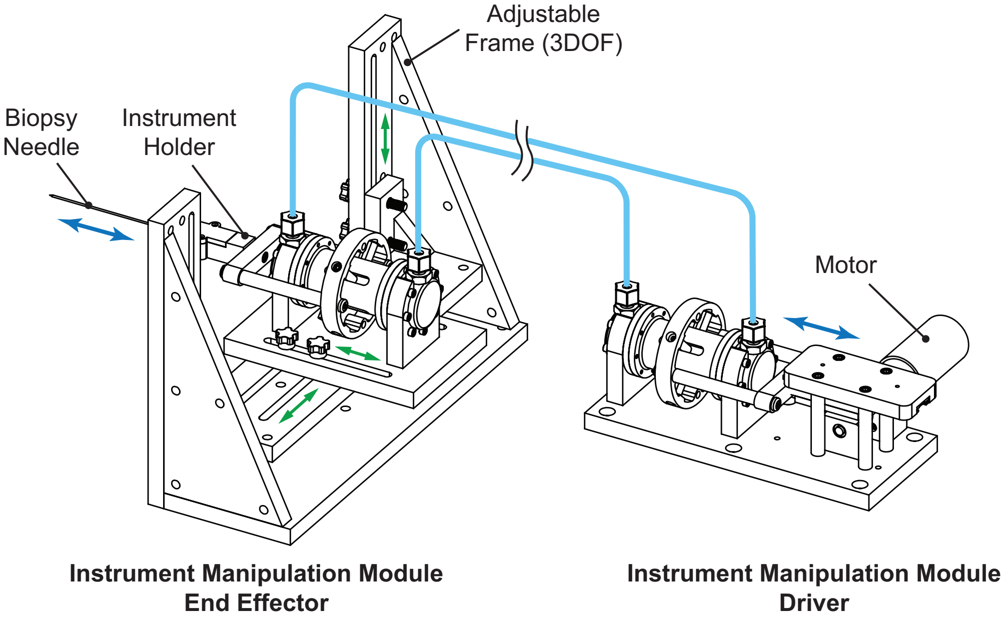


Figure 4.3: Instrument manipulation module. The insertion is controlled by a motor placed in the equipment room (actuated DOF marked in blue). The adjustable frame has three slots and thumb screws for instrument-target alignment in three translational DOFs (marked in green).

tuator uses rolling diaphragm as both sealing and guiding element. Detailed design and characterization can be found in [SLCed]. Through rack and pinion coupling to the motor output, the instrument (18 Gauge biopsy needle) tip can be modulated by the motion controller. The end effector of this module is mounted on a manually adjustable frame for initial alignment of feedback control study.

Shown in Fig. 4.4 is the target motion module based on the same fluid actuator. This module is used to reproduce reconfigurable and repeatable (0.31 mm precision) ground truth motion, e.g. patient breathing. The stroke length is 25.4 mm, and can be mechanically amplified via pulley mechanism as reported in [SLM17]. In [SLM17], it is shown that it can track profiles with different frequencies (0.2-0.85 Hz) with sub-pixel resolution of MRI (< 0.5 mm) via feedforward learning control. The end effector is coupled to a cart, which has four grooved wheels sitting on two round rails made of copper rods. Mounted on top of the cart is a acrylic phantom box containing a Delrin target plate. The phantom is filled with gel

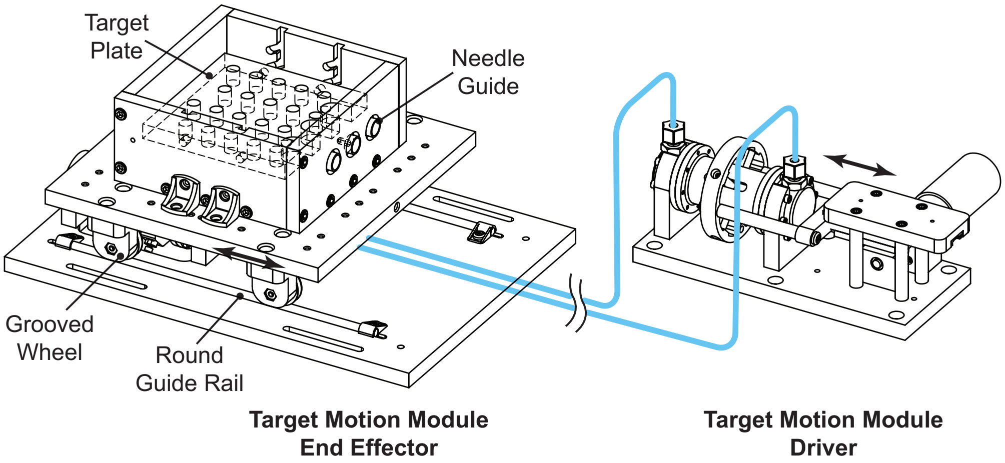

Target Plate
Needle Guide
Grooved Wheel
Round Guide Rail
Target Motion Module End Effector
Target Motion Module Driver

Figure 4.4: Target motion module. The emulated motion is replayed, and visualized by a gel-filled box containing a target plate. There are three needle guides on the side allowing instrument access inside the box. The platform sits on four grooved wheels and the motion is constrained by two round guide rails.

so the target plate is MR visible. Three plastic needle guides are on the side of the box to provide instrument access.

4.3 Sensing and Modeling of MRI

4.3.1 Multi-object Tracking

The multi-object tracking is implemented based on the real-time MRI pipeline established on a 3T MRI scanner (Prisma, Siemens Healthineers, Erlangen, Germany). The golden angle-ordered radial spoiled gradient echo sequence [LLM19] is applied to image the gel phantom on the target motion module in the coronal slice. Simultaneously, the needle inserting into the phantom is visualized passively due to the needle-induced susceptibility artifact. The real-time MRI data are streamed and reconstructed online using an open source reconstruction platform (Gadgetron) [HS13]. Then each image frame is passed into the image processing module to extract the needle motion and target motion on the phantom using template matching method as in [LLM19]. This is depicted in Fig. 4.19, and the motion coordinate

(X: left-right, Y: superior-inferior) represents the relative displacement of either needle or target to its location on initial image frame. A communication link (TCP/IP) is built between the image processing module and robotics control module at 10 Hz.

4.3.2 Modeling of MRI

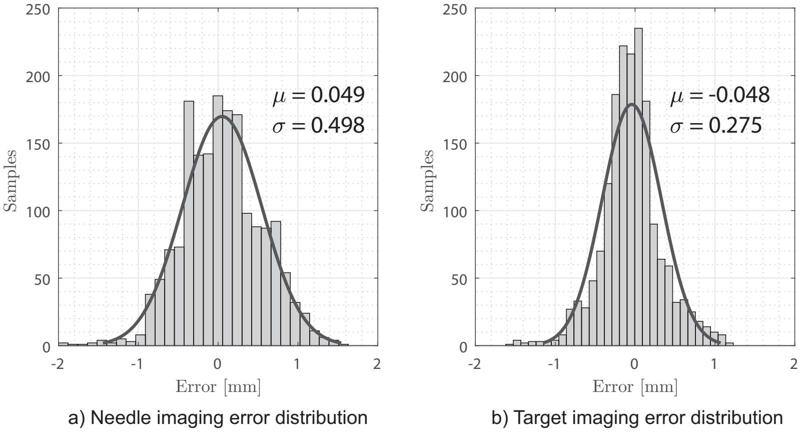
The image tracking process introduces processing delay, as well as tracking error as it deviates from the truth motion. As MRI feedback plays a critical rule in image-guided motion control, herein we develop an MRI model for high fidelity simulation when evaluating different control algorithms.

From experiment, the total MRI delay is determined to be 250 ms via temporal correlation, with 100 ms from image reconstruction and 150 ms from image tracking. This is cross-validated using a ground truth timestamp, generated by trigger signal converter (see [LLM19] for details). By comparing MR output to a truth measurement, the error can be regarded as the noise introduced by MRI. In Fig. 4.5, the error histograms suggest that they can be approximated as zero-mean normally distributed random processes. The MR tracking for needle has a larger variance because the feature size is smaller and more sensitive to background noise. To sum up, the MR imaging process can be modelled by a delay with additive noise, see Fig. 4.6 b). Note this is the image tracking induced noise, and should not be confused with the image background noise level.

4.4 Controller Design

The generic objective of MRI-guided intervention is to maintain a prescribed relative position between a target and the instrument. We will consider the challenging case for which the needle tip is regulated at a specified point on a moving target, and design a instrument motion controller. The target motion is assumed to be rigid body translation as described in Sec. 4.3.

Herein a Proportional-Integral (PI) controller is first designed, serving as the baseline



a) Needle imaging error distribution

b) Target imaging error distribution

Figure 4.5: Shown is the needle and target imaging tracking error distribution ($v_{ndl} = y - y_{true}$ and $v_{trgt} = r - r_{true}$) with fitted Gaussian density function $\mathcal{N} \sim (\mu, \sigma)$. The error is the difference of MR measurement from a truth (laser encoder) measurement outside scanner.

performance. Then, various add-on structures based on Adaptive Inverse Control (AIC) [Wid87] are investigated. These algorithms are first tested in simulation environment, and then verified by hardware-in-the-loop emulation on the physical instrument manipulation module. In the following, the sampling rate is chosen to be 10 Hz as the first harmonic of normal breathing is approximately 0.1-0.2 Hz.

4.4.1 Baseline PI Control

The MRI feedback PI control is shown in Fig. 4.6 a), wherein P_{mot} is the motor dynamics, C_{mot} the motor stabilizing controller, P_{trans} the transmission dynamics, D^{MR} the MRI delay, v_{ndl} the needle MR noise, and C_{PI} the baseline controller to be designed. The derivative control term is discarded due to known high image tracking variance (see Fig. 4.5) of MRI.

To identify the open-loop model $G_{mot}P_{tran}$ for design purpose, the transport delay in P_{tran} is first measured to be 50 ms via temporal correlation. Using the prior knowledge of motor loop G_{mot} and transmission dynamics (see [GGB04]), system order can be estimated. Then, a grey-box model fitting using frequency response data (FRD) from sine sweeping

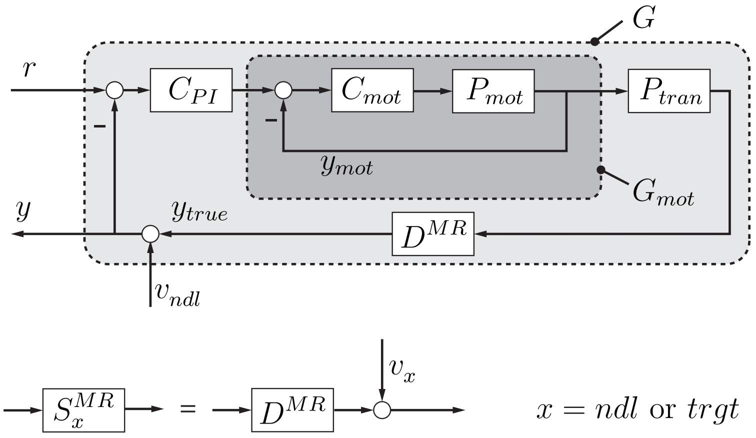


Figure 4.6: Shown in a) is the block diagram of the baseline PI control. G_{mot} is is referred to as "open-loop" and uses only motor encoder feedback. G is the outer loop that utilizes MRI feedback. Reference r can be predetermined, e.g. step input, or the target motion measured from the MRI. In b), the MRI processing (of either needle or target) is modelled as some delay with additive noise.

experiment is performed as in Fig. 4.7.

The PI control design shown here utilizes typical optimization technique. The cost function is the Integral Time Absolute Error (ITAE) $J = \sum_k k|e(k)|$ to penalize the steady-state error with respect to a step reference. The optimizer are proportional and integral gains, initialized by standard Ziegler-Nichols method. Gradient-descent method is used for cost minimization, and the iterative process is visualized in Fig. 4.8. The resultant controller has a gain margin of 6.43 dB and 58.8° phase margin.

To verify the result from offline design, the impulse response from PI design is compared to the MR experimental result (by one-step shift and subtraction of step response experiment), as shown in Fig. 4.9. The discrepancy is mostly attributed to transmission loss, either from mechanism or tubing, and is why an MRI feedback control is needed. Note the 300 ms delay due to MR image processing and transmission inherent in the system, which can result in significant phase error. In the following, add-on adaptive control will be employed to provide predictive action to enhance tracking performance. This will be added to the experimentally

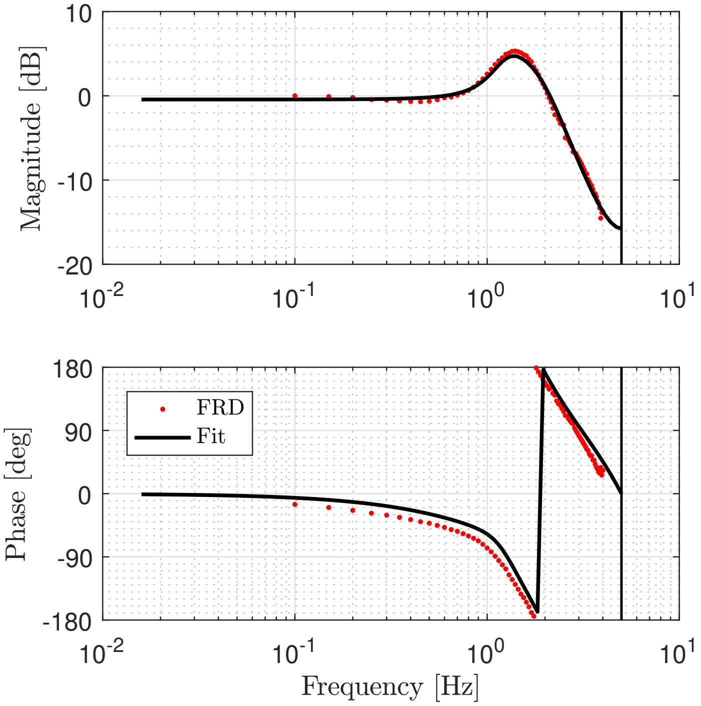
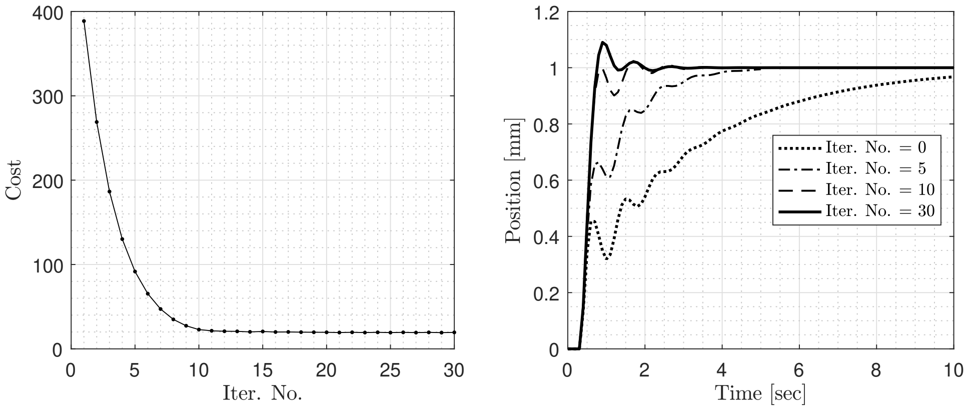


Figure 4.7: Frequency response of instrument manipulation module $G_{mot}P_{tran}$. Dots are FRD from sine sweep experiments, and the solid line is the fitted transfer function model.



a) Cost function convergence

b) Step response progression

Figure 4.8: Shown in a) is the learning curve of PI design using ITAE from an ideal step as cost function. In b), the progression of step response via PI tuning is presented.

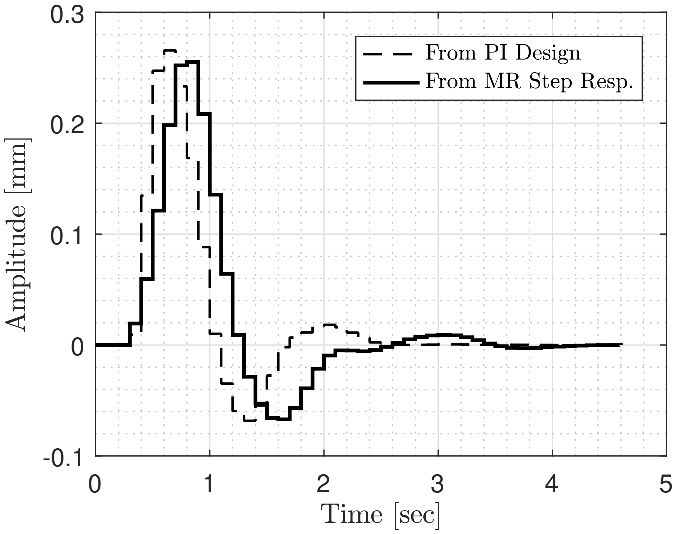


Figure 4.9: Shown is the impulse response predicted from PI design and the experimental result acquired using MRI feedback.

acquired Finite Impulse Response (FIR) model $\hat{G}_{ndl}^{MR}$ (solid line in Fig. 4.9).

4.4.2 Add-on Adaptive Control

The typical breathing motion is pseudo-periodic, and the center frequency of normal breathing (0.1-0.2 Hz) is unknown beforehand and may change over time [SRB91]. As there are also MRI delay and measurement noise, the dynamic target tracking problem becomes challenging. Herein we present three different adaptive control configurations similar to the study in [TCC12], and explains how they could adapt to the dynamic nature of the task using linear prediction.

First shown in Fig. 4.10 is the adaptive feedforward scheme. The adaptation is recursively solving the following:

$$\min_{C_{ff}} \quad ||H_{ff}(1 - C_{ff}\hat{G})y||_2 \tag{4.1}$$

wherein $\hat{G}$ is the the pre-stabilized closed-loop model, C_{ff} is a FIR filter whose tap weights

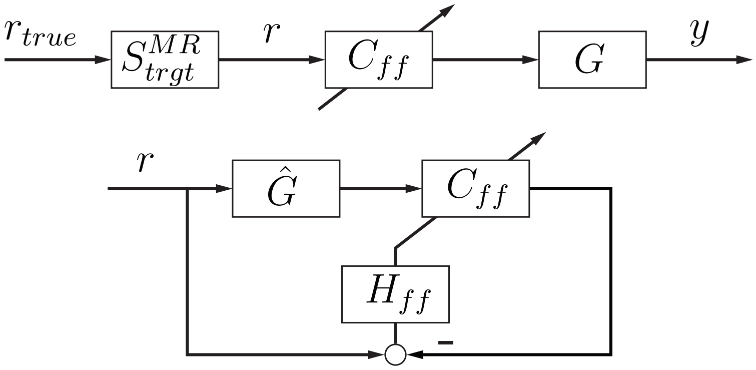


Figure 4.10: Block diagram of adaptive feedforward control.

are determined by the Recursive Least Squares (RLS) algorithm that minimize the error in inverting $\hat{G}$ with respect to the reference signal sensed r, and H_{ff} is a high-pass filter that penalizes the high-frequency contents of the control effort due to noisy r. This algorithm constructs a feedforward controller C_{ff} that best inverts the model regarding the reference history up to date. As the breathing profile is not purely stochastic, this provides some prediction capability in the dynamic tracking.

Shown in Fig. 4.11 is the adaptive feedback scheme. Again the RLS algorithm is used to minimize $||H_{fb}(1 - C_{fb}\hat{G})\eta||_2$, where the auxiliary signal $\eta = e + \hat{G}C_{fb}\eta$. With some algebraic manipulation this can be rewritten as:

$$\begin{aligned} &\min_{C_{fb}} \quad ||H_{fb}(1 - C_{fb}\hat{G})(1 - C_{fb}\hat{G})^{-1}e||_2 \\ &= \min_{C_{fb}} \quad ||H_{fb}e||_2 \end{aligned} \tag{4.2}$$

where C_{fb} is the feedback controller to be adapted, e is the needle-to-target error from MRI, w is the external disturbance (e.g. resistance from substrate medium), and H_{fb} is a frequency weighting that prevents excessively active control without introducing additional delays from linear-phase low-pass filtering, as is investigated in [PGT09]. In effect, the feedback controller C_{fb} uses signal η for predicting disturbance, breathing profile y, so that error e is minimized.

When these two channels are combined into a single controller, the structure of Fig. 4.12 is created. The transfer function from profile r and disturbance w to error e can be shown

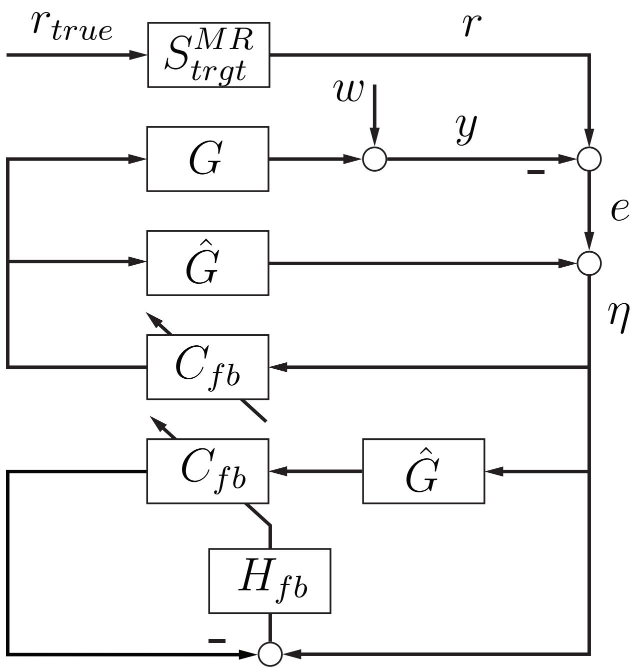


Figure 4.11: Block diagram of adaptive feedback control.

.

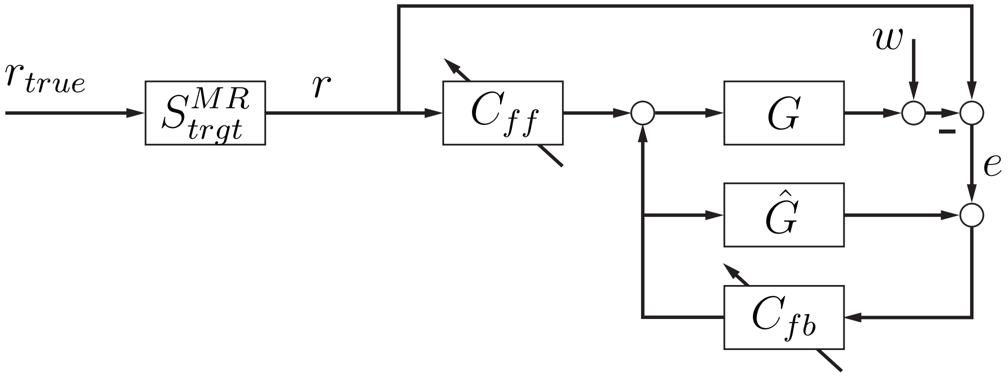


Figure 4.12: Block diagram of joint adaptive feedforward-feedback control.

to be

$$\begin{aligned} e = & \frac{(1 - C_{ff}G)(1 - C_{fb}\hat{G})}{1 + C_{fb}(G - \hat{G})} r \\ & - \frac{1 - C_{fb}\hat{G}}{1 + C_{fb}(G - \hat{G})} w. \end{aligned} \tag{4.3}$$

It can be seen that as long as $(1 - C_{fb}\hat{G})$ is small, the joint operation can achieve both reference tracking and disturbance rejection.

For the adaptive filter implementation, a rotation-based algorithm [RB91] is used with lattice filter structure. The order-recursive property, namely the leading FIR coefficients retain optimality when the tap length is reduced or extended, is tractable for filter order selection. A simulation example is shown in Fig. 4.13 where the order of C_{ff} in Fig. 4.10 is to be determined. The RMS value of true tracking error $e_{true} = r_{true} - y_{true}$ is shown as filter order changes. Clearly, the RMS error is decreasing as the filter order increases, as pointed out by the order-recursive property, and ten taps should be sufficient. The order of other configurations are determined similarly. For high performance, 20th order filter is used for all configurations.

In Fig. 4.14, we also demonstrate the importance of frequency weighting by simulating the effect of H_{ff} in Fig. 4.10. The high-pass weighting H_{ff} is chosen so because the frequency range for normal breathing is approximately 0.1-0.2 Hz. Because the the noisy content of y_{trgt}^{MR} in Eqn. 4.1 is suppressed, the control effort is mostly in the low-frequency region, resulting in smaller tracking error.

4.4.3 Hardware-in-the-loop Emulation

Our objective is to regulate the distance between a dynamic target and a controlled instrument. Considering the accessibility and setup cost in MRI suite, algorithms introduced in this section are first tested on the benchtop. This benchtop study includes the physical instrument manipulation module, with the end effector position y_{true} measured by a laser

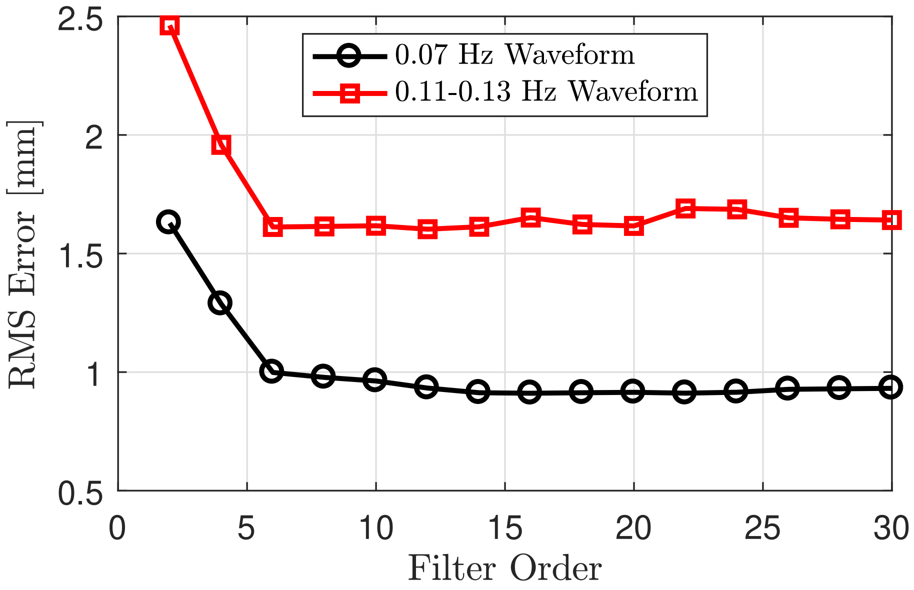


Figure 4.13: Simulating the effect of adaptive filter order of C_{ff} on tracking RMS error. Two waveforms, middle and bottom plots in Fig. 4.20 are tested. The fluctuation after 10^{th} order is due to the stochastic nature of the MRI noise.

encoder. The target motion r_{true} is artificially generated from pre-recorded breathing motion using high resolution offline image tracking. Both true target r_{true} and end effector y_{true} positions are corrupted via the MR image tracking model, delay plus additive Gaussian noise, as discussed in Sec. 4.3, to emulate MR output r and y. This hardware-in-the-loop emulation is illustrated in Fig. 4.15.

Shown in Fig. 4.16 is the tracking error history of the algorithms presented in this section. The reference profile tracked is the same as the middle plot in Fig. 4.20. The initial large error up to 15 second for adaptive controllers (FF, FB, and FF+FB) is from software protection, limiting the control effort so that input is not saturated in the learning phase. Here we focus on the steady-state performance, and note first the add-on adaptive controllers improve the baseline PI performance. Secondly, the adaptive feedback (FB) performs the worst amongst three adaptive schemes. This is because PI control has already used the noisy feedback y (see Fig. 4.6), and having another feedback with the same noisy measurement

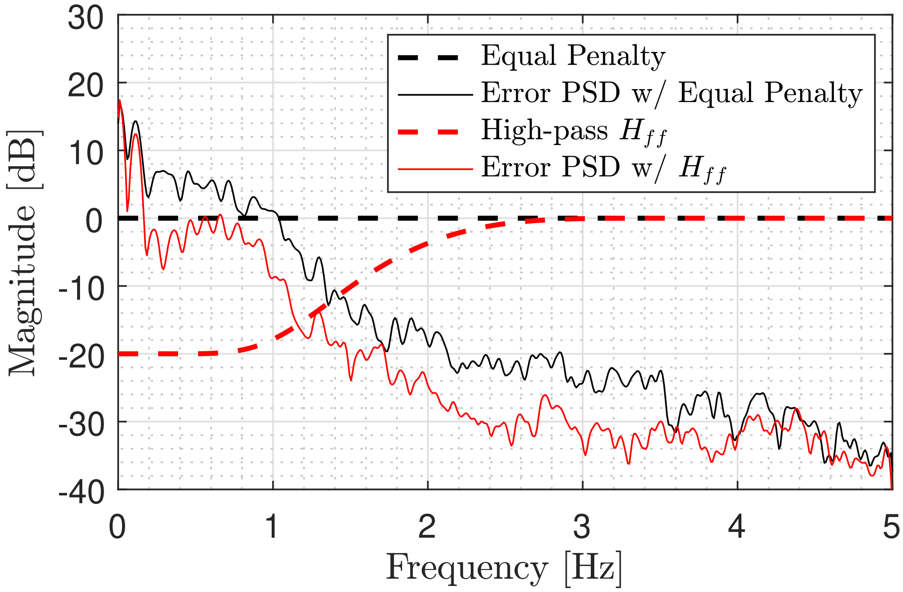


Figure 4.14: Simulating the effect of high-pass weighting H_{ff} on tracking error Power Spectral Density (PSD). Dashed lines are the frequency weighting functions, and solid lines are the corresponding error PSD. The error is effectively attenuated by penalizing the high frequency content from MRI noise.

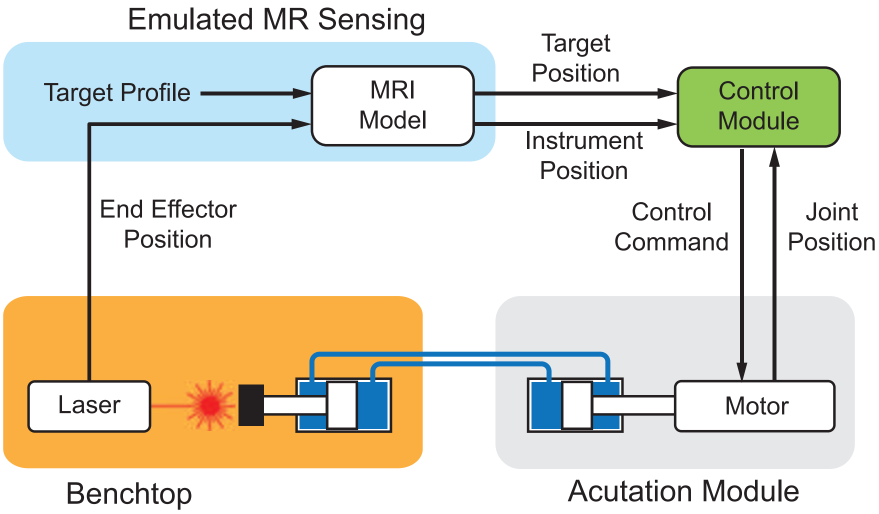


Figure 4.15: This shows the benchtop setup emulating MR feedback by corrupting a fictitious target reference and instrument end effector measurement using the MRI model described in Sec. 4.3.

(see Fig. 4.11) limits the improvement. This also explains why joint adaptive feedforward-feedback (FF+FB) does not do better than adaptive feedforward (FF) alone, as most of the low-frequency disturbance is already compensated by PI and the inclusion of feedback path introduces additional noise. In the next section, we will implement the most promising adaptive feedforward controller in MR environment.

4.5 Experimental Results

The system architecture is shown in Fig. 4.17. The instrument manipulation module and the target motion module described in Sec. 4.2 are controlled by National Instrument PXI-7833R and myRIO real-time targets, respectively. The end effector positions are sensed by the MR scanner, and the image is processed at 10 Hz on a host laptop that generates compensation signal, also at 10 Hz, to the instrument manipulation module. This host laptop also sends learned feedforward signal to the target motion module, which is independently controlled at 200 Hz, to reproduce pre-recorded breathing profile.

Shown in Fig. 4.18 is the test setup for motion synchronization of the dynamic target and the needle. The needle enters a plastic needle guide and is inserted into the gel-filled phantom. The tracked point on the target plate is the midpoint of the open slot aligned with the edge. It is chosen so that the needle tip always stays in the gel-filled region (MR visible) without colliding with the target plate. Detailed dimensions of the phantom can be found in Fig. 4.19, which also shows the pixel image with target and needle streamed from the MRI station.

The tracking results with the adaptive feedforward control (Fig. 4.10) for three different profiles are shown in Fig. 4.20. The experimental procedure is as follows: the real-time MR streaming is first established, then the target motion module is initiated with prescribed breathing profile, thereafter the adaptive control signal is gradually ramping in to prevent input saturation and large transient. The steady state performance is summarized in Tab. 4.1, which agrees well with the simulation results.

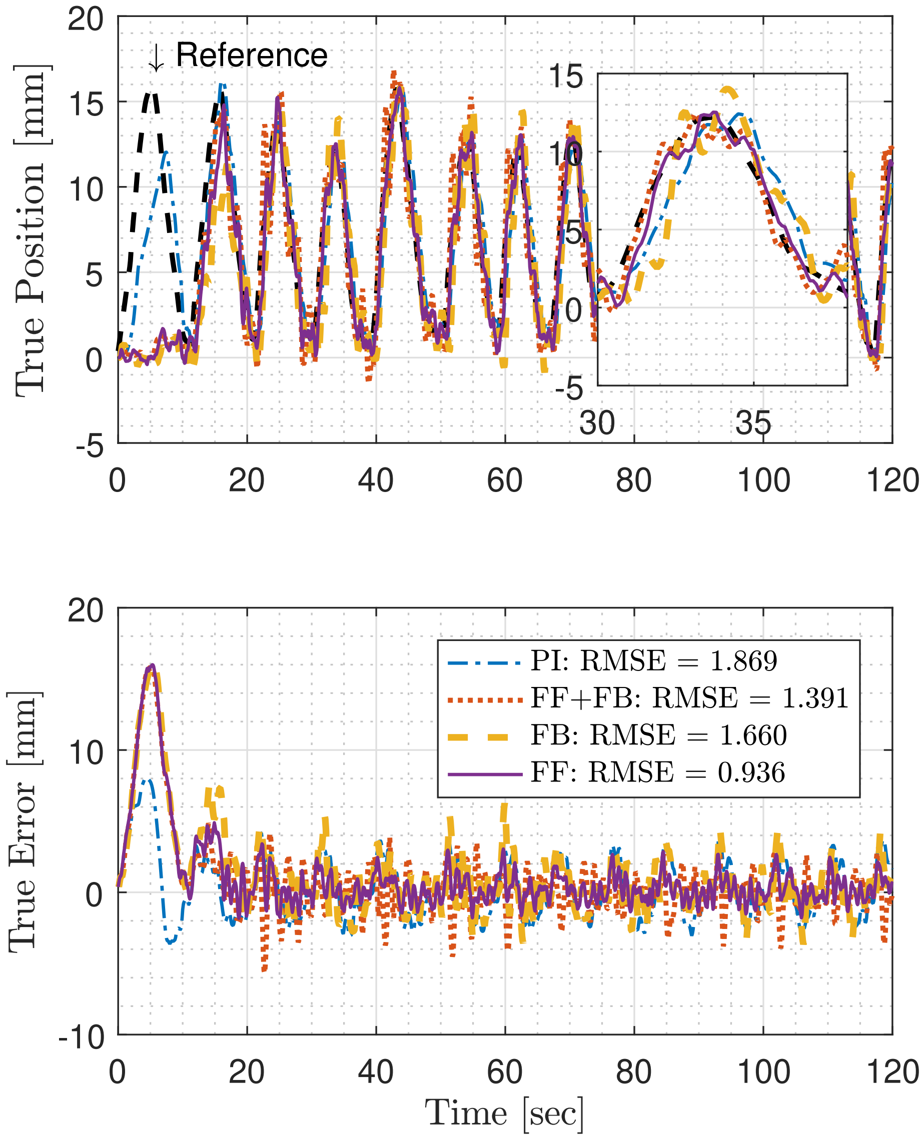


Figure 4.16: Shown is the benchtop tracking error $e_{true} = r_{true} - y_{true}$ history with the reference profile in the middle plot of Fig. 4.20. Algorithms discussed in Sec. 4.4 are examined. The RMS error is calculated for $t > 40$ seconds.

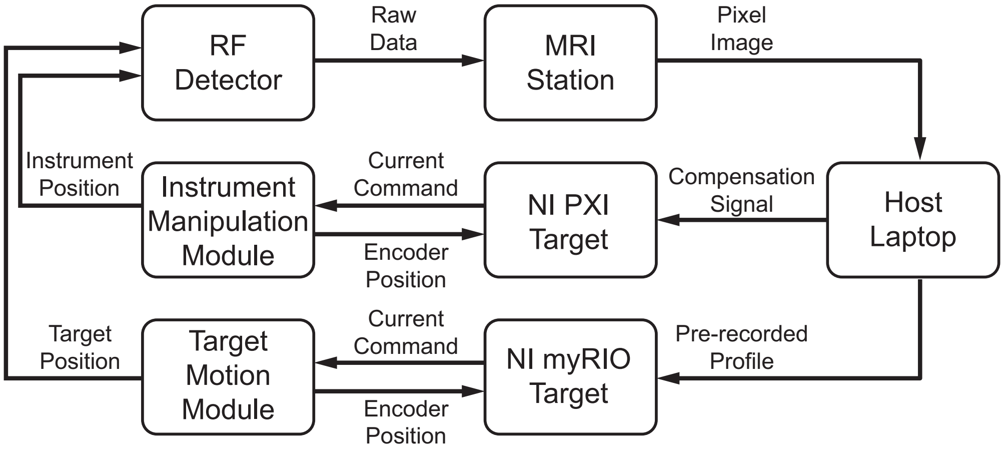
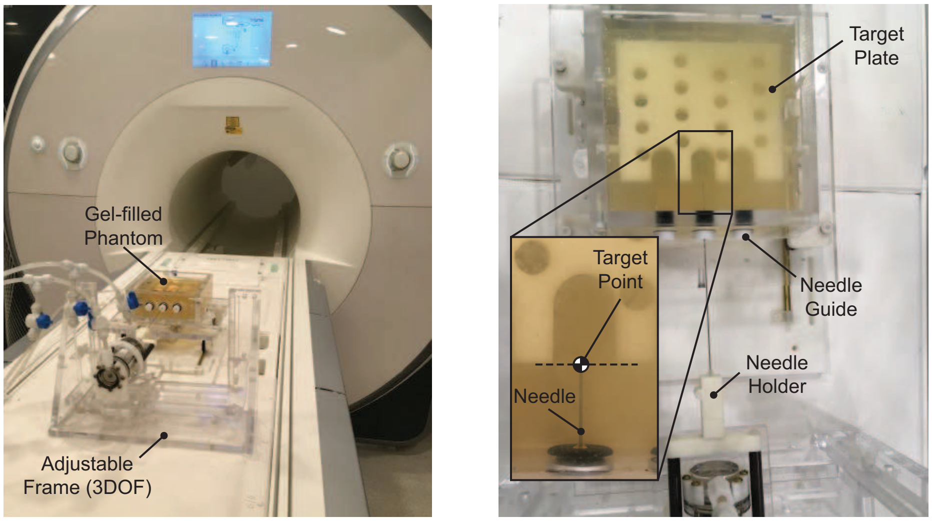


Figure 4.17: Shown is the system architecture for the MRI feedback motion control study.



a) End effector side of the hydrostatic actuated platform setup in MRI scanner

b) Top view of the needle-target tracking setup

Figure 4.18: The experimental setup for MRI feedback motion control study. The goal is to keep the needle tip at the target point at all times.

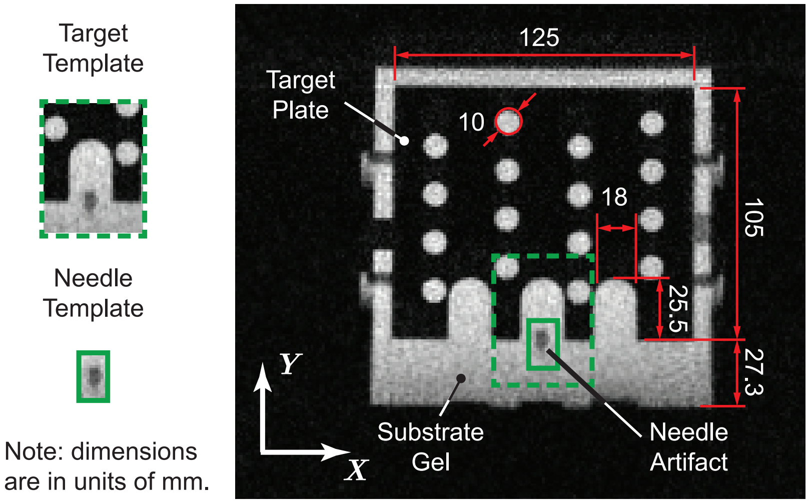


Figure 4.19: MR image for motion tracking using template matching. The templates used are shown, and the dimensions of the phantom is given. The dominant motion is in the Y direction.

Table 4.1: Needle tracking error statistics (calculated for $t > 60$ sec in Fig. 4.20) under closed-loop MRI feedback with adaptive feedforward control (see Fig. 4.10).

Profile Frequency [Hz]		Max. Error [mm]	Mean Error [mm]	RMS Error [mm]
0.07	Sim.	3.393	0.152	1.077
	Exp.	3.510	0.191	1.132
0.11-0.13	Sim.	4.431	0.138	1.165
	Exp.	4.095	0.139	1.197
0.17-0.19	Sim.	6.923	0.526	2.035
	Exp.	6.727	0.378	2.180

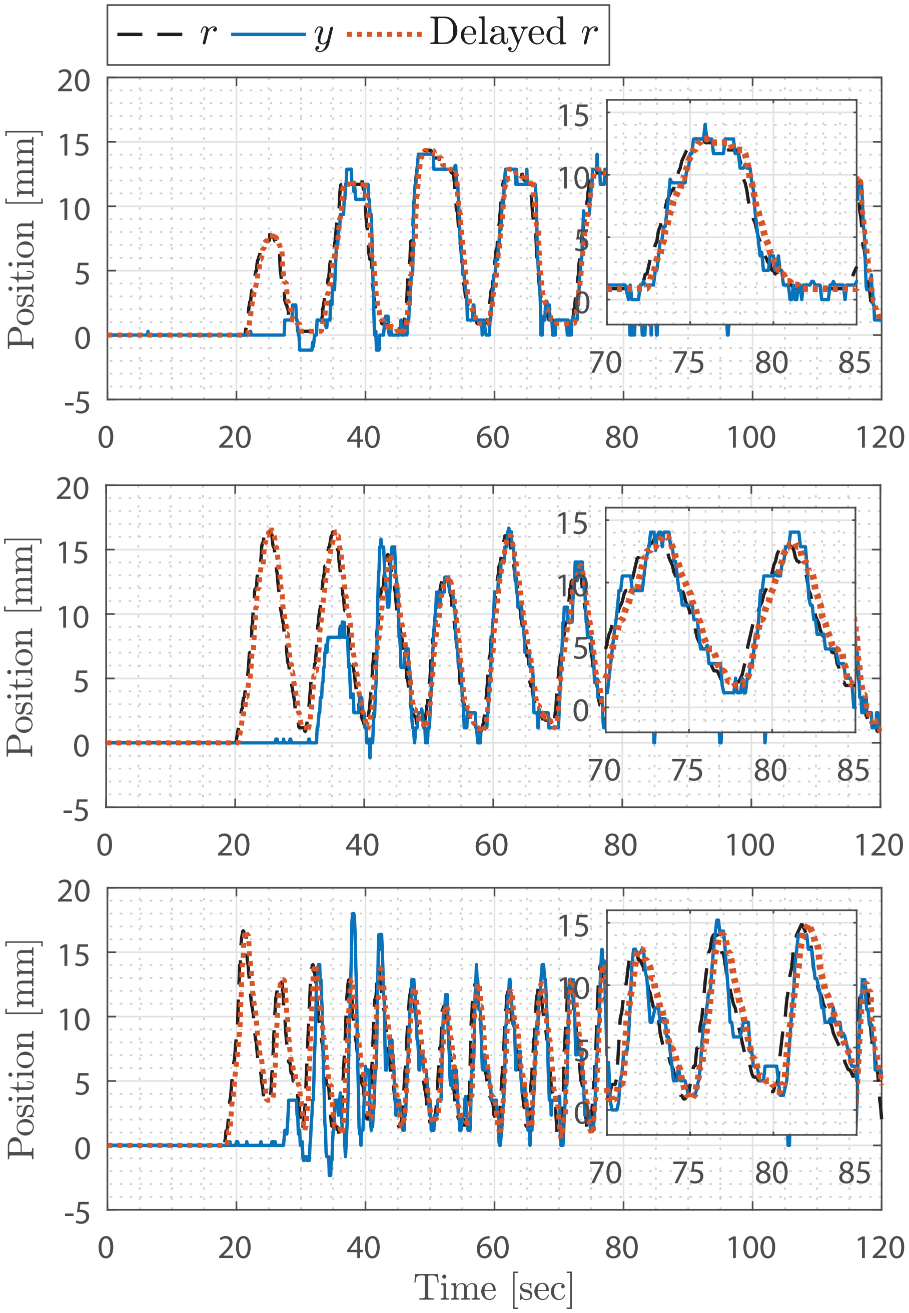


Figure 4.20: Tracking result y (in Y direction in Fig. 4.19) of adaptive feedforward control on 0.07 Hz (top), 0.11-0.13 Hz (middle), and 0.17-0.19 Hz (bottom) profiles, respectively. Delayed r is obtained by delaying signal r with system transmission and processing delays.

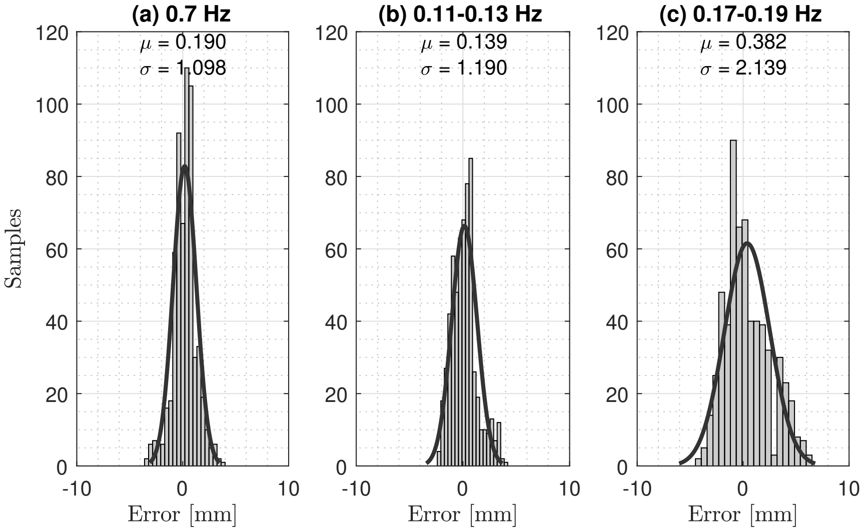


Figure 4.21: Error distribution of the steady-state ($t > 60$ s) tracking error in Fig. 4.20.

Shown further in Fig. 4.21 and Fig. 4.22 are the spatial distribution and power spectral density (PSD) of steady-state tracking error in Fig. 4.20. First notice how the spread widens as the profile becomes more challenging. Also note the prominent mode (0.2 Hz) in PSD for aggressive profile (0.17-0.19 Hz) due to the phase delay.

To demonstrate the effect of prediction from adaptive feedforward, a delayed reference signal (Delayed r) is also depicted in Fig. 4.20. This represents the performance of a perfect actuator with only processing delay. In Tab. 4.2, we see that the RMS error of adaptive feedforward is mitigated compared to the pure delay system, and is attributed to the predictive action.

To evaluate the feasibility for automated MRI-guided intervention, shown in Fig. 4.23 is the survival function with respect to the steady-state ($t > 60$ s) absolute tracking error from Fig. 4.20. In clinical practice, the smallest target size is about 5 mm diameter. For liver, lesion size less than 5 mm are usually benign [AND09]. Therefore, a 2.5 mm absolute error is acceptable and is drawn as the vertical red line. We see only a < 5% chance this requirement is not met for mild profiles, whereas it is about 15% for the aggressive profile (0.17-0.19 Hz). In the following, we will use our high fidelity simulation model to study how the limiting factors may be improved to make the performance comparable to those mild profiles.

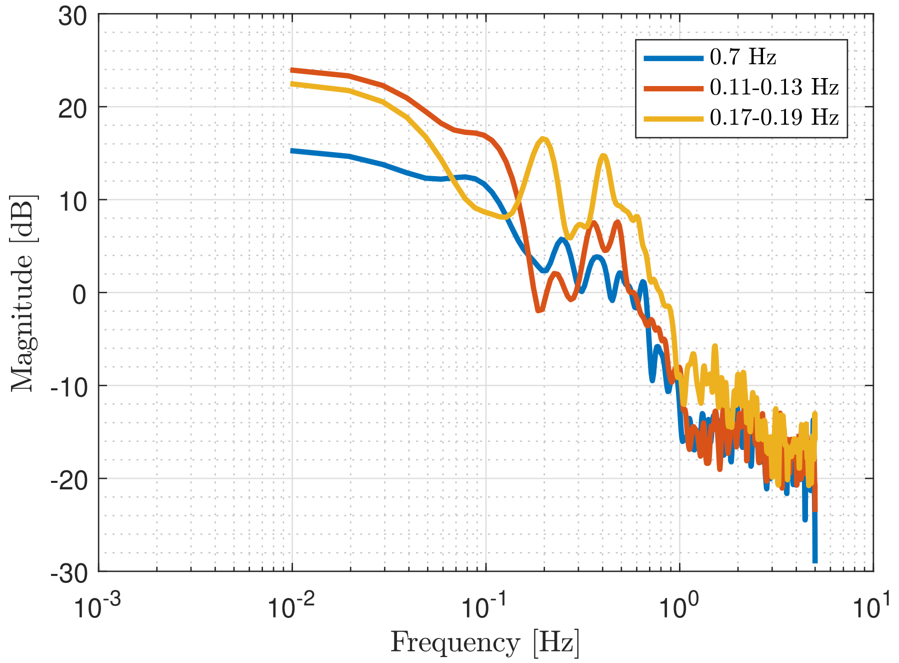


Figure 4.22: Power spectral density of the steady-state (t > 60 s) tracking error in Fig. 4.20.

Table 4.2: Tracking RMS error (calculated for t > 60 sec in Fig. 4.20) with adaptive feedforward and the pure delay system.

Profile	Scheme	RMS
Frequency [Hz]	Tested	Error [mm]
0.07	Adaptive FF	1.113
	Pure Delay	1.410
0.11-0.13	Adaptive FF	1.197
	Pure Delay	1.543
0.17-0.19	Adaptive FF	2.171
	Pure Delay	2.465

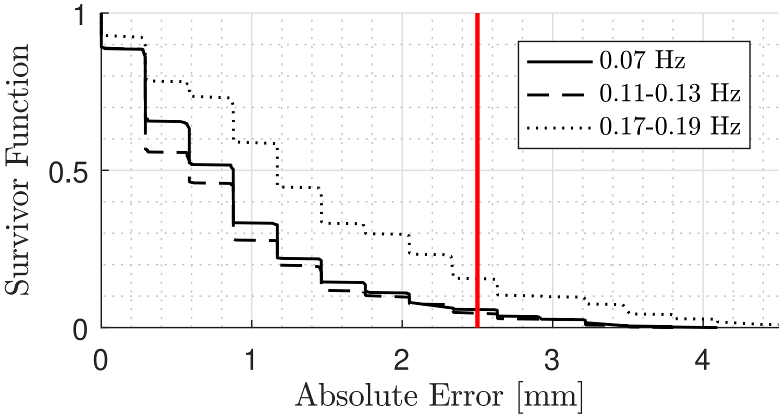


Figure 4.23: Survival function with respect to absolute tracking error. For a given $|e_0|$ on x-axis, the y-axis value denotes the probability $Pr(|e| > |e_0|)$, namely the chance of having absolute error larger than $|e_0|$.

4.6 Discussion

Now we examine two performance limiting factors, the system delay and the needle noise variance (see Fig. 4.5) induced by image processing, using the aggressive profile in the bottom plot of Fig. 4.20. Fig. 4.24 shows the simulation results. The range of delay is chosen so because MRI has a 100 msec acquisition time, and with image processing a 200-400 msec total delay is common [LLM19]. The lower bound for needle noise variance is set to be the same as the target noise variance 0.275 mm because this sub-pixel image tracking with large feature size is considered the best performance with current real-time MRI technology. It can be seen decreased delay results in improvement because the prediction horizon is shorter. Interestingly, when the needle noise variance decreases, the FB scheme improves significantly. This is because the adaptive feedback scheme in Fig. 4.11 utilizes heavily corrupted y in both inner loop and the adaptive loop, and thus is affected the most. The FF+FB is less sensitive because part of the control action is compensated by the feedforward controller. If both limits in delay and needle noise are achieved, the maximum tracking error on aggressive profile is reduced by 50%, and $Pr(e > 2.5\text{mm}) < 10\%$, comparable to the results of those mild ones. It should be noted that in the simulation $G = \hat{G}$ is used so it appears feedforward

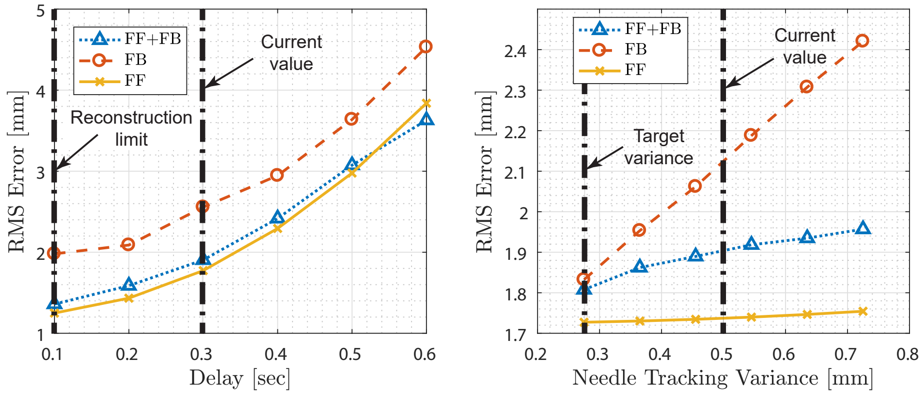


Figure 4.24: Effect of system delay and needle tracking variance on different adaptive control schemes. The original system delay is 300 msec, and the variance of needle imaging noise is 0.498 mm. Note in simulation we assume $G = \hat{G}$.

alone performs the best, but in reality modeling error exists and joint operation should give the best performance if the needle variance can be further mitigated.

4.7 Conclusion

The results of the paper can be summarized as follows:

1. Fully-automated dynamic target tracking in MR environment is realized via the integration of robotic platform, MR image tracking, and adaptive control algorithm. Navigating an instrument with image feedback can thus be more effective and streamlined.

2. Modeling of the MR image acquisition process and instrument manipulation module dynamics. With the high fidelity models, add-on adaptive controllers are analyzed, simulated, and validated.

3. Implementation of the adaptive controller on a hydrostatically actuated platform in MRI scanner. Breathing profiles with different spectral properties are tested and cross-validated by simulation. Limiting factors are also discussed.

Future work include several aspects. First is the hardware: multi-DOF platforms for both robot and target. Second is the MR sensing. We have developed a needle tracking algorithm [LRL19] that could account for a more general rigid body motion, and is working on the automated scanning plane alignment when the needle changes posture. In our recent work [LLM19] it is also shown possible to improve tracking accuracy and mitigate time delay via multi-rate fusion. If the sensing can be improved, a higher control rate could be used to deal with rapidly changing breathing profiles. Also, currently we do not assume any knowledge of the profile tracked. It is possible to include an internal model for breathing to further improve the tracking performance.

CHAPTER 5

Summary

In this research inversion-based control methods are studied for different systems and applications.

In Chapter 2, we study the tracking application of robot manipulator. Because the actuator dynamic compensation alone cannot guarantee end effector path accuracy due to static bias, either from imprecise kinematics or joint deflection. Therefore, a nested-loop iterative learning algorithm is proposed with an inner-loop motor dynamic compensation and an outer-loop static bias correction. The inner loop learning filter requires only simple impulsive inputs and has no fitting loss. Motion constraints of the system can also be incorporated. The outer loop uses robot inverse Jacobian for effective learning to cancel out static bias without exciting vibration modes. This is experimentally verified on a 6-DOF serial robot and the error is mitigated by an order of magnitude ($\approx 50 \mu m$) under variant payload. The following topics can be pursued going forward:

1. When the cross-coupling and system variation cannot be ignored, the system becomes multivariate and cross-coupled. An effective interpolation strategy of system inversion can further improve the performance.

2. Because the laser tracker is expensive to set up, a cost effective end effector performance evaluation is of practical interest. For example, vision-based sensor, inertia measurement [JTK09], and proximity sensor.

3. Another interesting aspect is when there exhibits robot-workpiece and/or robot-robot interaction. Position-based control is likely to induce large force, and therefore the

interaction needs to be addressed in the baseline controller as well as ILC design.

In Chapter 3, we propose a novel data-driven iterative learning control for nonlinear dynamic systems. Aiming to address the nonlinearity of real-world systems and mitigate the burden of tedious identification and control design process, this idea utilizes adaptive filter to identify the system inversion of the linearized system along the trajectory. Because this approach is inversion-based, it has fast convergence and good tracking performance. Also, the tracking capability and existing "fast" rotation-based adaptive filtering algorithm allows a memory efficient and computationally tractable implementation. A comparative study is given with the extension of an existing adjoint-based method for LTI systems [BKO18]. This is demonstrated on a 2-DOF laboratory robot with simulation and experimental results, showing the superior performance of the proposed adaptive ILC. Algorithms for multivariate systems are also developed and validated via simulation. The ongoing tasks include:

1. The implementation and comparative study of the MIMO algorithms.
2. A design guideline for excitation input, reference model, and low-pass filter selection. Proof of ILC convergence for general or a class of nonlinear systems.
3. A qualitative study of the non-smooth effect on each algorithm, for example, the effect of friction, dead zone, and finite word length.

In Chapter 4, we study the MRI-guided automated intervention, wherein the goal is to regulate the distance between a dynamic target and an instrument. This is set up as a unknown disturbance (target motion) rejection problem, and adaptive control schemes are investigated for their prediction capability. The concept is tested on a hydrostatically-actuated research platform, embodying an instrument manipulation module for intervention, and a target motion module emulating tissue motion [SLM17]. Through modeling of MR image processing and instrument module dynamics, the real-time MRI feedback motion control is simulated and experimentally verified in MR environment, showing great promise towards fully-automated MRI-guided intervention.

From the results of this study, the following recommendations can be made going forward:

1. Currently the adaptive control does not assume any knowledge of the breathing profile. However, the respiratory motion is not purely stochastic. Prediction can be made based on some model forms, either on the sensing side or the control side, to further enhance the performance.

2. With the knowledge of the transmission dynamics from motor output to end effector output, one can fuse the knowledge of encoder-based prediction and MR measurement to acquire better estimate. As the signal-to-noise-ratio improves, higher control rate can be used to extend system bandwidth.

3. A multi-degree-of-freedom robot and dynamic phantom development for a more relevant clinical setting.

4. For general instrument motion, image tracking algorithm needs to account for rigid body motion and to adaptively align the scanning plane with the instrument so that MRI can keep track of.

References

[AHF15] Maarten M Arnolli, Nevan C Hanumara, Michel Franken, Dannis M Brouwer, and Ivo AMJ Broeders. "An overview of systems for CT-and MRI-guided percutaneous needle placement in the thorax and abdomen." *The International Journal of Medical Robotics and Computer Assisted Surgery*, **11**(4):458–475, 2015.

[AND09] Nimer Assy, Gattas Nasser, Agness Djibre, Zaza Beniashvili, Saad Elias, and Jamal Zidan. "Characteristics of common solid liver lesions and recommendations for diagnostic workup." *World journal of gastroenterology: WJG*, **15**(26):3217, 2009.

[BFG17] Natalie Burkhard, Samuel Frishman, Alexander Gruebele, J Peter Whitney, Roger Goldman, Bruce Daniel, and Mark Cutkosky. "A rolling-diaphragm hydrostatic transmission for remote MR-guided needle insertion." In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1148–1153. IEEE, 2017.

[BKO18] Joost Bolder, Stephan Kleinendorst, and Tom Oomen. "Data-driven multivariable ILC: enhanced performance by eliminating L and Q filters." *International Journal of Robust and Nonlinear Control*, **28**(12):3728–3751, 2018.

[BO15] Joost Bolder and Tom Oomen. "Data-driven optimal ILC for multivariable systems: Removing the need for L and Q filter design." In *2015 American Control Conference (ACC)*, pp. 3546–3551. IEEE, 2015.

[BPA12] JA Butterworth, LY Pao, and DY Abramovitch. "Analysis and comparison of three discrete-time feedforward model-inverse control techniques for nonminimum-phase systems." *Mechatronics*, **22**(5):577–587, 2012.

[BWJ05] Chao Bi, Dezheng Wu, Quan Jiang, and Zhejie Liu. "Automatic learning control for unbalance compensation in active magnetic bearings." *IEEE Transactions on Magnetics*, **41**(7):2270–2280, 2005.

[BWK16] Lennart Blanken, Jeroen Willems, Sjirk Koekebakker, and Tom Oomen. "Design techniques for multivariable ILC: Application to an industrial flatbed printer." *IFAC-PapersOnLine*, **49**(21):213–221, 2016.

[CHX08] Ronghu Chi, Zhongsheng Hou, and Jianxin Xu. "Adaptive ILC for a class of discrete-time systems with iteration-varying trajectory and random initial condition." *Automatica*, **44**(8):2207–2213, 2008.

[CT14] Wenjie Chen and Masayoshi Tomizuka. "Dual-stage iterative learning control for mimo mismatched system with application to robots with joint elasticity." *IEEE Transactions on control systems technology*, **22**(4):1350–1361, 2014.

[CT17] Cheng-Wei Chen and Tsu-Chin Tsao. "Data-Driven Progressive and Iterative Learning Control." *IFAC-PapersOnLine*, **50**(1):4825–4830, 2017.

[DB02] Branko G Dijkstra and Okko H Bosgra. "Convergence design considerations of low order Q-ILC for closed loop systems, implemented on a high precision wafer stage." In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, volume 3, pp. 2494–2499. IEEE, 2002.

[EA01] Kaan Erkorkmaz and Yusuf Altintas. "High speed CNC system design. Part I: jerk limited trajectory generation and quintic spline interpolation." *International Journal of machine tools and manufacture*, **41**(9):1323–1345, 2001.

[Fis15] Gregory S Fischer. "Enabling closed-loop surgery in MRI." *Mechanical Engineering Magazine Select Articles*, **137**(09):S11–S14, 2015.

[FKI08] Gregory S Fischer, Axel Krieger, Iulian Iordachita, Csaba Csoma, Louis L Whitcomb, and Gabor Fichtinger. "MRI compatibility of robot actuation techniques–a comparative study." In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 509–517. Springer, 2008.

[Fle13] Roger Fletcher. *Practical methods of optimization.* John Wiley & Sons, 2013.

[GGB04] G Ganesh, Roger Gassert, Etienne Burdet, and Hannes Bleuler. "Dynamics and control of an MRI compatible master-slave system with hydrostatic transmission." In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 2, pp. 1288–1294. IEEE, 2004.

[Gor02] Dimitry Gorinevsky. "Loop shaping for iterative control of batch processes." *IEEE Control Systems Magazine*, **22**(6):55–65, 2002.

[Gus96] Fredrik Gustafsson. "Determining the initial states in forward-backward filtering." *IEEE Transactions on signal processing*, **44**(4):988–992, 1996.

[GYC06] Roger Gassert, Akio Yamamoto, Dominique Chapuis, Ludovic Dovat, Hannes Bleuler, and Etienne Burdet. "Actuation methods for applications in MR environments." *Concepts in Magnetic Resonance Part B: Magnetic Resonance Engineering: An Educational Journal*, **29**(4):191–209, 2006.

[Hay08] Simon S Haykin. *Adaptive filter theory.* Pearson Education India, 2008.

[HS13] Michael Schacht Hansen and Thomas Sangild Sørensen. "Gadgetron: an open source framework for medical image reconstruction." *Magnetic resonance in medicine*, **69**(6):1768–1776, 2013.

[HXV14] Deqing Huang, Jian-Xin Xu, Venkatakrishnan Venkataramanan, and The Cat Tuong Huynh. "High-performance tracking of piezoelectric positioning stage using current-cycle iterative learning control with gain scheduling." *IEEE Transactions on Industrial Electronics*, **61**(2):1085–1098, 2014.

[JTK09] Soo Jeon, Masayoshi Tomizuka, and Tetsuaki Katou. "Kinematic Kalman filter (KKF) for robot end-effector sensing." *Journal of dynamic systems, measurement, and control*, **131**(2):021010, 2009.

[KGM15] Elena A Kaye, Kristin L Granlund, Elizabeth A Morris, Majid Maybody, and Stephen B Solomon. "Closed-bore interventional MRI: percutaneous biopsies and ablations." *American Journal of Roentgenology*, **205**(4):W400–W410, 2015.

[KK96] Dong-Il Kim and Sungkwun Kim. "An iterative learning control method with application for CNC machine tools." *IEEE Transactions on Industry Applications*, **32**(1):66–72, 1996.

[LHD19] Yu-Hsiu Lee, Sheng-Chieh Hsu, Yan-Yi Du, Jwu-Cheng Hu, and Tsao Tsu-Chin. "Nested-Loop Iterative Learning Control for Robot Manipulators." *Joint Conference 8th IFAC Symposium on Mechatronic Systems (MECHATRONICS 2019), and 11th IFAC Symposium on Nonlinear Control Systems (NOLCOS 2019)*, Sept. 4-6, 2019.

[LLM19] X Li, YH Lee, S Mikaiel, J Simonelli, TC Tsao, and HH Wu. "Respiratory Motion Prediction using Fusion-Based Multi-Rate Kalman Filtering and Real-Time Golden-Angle Radial MRI." *IEEE transactions on bio-medical engineering*, 2019.

[LRL19] Xinzhou Li, Steven Raman, David Lu, Yu-Hsiu Lee, Tsu-Chin Tsao, and Holden Wu. "Real-time needle detection and segmentation using Mask R-CNN for MRI-guided interventions." In *Proceedings of the ISMRM 27th Annual Meeting, Canada*, p. 972, 2019.

[LT19] Yu-Hsiu Lee and Tsao Tsu-Chin. "Data-driven ILC for Trajectory Tracking in Nonlinear Dynamic Systems." *Proceedings of Proceedings of the ASME 2019 & Dynamic Systems and Control Conference (DSCC 2019)*, Oct. 9-11, 2019.

[MCS18] Reza Monfaredi, Kevin Cleary, and Karun Sharma. "MRI robots for needle-based interventions: systems and technology." *Annals of biomedical engineering*, **46**(10):1479–1497, 2018.

[MKM86] F Miyazaki, S Kawamura, M Matsumori, and S Arimoto. "Learning control scheme for a class of robot systems with elasticity." In *Decision and Control, 1986 25th IEEE Conference on*, volume 25, pp. 74–79. IEEE, 1986.

[MSK17] Pedro Moreira, Gert van de Steeg, Thijs Krabben, Jonathan Zandman, Edsko EG Hekman, Ferdinand van der Heijden, Ronald Borra, and Sarthak Misra. "The miriam robot: A novel robotic system for mr-guided needle insertion in the prostate." *Journal of medical robotics research*, **2**(04):1750006, 2017.

[MSL16] Samantha Mikaiel, James Simonelli, David Lu, Kyung Sung, Tsu-Chin Tsao, and Holden Wu. "Real-Time MRI-Guided Interventions using Rolling-Diaphragm

Hydrostatic Actuators." In *Proceedings of the ISMRM 24th Annual Meeting, Singapore*, p. 3588, 2016.

[MSL17] Samantha Mikaiel, James Simonelli, Yu-Hsiu Lee, Xinzhou Li, Yong Seok Lee, David Lu, Kyung Sung, Tsu-Chin Tsao, and Holden Wu. "Real-Time MRI-Guided Targeted Needle Placement During Motion using Rolling-Diaphragm Hydrostatic Actuators." In *Proceedings of the ISMRM 25th Annual Meeting, Honolulu*, p. 736, 2017.

[MW19] Evelyn Mendoza and John Peter Whitney. "A Testbed for Haptic and Magnetic Resonance Imaging-Guided Percutaneous Needle Biopsy." *IEEE Robotics and Automation Letters*, 2019.

[NG02] Mikael Norrlöf and Svante Gunnarsson. "Time and frequency domain convergence properties in iterative learning control." *International Journal of Control*, **75**(14):1114–1126, 2002.

[PGT09] Nestor O Perez-Arancibia, James S Gibson, and Tsu-Chin Tsao. "Frequency-weighted minimum-variance adaptive control of laser beam jitter." *IEEE/ASME Transactions On Mechatronics*, **14**(3):337–348, 2009.

[PKL15] Niravkumar A Patel, Tim van Katwijk, Gang Li, Pedro Moreira, Weijian Shang, Sarthak Misra, and Gregory S Fischer. "Closed-loop asymmetric-tip needle steering under continuous intraoperative MRI guidance." In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 4869–4874. IEEE, 2015.

[Ple01] Gregory L Plett. "Efficient linear MIMO adaptive inverse control." *IFAC Proceedings Volumes*, **34**(14):89–94, 2001.

[RB91] Phillip A Regalia and Maurice G Bellanger. "On the duality between fast QR methods and lattice methods in least squares adaptive filtering." *IEEE Transactions on Signal Processing*, **39**(4):879–891, 1991.

[RHB09] Michael Ruderman, Frank Hoffmann, and Torsten Bertram. "Modeling and identification of elastic robot joints with hysteresis and backlash." *IEEE Trans. Industrial Electronics*, **56**(10):3840–3847, 2009.

[SHV06] Mark W Spong, Seth Hutchinson, Mathukumalli Vidyasagar, et al. *Robot modeling and control*, volume 3. Wiley New York, 2006.

[SLCed] James Simonelli, Yu-Hsiu Lee, Cheng-Wei Chen, Xinzhou Li, Samantha Mikaiel, David Lu, Wu Holden, and Tsu-Chin Tsao. "Hydrostatic Actuation for Remote Operations in MR Environment." *IEEE/ASME Transactions on Mechatronics*, Submitted.

[SLM17] James Simonelli, Yu-Hsiu Lee, Samantha Mikaiel, Cheng-Wei Chen, Xinzhou Li, Kyung Sung, David Lu, Holden Wu, and Tsu-Chin Tsao. "An MR-Compatible Stage for Respiratory Motion Emulation." *IFAC-PapersOnLine*, **50**(1):6073–6078, 2017.

[SPS14] Tong Duy Son, Goele Pipeleers, and Jan Swevers. "Experimental validation of robust iterative learning control on an overhead crane test setup." *IFAC Proceedings Volumes*, **47**(3):5981–5986, 2014.

[SRB91] EA Simoes, R Roark, S Berman, LL Esler, and J Murphy. "Respiratory rate: measurement of variability over time and accuracy at different counting periods." *Archives of disease in childhood*, **66**(10):1199–1203, 1991.

[Tay04] Abdelhamid Tayebi. "Adaptive iterative learning control for robot manipulators." *Automatica*, **40**(7):1195–1203, 2004.

[TCC12] Kuo-Tai Teng, Kai-Hsiang Chang, Yung-Yaw Chen, and Tsu-Chin Tsao. "Respiration induced liver motion tracking control for high intensity focused ultrasound treatment." In *2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 57–62. IEEE, 2012.

[TMT07] Servet Tatli, Paul R Morrison, Kemal Tuncali, and Stuart G Silverman. "Interventional MRI for oncologic applications." *Techniques in vascular and interventional radiology*, **10**(2):159–170, 2007.

[TT15] Kuo-Tai Teng and Tsu-Chin Tsao. "A comparison of inversion based iterative learning control algorithms." In *American Control Conference (ACC), 2015*, pp. 3564–3569. IEEE, 2015.

[VDB11] JJM Van De Wijdeven, MCF Donkers, and OH Bosgra. "Iterative learning control for uncertain systems: Noncausal finite time interval robust control design." *International Journal of Robust and Nonlinear Control*, **21**(14):1645–1666, 2011.

[WGB14] John P Whitney, Matthew F Glisson, Eric L Brockmeyer, and Jessica K Hodgins. "A low-friction passive fluid transmission and fluid-tendon soft actuator." In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2801–2808. IEEE, 2014.

[Wid87] Bernard Widrow. "Adaptive inverse control." In *Adaptive Systems in Control and Signal Processing 1986*, pp. 1–5. Elsevier, 1987.

[WNL08] Clifford R Weiss, Sherif Gamal Nour, and Jonathan S Lewin. "MR-guided biopsy: a review of current techniques and applications." *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine*, **27**(2):311–325, 2008.

[WSG18] Marek Wartenberg, Joseph Schornak, Katie Gandomi, Paulo Carvalho, Chris Nycz, Niravkumar Patel, Iulian Iordachita, Clare Tempany, Nobuhiko Hata, Junichi Tokuda, et al. "Closed-loop active compensation for needle deflection and target shift during cooperatively controlled robotic needle insertion." *Annals of biomedical engineering*, **46**(10):1582–1594, 2018.

[WZW18] Cong Wang, Minghui Zheng, Zining Wang, Cheng Peng, and Masayoshi Tomizuka. "Robust Iterative Learning Control for Vibration Suppression of Industrial Robot Manipulators." *Journal of Dynamic Systems, Measurement, and Control*, **140**(1):011003, 2018.

[YW05] Yongqiang Ye and Danwei Wang. "Zero phase learning control using reversed time input runs." *Journal of Dynamic Systems, Measurement, and Control(Transactions of the ASME)*, **127**(1):133–139, 2005.

[ZBK16] Jurgen van Zundert, Joost Bolder, Sjirk Koekebakker, and Tom Oomen. "Resource-efficient ILC for LTI/LTV systems through LQ tracking and stable inversion: Enabling large feedforward tasks on a position-dependent printer." *Mechatronics*, **38**:76–90, 2016.

[ZLX15] Yi Min Zhao, Yu Lin, Fengfeng Xi, and Shuai Guo. "Calibration-based iterative learning control for path tracking of industrial robots." *IEEE Transactions on industrial electronics*, **62**(5):2921–2929, 2015.

[ZO18] Jurgen van Zundert and Tom Oomen. "On inversion-based approaches for feedforward and ILC." *Mechatronics*, **50**:282–291, 2018.