Robustification in Repetitive and Iterative Learning Control

Yunde Shi

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY
2013
ABSTRACT

Robustification in Repetitive and Iterative Learning Control

Yunde Shi

Repetitive Control (RC) and Iterative Learning Control (ILC) are control methods that specifically deal with periodic signals or systems with repetitive operations. They have wide applications in diverse areas from high-precision manufacturing to high-speed assembly, and nowadays these algorithms have even been applied to biomimetic walking robots, where tracking a periodic reference signal or rejecting periodic disturbances is desired. Compared to conventional feedback control designs (including the inverse dynamics method), RC and ILC improve the control performance over repetitions -- by learning from the previous input-output data, RC and ILC adaptively update the control input for the next run, aiming for zero tracking error in the hardware instead of in a model, as time goes to infinity. The stability robustness to model uncertainty however remains a fundamental topic as it determines the successful implementation of RC and ILC on any real-world system whose model dynamics cannot normally be determined precisely over all frequencies up to Nyquist. In the control field, there are various existing methods of robustification, such as Linear Matrix Inequality (LMI), μ-synthesis and H-infinity, but few of these methods offer intuitive information about how the stability robustness is achieved. In addition, many of these existing algorithms produce conservative stability boundaries, leaving room for further optimization and enhancement. In this study, several robustification approaches are developed, where better insight into the robustification design process and a tighter stability boundary are established. The first method presents an algorithm for RC compensator design that not only uses phase adjustments, but also
adjusts the learning rate as a function of frequency to obtain improved robustification to model parameter uncertainty. The basic objective of this algorithm is to make the system learn at each frequency at the maximum rate consistent with the need for robustness at that frequency. The second method, on the other hand, explores the benefits of compromising on the zero tracking error requirement for frequencies that require extra robustness, making RC tolerate larger model errors. The third topic focuses on the development of robustification algorithms for Iterative Learning Control that is analogous to the above two RC robustification designs, extending frequency response concepts to finite time problems. The final approach to robustification treated in this dissertation is based on Matched Basic Function Repetitive Control (MBFRC), which individually addresses each frequency, eliminating the need for a robustifying zero phase low pass filter and the need for interpolation in data as required in conventional RC design. Furthermore, this algorithm only uses the frequency response knowledge at the frequencies addressed, and as long as the phase uncertainties at those frequencies are within +/- 90 deg the system is guaranteed stable for all sufficiently small projection gains.
# Table of Contents

1 Introduction

1.1 Background ........................................................................................................ 1

1.2 Thesis Outline .................................................................................................... 1

1.3 References .......................................................................................................... 4

2 An Algorithm for Robustification of Repetitive Control to Parameter Uncertainties 7

2.1 Introduction ....................................................................................................... 7

2.2 Repetitive Control System Stability and Design ............................................... 7

2.3 Stability Robustness in Repetitive Control ....................................................... 14

2.3.1 The Importance of Robustness ..................................................................... 14

2.3.2 Approaches to Robustification .................................................................... 15

2.3.3 The Robustness Limit in Repetitive Control .............................................. 16

2.3.4 The Cost Function Averaging Approach ................................................... 17

2.4 Modifying the Objective to Further Improve Robustness ............................... 17

2.5 An Algorithm for Improved Robustification of RC Systems to Model Parameter Uncertainties ................................................................. 19

2.6 Numerical Examples ......................................................................................... 22

2.7 Conclusions ...................................................................................................... 26

2.8 References ........................................................................................................ 27

3 The Influence on Stability Robustness of Compromising on the Zero Tracking Error Requirement in Repetitive Control 30

3.1 Introduction ....................................................................................................... 30

3.2 An Effective Repetitive Control Design Procedure ......................................... 30
3.3 ILC Design by Quadratic Cost with Control Effort Penalty: Compromising on the Zero Error Requirement ................................................................. 33

3.3.1 The Optimization Criterion .......................................................... 34

3.3.2 The ILC Law .............................................................................. 35

3.3.3 Stability and Final Error Level ...................................................... 36

3.3.4 Robustness to Model Error ............................................................ 37

3.4 Creating RC Based on the Frequency Response of the ILC Design ....... 38

3.4.1 The Control Law in the Frequency Domain ................................. 38

3.4.2 Implementation with FIR Filters ............................................... 39

3.4.3 The Stability Condition and the Sensitivity Transfer Function ....... 40

3.4.4 Multiple Model Version of RC Law ............................................ 41

3.5 Numerical Examples ..................................................................... 42

3.5.1 Examples of Robust RC Design Using a Mathematical Model ....... 42

3.5.2 Examples of Robust RC Design Based on Multiple Frequency Response Tests ................................................................. 47

3.6 Conclusions ................................................................................. 50

3.7 References .......................................................... 52

4 Converting Repetitive Control Robustification Methods to Apply To Iterative Learning Control 54

4.1 Introduction ................................................................................. 54

4.2 Mathematical Formulation of the ILC Problem ............................. 55

4.3 Four Kinds of Robustification ....................................................... 56

4.4 Ordering the Singular Vectors and Singular Values ......................... 60

4.5 Equations for Updates Applied to Any Model ............................... 61
4.6 Making the ILC Problem Well Posed for Pole Excess Greater Than Two …… 62
4.7 Equations For Updates For Any Model With Rows Deleted .................. 64
4.8 A Process to Adjust ILC Gains Based On a Necessary Condition for Monotonic Decay ................................................................. 66
  4.8.1 The Necessary Conditions ....................................................... 66
  4.8.2 Adjusting the \( \gamma_i \) ........................................................... 67
  4.8.3 Adjusting the \( h_i \) ............................................................... 68
  4.8.4 Summary of Procedure .......................................................... 69
  4.8.5 Possible Additional Test Inputs ............................................... 69
4.9 Picking ILC Gains Based on the Frobenius Norm .............................. 70
4.10 A Condition for Learnability of a Model ....................................... 71
4.11 Examples .................................................................................... 72
4.12 Conclusions ................................................................................ 77
4.13 References .................................................................................. 78

5 Small Gain Stability Theory for Matched Basis Function Repetitive Control 82
  5.1 Introduction ................................................................................ 82
  5.2 The Matched Basis Function Repetitive Control Algorithm .............. 83
  5.3 The Structure of the Time Invariant Equivalent of MBFRCs ............. 87
  5.4 The Pole/Zero Repetitive Controllers for Each Frequency .................. 90
    5.4.1 The RC Transfer Functions For Each Frequency ....................... 91
    5.4.2 The Pole Locations For \( n = 1, 2, 3, \ldots, N \) .............................. 91
    5.4.3 The Zero Locations For \( n = 1, 2, 3, \ldots, N \) .............................. 93
  5.5 Small Gain Stability Theory for a Single Addressed Frequency ........... 94
5.5.1 The Characteristic Polynomial .................................................. 95
5.5.2 The Root Locus Departure Angle Condition ......................... 95
5.5.3 Departure Angle Contribution From $z_{n_1n_2}$ ................................. 97
5.5.4 The Special Case of $\cos(\phi_n - \tau_n) = 0$ ................................. 99
5.5.5 Departure Angle Contribution From $P_{n_1n_2}$, Real Root Case .... 100
5.5.6 Departure Angle Contribution From $P_{n_1n_2}$, Complex Root Case .... 101
5.5.7 Single Addressed Frequency Result ........................................... 101

5.6 Small Gain Stability Theorem for Arbitrary Number of Addressed Frequencies ................................................ 102

5.7 Numerical Examples ................................................................. 104

5.8 Conclusions ............................................................................. 107

5.9 References ............................................................................. 110

6 Conclusions ............................................................................. 115
Acknowledgements

I would like to express my sincere gratitude to all those who have helped me throughout my PhD research and thesis. My work would never be possible without their passionate encouragement and support.

I want to express my sincere gratitude to my thesis and research adviser Prof. Richard W. Longman. His guidance and patience has spurred me through the entire research. Without his help, I could not even start this work. As a hard-working and dedicated scholar who devotes his whole life to research, Prof. Richard W. Longman has continuously motivated me and has taught me many important things beyond the research itself.

I also want to thank my parents and brother for raising me in the hardship during my childhood in China. Their unconditional and endless love has accompanied me whenever I come across difficulties.

Finally, I would like to thank my lab members, Te Li, Jianzhong Zhu, Yui and Francesco, and my coauthors, Dr. Masaki Nagashima, Prof. Minh Phan and Prof. Remondo Betti for their enthusiastic help with various problems during this research.
Chapter 1

Introduction

1.1 Background

Repetitive control (RC) and Iterative Learning Control (ILC) are relatively new fields within control that aim to produce zero tracking error in repetitive situations. The desired output can be a periodic function, or the desired output can be a constant and there is a periodic disturbance of known period, or both. Early papers in this field include References [1-7] with applications to rectification of alternating current to DC, to periodic motions of robots, to tracking in computer disk drives, to non-circular machining, etc. Spacecraft applications include the use of active vibration isolation systems to isolate fine pointing equipment from spacecraft vibrations caused by slight imbalance in rotating parts such as a momentum wheel, reaction wheels, control moment gyros, cryo pumps, etc. References [8-11] give basic formulations and approaches to the design of RC systems. References [10], [12-18] give an overall presentation of design approaches recommended by the ILC and RC research group at Columbia University.

1.2 Thesis Outline

The unusual property of asking for zero tracking error for all frequencies or harmonics having the desired period all the way to Nyquist frequency makes the stability robustness to model errors an important issue to the Repetitive Control (RC) and Iterative Learning Control
(ILC). In this thesis, four methods are presented to address this specific robustification topic in Repetitive Control (RC) and Iterative Learning Control (ILC).

Chapter 2 first proposes an algorithm which is particularly useful for the robustness to parameter uncertainties in the model used to design the RC system. A series of publications, References [19-22], the authors demonstrate a remarkable improvement in the stability robustness of both RC and ILC by designing the control law based on averaging a cost function over the model distribution. One specifies the uncertainty in the model parameters, using whatever probability distribution is appropriate. Then one picks a representative sample of models from these distributions. Instead of designing the control action based on the nominal model using a quadratic cost function, one writes the cost function for each of the chosen models, and designs one control law that minimizes the sum or average of the costs over all models. The minimization problem is nonlinear in the parameters so that the control law that minimizes the average cost is not the same as the control law that minimizes the cost for the average model. Experience reported in these references shows a dramatic improvement in robustness. This chapter develops an understanding of the source of this robustification. It then uses this understanding to generate a more precise algorithm that adjusts both phase and learning rate as a function of frequency in order to optimize the interval of robustification achieved. Compared to other robustification approaches, this approach has the advantage that it deals with the actual stability boundary at each frequency, without needing to make any conservative approximations.

Chapter 3, on the other hand, explores the advantages of compromising on the usual the zero-tracking error requirement in typical RC designs and obtains extra stability robustness by extending the hard limit of the maximum model phase error. A quadratic cost design method is used here that includes a penalty on the size of the control action for frequencies that would
otherwise be cut out. The design is first done for ILC, and then the resulting frequency response behavior is used in designing RC. The results (e.g. error sensitivity transfer function) show that by compromising on the strict zero-tracking error requirement, the system obtains extra robustness and even improved tracking performance compared to the traditional design where a cutoff filter is used.

Robustification based on the above two algorithms is extended to apply to the sister field of ILC in Chapter 4, where the unique finite-time characteristic of ILC is taken into account. Some authors investigate the use of robust control theory based on linear matrix inequalities, for example Reference [23]. However, such an approach is complicated, and does not allow much intuition in understanding the limits to stabilization. This chapter develops a new method for ILC robustification that offers substantial insight during the design process.

Chapter 5 investigates the Matched Basis Function Repetitive Control (MBFRC) method as a method of achieving enhanced stability robustness. Unlike the previous RC methods that address all frequencies of a given period, the fundamental and all harmonics, MBFRC addresses each frequency individually. One advantage of MBFRC is that multiple unrelated frequencies are addressed in a simple manner, without the complexity needed in References [24~28]. Another advantage relates to interpolation, which may be required in the usual RC approaches when the period of the addressed frequency is not an integer number of time steps (Reference [29]). In MBFRC, the basis functions at each frequency allow one to “interpolate” with the actual frequency function of interest so that the concerns of deteriorated interpolation at high frequencies is also alleviated. A very strong small gain robustness result is obtained in this chapter, guaranteeing convergence to zero tracking error for the frequency components
addressed for all sufficiently small repetitive control gain, provided one knows the phase change through the system to within a tolerance of plus or minus 90 degrees at the addressed frequency.

Finally, Chapter 6 summarizes the research in this thesis.

1.3 References


Chapter 2

An Algorithm for Robustification of Repetitive Control to Parameter Uncertainties

2.1 Introduction

In this chapter, an algorithm is developed specifically for the robustification of Repetitive Control (RC) to model parameter uncertainties. It starts with the basic concepts underlying the cost function averaging in References [14~17] and develops ways to include the compensator magnitude response into the robustification process. First of all, the uncertainty in the model parameters is specified based on the knowledge of model probability distribution and a representative sample of models from these distributions is selected. The algorithm then creates a quadratic cost function which minimizes the sum or average of the costs over all models. Since the minimization problem is nonlinear in the parameters, the control law that minimizes the average cost is not the same as the control law that minimizes the cost for the average model. Based on this understanding, a precise algorithm that adjusts both phase and learning rate as a function of frequency is formulated (Equations (2-11), (2-12)) for the optimization of RC robustification. Compared to other robustification approaches, this method has the advantage of dealing with the real stability boundary at each frequency instead of using any conservative approximations.

2.2 Repetitive Control System Stability and Design

Consider a closed loop digital feedback control system (FBCS) whose transfer function is $G(z)$ and whose output is given by $Y(z) = G(z)U(z) + V(z)$. The $U(z)$ is the command input and
$V(z)$ is a periodic output disturbance. The disturbance to a feedback system usually enters before the plant in the feedback loop, but a periodic disturbance entering there or anywhere else around the loop, has an equivalent periodic disturbance added to the output after the feedback loop, and this model is used here. The periodic or constant desired output is $Y_D(z)$ and $E(z) = Y_D(z) - Y(z)$ is the error. The period of the periodic command and/or disturbance is considered to be $p$ time steps.

Figure 2-1. Block diagram of repetitive controller modifying the command to a feedback control system

The simplest form of repetitive control can be described in words as follows, if the output was 2 units too low at the current phase of the period during the last period, then add 2 units, or two units times a gain $\phi$, to the command at the current time step. Mathematically this is

$$u(k) = u(k - p) + \phi e(k - p + 1)$$

(2-1)

Assuming that there is a one time-step delay between the time step at which the input is changed and the time step an influence is first seen in the output, we have added one time step to the error examined in the previous period. Suppose that the DC gain of $G(z)$ were unity. Then setting $\phi = 1$ would make this RC law correct a constant error in the next time step once transients have decayed. Now suppose that the 2 unit error observed in the last period was a sample of an error that is oscillating at some frequency, and suppose that the frequency response of $G(z)$ has a 180 degree phase lag at that frequency. When the change in input dictated by Equation (2-1) goes through the system the phase lag is the equivalent of reversing the sign of the change in the output, and hence the RC law is increasing the error instead of decreasing it. In a rough manner,
one can consider that it does a perfect job of correcting the error when the phase through the system is zero, and does the correction exactly wrong when there is a 180 degree phase lag, so one might think that phase lag of 90 degrees might divide between reducing the error and adding to the error. For small signals this is true, but for larger signals one needs to have smaller phase error as described below. This thinking does suggest that phase error is the critical factor governing when errors in a repetitive control system converge and when they do not, and it also suggests that decreasing the gain in the repetitive control increases robustness to model error.

In order to make a repetitive control system converge to zero error for frequencies for which \( G(z) \) has substantial phase lag, it is necessary to design a compensator \( F(z) \) that supplies appropriate phase lead. One possibility is to try to make the compensator have a transfer function that is the inverse of the system. Usually this is not possible because the inverse of most digital system transfer functions is unstable, but one can choose to make the compensator match the inverse of the steady state frequency response of the system. Introducing this compensator into Equation (2-1) and shifting to the \( z \)-domain produces the general form of an RC law \( R(z) \)

\[
U(z) = z^{-\beta}[U(z) + F(z)E(z)] \\
U(z) = \left[ \frac{F(z)}{z^\beta - 1} \right] E(z) = R(z)E(z)
\] (2-2)

In practice it is not reasonable to expect that one knows the phase accurately all the way to Nyquist frequency, and as a result one expects to need to cut off the learning process above some frequency. To do this, one introduces a zero-phase low-pass filter \( H(z) \) that filters the command before it is applied to the system (multiplies the right hand side of the first equation in (2-2)), to prevent changes in the command at frequencies that do not learn. See References [7], [12], and [13]. Usually you do not know at what frequency your model becomes too poor to get decay of the error. Therefore, one designs the RC based on one’s model, applies the law to hardware, and
observes the output. Usually the output error grows slowly since the error is at high frequency with low amplitude, and one can do a frequency content analysis of the error to determine what frequency components are growing. Then pick the cutoff in $H(z)$ accordingly. Since this is done in hardware tuning, we do not consider $H(z)$ in the design stage discussed here.

Combine the equations of the system and the RC law (2-2) and solve for the error $E(z)$ as a function of the desired output $Y_d(z)$ and the periodic disturbance $V(z)$ to obtain

$$\{z^p - [1 - G(z)F(z)]\}E(z) = [z^p - 1][Y_d(z) - V(z)]$$

(2-3)

Because the desired trajectory and the disturbance are periodic with period $p$ time steps, the right hand side of this equation is zero, so it is a homogeneous equation for the error. When the expression in curly brackets is cleared of fractions, the resulting polynomial is the characteristic equation of the associated difference equation. If all roots are inside the unit circle, then the system is asymptotically stable and the error converges to zero as the time step number goes to infinity.

The number $p$ is the number of time steps in a period which is likely to be a large number. This fact together with the fact that there are $p$ poles of the transfer function on the unit circle makes it hard to apply typical stability analysis methods from control theory. Note that Equation (2-3) can be rewritten as

$$z^p E(z) = [1 - G(z)F(z)]E(z)$$

(2-4)

Heuristic thinking says that if one substitutes $z = e^{j\omega T}$ ($T$ is the sample time interval) into the square bracket term to form a frequency transfer function, then the magnitude $M(\omega)$ of this transfer function appears to be the factor by which the error decreases from one period to the next for each frequency $\omega$. Then if $M(\omega) < 1$ for all $\omega$ one predicts that all frequency components of the error go to zero as the periods progress. This thinking is not rigorous because
steady state frequency response only applies in steady state, and once one reaches steady state
the $E(z)$ on both sides of the equation is zero. So a quasi steady-state assumption is being made
going from one period to the next. References [18] and [19] study the settling time, or the decay
of each frequency component with periods, for repetitive control systems. In particular,
Reference [18] shows that for values of $p$ that are not particularly small, $M(\omega)$ is a very good
representation of the decay rate in repetitive control systems, in spite of the fact that the learning
may be dramatically violating the quasi steady state assumption. This result is justified
mathematically, showing the relationship between the decay of the solutions of the homogeneous
difference equation and the decay of the frequency components.

The actual stability boundary can be established using an approach that uses Nyquist
stability like thinking, modified to bypass the issue of having $p$ roots on the unit circle, as
presented in References [9] and [10]. The result is that the repetitive control system is
asymptotically stable if and only if the plot of $z^{-p}[1-G(z)F(z)]$ does not encircle the point +1 as
$z$ goes around the unit circle. From this one can state that asking for $M(\omega) < 1$, or
\[
\left| 1 - G(e^{i\omega T})F(e^{i\omega T}) \right| < 1
\]
for all $\omega$, guarantees asymptotical stability of the repetitive control system, i.e. it is a sufficient
condition. Furthermore, if one wants a repetitive control system that is asymptotically stable for
all possible periods $p$, then it is both a sufficient condition and a necessary condition. And
finally, one can state that for periods $p$ that are not a particularly small number of times steps, the
difference between condition (2-5) and the actual stability boundary is too small to be of
importance in applications. Reference [20] studies this difference. The phenomenon underlying
how small the difference usually is, comes from the fact that the phase of the factor $z^{-p}$ spins
through $p$ complete revolutions as $z$ goes once around the unit circle. Hence, if condition (2-5) is
violated for even a small interval in frequency, rotating the phase of the points violating Equation (2-5) by \( p \) revolutions most likely will cause the plot to encircle the point +1. Therefore, in making a repetitive control design one should aim to satisfy Equation (2-5). One can view this in terms of the diagram on the left of Figure 2-1. If the plot of the complex numbers \( G(e^{i\omega T})F(e^{i\omega T}) \) for all frequencies up to Nyquist, lies totally within the unit circle centered at +1, Equation (2-5) is satisfied and the RC system is asymptotically stable. Above we commented that the dividing line between decay of error and growth of error might occur as the phase change goes through -90 degrees. Here we see that the stability boundary approaches -90 degrees of phase \( \theta \) as the magnitude \( r_{GF} \) of \( G(e^{i\omega T})F(e^{i\omega T}) \) approaches zero. The magnitude limit on \( r_{FG} \) for this phase is length \( R \), and for \( R \) near zero the allowable phase approaches 90 degrees. Conversely, the bound on the phase gets progressively smaller as \( r_{GF} \) increases. This limit is shown on the right of Figure 2-1 and is further discussed below. Smaller \( r_{GF} \) values have larger tolerance to model error (also pointed out in Reference [8]) and we wish to make use of this in the robustification approach here.

![Diagram](image)

**Figure 2-2. Stability boundary of Equation (2-5) as a unit circle centered at +1, and allowable phase error from the positive real axis versus radial distance.**

The above thinking motivated the development of the very effective repetitive control design method presented in Reference [11]. One picks a compensator in the form of an FIR filter...
$F(z) = a_1 z^{m-1} + a_2 z^{m-2} + \cdots + a_m z^0 + \cdots + a_{n-1} z^{(n-m-1)} + a_n z^{-(n-m)}$

$$= (a_1 z^{n-1} + a_2 z^{n-2} + \cdots + a_m z^{n-m} + \cdots + a_{n-1} z^1 + a_n z^0) / z^{(n-m)}$$  \hspace{1cm} (2-6)

that in applications simply asks that you compute a linear combination of errors observed in the last period each time step. Then, for a given number $n$ of gains or coefficients, pick the coefficients (and the value $m$) to minimize the following cost

$$J = \sum_{j=1}^{N} [1 - G(e^{i\omega_j T}) F(e^{i\omega_j T})]^*[1 - G(e^{i\omega_j T}) F(e^{i\omega_j T})]$$  \hspace{1cm} (2-7)

The asterisk indicates the complex conjugate. The $\omega_j$ are a set of chosen frequencies going from zero to Nyquist frequency. One might prefer to use an integral, but the use of a sum of 180 frequencies works well in the examples. This makes a simple linear problem to solve. One can interpret the motivation for this cost function in several ways. One interpretation is that, as commented above, using compensator $F(z)$ equal to the inverse of the system $G(z)$ would make the repetitive controller learn fast, but the inverses of typical discrete time systems are unstable. Instead, cost function (2-7) aims to use the inverse of the steady state frequency response, which should produce zero error after transients die. A second interpretation is based on the above cited result that the left hand side of Equation (2-5), $M(\omega)$, is a good approximation of the amount of decay in the error for frequency component $\omega$. Therefore, cost function (2-7) is aiming to learn as fast as possible. Note that the method can be very effective as shown in Reference [11] or [10], where the use of 12 gains in compensator (2-6) applied to a third order system resulted in the plot of $G(e^{i\omega_j T}) F(e^{i\omega_j T})$ deviating from +1 by less than 1.5% over all frequencies, corresponding to very fast learning.
2.3 Stability Robustness in Repetitive Control

2.3.1 The Importance of Robustness

Repetitive control is an inverse problem: given the desired output, find an input to the system that would produce that output. As commented above, one cannot usually just invert the model because of instability, but one can invert the frequency response of the model. However, repetitive control does not just invert this, but instead it iterates with the world, getting data from the world behavior, with the aim of inverting the world behavior (for the given desired output) in the limit instead of inverting one’s model of the world. This is important in a number of ways. First to do the inversion based on a model, and not use data, also requires that one have a model of the periodic disturbance. This is harder to know accurately. But we also do not want the final error level we reach to be limited to the fidelity of the model we use in designing the repetitive controller. The final error level reached can only approximately reach the reproducibility level of the hardware, since it looks at error each period as if it will happen in the next period if one does not change the input, and hence one reacts to random effects by increasing their influence. Reference [21] used similar learning control methods on a robot, and reached error levels that were not below the reproducibility level of the hardware on a minute-to-minute basis, but were below the reproducibility level measured on a day-to-day basis. Therefore, the control action was fixing errors of the size of how differently the robot behaves tomorrow compared to today. One is not likely to be able to model such variations, but one can still reach error levels that correspond to inverting models of this fidelity without having such models. Insofar as one wants to achieve such high precision tracking error levels, the ability of RC to converge to zero error with substantial model inaccuracy becomes fundamental. One would like to make repetitive controllers that converge to zero error for as large a range of model error as possible.
2.3.2 Approaches to Robustification

There are two types of model error of importance in repetitive control. The first is robustness to high frequency model error, for example from unmodeled high frequency poles, sometimes called parasitic poles or residual modes. This type of robustness can be called robustness to singular perturbations. The fact that RC asks for zero error at the fundamental frequency of period $p$ time steps, and then at all harmonics up to Nyquist frequency, means that one needs the phase to be right within the tolerance dictated by Figure 2-2 for all these frequencies to Nyquist. As noted above, on the robot testbed in Reference [22], we could not take frequency response data above about 20 Hz, but Nyquist frequency was 200 Hz. Using long rich white noise inputs might get to somewhat higher frequencies, but the accuracy deteriorates quickly. Hence, one uses a cutoff of the learning with a zero phase filter as treated in References [12] and [13], in order to robustify to lack of knowledge of the model or knowledge that is too inaccurate at high frequencies. The second type of robustness is to parameter errors in one’s model, or uncertainty in the coefficients of the model. It is this type of robustness that is treated here. There are various papers that apply robust control methods to iterative learning control or repetitive control, for example using matrix inequality approaches. Such approaches normally are not appropriate to handle the first type of robustness, and often they are not well equipped to capture the actual physical uncertainty in the model without resulting in extra conservative results. Since we know rather precisely what the limits are on the allowable amplitude and phase error in repetitive control, our aim is to directly focus on using these limits, and not suffer from being overly conservative in the result.
2.3.3 The Robustness Limit in Repetitive Control

Various RC design methods approach the problem by simply aiming to produce zero
phase of the product $G(e^{i\omega T})F(e^{i\omega T})$, i.e. aim to make the phase of $F$ be the negative of that of $G$
(Reference [6] handles zeros outside the unit circle in this way). The method from Reference
[11] described above, using Equations (2-6) and (2-7) aim for zero phase and amplitude of unity
in $G(e^{i\omega T})F(e^{i\omega T})$ for all frequencies. Assuming that phase of zero is accomplished, then Figure
2-2 allows us to compute the allowable phase error at each frequency. The triangle in the left of
the Figure 2-is isosceles with two sides equal to unity. If we drop a perpendicular from their
vertex to the third side we create right triangles and can compute the length $R$ to the boundary of
the unit circle for any angle $\theta$ as $R = 2 \cos \theta$, or conversely one computes the phase tolerance as
plus or minus $\theta$ where $\cos \theta = R/2$. Then the allowable deviation from the positive real axis for
any point $G(e^{i\omega T})F(e^{i\omega T})$ with magnitude $r_{FG}$ is given in the right of Figure 2-2. Provided the
point stays inside the region defined by the curve, the system is asymptotically stable.

We can give some interpretation of these results. If one has a point at $+1$ with radial
distance $+1$, then the allowed phase error is $\pm 60$ degrees. As the radial distance tends to zero, the
allowable phase error tends to $\pm 90$ degrees, which is the boundary of the possible phase errors
that can be tolerated and still converge. For a radial distance of $r = 0.1$ then the allowable phase
error for convergence to zero error is $\pm 87.13$ degrees, and for $r = 0.05$ it is $\pm 88.57$ degrees. These
two correspond to learning rates per period of $1 - r_{FG}$, or a factor of $0.9$ and $0.95$ multiplying the
amplitude of the error going from one period to the next. By taking these numbers to higher and
higher powers one can determine how many iterations are necessary to reduce the error by a
factor of one third, or one tenth, etc., creating the analogy to time constants and settling times but
in iterations.
2.3.4 The Cost Function Averaging Approach

References [14] to [17] develop the method of averaging cost functions over model distributions for both iterative learning control and repetitive control for increased robustness. The approach is as follows. Specify any desired probability distribution for each of the uncertain parameters. These could be uniform distributions between upper and lower bounds on each parameter, or one can specify some more sophisticated distribution. Then pick \(2M+1\) models (we prefer an odd number of model) from these distributions of the parameters and denote them by \(G_k(z)\), \(k = 1, 2, ..., 2M + 1\). Good results can be obtained without needing to consider a large number of models. Then instead of minimizing the cost in Equation (2-7) one asks to find a single compensator (2-6) that minimizes cost (2-7) summed (or averaged) over the multiple models

\[
J_{mm} = \sum_{k=1}^{2M+1} \sum_{j=1}^{N} [1 - G_k(e^{i\omega_j}T)F(e^{i\omega_j}T)][1 - G_k(e^{i\omega_j}T)F(e^{i\omega_j}T)]^* \tag{2-8}
\]

This approach is simple, convenient, straightforward, and surprisingly effective. Note that it is trying to make as much as possible all \(G_k(e^{i\omega_j}T)F(e^{i\omega_j}T)\) look like unity, for all models and all frequencies. So it is trying to get these to lie on the positive real axis, and also to lie at the point +1, as much as possible.

2.4 Modifying the Objective to Further Improve Robustness

The cost function (2-7) and also the multiple model cost function (2-8) are both aiming for unit magnitude as much as possible for all frequencies and all models. The aim for +1 magnitude represents an aim to learn fast. The multiple model cost function will try to make the phase angle for all models average close to zero, so its effectiveness is based on adjusting the phase of \(F(z)\) to be close to the negative of the average phases of all models at a given
frequency, so as many models as possible will be inside the unit circle of Figure 2-2. But the approach does not consider the use of adjusting the learning rate by aiming for a radial distance smaller than unity. The robustness limits discussion above shows that this is an important parameter to adjust for improved robustness, and we take advantage of it in the improved algorithm presented here.

Note that one can improve the robustness of any repetitive control law by simply multiplying the RC update by a gain that is very small, making the whole plot of $G_k(e^{i\omega_j T}) F(e^{i\omega_j T})$ approach the origin, and the phase error tolerance relative to the positive real axis approach ±90 degrees. But this produces robustness at the expense of rate of convergence to zero error, for all frequencies up to Nyquist. The proposed algorithm starts like the multiple model approach of Equation (2-8), generating a set of $2M+1$ models representative of the distributions of parameter uncertainties in the model coefficients. The first objective in forming a proposed algorithm is that we want to learn as fast as possible at each frequency, consistent with the need for robustness at that frequency. A second objective is to have an appropriate method of handling the fact that it may not be possible to stabilize all models in the distributions. And a third aspect of the approach is to impose a limit on how slow the learning can get, i.e. a limit on the minimum value of $r_{FG}$ that one aims at.

In the cost function (2-8) we could reverse the order of the summations, so that the inner summation sums over all models for one frequency, and the outer sum is over all frequencies. So we can proceed by isolating each frequency $\omega_j$ and optimizing the design of the compensator $F(e^{i\omega_j T}) = r_{F,j} e^{i\theta_{F,j}}$ phase and amplitude to minimize the cost

$$J_j = \sum_{k=1}^{2M+1} [1 - G_{k,j} r_{F,j} e^{i\theta_{F,j}}] [1 - G_{k,j} r_{F,j} e^{i\theta_{F,j}}]^*$$

(2-9)
where $G_{k,j} = G_k(e^{j\omega_j T})$. After minimizing this at each $j$, we replace the value of $r_{F,j}$ by $\hat{r}_{F,j}$ which is a reduced value when needed to get all models inside the unit circle. Then design an implementable compensator in the form of the FIR filter Equation (2-6) aiming to make (2-6) match $\hat{r}_{F,j} e^{j\theta_{F,j}}$ over all frequencies, by minimizing

$$J_F = \sum_{j=1}^{N} [1 - F(e^{j\omega_j T})(1/\hat{r}_{F,j})e^{-j\theta_{F,j}}][1 - F(e^{j\omega_j T})(1/\hat{r}_{F,j})e^{-j\theta_{F,j}}]^*$$

over the coefficients in Equation (2-6) and examining the range of choices of $n$ and $m$. Thus, we no longer are aiming for +1 at all frequencies, but instead at the largest value less than or equal to +1 that is consistent with the uncertainty level. We also consider limiting $\hat{r}_{F,j}$ to a chosen $r_{\text{min},j}$, to allow the designer to make a choice between robustness and learning rate if the learning rate dictated is unreasonably slow.

### 2.5 An Algorithm for Improved Robustification of RC Systems to Model Parameter Uncertainties

**Step 1:** Setup. Specify the uncertainties of the parameters including any chosen limits and the distributions for each. Pick $2M+1$ representative models $G_k(z)$ from these distributions. If desired, for each frequency $\omega_j$ specify the minimum acceptable learning speed by picking $r_{\text{min},j}$. If the model is on the real axis, then the decrease in error at this frequency from one period to the next is $(1-r_{\text{min},j})$, and otherwise one can compute the associated learning rate.

**Step 2:** For each $\omega_j$ compute the $r_{F,j}, \theta_{F,j}$ that minimize Equation (2-9). Define

$$\sum_{k=1}^{2M+1} G_{k,j} = \rho_{G,j} \exp(i\phi_{G,j}).$$

Then $\partial J_j / \partial \theta_{F,j} = 0$ and $\partial J_j / \partial r_{F,j} = 0$ produce

$$\rho_{G,j} \sin(\phi_{G,j} + \theta_{F,j}) = 0$$

$$r_{F,j} = \left[ \rho_{G,j} e^{i(\phi_{G,j} + \theta_{F,j})} + \rho_{G,j} e^{-i(\phi_{G,j} + \theta_{F,j})} \right] / \left[ \sum_{k=1}^{2M+1} 2G_{k,j}G_{k,j}^* \right]$$

(2-11)
This tells us that the compensator phase \( \theta_{F,j} = -\phi_{G,j} \) should be minus the phase of the sum of the \( G_{k,j} \), which makes the phase of the average model lie on the positive real axis. Then

\[
\hat{r}_{F,j} = \left| \sum_{k=1}^{2M+1} G_{k,j} \right| \left[ \left| \sum_{k=1}^{2M+1} |G_{k,j}|^2 \right| \right]^{1/2}
\]

(2-12)

**Step 3:** In this step we make adjustments to the magnitude of \( F(e^{i\omega T}) \) that we aim form, modifying \( r_{F,j} \) to \( \hat{r}_{F,j} \). Note that a point \( G_k(e^{i\omega T})F(e^{i\omega T}) = r_{GF,j,k}(\exp(i\theta_{GF,j,k})) \) is inside the unit circle, and satisfies stability condition (2-5) provided that

\[
(r_{GF,j,k} \cos \theta_{GF,j,k} - 1)^2 + (r_{GF,j,k} \sin \theta_{GF,j,k})^2 < 1,
\]

which can be used to decide which models at each frequency are outside the unit circle in Figure 2-2. For each \( j \) this determines which models require one make \( \hat{r}_{F,j} \) smaller than \( r_{F,j} \), and one sees how much reduction is necessary for each such model. Figure 2-3 illustrates the reduction process.

Two conditions can apply for a given \( j \): (i) If no model is outside, then stability condition (2-5) is satisfied using \( r_{F,j} e^{i\theta_{F,j}} \) for the compensator, and in the next Step one aims for this point in designing the compensator using cost function (2-10). (ii) Otherwise, compute the amount of reduction in radial distance needed for each model to bring it into the unit circle. One must reduce the magnitude of \( G_k(e^{i\omega T})r_{F,j} e^{i\theta_{F,j}} \) to be less than \( 2 \cos \theta \) by reducing \( r_{F,j} \) to the needed value \( \hat{r}_{F,j} \), where \( \theta \) is the angle of \( G_k(e^{i\omega T})r_{F,j} e^{i\theta_{F,j}} \). Look through all models at frequency \( j \) and find the smallest such factor, and apply to all models in order to stabilize all models.

We can two kinds of additional modifications. Figure 2-3 shows the stability boundary as in the right part of Figure 2-2. Remember that we will be fitting an FIR filter to the chosen \( \hat{r}_{F,j} e^{i\theta_{F,j}} \) for all frequencies, and this will not be perfect. Hence, one may wish to include a phase tolerance, staying some distance away from the unit circle stability boundary. Figure 2-3 has a
dashed curve computed to allow a phase tolerance of $\pi/10$ radians (18 degrees). This particularly large tolerance is used for illustration purposes, and a value of 6 degrees in used in the example computations below. Regions III and IV cannot be stabilized by shrinking the amplitude of $F$. If any points in this part of the plot correspond to high frequencies, then the zero phase low pass filter $H$ can cut off the learning and stabilize these models if necessary. One might apply the designed law to hardware and only make such adjustments if hardware behavior necessitated it. Points in Region I (defined as inside the solid curve if not tolerance is used, and inside the dashed curve if a tolerance is used) need no adjustment. And points in Region II require shrinking the value of $\hat{r}_{F,j}$ until reaching either the solid or the dashed curve, whichever applies.

As a final consideration in making the adjustment of $\hat{r}_{F,j}$, one may examine the value of $\hat{r}_{F,j}$ for all $j$ to see if any are below the slowest acceptable learning rate $r_{\text{min},j}$ (Region V). If so one makes a decision whether to sacrifice trying to stabilize such models in favor of ensuring that the learning is accomplished in a reasonable number of iterations. If so, then one increases the $\hat{r}_{F,j}$ to this minimum acceptable value $r_{\text{min},j}$.

**Step 4:** It remains to design $F(z)$ of the form in Equation (2-6), finding values of the gains $a_i$ for $i = 1, 2, 3, \ldots, n$ that minimize the cost (2-10). The cost is a simple quadratic function of these coefficients so there is an immediate solution obtained by solving a set of simultaneous linear equations with dimension equal to the number of coefficients. One examines several values of $m$ to get the best solution for a given $n$, and one adjusts $n$ until one gets the desired accuracy.
Figure 2-3. Adjustment of compensator magnitude frequency response $\hat{r}_{F,j}$ for frequency $j$.

2.6 Numerical Examples

Consider a third order continuous time system whose Laplace transform is given by $G(s) = \frac{a}{s} \frac{\omega_0^2}{s^2 + 2\zeta \omega_0 s + \omega_0^2}$ whose input comes through a zero order hold sampling at 200 Hz sample rate. The nominal values of $a = 8.8$, $\omega_0 = 12\pi$, $\zeta = 0.1$. The undamped natural frequency is considered uncertain, and we pick 21 models spaced every $0.25\pi$ between $9.5\pi$ and $14.5\pi$. The left plot in Figure 2-4 gives the results of the design at Step 2 above. The unit circle is shown, and the dashed line gives the boundary associated with a 6 degree margin for extra phase error. There is one curve for each of the 21 frequency transfer functions $G_k(e^{i\omega_j T})r_{F,j}e^{i\theta_{F,j}}$. Note that a number of models go outside the unit circle and outside the dashed boundary. The small circles give the points for all 21 models that correspond to frequency 30.94 rad/sec, which is near the resonant peak of the system. The right part of Figure 2-4 give the corresponding results when the compensator is formed as in Equation (2-6) by minimizing (2-10) using $\hat{r}_{F,j}$ set
to \( r_{F,j} \), i.e. Step 3 is skipped and no adjustment of the magnitude is made. The value of \( n \) is 200 and \( m \) is 101. We observe that there is very little degradation in performance going from the desired frequency response for \( F \) to the frequency response of the FIR implementation of \( F \).

Figure 2-5 presents the corresponding plots including Step 3, and using the 6 degree tolerance. No \( r_{\min,j} \) was considered necessary. The left plot is for the chosen values of \( \hat{r}_{F,j}(\exp(i\theta_{F,j})) \) in Step 3, following Figure 2-3, and the right plot uses the FIR implementation of \( F \). This time there is degradation with one model going outside the dashed curve, but still remaining stable and inside the unit circle, demonstrating the benefit of incorporating the tolerance.

![Polar plot of \( G_k(e^{i\omega T})F(e^{i\omega T}) \) using the \( F(e^{i\omega T}) \) obtained in Step 2 (left) and using a 200 gain FIR approximation of \( F(e^{i\omega T}) \) from Equation (2-6) (right).](image)

**Figure 2-4.** Polar plot of \( G_k(e^{i\omega T})F(e^{i\omega T}) \) using the \( F(e^{i\omega T}) \) obtained in Step 2 (left) and using a 200 gain FIR approximation of \( F(e^{i\omega T}) \) from Equation (2-6) (right).
Figure 2-5. Same as Figure 2-4 except that the reduction in magnitude in Step 3 has been performed.

The results shown on the left in Figures 4 and 5 are plotted in a different manner in Figure 2-6. This time the format in Figure 2-3 and on the right in Figure 2-2 is used, again with the stability boundary of Equation (2-5) and the boundary including the 6 degree margin. The lines shown connect the values for all 21 models for a given frequency. The left plot does not make use of Step 3 and there are models that violate the stability boundary and the dashed curve boundary. The right plot makes the adjustments in Step 3 and stabilizes all models.

Figure 2-7 gives Bode plots all 21 models $G_k(e^{j\omega T})$. The dot-dash curve in the middle is the nominal model. The stars correspond the inverse of the compensator chosen in Step 3, while the nearby dashed curve is the corresponding result from Step 4 after designing the FIR form of the compensator. Note how the phase of the stars runs through the center of the phase spread of the models in the bottom plot. Also, the peak in the stars in the top plot correspond to the need to reduce the gain in F for these frequencies in order to stabilize certain models with large phase error.
Figure 2-6. The left plots in Figures 4 (left) and 5 (right) presented in the form of Figure 2-3.

Figure 2-7. Bode plots of the 21 models $G_k(e^{i\omega T})$, including nominal model (dot-dash), and the reciprocal of the compensator design in Step 3 (stars) and Step 4 (dashed curve).
Figure 2-8 presents the gains of the compensator Equation (2-6), \( a_i \), as a function of index \( i \). The \( i=1 \) corresponds to the most recent error and \( i=200 \) the oldest error used in the compensator computation. The nature of the plot suggests that the gains near the left and right edges of the plot might not be needed and might be poorly identified. Figure 2-9 displays the accuracy of Step 4 in making an FIR filter to implement the chosen frequency response function in Step 3. The left plot represents the phase difference between the desired compensator in Step 3 and the implemented compensator in Step 4. The dashed lines correspond to a phase difference of \( \pm 5 \) degrees. The right plot is the ratio of the magnitude of the compensator in Step 4 divided by the intended magnitude in Step 3. One wants this ratio to be unity.

2.7 Conclusions

(1) The cost function averaging approach to robustifying iterative learning and repetitive control can be surprisingly effective at stabilizing a set of models with uncertain parameters, but to accomplish its results it emphasizes adjustment of the phase of the compensator and still aims for fast learning.

(2) The proposed algorithm in addition adjusts the learning rate, so that it should be able to consistently outperform cost function averaging.

(3) The objective is to decrease the learning rate as a function of the uncertainty at each frequency. It does not slow down the learning at all frequencies, but rather slows down the learning in any frequency range where it is needed. For example when there is a sharp resonant peak which produces large and fast changes in phase, and when the location of this peak is somewhat uncertain, a slow learning rate is supplied to robustify convergence, allowing larger phase errors.
(4) The learning is slowed down selectively and slowed down only as much as is required for robustness to the uncertainty associated with each frequency.

(5) When some model requires a particularly large number of iterations to converge to zero error using the compensator designed for the set of models, the designer can evaluate the relative importance of stabilizing that model with resulting slow learning, or ignoring that possible model to have a reasonable learning rate for the remaining model candidates of the true world model.

(6) The choice of the phase change that one aims to achieve in the compensator design is seen to be decoupled from the choice of the magnitude change. The phase change is based on the average phase of all models. In some applications uncertainty distribution in the coefficients might be sufficiently skewed that one could make a better choice for the phase aim point.

2.8 References


Chapter 3

The Influence on Stability Robustness of Compromising on the Zero Tracking Error Requirement in Repetitive Control

3.1 Introduction

Chapter 2 has proposed an algorithm for Repetitive Control (RC) robustification particularly to model parameter uncertainties. There is however still a hard limit on the model phase error allowed. The purpose of this chapter is to investigate the benefit of compromising on the usual zero-tracking error requirement in standard RC designs for extra stability robustness. A quadratic cost design method is used here that also includes a penalty on the size of the control action for frequencies that would otherwise be cut out. The analysis is first based on ILC, and then the RC design is constructed using the resulting frequency response. The simulation results (e.g. error sensitivity transfer function) show that by compromising on the strict zero-tracking error requirement, the system obtains extra robustness and even improved tracking performance compared to the traditional design where a cutoff filter is used.

3.2 An Effective Repetitive Control Design Procedure

This section sets up the structure of the repetitive control problem together with the design process recommended in Reference [7]. This involves two separate steps, the design of a compensator by the method in [16], and the design of the zero-phase low-pass filter to cut off the learning process as presented in [8] and enhanced in [9]. Consider a digital feedback control system with a desired output $Y_D(z)$ in $z$-transform form, and a periodic output disturbance $W_o(z)$
that is equivalent to any periodic disturbance wherever it appears in the system. Each is considered to be periodic with \( p \) time steps with sample time interval \( T \) seconds. This includes the possibility that the desired output is a constant or zero which are periodic functions with any period. The actual system output and its error are given by

\[
Y(z) = G(z)U(z) + W_o(z) \\
E(z) = Y_D(z) - Y(z) \\
E(z) = -G(z)U(z) + [Y_D(z) - W_o(z)] = G(z)U(z) + W(z)
\]

where \( G(z) \) is a feedback control system transfer function and \( U(z) \) is its command input which is adjusted by the repetitive control aiming to improve the tracking accuracy. The simplest form of repetitive control uses the following logic, if the feedback control system output at this phase of the previous period was 2 units too low, then add 2 units, or 2 units multiplied by a gain \( \phi \), to the command this period. Assuming the time delay from change in command to the first time step one sees an influence of the change on the output is one time step, then one looks back one period and then forward one time step, resulting in the time domain rule at the \( k \)th time step given by \( u(k) = u(k - p) + \phi e(k - p + 1) \). This rule fails to work when the phase lag through the system becomes 180 degrees, in which case it adds to the error instead of reducing it. As a result one needs to design a compensator \( F(z) \) that multiplies the error and puts in appropriate phase lead (lag) that aims to cancel the phase lag (lead) going through the system. If the model is not accurate enough at high frequency for this to make high frequency components of the error decay, one needs a zero-phase low-pass filter \( H(z) \) to stop any adjustments of the command to the feedback control system for such frequency components. The resulting repetitive controller takes the form

\[
U(z) = z^{-p}H(z)[U(z) + F(z)E(z)] ; \quad U(z) = \left( \frac{H(z)F(z)}{z^p - H(z)} \right) E(z)
\]
and the difference equation for the error can be written as
\[
\{z^p - H(z)[1 - G(z)F(z)]\} E(z) = [z^p - H(z)] W(z)
\] (3-3)

When there is no cutoff, then the right hand side is zero since \( W(z) \) is considered periodic with period \( p \) time steps. To consider the influence of disturbances at frequencies that are not of period \( p \) time steps, we note the sensitivity transfer function from disturbance to error
\[
E(z) = S(z) W(z) = \left( \frac{z^p - H(z)}{z^p - H(z)[1 - G(z)F(z)]} \right) W(z)
\] (3-4)

This gives a particular solution for the difference Equation (3-3). The homogeneous equation can be written as
\[
z^p E(z) = H(z)[1 - G(z)F(z)] E(z)
\] (3-5)

which suggests that every frequency component of the solution to the homogeneous equation will decay with time if the magnitude of the frequency transfer function is less than unity at all frequencies, i.e. the repetitive control system will be asymptotically stable if
\[
\left| H(e^{j\omega T})[1 - G(e^{j\omega T})F(e^{j\omega T})] \right| < 1 \quad \forall \omega
\] (3-6)

This argument is not rigorous, but [6,7] establish that this is a necessary and sufficient condition for asymptotic stability of a repetitive control system when one wants asymptotic stability for all possible periods \( p \).

References [16] and [8] design FIR filters for \( F(z) \) and \( H(z) \). The compensator has the form below and optimizes the following cost
\[
F(z) = a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{m-1} z^{-(n-m-1)} + a_m z^{-(n-m)}
\]
\[=(a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{m-1} z^{-(n-m-1)} + a_m z^{-(n-m)}) / z^{(n-m)}\quad (3-7)
\]
\[
J = \sum_{j=0}^{N} [1 - G(e^{j\omega T})F(e^{j\omega T})] W_j [1 - G(e^{j\omega T})F(e^{j\omega T})]^{*}
\]
The sum is taken over an appropriately chosen discrete set of frequencies between zero and Nyquist. This produces an FIR filter that aims to match the inverse of the steady state frequency response of the system. The low pass filter takes the form below and optimizes the cost below

$$H(z) = \sum_{k=-n}^{n} a_k z^k \quad ; \quad J_H = \alpha \sum_{j=0}^{J} [1 - H(e^{i\omega_j T})][1 - H(e^{i\omega_j T})]^* + \sum_{j=1}^{N-1} [H(e^{i\omega_j T})][H(e^{i\omega_j T})]^*$$  \hspace{1cm} (3-8)

The first sum is over the pass band frequencies, and the second over the stop band frequencies, and the coefficients with negative subscripts must be the same as those with positive subscripts in order to produce zero phase.

**3.3 ILC Design by Quadratic Cost with Control Effort Penalty: Compromising on the Zero Error Requirement**

We first design an iterative learning control law that compromises on the zero error requirement, and then convert to a repetitive control law. A quadratic cost function is used. The normal quadratic cost in ILC has a quadratic penalty on the sum of the squares of the error and a similar term on the sum of the changes in the command input to the feedback control system from the previous repetition [17,18]. Thus only the amount of change in the control is penalized, the control can accumulate to whatever value it likes, and the design aims for zero tracking error. The change in the control penalty serves the purpose of controlling the transients during the learning.

Consider a system in state variable form

$$x(k+1) = Ax(k) + Bu(k)$$
$$y(k) = Cx(k) + w_o(k)$$  \hspace{1cm} (3-9)

and define underbar column matrices whose entries are the history of the corresponding variable for all $p$ time steps of a run.
\[
\begin{align*}
\begin{array}{c}
\begin{pmatrix}
\vdots \\
y(1) \\
\vdots \\
y(p) \\
\end{pmatrix} = \\
\begin{pmatrix}
\vdots \\
u(0) \\
\vdots \\
u(p-1) \\
\end{pmatrix}
\end{array}
= \\
\begin{pmatrix}
\begin{array}{c}
\vdots \\
\begin{pmatrix}
CA \\
\cdots \\
\end{pmatrix} \\
\end{array}
\begin{pmatrix}
\begin{array}{c}
\vdots \\
CB \\
\cdots \\
\end{pmatrix}
\begin{pmatrix}
\begin{array}{c}
\vdots \\
\begin{pmatrix}
CA > & 0 & \cdots & 0 \\
\end{pmatrix} \\
\end{array}
\begin{pmatrix}
\begin{array}{c}
\vdots \\
\begin{pmatrix}
CA > & 0 & \cdots & 0 \\
\end{pmatrix}
\end{array}
\begin{array}{c}
\vdots \\
\begin{pmatrix}
CA > & 0 & \cdots & 0 \\
\end{pmatrix}
\end{array}
\end{array}
\end{array}
\end{align*}
\]

(3-10)

Column matrices \(y_D, w_o, w, e\) are defined analogous to \(y\). Then Equations (3-1) for this ILC problem become

\[
y = Pu + Ax(0) + w_o ; \ e = y_D - y = -Pu + [y_D - w_o - \bar{A}x(0)] = -Pu + \bar{w}
\]

(3-11)

Matrix \(P\) is a Toeplitz matrix of Markov parameters and the product \(Pu\) produces the zero initial condition particular solution corresponding to \(G(z)U(z)\) where \(G(z) = C(zI - A)^{-1}B\).

References [19, 15] established the relationship between the singular value decomposition \(P = USV^T\) and the frequency response of \(G(z)\). As the value of \(p\) tends to infinity, the singular values in \(S\) converge to the steady state magnitude frequency response of the transfer function, at the discrete set of frequencies observable from \(p\) time steps of data. The right and left singular vectors converge to sinusoids at these frequencies, and the phase difference between the sinusoid in the input singular vector (column of \(V\)) and the sinusoid in the corresponding output singular vector (column of \(U\)) contains the phase change through the system.

### 3.3.1 The Optimization Criterion

Introduce a subscript \(j\) to denote the run or iteration number in ILC. Also define a delta operator in the iteration domain which when applied to any quantity \(\xi\) produces \(\delta \xi = \xi_j - \xi_{j-1}\).

The usual quadratic cost in ILC only penalizes this difference of the command input, not the command input itself. We introduce the latter penalty so that the minimization does not converge to zero error. At iteration \(j\) the cost function to minimize is then

\[
J_j = \xi_j^T \xi_j + \delta_j^T \bar{R}_j \delta_j + u_j^T \bar{R}_u u_j
\]

(3-12)
The matrix $\mathbf{R}_d$ controls the amount of learning for each iteration. Slower learning improves robustness to model error. The new matrix $\mathbf{R}_u$ governs how far from zero error one will be once the learning process converges. Define hat quantities as follows

$$
\hat{e}_j = U^T e_j ; \quad \hat{u}_j = V^T u_j ; \quad \hat{w} = U^T w ; \quad \mathbf{R}_d = V R_d V^T ; \quad \mathbf{R}_u = V R_u V^T
$$

(3-13)

By converting to the hat variables we convert to what become the discrete frequency components (for the discrete frequencies that can be computed from $p$ time steps of data) as the value of $p$ gets large. We will denote the $k^{th}$ component of the hat variables as an argument, where we start with one and progress to $p$ for all variables including $\hat{u}_j$. Then as $p$ gets large, $k$ denotes a frequency component, each frequency having two components corresponding to the need for sine and cosine to span each frequency’s space. The cost function can now be written in terms of these new variables as

$$
J_j = \hat{e}_j^T \hat{e}_j + \delta_j \mathbf{R}_d \hat{u}_j + \hat{u}_j^T \mathbf{R}_u \hat{u}_j
$$

(3-14)

$$
\mathbf{R}_d = \text{diag}(r_d(1), r_d(2), \cdots, r_d(p)) ; \quad \mathbf{R}_u = \text{diag}(r_u(1), r_u(2), \cdots, r_u(p))
$$

We choose to pick the weight matrices diagonal in which case, as $p$ gets large we are supplying weighting factors for each frequency component. As a result, the cost can be decomposed frequency by frequency, so that one solves the optimization problem independently for each $k$

$$
J_j = \sum_{k=1}^{p} J_j(k) ; \quad J_j(k) = \hat{e}_j^2(k) + r_d(k)(\delta \hat{u}(k))^2 + r_u(k)(\hat{u}_j(k))^2
$$

(3-15)

### 3.3.2 The ILC Law

In order to determine the optimal change to make in the command input, $\delta \hat{u}(k)$, one computes $dJ_j(k)/d\delta \hat{u}(k) = 0$. The dependence of the last term on the change in command is $\hat{u}_j(k) = \hat{u}_{j+1}(k) + \delta \hat{u}(k)$. The dependence in the first term is computed as follows. From Equation
(3-11), \( \delta_j \varepsilon = -P \delta_j u \) or \( \varepsilon_j = \varepsilon_{j-1} - P \delta_j u \). Then converting to hat variables results in the uncoupled set of equations \( \hat{\varepsilon}_j = \hat{\varepsilon}_{j-1} - S \delta_j \hat{u} \). The needed relationship is then the \( k \)th component given as \( \hat{\varepsilon}_j(k) - \sigma(k) \delta_j \hat{u}(k) \) where \( \sigma(k) \) is the \( k \)th singular value. The result for \( k \) is

\[
\hat{u}_j(k) = H(k)[\hat{u}_{j-1}(k) + F_1(k) \hat{\varepsilon}_{j-1}(k)]
\]

\[
H(k) = \frac{\sigma^2(k) + r_u(k)}{\sigma^2(k) + r_u(k) + r_a(k)}; \quad F_1(k) = \frac{\sigma(k)}{\sigma^2(k) + r_a(k)}
\]

(3-16)

One can return to the original un-hatted variables

\[
u_j = H[u_{j-1} + V^T diag(1/(\sigma^2(k) + r_u(k)))V^T SU^T \varepsilon_{j-1}]
\]

\[
\hat{u}_j = H[u_{j-1} + EP^T \varepsilon_{j-1}]
\]

\[
H = V^T diag(H(k))W; \quad F = V^T diag(1/(\sigma^2(k) + r_a(k)))V^T
\]

3.3.3 Stability and Final Error Level

In order to examine stability, consider Equation (3-16) and eliminate the error component in favor of the input forcing function. Starting with \( \varepsilon_{j-1} = - Pu_{j-1} + w \) and converting to hatted variables produces \( \hat{\varepsilon}_{j-1} = -S \hat{u}_{j-1} + \hat{w} \). This results in the difference equation for the control input

\[
\hat{u}_j(k) = [H(k) - \sigma(k)F_1(k)] \hat{u}_{j-1}(k) + H(k)F_1(k) \hat{w}(k)
\]

\[
[H(k) - \sigma(k)F_1(k)] = r_a(k)/(\sigma^2(k) + r_a(k) + r_u(k))
\]

(3-18)

Clearly this is a stable difference equation ensuring that the command input \( \hat{u}_j(k) \) converges as \( j \) tends to infinity, and if the command converges so does the error history, producing a stable ILC law.

The final value of the command input is obtained by recognizing that \( \hat{u}_j(k) \) and \( \hat{u}_{j-1}(k) \) will both equal the same converged value in the limit, \( \hat{u}_\infty(k) \). Solving for this and also substituting it into \( \hat{\varepsilon}_{j-1} = -S \hat{u}_{j-1} + \hat{w} \) gives the final value of the command input and the final error

\[
\hat{u}_\infty(k) = (\sigma(k)/(\sigma^2(k) + r_u(k)))\hat{w}(k)
\]

\[
\hat{e}_\infty(k) = (r_a(k)/((\sigma^2(k) + r_u(k)))\hat{w}(k)
\]

(3-19)
3.3.4 Robustness to Model Error

It is of interest to be able to study the effect of model error on the stability of the iterative learning control law (16) or (17). Define the singular value decomposition of the matrix $P$ associated with the model used in designing the learning law as $P_M = U_M S_M V_M^T$, and similarly define the singular value decomposition of the matrix $P$ that applies to the actual world behavior as $P_W = U_W S_W V_W^T$. Because these are different we cannot convert to the decoupled form when studying the behavior of an ILC law designed from model $P_M$ when applied to system $P_W$. So we start with the cost function in the form of Equation (3-12). Since we are designing based on the model, the first term in the cost will use $\epsilon_j = \epsilon_{j-1} - P_M \delta_j u$ although of course it is not correct.

Then the control update equation is obtained by computing $dJ_j / d(\delta_j u) = 0$,

$$u_j = \{I + [P_M^T P_M + \bar{R}_d + \bar{R}_u]^{-1} \bar{R}_u\} u_{j-1} + [P_M^T P_M + \bar{R}_d + \bar{R}_u]^{-1} P_M^T \epsilon_{j-1}$$  \hspace{1cm} (3-20)

In order to study stability, we again eliminate the error term in favor of an expression with the forcing function, $\epsilon_{j-1} = -P_W u_{j-1} + w$, producing

$$u_j = \{I - [P_M^T P_M + \bar{R}_d + \bar{R}_u]^{-1} [P_M^T P_M + \bar{R}_u]\} u_{j-1} + [P_M^T P_M + \bar{R}_d + \bar{R}_u]^{-1} P_M^T w$$  \hspace{1cm} (3-21)

This is a difference equation for updating $u_j$, and it will converge asymptotically as $j$ tends to infinity if all eigenvalues of the matrix in curly brackets are less than unity in magnitude. Substitute the singular value decompositions into this matrix and factor out a $V_M$ in front of the matrix and a $V_W^T$ to the right of the matrix, and what remains in the center must have all eigenvalues less than unity in order for the ILC to be asymptotically stable. This matrix is

$$\{I - \text{diag}(1/(\sigma_M^2(k) + r_d(k) + r_u(k))) \{[(U_M^T U_W) S_W (V_M^T V_W) + R_u]\} \}$$  \hspace{1cm} (3-22)

If there were no model error, then $(U_M^T U_W)$ and $(V_M^T V_W)$ would be identity matrices, and the result reverts to the previous stability result. As noted before, the phase information is contained in the
relationship between corresponding input and output singular vectors, so these terms are related
to phase errors. Note that when one adds a constant \( \alpha \) to every term on the diagonal of a matrix,
then one adds the same amount to every eigenvalue of the matrix. Suppose we pick matrix \( R_u \) to
have \( \alpha \) for every entry on the diagonal. We could pick the elements on the diagonal of \( R_u \) such
that the \( R_u \) factor subtracts the same constant from every term on the diagonal. This tells us that
we can always stabilize the ILC law by using big enough values in \( R_u \). Of course this will most
like produce large final error levels so more finely tuned values would be appropriate. One can
find an expression for the final error level by again finding \( u \) from (21) substituting into
\[
e_{j-1} = -P_w u_{j-1} + \omega.
\]

3.4 Creating RC Based on the Frequency Response of the ILC Design

3.4.1 The Control Law in the Frequency Domain

Iterative learning control asks to converge to zero error at every time step of a finite time
trajectory. Hence, it aims for zero error during the transients as well as during whatever part of
the trajectory might be considered steady state, i.e. after the transients have decayed. Repetitive
control on the other hand asks to converge as the time step number tends to infinity. Hence, it is
appropriate to study repetitive control using steady state frequency response concepts. Although
the control update rule developed above is not designed for a repetitive control formulation we
can use its frequency response characteristics for repetitive control. To do this first recognize that
the product \( P_u \) forms the convolution sum solution for zero initial conditions which in the \( z \)-
transform domain is the product of \( G(z) \) with the transform of \( u \). As \( j \) tends to infinity, singular
values converge to the frequency magnitude response so that \( \sigma(k) \) corresponds to \( |G(e^{i\omega T})| \). The
Taylor series expansion \( G(z) = CBz^{-1} + CABz^{-2} + CA^2Bz^{-3}L \) helps show the relationship of \( G(z) \) to the \( P \) matrix. Note that \( G(z^{-1}) = CBz^{-1} + CABz^{-2} + CA^2Bz^{-3}L \) corresponds to \( P^T \). Combining these statements gives the command update formula in the frequency domain as

\[
U_j(e^{i\omega kT}) = H(e^{i\omega kT})[U_{j-1}(e^{i\omega kT}) + F(e^{i\omega kT})E_{j-1}(e^{i\omega kT})]
\]

\[
H(e^{i\omega kT}) = \frac{|G(e^{i\omega kT})|^2 + r_d(k)}{|G(e^{i\omega kT})|^2 + r_d(k) + r_u(k)}; \quad F(e^{i\omega kT}) = \frac{G(e^{-i\omega kT})}{|G(e^{i\omega kT})|^2 + r_d(k)} \quad (3-23)
\]

Note that the \( H \) is a zero phase filter as before. Also, the \( F \) design is independent of the choice of the control penalty \( R_u \) as it was before. The repetitive control design method described in the second section of this paper was a two step process, first design the compensator \( F \), and then design a cutoff filter. Although the cost function used here designs both filters, the compensator is still an independent design. What is new is that \( H \) is no longer considered to be a cutoff filter, but rather something that is used to stabilize. And perhaps not cutting off sharply can have advantages.

### 3.4.2 Implementation with FIR Filters

In order to create an implementable repetitive controller based on this frequency response behavior, we design an FIR filter and FIR compensator as in Equations (3-8) and (3-9). So we find coefficients for the zero phase filter to minimize the cost below

\[
H_a(z) = \sum_{k=-n}^{n} a_k z^k; \quad J_H = \sum_{k=1}^{n} |1 - H_a(e^{i\omega kT})H^{-1}(e^{i\omega kT})|^2 \quad (3-24)
\]

Again, the \( a_k \) with negative subscripts are equal to their counterpart with positive subscripts in order to form a zero phase filter. And similarly the compensator is designed as follows

\[
F_a(z) = a_1z^{-n-1} + a_2z^{-n-2} + \cdots + a_mz^0 + \cdots + a_{n-1}z^{-(n-m)} + a_nz^{-(n-m)}
\]

\[
= (a_1z^{-n-1} + a_2z^{-n-2} + \cdots + a_mz^{-m} + \cdots + a_{n-1}z^{0} + a_nz^{0}) z^{-(n-m)} \quad (3-25)
\]

\[
J_F = \sum_{k=1}^{n} |1 - F_a(e^{i\omega kT})F^{-1}(e^{i\omega kT})[1 - F_a(e^{i\omega kT})F^{-1}(e^{i\omega kT})]^*|\]
Then the repetitive control law is given as

$$U(z) = H_\alpha(z)[z^{-p}U(z) + F_u(z)z^{-p}E(z)]$$  \(3-26\)

The variables from the previous run have been replaced by the variables in the previous period.

### 3.4.3 The Stability Condition and the Sensitivity Transfer Function

In order to understand when this design process will produce a convergent stable process, use $$E(z) = -G_w(z)U(z) + W(z)$$ which represents the real world dynamics, and substitute it into (26). The result is

$$z^pU(z) = H_\alpha(z)[1 - F_u(z)GW(z)]U(z) + H_\alpha(z)F_u(z)W(z)$$  \(3-27\)

Note that the product $$F(z)GW(z)$$ contains $$GM(z^{-1})GW(z)$$ which is real and positive for $$z$$ on the unit circle when there is no model error. One can determine stability of the repetitive control system by examining the homogeneous equation for (27). Using the same heuristic argument as for Equation (3-6) gives the stability condition as

$$\left|H_\alpha(e^{i\omega T})[1 - F_u(e^{i\omega T})GW(e^{i\omega T})]\right| < 1 \quad \forall \omega$$  \(3-28\)

And as before, using the logic presented in references [6] or [7] establishes that this is a necessary and sufficient condition for asymptotic stability for all possible periods $$p$$.

In the design process one first deals with adjusting the weights to robustify as needed. For this purpose we write the stability condition in terms of the original frequency components. After making the design and creating the FIR filters, one again checks for satisfaction of (28). The condition for picking the weights in the cost function is then

$$\left|1 - \frac{G_u(e^{-i\omega T})G_w(e^{i\omega T})}{GM(e^{i\omega T}) + r_u(k)}\right| < \frac{1}{H(e^{i\omega T})} = 1 + \frac{r_u(k)}{GM(e^{i\omega T}) + r_u(k)} \quad \forall k$$  \(3-29\)
At high frequencies one may not have a good system model, but the amplitude of the output is also very small. This means that it will often be true that the left hand side goes above unity by only a small amount. In this case, and small $r_u(k)$ can be enough to stabilize the system.

It is then important to determine how much error is produced by allowing a nonzero value of $r_u(k)$. To evaluate this, compute the sensitivity transfer function. Solve (28) for $U(z)$ and substitute into $E(z) = -G_w(z)U(z) + W(z)$ to obtain

$$E(z) = \left[ \frac{z^p - H(z)}{z^p - H(z)[1 - F(z)G_w(z)]} \right] W(z)$$

$$H(e^{i\omega_T}) = \frac{\left| G_M(e^{i\omega_T}) \right|^2 + r_u(k)}{\left| G_M(e^{i\omega_T}) \right|^2 + r_u(k) + r_d(k)} ; \quad F(e^{i\omega_T}) = \frac{G_M(e^{-i\omega_T})}{\left| G_M(e^{i\omega_T}) \right|^2 + r_d(k)}$$

(3-30)

### 3.4.4 Multiple Model Version of RC Law

References [10~13] averaged some cost function over a set of models representative of the distributions of the uncertainties in model parameters. Note that the extra adjustment of the speed of learning that is used for further robustification in [14] is accomplished here by adjusting the $r_u(k)$. One can do the averaging here by averaging the original cost function over a set of $N$ models $P_{M\ell}$ or $G_{M\ell}(e^{-i\omega_T})$, $\ell = 1, 2, ..., N$, picked from the uncertainty distributions. The result is to replace $\left| G_M(e^{i\omega_T}) \right|^2$ and $G_M(e^{-i\omega_T})$ by

$$\frac{1}{\ell} \sum_{\ell=1}^{N} \left| G_{M\ell}(e^{-i\omega_T}) \right|^2 ; \quad \frac{1}{\ell} \sum_{\ell=1}^{N} G_{M\ell}(e^{-i\omega_T})$$

(3-31)
3.5 Numerical Examples

3.5.1 Examples of Robust RC Design Using a Mathematical Model

Consider designing a repetitive control system based on a third order model, but the actual system is fifth order with an extra mode at high frequency

\[
G_M(s) = \left( \frac{a}{s + a} \right) \left( \frac{\omega_a^2}{s^2 + 2\zeta\omega_a s + \omega_a^2} \right)
\]

\[
G_W(s) = \left( \frac{a}{s + a} \right) \left( \frac{\omega_b^2}{s^2 + 2\zeta\omega_b s + \omega_b^2} \right) \left( \frac{\omega_a^2}{s^2 + 2\zeta\omega_a s + \omega_a^2} \right)
\]

where \( a = 8.8, \, \zeta = 0.5, \, \omega_a = 6(2\pi), \, \omega_b = 18(2\pi) \) and we consider that these systems are fed by a zero order hold running at 100 Hz. In practice we will not know the fifth order system, but here we use the information to examine possible behaviors when designs are applied in the world.

First we can design an \( F(z) \) as above using \( R_d \) as the identity matrix. The polar plot of \( F(z)G_w(z) \) is given in Figure 3-1. In the view on the left it is not obvious that the RC system is unstable, but the enlargement on the right shows that the plot does in fact go outside the unit circle centered at +1, and therefore the system is unstable (at least for reasonable choices of period \( p \)). Before making use of the design method here, let us consider the very simple approach of using \( H \) equal to a constant. We pick the constant as one over the maximum value of \( |1 - F(z)G_w(z)| \) over all frequencies. This value is \( H = 0.99998 \). The resulting sensitivity transfer function is given in Figure 3-2. We see that the error at the addressed frequencies is attenuated for all frequencies up to some value, after which they are amplified. Figure 3-3 shows the same plot for only the frequencies having period \( p \) time steps, where \( p = 100 \). In the detail plot on a log scale we see that we are not getting zero error anywhere and the very slight change in the value of \( H \) in the 5th decimal place, has produced errors of the size of about \( 10^{-1} \) in the mid
teen range of Hertz. This suggests that one might prefer to be more selective in the design and make use of the ability to adjust $R_u$ values as a function of frequency.

Now let us compare the performance of a near perfect cutoff and near minimal adjustment of the value of $H$ at each frequency. Figures 4 displays the value of $|1 - F(z)G_w(z)|$. The dashed curves display the values of $1/H$ as a function of frequency for the sharp cutoff case on the left and the minimal adjustment of $H$ on the right. The minimal adjustment uses a tolerance of $10^{-5}$ above the curve of $|1 - F(z)G_w(z)|$, and the perfect cutoff is made at the frequency necessary for the same tolerance. From these curves one can back calculate the needed values of the $r_u(k)$, and these are given in Figure 3-5, where the choice on the left simply picks a very large number for frequencies that are to be cutoff.

Figure 3-6 presents the corresponding sensitivity transfer function behavior. The frequencies being addresses are at the bottoms of the valleys. And many disturbances at frequencies between these valleys are amplified. The value of $r_u(k)$ can be adjusted to change the rate of convergence at each frequency, and this can adjust the maximum amplification seen at the between frequencies. The value chosen here has already decreased these peaks by about as much as is possible. Note that in the range of frequencies when $r_u(k)$ is zero, the plots look identical. Above the frequency at which a cutoff needed to start, the perfectly sharp cutoff makes the sensitivity transfer function equal unity, while the minimally reduced $H$ is seen to continue to decrease the errors at the harmonics for period $p$ time steps. However, we pay for this improved performance by amplifying still higher frequencies having this period. In many physical systems the amplitudes of the errors decrease quickly with frequency, and then the attenuation at lower frequencies will have a stronger effect on the overall error than the amplification of the smaller errors at higher frequencies. This tradeoff underlies all feedback control systems according to the
Bode integral theorem (the waterbed effect), so we should not be surprised to see it arise here. But it does require that the design analyze the situation to be sure that the design produces an overall improvement in tracking error.

The example presented here uses knowledge of the 5\textsuperscript{th} order system, and uses fine tolerance that would not be possible in applications. In previous work, $H$ was simply a cutoff based on when the model became too poor to have convergence. The cutoff value could be adjusted in hardware, since one does not know what is wrong with one’s model. As stated previously, the instability is normally slow at high frequencies, and one can apply the RC law and either start with a low cutoff and raise it until some instability is observed, or one can start with a high cutoff, observe the error and compute the frequencies of the components that are growing, and then make the cutoff accordingly. To apply such methods to the current design process, one should not only observe which frequencies are growing, but observe how fast they are growing as a function of time measured in periods. As discussed in [7], the value of $|H(z)[1 - F(z)G_w(z)]|$ at any frequency is a good estimate of the change in amplitude of the error from period to period. Thus this extra experimental information can be used to pick the needed magnitude of $H$ as a function of frequency.
Figure 3-1. Polar plot of the compensator times the true world transfer function. Detail shown on the right. Dotted curve shows tolerance used.

Figure 3-2. Magnitude sensitivity transfer function plot using constant $H = 0.99998$. 
Figure 3-3. Sensitivity transfer function magnitude at addressed frequencies only, for constant $H$ and using the $H$ with $10^{-5}$ tolerance.

Figure 3-4. Values of $|1 - F(z)G_n(z)|$ as a function of frequency, sharp cutoff $1/H$ left, and minimum $1/H$ on right.

Figure 3-5. Plots of $r_u(k)$ vs. $\omega_k$, sharp cutoff left, minimal $1/H$ with tolerance right.

Figure 3-6. Sensitivity transfer function plots for sharp cutoff (left).
and minimal $1/H$ with tolerance (right).

3.5.2 Examples of Robust RC Design Based on Multiple Frequency Response Tests

The design process used here only needs to know the frequency response of the system, and one need not develop a model. One can apply a rich input signal obtained by generating pseudo random numbers, and apply to the hardware to obtain the system frequency transfer function, i.e. the magnitude and phase of the system transfer function. Here we consider taking 10 sets of 5000 data points this way. Each set defines one frequency response model which we obtain using the TFE algorithm in Matlab. It is typical for people to average such multiple data sets in order to obtain a more accurate model. Reference [13] shows that it is best to use all 10 models and average the cost function instead, resulting in a more robust repetitive control law.

Figure 3-7 gives the magnitude and phase plots for all 10 data sets. Clearly the phase at high frequency is not well known. For this investigation we consider that the world is the 5th order system, the damping ratios changed to 0.01 for each mode, and all we know about the system is the results of the 10 frequency response tests. We want to consider how one might apply the methods presented here to this situation. Figure 3-8 presents the design for the compensator $F(z)$ (we do not take the next step to design the associated FIR filter $F_a(z)$) that again uses $R_d$ equal the identity matrix. To clean up this signal, a moving average of the complex frequencies was made. At any time step, the value used was the average of 20 forward points, 20 backward points, and the point itself. The resulting plots of magnitude and phase are given in Figures 9.

Figure 3-10 present choices for the zero-phase filter $H$. The plot on the left presents the values of $|1 - F(z)G_{ml}(z)|$ for all 10 frequency response models using the $F$ designed from the average cost. Based on this, the cutoff frequency is shown as indicated by the dotted line. The
dashed plot on the right does the same kind of windowing, but this time it is averaging the peak valued of all 10 models, and this results in the desired $1/H$ values. Then Figure 3-11 presents the results of using the sharp cutoff and the gentle cutoff on the right and left of Figure 3-10. We see that with the $1/H$ curve on the right of Figure 3-10, we are able to decrease the error at all addressed frequencies above the cutoff at 30 Hz on the right in Figure 3-11. In this case, the averaging of the cost has resulted in much improved performance that attenuates errors at all addressed frequencies up to Nyquist. This demonstrates the ability of the approach to improve performance when the uncertainty is characterized by the frequency response tests, as it might often be in applications. Note however, that we could perform the same operations directly without making use of the quadratic cost problem, by designing a compensator, and then looking at the data to design the $H$. The bottom line is therefore, that it can be advantageous to not use a sharp cutoff to robustify the repetitive control law to model error, and instead adjust $H$ minimally downward based on one’s understanding of the uncertainties in the model.

![Figure 3-7. Magnitude and frequency plots of the 10 models $G_M(e^{j\omega})$.](image)

Figure 3-7. Magnitude and frequency plots of the 10 models $G_M(e^{j\omega})$. 
Figure 3-8. The compensator design $F(e^{i\omega T})$ using the cost summed over 10 models.

Figure 3-9. Magnitude and phase plots of $F(e^{i\omega T})$ after using a 51 point average windowing.

Figure 3-10. Plots of $|1 - F(z)G_{Mi}(z)|$ left with dashed sharp cutoff, and of windowed values and windowed maximum values (dashed) on right.
3.6 Conclusions

Some conclusions are as follows:

- A method is presented to design repetitive controllers and also learning controllers that compromises on the zero error objective in order to obtain improved robustness to model error. Previous work [10-13] has used cost averaging to improve robustness, but the emphasis of these works is in adjusting the phase of the compensator to make as much of the anticipated model uncertainty fall within the stability boundary as possible. Relatively little penalty is paid for this robustification. More recent work [14] added to this, adjustment of the learning rate to improved robustness. Here one obtains the robustification in exchange for slower learning in some frequency ranges. The approach presented here can be used in addition to both previous approaches. The additional robustification is accomplished in exchange for no longer converging to zero error in some frequency ranges.

- The repetitive control design approach of [16] and [8,9] first designs a compensator, and then based on the performance of the compensator designs a cutoff filter. In order to learn up to as
high a frequency as possible, the filter is designed with as sharp a cutoff as possible. The design approach here initially appears to be designing the compensator and the cutoff simultaneously. However, the final result obtained here has the property that the compensator is determined completely by the choice of the $R_d$ weights in the cost function, and is independent of the compromise made with zero error which is determined with knowledge of $R_d$ by the choice of $R_u$. Therefore, the approach presented here can also be thought of as a two stage process. What is new is the perspective that one does not need to make a sharp frequency cutoff filter, but rather a very small decrease in the effort to get to zero error is likely enough to stabilize the repetitive control system. One can take this approach using the method presented here, or simply do it directly in a two stage design. But the method here does make it clear how to adjust gains for the amount of tolerance needed at each frequency.

- What can potentially be accomplished by using a minimal amount of attenuation instead of a sharp cutoff, is given by the sensitivity transfer function from output disturbance or command to error. Using a sharp cutoff, the sensitivity transfer function transitions to unity above the cutoff, without attenuation of error but also without amplification. Using a minimal enlargement of the unit circle stability boundary for the frequencies that need it, appears to allow a region where the error at addressed frequencies are attenuated, and this appears to be a possibly significant benefit. This benefit is paid for by the waterbed effect, that makes the design amplify errors at higher addressed frequencies. Whether one is winning in this tradeoff is determined by the frequency distribution of error amplitudes. One needs that the errors at addressed frequencies that are amplified are smaller than those that are attenuated, and this could often be true because errors tend to get smaller as the frequency goes up.
- We note that the lower frequencies when no penalty is applied in $R_u$ exhibit the same waterbed effect with or without the $R_u$ being applied. One might have hoped that the changes introduced by having a nonzero $R_u$ at higher frequencies might have allowed the amplification of signals between addressed frequencies to be shifted to the higher frequency range where they are less important, but this does not happen.

We conclude that the compromise in not asking for zero error for certain frequencies can be an effective tool in robustification. It can be combined with cost averaging and adjustment of the learning rate. By comparison with the use of a sharp cutoff for robustification, this approach can result in better performance in a frequency range just above what would have been the cutoff.

3.7 References


Chapter 4
Converting Repetitive Control Robustification Methods to Apply To Iterative Learning Control

4.1 Introduction

The previous two chapters have developed two different robustification methods for Repetitive Control (RC). While each chapter has explored extensively the possible limits for robustification, both of them are based on the steady state frequency response models, which could be rigorously applied to RC designs as they aim for zero error as time goes to infinity. ILC, on the other hand, asks for zero error in finite time problems that are repeated, each time starting from the same initial condition, and hence they technically never reach steady state. References [22,25] develop the relationship between frequency response and the singular value decomposition of the Toeplitz matrix of system unit pulse response. This allows one to extend the usefulness of frequency response thinking into the finite time ILC problems. It is the purpose of this paper to make use of this extension to finite time problems, in order to develop the ILC versions of the three kinds of robustification detailed above. It can be important to make use of the results in References [26,27] that delete one or more time steps from consideration for zero tracking error, in order to eliminate badly behaved singular values and singular vectors of the Toeplitz matrix that do not relate to frequency response, but rather to instability of the inverse system.
4.2 Mathematical Formulation of the ILC Problem

Consider a discrete time state variable model

\[
x(k+1) = Ax(k) + Bu(k) \\
y(k) = Cx(k) + v(k)
\]  

(4-1)

of a feedback control system. The desired output is \( y^*(k), \ k = 1, 2, 3, \ldots, p \). The \( v(k) \) represents any disturbance that occurs every run (also called repetition or iteration). Wherever the disturbance occurs, there is an equivalent output disturbance which is used here. A subscript \( j \) is applied to variables to indicate the run, repetition, or iteration number. The error is defined as \( e(k) = y^*(k) - y(k) \). Underbars are used on variables to indicate a column vector of the associated variable for each time step of a run, i.e. for the input variable \( u(k) \) the time step argument runs \( k = 0, 1, 2, \ldots, p-1 \), and for output variables such as \( y, \ v, \ y \), \( k \) has the range \( k = 1, 2, 3, \ldots, p \). Then the error for run \( j \) can be written in the form

\[
e_j = -Pu_j + f
\]  

(4-2)

\[
f = y^*_j - y - \bar{A}x(0)
\]

\[
P = \begin{bmatrix} CB & 0 & \cdots & 0 \\ C^2B & CB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{p-3}B & CA^{p-2}B & \cdots & CB \end{bmatrix} \quad \bar{A} = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^p \end{bmatrix}
\]

The singular value decomposition of \( P \) is given by

\[
P = USV^T \quad ; \quad S = diag(\sigma_1, \sigma_2, \ldots, \sigma_p)
\]  

(4-3)

The form of the learning law considered here is given as

\[
L = rVTU^T \quad ; \quad \Gamma = diag(\gamma_1, \gamma_2, \ldots, \gamma_p)
\]  

(4-4)

\[
u_{j+1} = H(u_j + rVTU^T \varepsilon_j)
\]  

(4-5)
The $r$ and $\Gamma$ entries are design parameters, and $H$ is introduced as a finite time version of a zero phase low pass filter used to robustify to unmodeled high frequency dynamics. By making different choices for the values of the gammas, one can get the partial isometry ILC law, or the Euclidean norm ILC law, or an inverse ILC law that is too aggressive to be useful (see References [2], [9], [22]). It is convenient to change the basis used for the various vectors and matrices involved, according to

$$\hat{e}_j = U^T e_j \quad \hat{u}_j = V^T u_j \quad \hat{f} = U^T f \quad \hat{H} = V^T H \quad \hat{H} = \text{diag}(h_1, h_2, ..., h_p)$$

$(4-6)$

$$\hat{e}_j = -S \hat{u}_j + \hat{f}$$

$$\hat{u}_{j+1} = \hat{H}(\hat{u}_j + r\Gamma \hat{e}_j)$$

$$\hat{u}_{j+1} = \hat{H}(I - r\Gamma S)\hat{u}_j + \hat{H}\hat{f}$$

The learning law above will converge if the homogeneous part of this equation has coefficient matrix with all eigenvalue magnitudes less than unity (and it will converge monotonically because the singular values are the absolute values of the eigenvalues since the matrix is diagonal).

For this situation we can also easily get the error propagation equation

$$\hat{e}_{j+1} = \hat{H}(I - r\Gamma S)\hat{e}_j + (I - \hat{H})\hat{f}$$

$(4-7)$

If $I - \hat{H} = 0$, the error converges to zero when stable. For this diagonal situation, it is the same matrix determining stability for both error and control difference equations.

### 4.3 Four Kinds of Robustification

From the above, the condition for convergence is

$$|h_i(1 - r\gamma_i\sigma_i)| < 1 \quad \forall \quad i$$

$(4-8)$
and if one does not use the cutoff filter $h_i$, it is the condition for convergence to zero error for all possible initial error histories. References [18~21] consider designing repetitive control and iterative learning control based on averaging a cost function over a distribution of models. One specifies the uncertainty in parameters of the model in terms of a probability distribution, and then picks a finite set of representative models from the distribution. The control design is based on minimizing the associated design cost function summed over this set of models.

Previous publications have considered four kinds of robustification:

(1) ILC laws and RC laws have used zero-phase low-pass filters to cut off the learning at high frequencies, producing stability robustness at the expense of not attempting to eliminate error components above the cutoff frequency (see References [11~17]). This can be considered robustification to singular perturbations of the model.

(2) Reference [23] establishes that averaging of the phase over models is the main underlying robustification mechanism of the averaging cost functions approach in References [18~21].

(3) Reference [23] then introduced adjustment of the learning rate as a function of frequency to try to get all models in the distribution to be within the unit circle.

(4) When (3) is not good enough, Reference [24] introduced expansion of the unit circle to include more of the distribution, at the expense of not converging to zero error.

The stability for RC is studied by examining a polar frequency response plot of the product of the feedback control system transfer function times the RC compensator design. Examining the RC approach together with the ILC formulation created here, we make the following observations:

• Equation (4-8) is the ILC analog of the unit circle stability condition used in References [23], [24], and [9] for RC.
• The $\sigma_i$ is the discrete version of the magnitude frequency response of $PL$ for “frequency $i$”.

• The $r$ is an overall learning gain.

• The $\gamma_i$ is a separate gain for each “frequency” to keep within the unit circle.

• The $h_i$ should be unity in order to have convergence to zero error.

• The choice of $h_i$ can be used to cut out high frequencies, setting values above some “frequency” to zero corresponds to the zero-phase low-pass filter for discrete time.

• But $h_i$ can also be used as a way to expand the unit circle to avoid divergence at some frequency, analogous to Reference [24] for RC.

Relating these observations to the four robustification methods, we conclude that for ILC:
- The $h_i$ form the discrete frequency version of a zero-phase filter cutoff, associated with the finite number of frequencies one can observe in a finite time signal, and their choice can addresses Robustification Item (1) above.
- The $\gamma_i$ adjust the learning rate for each discrete frequency for Robustification Item (3) above.
- The $h_i$ are also used for Robustification Item (4) above, done frequency by frequency.
- It remains to develop the averaging of phase needed for Robustification Item (2).

According to References [25] and [22], the phase information is contained in the $U$ and $V$ matrices. Specifically, as the number of time steps in the trajectory, $p$, becomes long, the columns of $V$ and the columns of $U$ approach sinusoids at each of the discrete frequencies observable in that number of time steps, roughly $p/2$ frequencies. The phase change going through the system appears in this finite time formulation as a phase difference between the input phase of the $i^{th}$ column of $V$, and the output phase of the $i^{th}$ column of $U$, for what we might call the $i^{th}$ frequency. In the limit there are two entries for each frequency which together span the
needed space, analogous to a sine and a cosine at each frequency in a Fourier series. For Robustification Item (2) we want to create an average relating to this without involving the magnitudes, which would distort the result away from our purpose, we aim to adjust magnitudes separately as needed.

We consider two candidates for obtaining the phase average equivalent for finite time problems. Find the singular value decomposition of each model $P_m = U_m S_m V_m^T$, $m=1,2,3,\ldots,M$. The subscript represents the model number from the distribution of models.

(A1) Find the average over $m$ of $U_m V_m^T$. Note that the product for each $m$ has no magnitude information, it only produces the finite time version of phase change from input to output. Then one may want to take the SVD of the resulting average, and set all of its singular values to unity (they should stay close to unity, but will certainly deviate). We don’t want the matrix to be changing the magnitudes of vectors. From this SVD we obtain two separate matrices $U_{A1}$ and $V_{A1}$ associated with the average of the product. And these are used in the ILC law, Equation (4-5).

(A2) Another option is to compute the average of $U_m$ over all models, and compute the average of $V_m$ over all models. One might want to at least renormalize to unit vector columns. Then we have $U_{A2}$ and $V_{A2}$.

Now that we have a way to approach Robustification Item (2), we want to create an algorithm to design the ILC law, trying to imitate the frequency-by-frequency design of Reference [23]. The problem for finite time is not as clean as that for steady state frequency response in RC because the frequencies from input to output are no longer fully decoupled, i.e. a single “frequency” input can produce outputs at neighboring frequencies as well.
4.4 Ordering the Singular Vectors and Singular Values

References [25] and [22] show how the singular values and singular vectors of matrix $P$ are related to frequency response, with each being associated with a specific frequency or with two neighboring frequencies. Of course, when we do a frequency response analysis we know the result for each frequency. But singular value decomposition algorithms deliver the singular values in the order of decreasing value. If all models happen to have monotonically decreasing magnitude frequency response, then we expect that the singular value order matches the discrete frequencies observable in the number of time steps $p$. When the magnitude frequency response is not monotonically decaying for each model, we need to determine what frequency each singular value and each right and left singular vector applies to. And then we need to reorder the entries in the $U$, $V$, and $S$ matrices to correspond to the frequencies in ascending order. To determine the associated frequency for each singular value, right singular vector, left singular vector triplet, perform a discrete Fourier transform on the singular vectors and examine the frequency of the peak. References [25] and [22] observe that every other singular value is associated with a clean sharp peak, and the singular value between these has a peak that has two frequencies in it, i.e. it bridges to the next frequency. In the limit there must be two singular values and two singular vectors for each frequency. This reordering of the columns of $U$ and $V$ is necessary before one does the averaging discussed in (A1) and (A2), so that the averaging has the intended meaning.

Reference [25] develops the following formula for the phase change through the system $P$ obtained for the finite time $p$ step ILC problem. The magnitude information is obtained as described above by reordering the singular values as needed. The phase change $\theta_i$ at the $i^{th}$ frequency is given by Equation (4-24) of that reference, where the subscript $i$ counts the discrete frequencies starting at dc going completely around the unit circle. The second entry corresponds
to the same frequency as the last entry, and similarly for other entries. Using the $U$ and $V$
matrices of a singular value decomposition of matrix $P$, these phase changes are given by

$$\text{diag}(e^{i\theta_0}, e^{i\theta_1}, \ldots, e^{i\theta_{p-1}}) = \hat{H}UV^T (\hat{H}^*)^T$$

(4-9)

$$\hat{H} = (1 / \sqrt{p})H \quad ; \quad \hat{H}^{-1} = (\hat{H}^*)^T \quad ; \quad z_o = e^{i\theta_0} \quad ; \quad \omega_o = (2\pi / p)$$

$$H = \begin{bmatrix}
(z_o^0)^0 & (z_o^0)^{-1} & \cdots & (z_o^0)^{-(p-1)} \\
(z_o^1)^0 & (z_o^1)^{-1} & \cdots & (z_o^1)^{-(p-1)} \\
\vdots & \vdots & \ddots & \vdots \\
(z_o^{p-1})^0 & (z_o^{p-1})^{-1} & \cdots & (z_o^{p-1})^{-(p-1)}
\end{bmatrix}$$

4.5 Equations for Updates Applied to Any Model

Consider using $U_A$ and $V_A$ produced from the averaging as in (A1), and consider the
resulting ILC updates each iteration for any model $P_m$. The math below needs to have the inverse
equal the transpose for these two matrices, as discussed in (A1). For each model, define

$$\hat{e}_j = U_A^T e_j \quad \hat{u}_j = V_A^T u_j \quad \hat{f} = U_A^T f \quad \hat{H} = V_A^T HV_A \quad \hat{H} = \text{diag}(h_1, h_2, \ldots, h_p)$$

(4-10)

$$e_j = -P_m u_j + \hat{f}$$

$$\hat{e}_j = -U_A^T P_m V_A \hat{u}_j + \hat{f}$$

$$\hat{u}_{j+1} = \hat{H}(\hat{u}_j + r\Gamma \hat{e}_j)$$

Note that this ILC law is defined in “hat” space. The control updates satisfy

$$\hat{u}_{j+1} = \hat{H}(I - r\Gamma U_A^T P_m V_A)\hat{u}_j + \hat{f}$$

(4-11)

Convergence is obtained for this model if the spectral radius of the matrix

$$E_m = \hat{H}(I - r\Gamma U_A^T P_m V_A)$$

(4-12)

is less than unity, and convergence is monotonic in the Euclidean norm if the maximum singular
value is less than unity. Because the $\hat{H}$ does not commute with the non-diagonal matrix it
multiplies, it is a problem to find a corresponding matrix for the error update from iteration to iteration, as was done in Equation (4-7). But for convergence we do not need to find this matrix.

4.6 Making the ILC Problem Well Posed for Pole Excess Greater Than Two

When the discrete time model $P$ involves a continuous time system fed by a zero order hold whose output is sampled, the matrix $P$ will normally be badly ill conditioned if the continuous time system has a pole excess of 3 or more, i.e. at least three more poles than zeros. The mathematical proofs of convergence for ILC establish convergence to zero tracking error, but this can only be accomplished after a very large number of iterations resulting in prohibitively large control actions. What usually happens in practice is that the error seems to stop improving at some level that is far from zero. But this can still be a very useful result, with substantial improvement in tracking accuracy accomplished with relatively little effort.

Therefore, it can be reasonable to apply the results presented above to get improved performance of feedback control systems doing repetitive tasks. But the mathematical problem involved is actually ill posed for pole excesses of 3 or more, and one does not get the accuracy improvement expected. An understanding of this phenomenon is given in References [26] and [27]. Also presented is a method of eliminating the ill conditioning, with the result that the final error levels actually can approach zero.

To address this, we delete the first few rows of the output matrix from consideration, and no longer ask for zero error at these initial time steps. The number of rows to be deleted is equal to or greater than the number of zeros outside the unit circle in the $z$-transfer function corresponding to the differential equation that is fed by a zero order hold. For a continuous time third order system with no zeros, this number is one. In this section we suppose that our model is
perfect and rewrite the equations for the ILC updates for the case when there are initial rows removed from the matrix $P$, producing matrix $P_D$. This is the generalization of Equations (4-1) through (4-7). In the next section we consider what happens when the world is different from our model as in the previous section above. The original dimension of $P$ is $p$, and the row dimension after removing rows is $q$, so $p - q$ rows are removed. We partition matrices into those parts related to the removed rows with subscript $R$, and the remaining parts with subscript $D$, representing the original matrices with parts deleted:

\[
\begin{bmatrix}
  \hat{e}_{k_0} \\
  \hat{e}_{l_0}
\end{bmatrix} = - \begin{bmatrix}
  P_R \\
  P_D
\end{bmatrix} u_j + \begin{bmatrix}
  f_R \\
  f_D
\end{bmatrix}
\]

\[
P_D = U_D \begin{bmatrix}
  S_D & 0
\end{bmatrix} V_D^T
\]

\[
S_D = \text{diag}(\sigma_1, \sigma_2, ..., \sigma_q)
\]

(4-13)

\[
\hat{e}_{0j} = - P_D u_j + f_D = - U_D \begin{bmatrix}
  S_D & 0
\end{bmatrix} V_D^T u_j + f_D
\]

Note that $S_D$ is not $S$ with one or more singular value deleted, the dimension of $U_D$ is $q$ by $q$, and $V_D^T$ is $p$ by $p$, and is not related to $V^T$ by deleting anything, and similarly for $\hat{H}_D$. Define

\[
\hat{e}_{0j} = U_D^T e_{0j} \quad \hat{u}_j = V_D^T u_j \quad \hat{f}_D = U_D^T f_D \quad \hat{H}_D = V_D^T H V_D
\]

(4-14)

\[
\hat{H}_D = \text{diag}(h_1, h_2, ..., h_p)
\]

The ILC law becomes

\[
L_D = rV_D \begin{bmatrix}
  \Gamma_D \\
  0
\end{bmatrix} U_D^T \quad \Gamma_D = \text{diag}(\gamma_1, \gamma_2, ..., \gamma_q)
\]

(4-15)

\[
\hat{u}_{j+1} = H(u_j + L_D \hat{e}_{0j})
\]

resulting in

\[
V_D^T \hat{u}_{j+1} = V_D^T H (V_D V_D^T u_j + rV_D \begin{bmatrix}
  \Gamma \\
  0
\end{bmatrix} U_D^T e_{0j})
\]

(4-16)
\[
\hat{u}_{j+1} = \hat{H}_D (\hat{u}_j + r \begin{bmatrix} \Gamma_D & \\
0 & 0 \end{bmatrix} \hat{e}_{Dj})
\]

\[
\hat{e}_{Dj} = - [S_D \ 0] \hat{u}_j + \hat{f}_D
\]

Control input convergence is governed by the equation

\[
\hat{u}_{j+1} = \hat{H}_D \left( I - r \begin{bmatrix} \Gamma_D S_D & 0 \\
0 & 0 \end{bmatrix} \right) \hat{u}_j + r \hat{H}_D \begin{bmatrix} \Gamma_D & \\
0 & 0 \end{bmatrix} \hat{f}_D
\]  

(4-17)

Written in two parts, the \( R \) part is not updated, and the \( D \) part is. Pick

\[
\hat{H}_{DD} = \text{diag}(h_1, h_2, \ldots, h_q)
\]  

(4-18)

and let the remaining entries of \( \hat{H}_D \) be unity.

\[
\begin{align*}
\hat{u}_{p,j+1} &= \hat{H}_{DD} (I - r \Gamma_D S_D) \hat{u}_{Dj} + \hat{H}_{DD} \Gamma_D \hat{f}_D \\
\hat{u}_{k,j+1} &= \hat{u}_{kj}
\end{align*}
\]  

(4-19)

Convergence is obtained provided all eigenvalues of the diagonal matrix \( \hat{H}_{DD} (I - r \Gamma_D S_D) \) are less than unity in magnitude

\[
|h_i (1 - r \Gamma_i \sigma_i)| < 1 \quad \forall i = 1, 2, \ldots, q
\]  

(4-20)

### 4.7 Equations For Updates For Any Model With Rows Deleted

Now consider the generalization of Equations (4-10) through (4-12) to the case when initial rows are deleted, applying a learning law to any chosen model in the distribution of models considered. Paralleling the previous development, we produce the following sequence of equations

\[
\begin{align*}
\hat{e}_{Dj} &= U_D^T e_{Dj} \quad \hat{u}_j = V_D^T u_j \quad \hat{f}_D = U_D^T f_D \\
\hat{H}_D &= V_D^T H V_D
\end{align*}
\]  

(4-21)

\[
\begin{align*}
\varepsilon_j &= - P_m u_j + \bar{f} \\
\varepsilon_{Dj} &= - P_{Dm} u_j + \bar{f}_D
\end{align*}
\]
\[
U_D^T \tilde{e}_{Dj} = -U_D^T P_{Dm} V_D \hat{u}_j + U_D^T \tilde{f}_D
\]

\[
\hat{e}_{Dj} = -U_D^T P_{Dm} V_D \hat{u}_j + \hat{f}_D
\]

\[
\hat{u}_{j+1} = \hat{H}_D(\hat{u}_j + r \begin{bmatrix} \Gamma_D \\ 0 \end{bmatrix} \hat{e}_{Dj}) = \hat{H}_D(\hat{u}_j - r \begin{bmatrix} \Gamma_D \\ 0 \end{bmatrix} U_D^T P_{Dm} V_D \hat{u}_j + r \begin{bmatrix} \Gamma_D \\ 0 \end{bmatrix} \hat{f}_D)
\]

\[
\hat{H}_D = \text{diag}(\hat{H}_{DD}, I)
\]

\[
\hat{u}_j = \begin{bmatrix} \hat{u}_{Dj} \\ \hat{u}_{Rj} \end{bmatrix}
\]

\[
V_D = \begin{bmatrix} V_{DD} & V_{DR} \end{bmatrix}
\]

Then the resulting control update law is

\[
\hat{u}_{D,j+1} = \hat{H}_{DD}(I - r \Gamma_D U_D^T P_{Dm} V_{DD}) \hat{u}_{Dj} - \hat{H}_{DD}(r \Gamma_D U_D^T P_{Dm} V_{DR}) \hat{u}_{Rj} + r \hat{H}_{DD} \Gamma_D \hat{f}_D
\]

\[
\hat{u}_{R,j+1} = \hat{u}_{Rj}
\]

Convergence is determined by the eigenvalues, and monotonic convergence in the sense of the Euclidian norm is guaranteed by the singular values of either of the following

\[
E_{Dm} = \hat{H}_{DD}(I - r \Gamma_D U_D^T P_{Dm} V_{DD})
\]

\[
E_{Dm}^T = (I - r V_{DD}^T P_{Dm} U_D^T \Gamma_D) \hat{H}_{DD}
\]

If we compare the update equation for error above to the perfect model case

\[
\hat{e}_{Dj} = -[S_D, 0] \hat{u}_j + \hat{f}_D,
\]

we see that the imperfect model can disturb the final error level. Therefore, in the presence of model error, convergence no longer guarantees zero error at addressed frequencies. ILC and RC are different in this respect.
4.8 A Process to Adjust ILC Gains Based On a Necessary Condition for Monotonic Decay

The ILC law updates the command according to \( \hat{u}_{j+1} = \hat{H}(\hat{u}_j + r\hat{e}_j) \) for the case when no rows are deleted from consideration, and according to the corresponding equation in Equation (4-16) when rows are deleted. The design process requires us to tune \( \Gamma, \hat{H} \), and \( r \), or \( \Gamma_D, \hat{H}_D \), and \( r \). This corresponds to tuning the scalars \( \gamma_i, h_i \) for \( i = 1, 2, \ldots, p \) or to \( p - q \), and \( r \). The initial objective is to tune these scalars to produce convergence to zero error for all models. We can further aim for monotonic decay of the Euclidean norm of the error from run to run, in which case we would like the maximum singular value of \( E_m \) or \( E_{Dm} \) to be less than one for all possible models \( m \). We could compute the singular value for each model and try to generate some overall optimization algorithm that tries to get the largest singular value in each case to be less than unity by choice of these variables. But it is not obvious how to create an algorithm to do that.

In this section we consider a method of adjusting the gains based on necessary conditions for monotonic decay. The approach seeks to preserve as much as possible of the frequency-by-frequency design method of References [23] and [24]. After satisfying the necessary conditions, one still needs to test the maximum singular value for each model to determine whether stability has been achieved. A concept for further tuning is also presented if it is not achieved.

4.8.1 The Necessary Conditions

We want \( E_m \) to be a contraction mapping, and if it is a contraction mapping then so is its transpose \( E_{Dm}^T = (I - rV_U^TP_m^U\Gamma_D)\hat{H}_D \) which has the same singular values. In a linear equation \( Ax = b \), matrix \( A \) is a contraction mapping if for all \( x \), the associated \( b \) has the property that \( \|b\| < \|x\| \). The initial set of necessary conditions considers a set \( x_1, x_2, \ldots, x_p \) of orthonormal vectors that span the input space, producing outputs \( b_1, b_2, \ldots, b_p \). A necessary condition for the
maximum singular value of $A$ to be less than unity is that $b_i^T b_i = x_i^T A^T Ax_i < x_i^T x_i \forall \ i$. Note that this is not a sufficient condition to establish a contraction mapping. As a counterexample, let matrix $A$ have a singular valued decomposition

$$A = \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

(4-24)

where the usual $U$ is the identity matrix. The magnitudes of $Ax$ when $x$ is the first column and the second column of the identity matrix are the same, given by $(1/2)(s_1^2 + s_2^2)$. If $s_1 = \sqrt{0.8}$ and $s_2 = \sqrt{1.1}$ matrix $A$ produces contraction for these two $x$ vectors, but one singular value is larger than unity.

### 4.8.2 Adjusting the $\gamma_i$

If $A$ is $E_{thm}^T$ and the inputs vector is the $i^{th}$ column of the identity matrix, which we denote by $I_i$, then the resulting $b_i$ output is the $i^{th}$ column of $E_{thm}^T$. We want its norm to be less than unity. Note that if the model $m$ happens to be the same as that used to design the ILC, then there is only one entry in this output column vector. But a difference between these two models will spread the non zero outputs to neighboring entries in the output vector. Observe that by picking columns of the identity matrix, we are examining each discrete time “frequency” individually, as is done in the repetitive control problem in Reference [23].

The $\gamma_i$ influences the $i^{th}$ column of $rV_{DD}^T P_m^T U_D$, and no other column. The same is true of $h_i$. Our first objective is to adjust $\gamma_i$ to make the $i^{th}$ column of $(I - rV_{DD}^T P_m^T U_D \Gamma_D)$ have magnitude less than one. The magnitude squared of the output vector is

$$|I_i - rF_m \gamma_i|^2 = [I_i - rF_m \gamma_i]^T [I_i - rF_m \gamma_i] = 1 - 2rF_m \gamma_i + r^2 F_m^T F_m \gamma_i^2$$

$$F_m = V_{DD}^T P_m^T U_D$$

(4-25)
where subscript \( i \) indicates the \( i^{th} \) column. To have this less than unity, we want
\[
(F_{mi}^TF_{mi})r^2\gamma_i^2 < 2rF_{mi}\gamma_i.
\]
Since \( r \) and \( \gamma_i \) must be greater than zero, we need to satisfy
\[
(F_{mi}^TF_{mi})r\gamma_i < 2F_{mi}.
\]
Therefore, we ask that \( r\gamma_i \) satisfy
\[
r\gamma_i < \min_m[2F_{mi}/(F_{mi}^TF_{mi})]
\]
By making \( r\gamma_i \) sufficiently small we can always satisfy this condition, provided the \( i^{th} \) entry in \( F_{mi} \), denoted by \( F_{mii} \), is positive.

One may consider several extra issues when picking the \( \gamma_i \). One may not want to make the \( \gamma_i \) too small since this results in a very slow learning/convergence rate. Therefore, one might put a lower limit on this value. We can always satisfy the desired condition by reducing the \( h_i \) instead, but of course this sacrifices zero tracking error to produce the monotonic error decay. Also, if one no longer aims for zero error for a model in the distribution, one could consider re-evaluating the phase adjustment (A1) so that it no longer averages over this model.

4.8.3 Adjusting the \( h_i \)

After picking the values of \( \gamma_i \) going down to the smallest value we wish to consider based on learning rate, one considers each model in the set of models from the distribution that is not stabilized. One can then make a decision whether one wants to sacrifice zero tracking error for that “frequency”, or whether one prefers to ignore the possibility that this model actually applies to the real world, in which case one would ignore this model. If one wants to address it, then pick the values of \( h_i \) to make the necessary conditions satisfied, i.e. pick it to satisfy
\[
(1-2rF_{mi}\gamma_i + r^2F_{mi}^TF_{mi}\gamma_i^2)h_i^2 < 1
\]
\[
h_i < \{[I_i - rF_{mi}\gamma_i]^{-\gamma_i}[I_i - rF_{mi}\gamma_i]^{-\gamma_i}\}^{-\gamma_i}
\]
(4-28)
After making the choices of $\gamma_i$ and $h_i$ to satisfy the $p$ or the $p-q$ necessary conditions, one next examines the maximum singular value of $E_m^T$ or $E_{Dm}^T$ for each model considered to determine whether they are all stabilized.

### 4.8.4 Summary of Procedure

First determine values of $U_D$ and $V_D^T$. Do the following column by column. For the $i^{th}$ column of $(I - r V_{DD}^T P_m U_D \Gamma_D)$, examine this column for all $M$ models, and adjust $r\gamma_i$ to make the magnitudes of each column less than one using Equation (4-26), after eliminating any models that produce negative right hand sides of this equation. In addition one can eliminate any model for which the needed $\gamma_i$ corresponds to too small a learning rate. After making the choice of models to keep, one might decide to recompute the $U_D$ and $V_D^T$ for the models kept, and re-evaluate.

This is a candidate design to try to stabilize the models being considered. Concerning the models that corresponded to a negative right hand side of Equation (4-26), make a decision whether one wants to sacrifice zero tracking error for that “frequency”, or whether to ignore the possibility that this model actually applies to the world. In the former case adjust $h_i$ to get the $i^{th}$ column of $(I - r V_{DD}^T P_m U_D \Gamma_D) \hat{H}_{DD}$ less than one in magnitude for all models.

To know if the design works, one has to find the largest singular value for each model to see if they are all less than unity. If not, for all models that are stabilizable one can decrease the value of $r$ until stability is reached, and for the models that are not stabilizable by choice of $\gamma_i$, one can always make further decreases in the values of the $h_i$ until stability is achieved.
4.8.5 Possible Additional Test Inputs

If the initial design fails to get all singular values of the models less than unity for the addressed models, it is due to the phenomenon described by the example in Equation (4-24). The choice of the input set did not capture the input that is amplified. In the limit as \( p \) tends to infinity, we have pure frequency response of a linear system, and a single frequency input produces a single frequency in the output. And the method works for this situation. As the size of the matrix \( P \) decreases, crosstalk with neighboring frequencies develops which complicates the problem. Since one expects the extra “frequencies” in the output to be associated with spreading to neighboring frequencies, one might consider extra input vectors beyond the original \( p \) or \( p-q \), having the extra inputs span from one “frequency” to the next. We generalize the choice of inputs to include combinations of columns. For simple illustration, consider \( Ax=b \) as a 2 by 2 case, and let

\[
Ax = [I - rF_m^\Gamma_B](\cos \theta I_1 + \sin \theta I_2) = c_0 - \hat{\gamma}_1 F_{m1} - \hat{\gamma}_2 F_{m2}
\]

\[
c_0 = [\cos \theta \quad \sin \theta]^T \quad \hat{\gamma}_1 = \varepsilon r \gamma_1 \cos \theta \quad \hat{\gamma}_2 = \varepsilon r \gamma_2 \sin \theta
\]

When \( \theta \) is zero we have the first column of the identity matrix as input, and when it is 90 degrees we have the second column. So we can investigate all angles between. We assume that \( b_1^T b_1 \) and \( b_2^T b_2 \) are less than unity. The \( \varepsilon \) is a scale factor introduced to reduce both \( \gamma_i \) gains if needed to ensure that this new test input also satisfies the necessary condition. This is done by ensuring that

\[
\varepsilon (\hat{\gamma}_1 F_{m1} + \hat{\gamma}_2 F_{m2})^\top (\hat{\gamma}_1 F_{m1} + \hat{\gamma}_2 F_{m2}) < 2 (\hat{\gamma}_1 F_{m1} + \hat{\gamma}_2 F_{m2})^\top c_0
\]

and scaling the chosen values of \( \gamma_i \) by this factor, which can always be done.
4.9 Picking ILC Gains Based on the Frobenius Norm

The Frobenius norm of a matrix $A$ is an upper bound on the maximum singular value of the matrix. The square of the Frobenius norm is $\|A\|_F^2 = tr(A^T A)$. References [20] and [9] consider RC design for MIMO systems using a Frobenius norm instead of the maximum singular value, and show that averaging the cost function over models in a distribution of model parameters is very effective at robustifying the RC stability. We consider the ILC version of the same approach here, as an alternative to the procedure detailed above. The cost function is

$$J = \sum_m tr(E_{Dm} E_{Dm}^T) = \sum_m tr(I - r \Gamma_D F_m^T - r F_m \Gamma_D + r^2 \Gamma_D F_m^T F_m \Gamma_D)$$

(4-31)

$$E_{Dm} = I - r \Gamma_D F_m^T$$

The cost is quadratic in the gains $r \gamma_i$ to be chosen. Computing the derivative $\partial J / \partial \gamma_i = 0$ produces

$$\left( \sum_m F_m^T F_m \right) r \gamma_i = \left( \sum_m F_m \right)$$

(4-32)

One can use this to pick the values of the $\gamma_i$. It is interesting to compare this result to that obtained above. Consider summing Equation (4-26) over all models $m$, the result is

$$\left( \sum_m F_m^T F_m \right) r \gamma_i < 2 \left( \sum_m F_m \right)$$

(4-33)

4.10 A Condition for Learnability of a Model

We can learn monotonically in a model if the maximum singular value of $E_{Dm}$ is less than one. The singular values squared are the eigenvalues of

$$E_{Dm} E_{Dm}^T = I - r \Gamma_D F_m^T - r F_m \Gamma_D + r^2 \Gamma_D F_m^T F_m \Gamma_D$$

(4-34)
Consider $\Gamma_D$, the identity matrix, and study adjusting the overall gain $r$. Note that for sufficiently small $r$, the $r^2$ term is negligible compared to the linear in $r$ terms (provided they are not zero). Then

$$E_{Dm}^T E_{Dm} = I - r(F_m^T + F_m) \quad (4-35)$$

where $(F_m^T + F_m)$ is a symmetric matrix. Hence, it can be diagonalized by a unitary matrix $M$

$$E_{Dm}^T E_{Dm} = I - rM^T \Lambda M = M^T (I - r \Lambda) M \quad (4-36)$$

The diagonal elements of $(I - r \Lambda)$ are the square of the singular values of $E_{Dm}$ for small $r$.

Therefore, any model can be included in the set of models that can be learned (without needing to employ $h_j$ to get convergence), provided

$$F_m^T + F_m = U_D^T P_{Dm} V_{Dm} + V_D^T P_{Dm} U_D \quad (4-37)$$

is positive definite, i.e. all eigenvalues are positive. If so there exists an $r > 0$ sufficiently small that one is guaranteed convergence. Note that a modified form of this condition can be employed, making use of any design for $\gamma_i$ or $\Gamma_D$ that one has made. One simply replaces Equation (4-35) by

$$E_{Dm}^T E_{Dm} = I - r \Gamma_D F_m^T - rF_m \Gamma_D \quad (4-38)$$

and again one sees that there exists an $r$ sufficiently small to stabilize the system.

As a side comment, note that Equation (4-27) exhibits the same kind of property, that for $F_{mi}$ positive, one can always satisfy the desired criterion by using a sufficiently small value for $r \gamma_i$, or for $r$. However Equation (4-27) is based on necessary conditions only, and this section creates a necessary and sufficient condition to be able to have monotonic convergence of the solution of the homogeneous equation for the control input, for all sufficiently small $r > 0$.  

4.11 Examples

Consider the following system

\[ G(s) = \left( \frac{a}{s+a} \right) \left( \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \right) \]  

(4-39)

where \( a = 8.8 \), \( \zeta = 0.5 \), and nominally \( \omega_n = 12\pi \). This is a reasonably good model of the command to response of the control systems for each axis of the 7 degree of freedom Robotics Research Corporation robot used in the ILC experiments reported in References [11], [12], and [4]. Consider that the input to this transfer function is digital and comes through a zero order hold sampling at 100Hz sample rate. The desired trajectory is taken to be one second long, or \( p = 100 \). It is the job of the iterative learning control law to use the stored 100 data points from the previous run to determine an adjusted 100 time step command that will improve the tracking error.

The parameter \( \omega_n \) is considered to be uncertain and to have a uniform distribution over the range \( \pm 2.5\pi \) about the nominal value given above. The ILC law averages over this distribution as represented by a set of models that have values of \( \omega_n \) in uniform \( 0.5\pi \) increments. This produces 11 models. The averaging is performed as in (A1). Note that the frequency response plot of the continuous time system is monotonically decaying with frequency, the damping ratio of 0.5 is sufficiently high that the plot continues the decay after the bandwidth of 1.4 Hz (from the first order term) as it goes through the resonant frequency. This means that the singular values, and the associated singular vectors given in the order of decreasing singular values maps to the discrete frequencies in the frequency response that are visible using 100 data point, without needing adjustment.
The examples are run using the full matrix $P$. As discussed, one might want to delete the first row of $P$ in order to eliminate one particularly small singular value for the pole excess of 3 in the original continuous time system. This one singular value is not related to frequency response, and is instead related to the zero outside the unit circle. Reference [27] discusses the issues involved. Figure 4-1 plots the frequency magnitude response curve for each of the 11 models, as dot and dash curves. The singular values of the initial averaged $U_m V_m^T$ is also computed and shown at the top of the plot, and we see that it is near unity. Following (A1) the singular value decomposition is made and these magnitudes set to unity to form the ILC law.

![Figure 4-1. Discrete magnitude frequency response of the 11 models, and that of the averaged $U_m V_m^T$.](image)

A key characteristic of the design approach is to make the phase response of the finite time / discrete time learning law (according to Equation (4-9)), when its sign is reversed, lie centered in the distributions of the corresponding phase responses of each model considered, frequency by frequency (the 11 dot and dash curves). This allows for a maximum possible phase error at each frequency. Figure 4-2 shows these phases for each of the 11 models. Also shown as a line with asterisks, is the negative of the phase response of the control design. We see that it does lie in the middle of the distribution as desired.

The next stage in designing the ILC law is to examine whether each model can be stabilized for each frequency. Set $r = 1$ and study the range of values of $\gamma_i$ to determine whether
all models are stabilizable, and if so what values produce stability. Figure 4-3 gives the results, representing the right hand side of Equation (4-27) that is the limit of the stabilizing value for this necessary condition. The horizontal axis is the index number for the singular values. The last value corresponds to the particularly small singular value associated with the zero outside the unit circle, and does not relate to frequency response. All of the actual frequencies are stabilizable.

Now examine stability of each model by computing the maximum singular value of

\[ E_m = \hat{H} (I - r \Gamma U_A^T P_m V_A) \quad \text{with} \quad \hat{H} = I. \]

With \( r = 1 \) the singular values versus singular value index is shown in Figure 4-4 for each of the 11 models. The values chosen for \( \gamma_i \) are those for the limiting value in Equation (4-27), i.e. using and equal sign, and hence we know that we cannot have stability. Singular values below unity guarantee monotonic decay of the Euclidean norm of the error. Singular values substantially above unity may produce bad transients or correspond to an unstable system.

![Figure 4-2. Discrete finite time phase response of the 11 models and the negative of the ILC law phase response.](image)
When the overall gain is turned down to $r = 0.1$, the singular value plot of the 11 models is shown in Figure 4-5. Provided the plot stays less than one, it corresponds to monotonic decay.

It is not clear from Figure 4-5 if the plot is actually less than one or not, so Figure 4-6 gives a detailed view of the first part of the plot. And it is clear that this plot is in fact less than unity for all index values except for the first. This first value is the result of the zero outside the unit circle in the continuous time transfer function, and it produces a singular value of $P$ that is unrelated to frequency response. Its value is estimated as smaller than $10^{-50}$ in Reference [27], and it is not possible to learn this part of the error space. Deleting the first row of matrix $P$ as discussed above eliminates this problem.
4.12 Conclusions

Previous literature has demonstrated a type of robustification to parameter uncertainty for repetitive control. One specifies the uncertainty of the parameters in terms of a distribution, and then creates a finite set of models that together represent the distribution. Then one designs the repetitive control law to minimize a cost function, but the cost function is averaged over the set of models. The design that minimizes the cost averaged over models is different than the design that minimizes the cost of the average model. Considerable experience has demonstrated that this simple process is surprisingly effective at improving robustness to model error. The same
Improvement in robustness was observed in the design of iterative learning controllers when the averaging method was applied.

Several papers by the authors investigated the method for repetitive control, making use of frequency response thinking. Based on the understanding obtained, it was possible to create design procedures that improved on the robustification properties of simply averaging the cost. These included applying the averaging process to model phase at each frequency, then independently for each frequency adjusting the learning rate to achieve stability, and for models not stabilizable by these methods, one can relax the condition requiring convergence to zero error for selected frequency components of the error.

This paper develops the extension of these three kinds of robustification from the repetitive control problem to the iterative learning control problem. The RC results are based on steady state frequency response properties. ILC is a finite time problem, and hence it is never completely in steady state. The singular value decomposition of the Toeplitz matrix of Markov parameters, that produces the convolution sum particular solution for the output, can be related to frequency response, and in the limit as the number of time steps in the trajectory gets large, the singular value decomposition of this matrix converges to the frequency response of the system. This paper makes use of this understanding. And as much as possible, it creates the finite time design analog for ILC of the robustification results for RC.

4.13 References


Chapter 5
Small Gain Stability Theory for Matched Basis Function Repetitive Control

5.1 Introduction

Repetitive Control (RC) methods normally only address one period, so in spacecraft applications they can apply to disturbance environments such as a cryogenic pump or a momentum wheel. References [13~18] present RC methods that address multiple unrelated periods, as would be needed to handle disturbances from imbalance in three reaction wheels or four CMGs. Until one introduces a frequency cutoff, these methods are addressing all harmonics of each period included. Robustness to model error deteriorates as more periods are included (Reference [18]).

Various Iterative Learning Control (ILC) and Repetitive Control (RC) approaches make use of the concept of basis functions, as in References [19~22]. The most useful basis functions are simple sine and cosine functions of the frequencies of interest. References [23~28] present Matched Basis Function Repetitive Control (MBFRC), which uses the projection algorithm commonly applied in adaptive control (Reference [29]) to obtain the components of the error on sines and cosines of the frequencies of interest, and applies sine and cosine modifications to the system input that include adjustment of the amplitude and phase change going through the system in order to have the output error be cancelled. These adjustments define the matched basis functions, matching feedback control system input sinusoids to their resulting control system output sinusoids. As in other forms of RC, an integration is included to create
convergence to zero error at the addressed frequencies in spite of substantial model error. Reference [25] reports experimental tests on a Stewart platform, of a type that has been flown on a spacecraft to test vibration isolation algorithms. The same platform is used in Reference [2].

Multiple period RC and MBFRC each have their own potential advantages. Multiple period RC simultaneously addresses all frequencies of the periods considered until one cuts out high frequencies with a cutoff filter. MBFRC on the other hand, introduces a separate RC controller for each frequency to be addressed, requiring many controllers for many harmonics. If there are many harmonics that need to be addressed the former approach has an advantage. In exchange for the added complexity of one controller for each frequency, the problem of robustness to high frequency model error is alleviated when using MBFRC. One expects potential improvement in the waterbed effect using MBFRC by allowing high frequencies, above the desired cutoff, to absorb some of the required amplification. An advantage of MBFRC is that multiple unrelated frequencies are addressed in a simple manner, without the complexity needed in References [14~18]. Yet another advantage relates to interpolation. The usual RC approaches require interpolation when the period of the addressed frequency is not an integer number of time steps (Reference [30]). And the interpolation deteriorates at high frequencies. In MBFRC, the basis functions at each frequency allow one to “interpolate” with the actual frequency function of interest.

MBFRC projects the error onto sinusoids and then applies the matched sinusoids to the system. This results in linear equations but with periodic coefficients. References [23,24,26] use Floquet theory, or time domain raising, to study stability when the frequencies of interest have periods that are integer multiples of the sampling time interval. Under the same assumption, Reference [27] developed stability analysis using the frequency raising technique. A very
interesting result of this approach is that the controller involving linear equations with periodic coefficients related to the basis functions, is seen to have a linear time invariant equivalent model.

The purpose of this chapter is to use the time invariant repetitive controller representation to develop very general and simple small gain stability robustness results. The result is obtained by using the departure angle condition from the theory of root locus plots. This approach was used previously to study the simplest form of RC in Reference [31]. It represents a strong robustness result guaranteeing convergence to zero tracking error for all sufficiently small gain, provided the phase information about your system response for each addressed frequency is accurate to within ±90 degrees. The result is independent of the system behavior at any other frequency. An additional bonus for the design method presented here is that it no longer requires that the frequencies being addressed have periods that are integer multiples of the sample time interval.

In the next sections, first the MBFRC algorithm is presented, then the equivalent time invariant control laws for each frequency. Then the case of addressing one frequency only is treated to determine the departure angles from poles on the unit circle, which is then generalized to apply to any number of addressed frequencies. Numerical examples are presented.

### 5.2 The Matched Basis Function Repetitive Control Algorithm

This section summarizes the MBFRC algorithm. Usually the RC controller adjusts the command to a feedback control system, although it need not be a feedback system. Consider a single-input, single-output system

\[
\begin{align*}
x(k+1) &= Ax(k) + Bu(k) \\
y_{fb}(k) &= Cx(k)
\end{align*}
\]  

(5-1)
whose transfer function is given by $G(z)$. The actual output $y(k) = y_{fb}(k) + w(k)$ contains the deterministic disturbance $w(k)$ given as an output disturbance. Wherever the disturbance enters in the feedback control system there is an equivalent disturbance that can be added to the output as is done here. The desired output is $y_d(k)$ and the associated tracking error is $e(k) = y_d(k) - y(k)$.

The disturbance is the sum of sinusoids at $N+1$ frequencies $\omega_n$ where $n = 1, 2, 3, ..., N$, and $n = 0$ is used for DC or zero frequency. The desired output $y_d(k)$ may be a constant, or can be a sum of sinusoids at these frequencies. Of course, one can also consider commands $y_d(k)$ and disturbances in $w(k)$ at frequencies not being addressed in this set of $N+1$, and study the behavior of the resulting design for such situations.

The frequency response of the system is such that an input $u(k) = \cos(\omega_n kT) = \cos(\phi_n k)$ where $T$ is the sample time interval of the digital control, results in a steady state output given by $y_{fb}(k) = r_n \cos(\phi_n k + \tau_n)$. The radian frequencies addressed, $\omega_n$, have an upper limited of Nyquist frequency $\pi / T$, and the normalized frequencies $\phi_n$ then range from 0 to $\pi$.

The matched input and output basis function are then given as follows. The terms input and output refer to the input and output of the feedback control system. For frequency $n$ one needs two output basis functions given in matrix $H_n(k)$, and the matched input basis function are then given in $F_n(k)$

$$H_n(k) = \begin{bmatrix} \cos(\phi_n k) & \sin(\phi_n k) \end{bmatrix}$$

$$F_n(k) = \begin{bmatrix} (1/r_n) \cos(\phi_n k - \tau_n) & (1/r_n) \sin(\phi_n k - \tau_n) \end{bmatrix}$$

The projection algorithm finds the components of the output error on the output basis functions
\[
\beta_n(k+1) = [I - aH_n^T(k+1)H_n(k+1)]\beta_n(k) + aH_n^T(k+1)e(k+1) \\
= A_{jn}(k)\beta_n(k) + B_{jn}(k)e(k+1) \quad n = 1, 2, 3, \ldots, N
\] (5-3)

The algorithm converges for \(0 < a < 2\). To handle DC, note that \(\phi_0 = 0\) produces \(H_0 = 1, F = 1/r_0\) which are now scalars instead of matrices.

Repetitive control uses the discrete form of an integrator to force convergence to zero error. In the present context this becomes the discrete form of an integral of the output basis function components \(\alpha_n(k+1) = \alpha_n(k) + \Lambda_n\beta_n(k)\). Then the command input is formed from the linear combination of the input basis functions.

\[
u(k) = \sum_{n=0}^{N} F_n(k)\alpha_n(k)
\] (5-4)

For future reference, the gain \(\Lambda_n = \Phi\lambda_n\) is split into an overall gain \(\Phi\) and a separate gain to use for each frequency \(\lambda_n\). The algorithm is summarized in Figure 5-5-1.
5.3 The Structure of the Time Invariant Equivalent of MBFRCs

Reference [27] used frequency raising as a method to study the stability of MBFRC. To do so it had to make the assumption that the period of the periodic function of interest is an integer $N$ number of sample times $T$. This meant that $\phi_n$ must be an integer multiple of $\theta_n = 2\pi / N$. Under these conditions it was possible to develop a time invariant relationship between the error $e(k)$ to the command $u(k)$, in spite of the periodic coefficients appearing in intervening steps. To do this was sufficiently complicated that Mathematica was used to handle the algebra. The result is an expression for the MBFRC controllers that is simply a well chosen
pole/zero configuration. And now, one can make use of this pole/zero version of MBFRC to study stability, which is the subject of this paper.

An important additional benefit of the pole/zero design is that it can be used for any frequency of interest and is not restricted to an integer multiple of $\theta_\eta$. This fact directly addresses the interpolation problem that is present in other forms of RC, in effect it is doing the interpolation using the basis functions which is precisely what is needed for a perfect interpolation.

The basic structure of the new version of MBFRC is given in Figure 5-2. Each of the $T_n(z)$ is the pole/zero compensator needed for frequency $\phi_n$, and these will be discussed in detail in the next section. A new aspect has been introduced into the block diagram, the feedforward path that adds $Y_d(z)$ to $U(z)$, see Reference [11] for discussion of use of this loop in other forms of RC.

\begin{equation}
U(z) = \Phi R(z) E(z) \quad \text{where}
\end{equation}

\begin{equation}
R(z) = \sum_{n=0}^{N} \lambda_n T_n(z)
\end{equation}
First consider the case when the feedforward loop is not present. Then block diagram algebra establishes that the transfer functions from command and output disturbance to output and to error are given by

\[
Y(z) = T_{cl}(z)Y_d(z) + T_s(z)W(z) \\
E(z) = T_s(z)[Y_d(z) - W(z)]
\]  
(5-6)

where the closed loop command to output transfer function \( T_{cl}(z) \) and the repetitive control sensitivity transfer function from output disturbance to error are

\[
T_{cl}(z) = \frac{\Phi R(z)G(z)}{1 + \Phi R(z)G(z)} \quad T_s(z) = \frac{1}{1 + \Phi R(z)G(z)}
\]  
(5-7)

Now consider the case when the feedforward loop is present

\[
Y(z) = T_{ff}(z)Y_d(z) + T_s(z)W(z) \\
E(z) = T_s(z)(1 - G(z))Y_d(z) - T_s(z)W(z) \\
T_{ff}(z) = [1 + \Phi R(z)]G(z) / [1 + \Phi R(z)G(z)]
\]  
(5-8)

To interpret this result, consider that the feedback control system consists of a controller \( C(z) \), a plant \( P(z) \), and unity feedback. Also consider that the actual disturbance is \( \widetilde{W}(z) \) which enters the feedback control system in the usual location, between the controller and the plant, so that the equivalent output disturbance is \( W(z) = S(z)\widetilde{W}(z) \), where \( S(z) \) is the sensitivity transfer function of the feedback control system. It is given by \( S(z) = 1/[1 + P(z)C(z)] \), and the feedback control system closed loop transfer function is \( G(z) = P(z)C(z) /[1 + P(z)C(z)] \). Therefore \( S(z) = 1 - G(z) \). We conclude that

\[
E(z) = T_s(z)S(z)[Y_d(z) - \widetilde{W}(z)]
\]  
(5-9)

The difference between the transfer function from command to error with the feedforward signal and without the feedforward signal is of no significance if the command is composed only of frequencies being addressed. But if one wishes to give commands that are unrelated to the
frequencies addressed, and are using the RC for the purpose of eliminating periodic disturbances, then one should use the feedforward signal to get the benefit of the feedback control system performance for these frequencies.

5.4 The Pole/Zero Repetitive Controllers for Each Frequency

The repetitive control systems designed by MBFRC have poles on the unit circle at the frequencies for which we seek zero tracking error. This creates a discrete version of an integral, and just as integral control with a pole on the unit circle at \( z = +1 \) will not tolerate a constant error, the poles on the unit circle at nonzero frequencies will not tolerate a steady state error at these frequencies. MBFRC only places poles at the addressed frequencies, while the usual RC designs that address all frequencies of a given period have poles evenly spaced around the unit circle, for the fundamental and all harmonics and DC. For stability, we need all poles on the unit circle to move toward the inside of the unit circle when the gain is increased from zero. We will prove that the MBFRC design method has the property that all such poles depart radially inward. We comment that if one wanted to design a controller by placing poles and zeros, with the aim of making every pole on the unit circle depart inward, it is not obvious how to accomplish this, especially with many frequencies being addressed. By taking the seemingly unlikely circuitous path of going to the projection algorithm and the matched basis functions in MBFRC, and then using frequency raising, followed by complex computations using Mathematica, we are able to design pole/zero configurations that do precisely what we want – depart radially inward (Reference [27]).
5.4.1 The RC Transfer Functions For Each Frequency

The repetitive controller for the \( n \)th addressed frequency has the following transfer function

\[
T_n(z) = \left(\frac{a_n}{r_n}\right) \frac{[\cos(\phi_n - \tau_n)z^2 - 2\cos(\tau_n)z + \cos(\phi_n + \tau_n)]z}{[z^2 + (a_n - 2)\cos(\phi_n)z + (1 - a_n)][z^2 - 2\cos(\phi_n)z + 1]}
\]

\( n = 1, 2, 3, ..., N \) (5-10)

This transfer function simplifies for the case of DC, \( n = 0 \), by setting \( \phi_0 = \tau_0 = 0 \), and then noting that a factor of \([z-1]^2\) cancels from the numerator and denominator to result in the following

\[
T_0(z) = \frac{\left(\frac{a_0}{r_0}\right)z}{[z^2 + (a_0 - 2)z + (1 - a_0)]}
\]

(5-11)

Examine the poles and zeros of this case. There is one zero located at the origin. The two poles are located at +1 and \((1-a_0)\). Recall that the gain \( a_0 \) must be in the open interval from zero to two, so the second pole starts at +1 and goes to -1 as \( a_0 \) increases from zero to two. When \( a_0 = 1 \), the pole is at the origin on top of the zero.

5.4.2 The Pole Locations For \( n = 1, 2, 3, ..., N \)

The integrator poles on the unit circle at the addressed frequency \( \phi_n \) are given by

\[
[z^2 - 2\cos(\phi_n)z + 1] = (z - P_n)(z - \overline{P}_n) \quad P_n = e^{i\phi_n} \quad \overline{P}_n = e^{-i\phi_n}
\]

(5-12)

The other two poles are

\[
p_{n1,n2} = -(\frac{1}{2})(a_n - 2)\cos(\phi_n) \pm (\frac{1}{2})\sqrt{(a_n - 2)^2 \cos^2(\phi_n) - 4(1 - a_n)}
\]

(5-13)

Note that when \( a_n = 0 \) these poles are at \( P_n \) and \( \overline{P}_n \). When \( a_n = 1 \) the roots are real and are at 0 and \( \cos(\phi_n) \). When \( \phi_n \) corresponds to 90 degrees, i.e. for half Nyquist frequency, there is a repeated pole at the origin. When \( a_n = 2 \) the poles are at \( \pm 1 \). Figure 5-3 presents the root locus
for these two roots for different values of the addressed frequency $\phi_n$ as $a_n$ goes from zero to two.

Figure 5-3. Root locus for pole $p_{n1,n2}$ locations for different addressed frequencies $\phi_n$.

Figure 5-4 gives the values of $a_n$ at which the roots enter the real axis for different addressed frequencies $\phi_n$. This happens when the square root in Equation (5-13) becomes zero. Note that when $1 \leq a_n \leq 2$ the roots are always real, and otherwise they are complex when $\phi_n$ satisfies

$$\cos^{-1}\left(\frac{2\sqrt{1-a_n}}{2-a_n}\right) < \phi_n < 180^\circ - \cos^{-1}\left(\frac{2\sqrt{1-a_n}}{2-a_n}\right)$$

(5-14)

Figure 5-5 gives the location of the entry to the real axis as a function of the addressed frequency $\phi_n$. This is the value of the first term on the right of Equation (5-13) when the value of $a_n$ makes the square root zero. This $a_n$ is given by $2\sin\phi_n(1-\sin\phi_n)/\cos^2\phi_n$ which takes on the value +1 when $\cos^2\phi_n = 0$. The resulting arrival position on the real axis is given by $(1-\sin\phi_n)/\cos\phi_n$ which has the value zero when the denominator is zero. Note that the arrival location to the real axis is approximately a linear function of the addressed frequency.
5.4.3 The Zero Locations For $n = 1, 2, 3, \ldots, N$

The poles discussed above are functions of the addressed frequency $\phi_n$ and the projection algorithm gain $a_n$. The zeros are independent of this gain, but are functions of the phase change through the system $\tau_n$. One zero is always located at the origin. The two remaining zeros are located at

$$z_{n1,n2} = \frac{\cos \tau_n \pm \sqrt{\cos^2 \tau_n - \cos(\phi_n - \tau_n) \cos(\phi_n + \tau_n)}}{\cos(\phi_n - \tau_n)}$$

(5-15)

Consider the term inside the square root. Use the formula for $\cos(x \pm y)$, replace $1 - \cos^2 \phi_n$ by $\sin^2 \phi_n$, and $\cos^2 \tau_n + \sin^2 \tau_n$ by unity to find that this term is equal to $\sin^2 \phi_n$. Then the zeros are

$$z_{n1,n2} = \frac{\cos \tau_n \pm \sin \phi_n}{\cos(\phi_n - \tau_n)}$$

(5-16)

Note that these zeros are independent of $a_n$, and are always real. Figures 5-6 and 5-7 show the $z_{n1}$ and $z_{n2}$ locations as a function of $\phi_n - \tau_n$ for different values of the phase change $\tau_n$ through
the system for the addressed frequency. Because the coefficient of $z^2$ goes to zero when $\phi_n - \tau_n$ reaches 90 degrees, the root $z_{n1}$ goes to plus infinity as the angle approaches 90 degrees from below, and once it passes 90 degrees and the coefficient is now negative, the roots start coming in from minus infinity. The other root $z_{n2}$ stays finite, staying between -1 and approximately 1.5.

Figure 5-6. The location of the zero $z_1$ as a function of $\phi_n - \tau_n$ for different values of $\tau_n$.

Figure 5-6. The location of the zero $z_2$ as a function of $\phi_n - \tau_n$ for different values of $\tau_n$. 
5.5 Small Gain Stability Theory for a Single Addressed Frequency

In this section we develop a small gain stability result for MBFRC for the case when it addresses only one frequency, which can be DC. We use the departure angle condition of root locus plots to show that the integrator poles on the unit circle depart radially inward, becoming stable, as the overall gain $\Phi$ increases from zero, thus guaranteeing that for sufficiently small gain the MBFRC system is asymptotically stable. In the next section we then show that the MBFRC with as many frequencies as desired has the same property that all poles on the unit circle depart radially inward. This property is not influenced by what and how many frequencies are being addressed. Note that unlike the usual RC that applies to one period, where the frequencies all are fundamental and harmonics, the frequencies that MBRFC can address can be totally independent of each other.

5.5.1 The Characteristic Polynomial

The characteristic polynomial from Figure 5-2 for the case of addressing one frequency $\phi_n$ can be written in the form needed for a root locus plot for gain $\Phi$ as

$$\Phi \lambda_n T_n(z) G(z) = -1 \quad (5-17)$$

First we treat the special case of DC, and obtain the needed result. Using Equation (5-11) in (5-17)

$$\frac{\Phi (\hat{\lambda}_n a_o / r_o) z G(z)}{[z-1][z-(1-a_o)]} = -1 \quad (5-18)$$

We assume that $G(z)$ is asymptotically stable, and for $a_o$ in the required range for the projection algorithm, the root $1-a_o$ is inside the unit circle. The root locus condition that the locus exists on the real axis to the left of an odd number of real zeros plus real poles, indicates that the one
root on the imaginary axis, at \( z = 1 \), departs radially inward along the real axis, giving the desired result for DC.

### 5.5.2 The Root Locus Departure Angle Condition

Now consider any nonzero frequency \( \phi_n \). Writing Equation (5-17) in detail gives

\[
\Phi \left( \frac{a_n}{r_n} \right) \left( \lambda_n \cos(\phi_n - \tau_n) \right) \frac{[z - z_{1n}][z - z_{2n}]zG(z)}{[z - p_{1n}][z - p_{2n}][z - \tilde{p}_n][z - \tilde{\tilde{p}}_n]} = -1 \quad (5-19)
\]

\[
\Phi \left( \frac{a_n}{r_n} \right) \left( \lambda_n \cos(\phi_n - \tau_n) \right) \frac{\rho_{n_1}e^{i\theta_{n_1}}\rho_{n_2}e^{i\theta_{n_2}}\rho_{n_3}e^{i\theta_{n_3}}}{\rho_{np_1}e^{i\theta_{np_1}}\rho_{np_2}e^{i\theta_{np_2}}\rho_{np_3}e^{i\theta_{np_3}}}z = -1 \quad (5-20)
\]

Any \( z \) satisfying Equation (5-19) is on the root locus for some gain \( \Phi \). In Equation (5-20) the complex number represented by each factor in (19) is written in polar form. Note that the phase response of \( G(z) \) is indicated here by \( \tau_{nn} \). For later use this indicates the actual phase response for the world, and \( \tau_{nm} \) will be used to indicate the phase response of the model we use to design the MBFRC which might be incorrect. For purposes of this section we assume that they are equal.

We assume that \( \Phi \) is positive, but because \( \cos(\phi_n - \tau_n) \) could be either positive or negative, we consider the possibility that one might want \( \lambda_n \) to be negative. Then when the product is positive, replace \( -1 \) by \( \exp(i(\pi \pm 2\pi \ell)) \) where \( \ell \) could be any integer. In the case the product is negative, replace \( -1 \) by \( \exp(i(\pm 2\pi \ell)) \) and introduce absolute values on \( \lambda_n \cos(\phi_n - \tau_n) \).

Equating the angle of the complex number on the left of the equality to that on the right produces the root locus formula for the problem of interest

\[
[(\theta_{n1} + \theta_{n2}) + \theta_{n3} + \theta_{nm}] - [(\theta_{np1} + \theta_{np2}) + \theta_{np3} + \theta_{nm}] = [(\theta_{n1} + \theta_{n2} + \phi_n + \tau_{nm}) - [(\theta_{np1} + \theta_{np2} + \phi_n + \tau_{nm}) + \theta_d + 90^\circ]]
\]

\[
= \begin{cases} 
180^\circ \pm \ell 360^\circ & \text{for } \lambda_n \cos(\phi_n - \tau_{nm}) > 0 \\
\pm \ell 360^\circ & \text{for } \lambda_n \cos(\phi_n - \tau_{nm}) < 0
\end{cases}
\]

(5-21)
The $\ell$ can be any integer. The first expression on the left applies to any $z$ on the locus. Our interest is the departure angle from the pole at $z = e^{i\phi}$ so that all angles are computed for $z$ arbitrarily close to this value. Note that the departure angle of interest is $\theta_{nP}$, which for clarity we denote by $\theta_d$. Of course, the other departure angle of interest $\theta_{nP}$, can be obtained immediately by the complex conjugate nature of the roots, once $\theta_d$ has been found. For the $z$ of interest, $\theta_{nP} = 90^\circ$ and $\theta_{z3} = \phi_n$, as indicated on the second line of Equation (5-21).

We seek to prove that the departure angle is radially inward. The complex number $z = e^{i\phi} = P$ is a phasor that is radially outward for this pole on the unit circle. Therefore the desired departure angle is $\theta_d = \phi_n + 180^\circ$ modulo $360^\circ$. In the next three subsections, we will show that

\[
(\theta_{nc1} + \theta_{nc2}) = \begin{cases} 
\phi_n - \tau_{nm} + 180^\circ & \text{for } \lambda_n \cos(\phi_n - \tau_{nm}) > 0 \\
\phi_n - \tau_{nm} & \text{for } \lambda_n \cos(\phi_n - \tau_{nm}) < 0
\end{cases}
\tag{5-22}
\]

\[
(\theta_{np1} + \theta_{np2}) = \phi_n + 90^\circ \forall \ a_n
\tag{5-23}
\]

Substituting these into the second version of Equation (5-21) produces the following result

\[
(\phi_n - \theta_d) - (\tau_{nm} - \tau_{nw}) = 180^\circ \pm \ell 360^\circ
\tag{5-24}
\]

Therefore, when the phase change through the system in the model used to design the MBFRC matches that in the real world, so that $(\tau_{nm} - \tau_{nw}) = 0$, the departure angle is the desired radially inward direction from the poles on the unit circle. And when there is error in the phase information, this error is the amount of deviation from radially inward departure.
5.5.3 Departure Angle Contribution From $z_{n1,n2}$

The objective of this section is to establish Equation (5-22). We examine the departure from the pole at $P_n$, so that the angles $\theta_{n1,n2}$ represent the angles the complex number $P_n - z_{n1,n2}$ makes with the positive real axis, and hence can be written as

$$\theta_{n1,n2} = \angle\{P_n - z_{n1,n2}\} = \angle\left[ \frac{\cos \phi_n \cos(\phi_n - \tau_n) - \cos \tau_n \pm \sin \phi_n}{\cos(\phi_n - \tau_n)} + i \sin \phi_n \right]$$

(5-25)

We adopt the convention that the angle for the first subscript, $n1$, refers to the upper sign, and the second subscript to the lower sign when there are sign choices in the equation. Examine the numerator in the real part of the complex number. Using the trigonometric identity to write $\cos(\phi_n - \tau_n)$ in terms of trigonometric functions of each angle, factoring out $\sin \phi_n$, and recognizing $\sin(\phi_n - \tau_n)$ in the result, produces the numerator in the form $\sin \phi_n [\mp 1 - \sin(\phi_n - \tau_n)]$. Note that both the real part and the imaginary part contain the factor $\sin \phi_n$ which is always positive (DC has been treated separately), so this factor can be eliminated without changing the angle. Since $\cos(\phi_n - \tau_n)$ can be both positive and negative, let $\Delta = \text{sgn}[\cos(\phi_n - \tau_n)]$. Then multiply both real and imaginary parts by $|\cos(\phi_n - \tau_n)|$ and factor out $\Delta$ to obtain

$$\theta_{n1,n2} = \angle \Delta [\mp 1 - \sin(\phi_n - \tau_n)] + i \cos(\phi_n - \tau_n) \}$$

(5-26)

For later use, let $w_{n1}$ and $w_{n2}$ be the magnitudes of the complex numbers for each of the angles respectively, then

$$w_{n1,n2} = [2 \pm 2 \sin(\phi_n - \tau_n)]^{1/2}$$

$$w_{n1}w_{n2} = [4 \cos^2(\phi_n - \tau_n)]^{1/2} = 2 \Delta \cos(\phi_n - \tau_n)$$

(5-27)

Since these are magnitudes, we must use the positive square root as indicated.
Our objective is to compute both of the following

\[
\sin(\theta_{nc1} + \theta_{nc2}) = \sin \theta_{nc1} \cos \theta_{nc2} + \cos \theta_{nc1} \sin \theta_{nc2} \\
\cos(\theta_{nc1} + \theta_{nc2}) = \cos \theta_{nc1} \cos \theta_{nc2} - \sin \theta_{nc1} \sin \theta_{nc2}
\]

(5-28)

From the complex numbers for each angle

\[
\sin \theta_{nc1} = \frac{[\Delta \cos(\phi_n - \tau_n)]}{w_{n1}} \quad \cos \theta_{nc1} = \frac{\Delta [-1 - \sin(\phi_n - \tau_n)]}{w_{n1}} \\
\sin \theta_{nc2} = \frac{[\Delta \cos(\phi_n - \tau_n)]}{w_{n2}} \quad \cos \theta_{nc2} = \frac{\Delta [-1 - \sin(\phi_n - \tau_n)]}{w_{n2}}
\]

(5-29)

one computes that

\[
\sin(\theta_{nc1} + \theta_{nc2}) = -\Delta \sin(\phi_n - \tau_n) \\
\cos(\theta_{nc1} + \theta_{nc2}) = -\Delta \cos(\phi_n - \tau_n)
\]

(5-30)

If \( \Delta \) is negative, it is evident that \( \theta_{nc1} + \theta_{nc2} = \phi_n - \tau_n \), and otherwise one adds 180° to the right hand side. This establishes Equation (5-22).

5.5.4 The Special Case of \( \cos(\phi_n - \tau_n) = 0 \)

For this special case, the compensator becomes

\[
T_n(z) = \left( \frac{a_n}{r_n} \right) \frac{[-2 \cos(\tau_n)][z - \cos(\phi_n + \tau_n)]/(2 \cos(\tau_n))z}{[z^2 + (a_n - 2) \cos(\phi_n)z + (1 - a_n)][z^2 - 2 \cos(\phi_n)z + 1]}
\]

(5-31)

One of the zeros has disappeared. The angle for the remaining zero is

\[
\angle[\exp(\phi_n) - \cos(\phi_n + \tau_n)/2 \cos(\tau_n)] = \angle \left[ \frac{\cos \phi_n \cos \tau_n + \sin \phi_n \sin \tau_n}{2 \cos(\tau_n)} + i \sin \phi_n \right]
\]

(5-32)

The condition \( \cos(\phi_n - \tau_n) = 0 \) means that \( \phi_n = \tau_n + 90° \pm \ell' 180° \) for some integer value of \( \ell' \). If, \( \phi_n = \tau_n + 90° \) then \( \cos \phi_n = -\sin \tau_n \) and \( \sin \phi_n = \cos \tau_n \). The signs are reversed if \( \phi_n = \tau_n - 90° \).

Other values of \( \ell' \) repeat these two possibilities. In both cases the real part is zero in Equation (5-32), and hence the angle contribution for this zero, denoted \( \theta_{nc12} \), is \( \angle[i \sin \phi_n] \), or 90°. Note
that the gain for this root locus, \( \Phi \lambda_n [-2 \cos(\tau_n)] \), could be either positive or negative. Since there is only one instead of two extra zeros in this case, angle condition Equation (5-21) becomes

\[
[\theta_{n12} + \theta_c + \tau_{nw} - [(\theta_{np1} + \theta_{np1}) + \theta_{ap} + \theta_{ap}]] = [90^\circ + \phi_n + \tau_{nw}] - [(\phi_n + 90^\circ) + \theta_d + 90^\circ]
\]

\[
= \begin{cases} 
0^\circ \pm 1360^\circ & \text{for } -\lambda_n \cos(\tau_{nw}) > 0 \\
180^\circ \pm 1360^\circ & \text{for } -\lambda_n \cos(\tau_{nw}) < 0
\end{cases} \quad (5-33)
\]

where \( \theta_{n12} = 90^\circ \) from above and \( (\theta_{np1} + \theta_{np1}) = (\phi_n + 90^\circ) \) (see Equation (5-23)). Then

\[
\theta_d - \phi_n = \begin{cases} 
(\tau_{nw} - \phi_n) - 90^\circ \mp 1360^\circ & \text{for } -\lambda_n \cos(\tau_{nw}) > 0 \\
(\tau_{nw} - \phi_n) - 90^\circ \mp 1360^\circ - 180^\circ & \text{for } -\lambda_n \cos(\tau_{nw}) < 0
\end{cases} \quad (5-34)
\]

Pick \( \text{sgn}(\lambda_n) = +\text{sgn}(\tau_{nw}) \) when \( \tau_{nw} - \phi_n = +90^\circ \), and \( \text{sgn}(\lambda_n) = -\text{sgn}(\tau_{nw}) \) when \( \tau_{nw} - \phi_n = -90^\circ \). In either case \( \theta_d - \phi_n = -180^\circ + 1360^\circ \). Therefore, even if \( \cos(\phi_n - \tau_n) = 0 \), the departure angle is still radially inward provided one picks the sign of \( \lambda_n \) according to

\[
\text{sgn}(\lambda_n) = \text{sgn}(\cos(\tau_{nw})) \text{sgn}(\phi_n - \tau_{nw}) \quad (5-35)
\]

If \( \tau_{nw} - \phi_n = +90^\circ \) and \( \cos(\tau_n) > 0 \), then \( \lambda_n > 0 \), and the same is true with \(-90^\circ \) and \( < 0 \). And with \(+90^\circ \) and \( < 0 \), or \(-90^\circ \) and \( > 0 \) one uses \( \lambda_n < 0 \).

### 5.5.5 Departure Angle Contribution From \( p_{n1,n2} \), Real Root Case

This section establishes Equation (5-23) for the case that the poles \( p_{n1,n2} \) are real so that

\[
p_{n1,n2} = -(\frac{1}{2})(a_n - 2) \cos \phi_n \pm (\frac{1}{2})\sqrt{\Gamma} \quad \Gamma = (a_n - 2)^2 \cos^2 \phi_n - 4(1 - a_n) \quad (5-36)
\]

\[
P_n - p_{n1,n2} = \left[ (\frac{1}{2})a_n \cos \phi_n \mp (\frac{1}{2})\sqrt{\Gamma} \right] + i \sin \phi_n \quad (5-37)
\]

\[
\theta_{np1} = \angle(P_n - p_{n1}) \quad, \quad \theta_{np2} = \angle(P_n - p_{n2})
\]

Instead of \( w_{n1} \) and \( w_{n2} \), denote the magnitudes of the complex number in Equation (5-37) by \( r_{n1} \) and \( r_{n2} \) which are given by
\[ r_{n1,n2} = (1/2)\left\{a_n^2 \cos^2 \phi_n + \Gamma + 4 \sin^2 \phi_n \pm 2a_n \cos \phi_n \sqrt{\Gamma}\right\}^{1/2} \quad (5-39) \]

The first three terms in the curly brackets equal \(((a_n^2/2) - a_n)\cos^2 \phi_n + a_n\) including the \((1/2)\) factor. Then \(r_{n1}r_{n2} = a_n \sin \phi_n\), making use of the fact that the product is a magnitude and that \(\sin \phi_n\) is always positive. Then we write Equation (5-28) for the argument \((\theta_{np1} + \theta_{np2})\) and in place of Equation (5-26) we have

\[
\begin{align*}
\sin \theta_{np1} &= \sin \phi_n / r_{n1} \\
\cos \theta_{np1} &= (1/2)\left[a_n \cos \phi_n - \sqrt{\Gamma}\right] / r_{n1} \\
\sin \theta_{np2} &= \sin \phi_n / r_{n2} \\
\cos \theta_{np2} &= (1/2)\left[a_n \cos \phi_n + \sqrt{\Gamma}\right] / r_{n2}
\end{align*}
\]

producing the following result that establishes Equation (5-23)

\[
\begin{align*}
\sin(\theta_{np1} + \theta_{np2}) &= \cos \phi_n \\
\cos(\theta_{np1} + \theta_{np2}) &= -\sin \phi_n
\end{align*}
\]

5.5.6 Departure Angle Contribution From \(p_{n1,n2}\), Complex Root Case

When the square root in Equation (5-36) is the root of a negative number, replace \(\sqrt{\Gamma} = i\sqrt{\Omega}\) where \(\Omega = -\Gamma\), and then Equation (5-37) becomes

\[
P_n - p_{n1,p2} = (\sqrt{\Omega})a_n \cos \phi_n + i\left[\sin \phi_n \mp (1/2)\sqrt{\Gamma}\right]
\]

The magnitudes of these two complex numbers are given by

\[
s_{n1,n2} = \left\{(1/4)a_n^2 \cos^2 \phi_n + \sin^2 \phi_n + \Omega / 4 \mp 2 \sin \phi_n \sqrt{\Omega}\right\}^{1/2}
\]

The first three terms are equal to \((2 - a_n)\sin^2 \phi_n\), and the product of the magnitudes matches the product obtained in the real root case. In the \((\theta_{np1} + \theta_{np2})\) version of Equation (5-28) we substitute

\[
\begin{align*}
\sin \theta_{np1} &= [\sin \phi_n - \sqrt{\Omega} / 2] / s_{n1} \\
\cos \theta_{np1} &= (1/2)a_n \cos \phi_n / s_{n1} \\
\sin \theta_{np2} &= [\sin \phi_n + \sqrt{\Omega} / 2] / s_{n2} \\
\cos \theta_{np2} &= (1/2)a_n \cos \phi_n / s_{n2}
\end{align*}
\]
and after simplifying, obtain Equation (5-41), which establishes the desired result.

### 5.5.7 Single Addressed Frequency Result

The previous subsections have proved the following asymptotic stability theorem.

**Theorem 1:** Given a MBFRC system as in Figure 5-2:

(i) With or without the feedforward loop.

(ii) Which addresses a single frequency $\phi_n$, which can be DC.

(iii) The system transfer function $G(z)$ is known to be asymptotically stable.

(iv) The phase response $\tau_n$ is known at the addressed frequency, where $G(e^{j\omega}) = r_n e^{j\zeta}$ (with $r_n > 0$ by definition).

(v) And $\text{sgn}(\lambda_n)$ is chosen according to Equation (5-35) as needed.

Then the MBFRC system has the following properties:

1. For any $a_n \in (0, 2)$ and any $r_n > 0$, the system is asymptotically stable for all sufficiently small gains $\Phi > 0$.

2. If the disturbance $w(k)$ and the desired output $y_d(k)$ are either zero or signals of the addressed frequency, then the output will converge to the desired output as $k \to \infty$ producing zero tracking error for all sufficiently small gains $\Phi > 0$.

Note that the only knowledge needed about the system is that it is linear, time invariant, and asymptotically stable, and one needs the phase change through the system at the addressed frequency. No other knowledge is needed and hence the asymptotic stability result is robust to inaccuracy to all other system properties. The robustness to error in the phase information is given by Equation (5-24).
Theorem 2: Asymptotic stability and convergence to zero tracking error according to Theorem 1, is also achieved for all phase discrepancies between the phase used to design the MBFRC law, and the real world phase at the addressed frequency, that satisfy

$$-90^\circ < (\tau_{nm} - \tau_{nv}) < 90^\circ$$  \hspace{1cm} (5-45)

We comment that this condition should easily be satisfied in applications.

5.6 Small Gain Stability Theorem for Arbitrary Number of Addressed Frequencies

This section generalizes the results to the case when the MBFRC addresses an arbitrary number of frequencies $N$ and also DC if desired. The characteristic polynomial becomes

$$\Phi[\hat{\lambda}_0 T_0(z) + \hat{\lambda}_1 T_1(z) + \cdots + \hat{\lambda}_N T_N(z)]G(z) = -1$$  \hspace{1cm} (5-46)

The system $G(z)$ is assumed asymptotically stable so it has all poles inside the unit circle. Each $T_n(z)$ has four poles, two are inside the unit circle, and two are on the unit circle at $P_n = e^{i\phi_n}$ and $\bar{P}_n = e^{-i\phi_n}$. When the left hand side is put over a common denominator, the resulting set of poles is composed of all of the poles of each term. When the gain $\Phi = 0$, all of the roots of the MBFRC system are at these poles. To study the departure angle from any chosen pole on the unit circle $P_n$, one picks a $z$ arbitrarily close to this pole but not at the pole. Then the denominator of $T_n(z)$ contains the factor $z - P_n$ which must approach zero as $\Phi$ approaches zero. All other terms $T_m(z)$, $m \neq n$ will have denominators bounded away from zero. We can rewrite Equation (5-46) as

$$\Phi[\hat{\lambda}_0 T_0(z) + \cdots + \hat{\lambda}_{n-1} T_{n-1}(z)]G(z) + \Phi \hat{\lambda}_n T_n(z)G(z) + \Phi[\hat{\lambda}_{n+1} T_{n+1}(z) + \cdots + \hat{\lambda}_N T_N(z)]G(z) = -1$$  \hspace{1cm} (5-47)
As $\Phi$ approaches zero, the first and third terms on the left of this equation are $\Phi$ times bounded functions, and hence they approach zero. As zero is approached only the middle term is able to match the -1 on the right hand side, since it is formed as a product of $\Phi$ going to zero times something with $z - P_n$ in the denominator, which is also going to zero. Hence, to study the departure angle from $P_n$ one only needs to examine the equation

$$\Phi \lambda_n T_n(z) G(z) = -1$$

Therefore, Theorem 1 derived for the case of addressing only one frequency, also applies to each frequency independently when addressing multiple frequencies.

**Theorem 3:** The MBFRC of Figure 5-2 having an arbitrary number of addressed frequencies, including DC if desired, and satisfying the conditions of Theorem 1 for each frequency independently, has the following properties:

1. The MBFRC system is asymptotically stable for all sufficiently small $0 < \Phi$, for correctly chosen $\text{sgn}(\lambda_n)$, for any $r_n > 0$, and for any choice of $a_n \in (0,2)$, for all $n$,

2. If the disturbance $w(k)$ and the desired output $y_d(k)$ are zero or signals composed of linear combinations of the addressed frequencies, then the tracking error will approach zero as $k \to \infty$ for all sufficiently small $0 < \Phi$.

3. The above two properties are maintained in the presence of modeling errors of the phase change through the system at the addressed frequencies, satisfying Equation (5-45) at all $n$. Convergence is independent of the frequency response at any other frequency, and independent of the magnitude response at each addressed frequency.

### 5.7 Numerical Examples

Consider applying MBFRC to a third order system given by
where \( b = 44 \), \( \zeta = 0.5 \), and \( \omega_u \) corresponds to \( 29.5Hz \). This system is fed by a zero order hold sampling at \( 100Hz \). The MBFRC law uses \( a_n = 0.7 \) and \( \lambda_n = 1 \) for all addressed frequencies which include 10 frequencies, \( 0 \) Hz up to \( 9 \) Hz in increments of \( 1 \) Hz. The overall gain is \( \Phi = 0.01 \). Figure 5-8 shows the sensitivity transfer function \( T_s(z) \) for the system in Figure 5-2 without the feedforward signal, giving the error as a function of frequency according to Equation (5-6). We see that it does produce zero error at each of the 10 addressed frequencies. The waterbed effect (Bode integral theorem) produces significant amplification of errors at frequencies between those addressed. The feedforward signal is introduced in Figure 5-9, which shows the frequency response of the new sensitivity transfer function \( T_s(z)S(z) \) according to Equation (5-9). For disturbances \( \hat{W}(z) \) occurring in the usual location between controller and plant in the feedback controller, there is no influence of the feedforward signal. Using the feedforward signal is only important if one may want to apply commands that are not restricted to linear combinations of the addressed frequency, in which case we want the sensitivity transfer function \( S(z) \) exhibiting the performance of the feedback controller response to commands to be present as in Equation (5-9).
Figure 5-8. Sensitivity transfer function magnitude response without the feedforward signal.

Figure 5-9. Sensitivity transfer function magnitude response with feedforward signal.

Figure 5-10. Error vs. time using MBFRC that is turned on at 4 sec.

Figure 5-10 shows the performance of the MBFRC when the command is set to zero, and the disturbance $w(k)$ is a linear combination of cosines of the 10 addressed frequencies above, with amplitudes 0.7, 0.5, 0.3, 0.4, 0.2, 0.1, 0.2, 0.1, 0.1, 0.15 for the frequencies in increasing order. When the MBFRC controller is turned on at 4 seconds the tracking error decreases to a small value relatively quickly.
Figures 5-11 and 5-12 give examples of the root locus plots. Figure 5-11 considers two addressed frequencies at 30% and 48% Nyquist frequency. The phases of the system frequency response for these two frequencies are $-126.38^\circ$ and $-186.82^\circ$ respectively. Figure 5-12 shows the changed when DC is included. The corresponding gains $\lambda_n$ are 0.7 for DC, and 1 and 0.7 for the other frequencies. All projection gains $a_n$ are 0.7. We observe the radially inward departures from the poles on the unit circle. The poles are all ones that we know from the denominators of the compensators and the system $G(z)$. The zeros however are altered when the controllers are added together in Equation (5-46). Note that introducing DC into the control law modifies the zero location that was on the positive real axis. Without this happening, the departure angle from the DC pole at +1 would be in the wrong direction.

Figure 5-11. Root locus plot for MBFRC addressing two frequencies.
5.8 Conclusions

Repetitive control aims to produce zero tracking error to periodic commands and to do so in the presence of periodic disturbances. Matched basis function repetitive control was initially based on using the projection algorithm from adaptive control to determine the components of the error on frequencies of interest. To get the repetitive control property of convergence to zero error, it introduced something equivalent to an integral at any frequency of interest. The resulting controller equations are linear with periodic coefficients. In a previous work the authors used frequency raising and the assumption that each addressed period was an integer number of time steps, and obtained a linear time invariant pole/zero transfer function that is equivalent to the periodic coefficient repetitive controller. This paper examines this pole/zero design. For each frequency addressed, it uses two poles on the unit circle at this frequency which produce the error integral that demands convergence to zero tracking error at this frequency. There are two additional poles inside the unit circle, one zero at the origin, and two additional zeros. Thinking in terms of classical control system design, when there are many unrelated poles on the unit circle, corresponding to many addressed frequencies, it is very hard to find a compensator that will pull these roots on the unit circle stability boundary into the stable region inside the unit
circle as the gain is turned up. This circuitous route of the projection algorithm, integration, periodic coefficient equations, and frequency raising succeeds in producing a simple set of pole/zero locations to do this, and they are uncoupled, one frequency at a time.

The main result of this paper establishes that this design approach is guaranteed to produce asymptotic stability for all sufficiently small gains. The only information that one needs to know about is the phase change from command to response of the feedback control system being used (and knowledge that it is an asymptotically stable). The result does not require any additional knowledge about the system behavior at any other frequency, does not require knowledge of the order, or number and locations of system zeros or poles. And the approach allows one to address an arbitrary number of frequencies that can be totally unrelated, e.g. they need not be harmonics. Concerning the accuracy needed for this one piece of information required, the phase change through the system at frequencies of interest, it is shown that asymptotic stability and convergence to zero tracking error is obtained for all phases used in the design, provided they do not differ by more than ±90 degrees from the true phase. For each frequency one chooses to address, one expects to be able to determine the phase information to within this generous accuracy limit. Thus, the small gain stability and the convergence to zero error properties are extremely robust.

A secondary result of this paper is to eliminate the assumption needed in stability analysis using either time domain raising or frequency domain raising, that the periods of the frequencies being addressed are all an integer number of time steps. Frequency raising showed us the pole zero design under this assumption. All of the small gain stability results here do not need this assumption. Hence, the approach developed here proves asymptotic stability for situations for which we do not have a proof for the repetitive controllers employing the projection algorithm.
One can compare the MBFRC design to more standard repetitive control design methods and to the filtered $x$-LMS approach. Concerning the latter, MBFRC assumes that you can stay synchronized to the disturbance frequency using for example and index pulse for each revolution of a momentum wheel. And it uses integral action to converge to zero tracking error employing knowledge of the phase change through the system at addressed frequencies. Convergence to zero error is obtained by what might be considered a direct approach. Filtered $x$-LMS instead requires a disturbance correlated signal. And it adaptively produces a finite impulse response model of the system which produces the needed phase information. It can be analogous to indirect adaptive approaches, and the imperfect FIR model structure could compromise the zero error performance. Both approaches need to use one controller for each frequency to be addressed. We can make a series of comments to compare the approach developed here to other repetitive control design approaches as in Reference [11].

**The Price Paid:** The price paid to obtain all of the good robustness properties listed above, is mainly related to the requirement of sufficiently small gain. Nothing is saying how small the gain has to be to have the stability and robustness properties. Whereas the robustness property is independent of the system being controlled in nearly all aspects except the phase information at addressed frequencies, the gain limit is likely very much system dependent. Limited experience suggests that this gain limit may be influenced adversely when one needs to address many frequencies. If the limit is low, then one is forced to have slow convergence.

**Frequency vs. Period:** MBFRC uses one controller for each frequency addressed, while in typical RC all frequencies with the same period are addresses simultaneously, i.e. DC, the fundamental, and all harmonics up do Nyquist frequency. Depending on how many harmonics are of interest, handing all harmonics at once can be an advantage.
**Cutoff Filter:** One very likely has large error in any model at high frequencies, due to inability to measure response at very high frequencies, due to parasitic or residual modes, etc. Since usual RC methods address all harmonics up to Nyquist, it is necessary for stability robustness to high frequency model error, to introduce a zero phase low pass filter. The model independence of the small gain results here indicate that no such filter is needed in MBFRC.

**Controller Order:** The usual RC design methods require a controller order higher than the number of time steps in a period. In addition, the zero-phase cut-off filter can significantly increase the order of the controller. Unless, the MBFRC is addressing very many harmonics, its order is likely to be significantly smaller than the RC design.

**Interpolation:** Usual RC design methods require interpolation of error signals when the frequencies of interest do not have periods that are an integer number of time steps. Interpolation compromises performance. MBFRC uses sine and cosine functions of the frequency of interest to perform the interpolation, and these are precisely the functions that should be used.

**Multiple Unrelated Periods:** Usual RC design methods need a special structure to address multiple periods (References [14~17] or [11]). This adds complexity to the control law, and also has an adverse influence on robustness to model error (Reference [18]). The MBFRC approach handles multiple unrelated frequencies as effortlessly as it does frequencies that are harmonics.

### 5.9 References


Chapter 6

Conclusions

Due to the unusual requirement of asking for zero tracking error for all frequencies or harmonics having the desired period all the way to Nyquist frequency, the stability robustness to model errors remains an important issue in Repetitive Control (RC) and Iterative Learning Control (ILC). Four methods have been developed in this thesis to specifically address the topic of stability robustification of RC and ILC.

Chapter 2 presents a new approach to robustifying repetitive control for models with uncertain parameters, generalizing an algorithm based on averaging cost functions over model parameter distributions. It emphasizes adjustment of the phase of the compensator at each frequency and still aims for fast learning. But in addition, the algorithm adjusts the learning rate, with the objective of adjusting the learning rate as a function of the parameter uncertainties at each frequency. This slows down the learning only at frequencies that need additional robustness, and maintains the faster learning rate for other frequencies. As a result, the method outperforms the prior robustification method based on averaging alone. This method is very effective for models with uncertain parameters and it produces a robust RC compensator with a fast learning speed that takes full advantage of frequency response based knowledge of the actual stability boundary.

Chapter 3, on the other hand, presents a method of extending the robustness to model error when the parameter uncertainties are too large to be handled by the methods of the previous chapter. It proposes an RC design that compromises on the zero error objective in order to
obtain improved robustness to model error. The approach presented here can be used in addition to the previous approach in Chapter 2. The additional robustification is accomplished in exchange for no longer converging to zero error in some frequency ranges. The method in Chapter 3 initially appears to be designing the compensator and the cutoff simultaneously. However, the final result obtained here has the property that the compensator is determined completely by the choice of the $R_d$ weights in the cost function, and is independent of the compromise made with zero error which is determined with knowledge of $R_d$ by the choice of $R_u$. Therefore, this approach here can also be thought of as a two stage process. What is new is the perspective that one does not need to make a sharp frequency cutoff filter as required in previous designs, but rather a very small decrease in the effort to get to zero error is likely enough to stabilize the repetitive control system. The compromise in not asking for zero error for certain frequencies is shown to be an effective tool in robustification. It can be combined with cost averaging and adjustment of the learning rate. By comparison with the use of a sharp cutoff for robustification, this approach can result in better performance in a frequency range just above what would have been the cutoff.

Chapter 4 investigates the generalization of the above two methods for robustification of the Repetitive Control (RC) problem to apply to the sister field of Iterative Learning Control (ILC). The RC results are based on steady state frequency response properties, while ILC is a finite time problem which is never completely in steady state. The singular value decomposition of the Toeplitz matrix of Markov parameters, that produces the convolution sum particular solution for the output, can be related to frequency response, and in the limit as the number of time steps in the trajectory gets large, the singular value decomposition of this matrix converges to the frequency response of the system. This chapter makes use of this understanding, and as
much as possible, it creates the finite time design analog for ILC of the robustification results for RC.

Chapter 5 develops a small gain theory for the stability of the Matched Basis Function Repetitive Control (MBFRC) algorithm, for stability robustness. MBFRC is formulated to project the error onto periodic basis functions, and this produces linear system equations with periodic coefficients. By use of the frequency raising technique one can replace the periodic coefficient system by a larger linear system. The result is a time invariant pole/zero controller design, one controller for each frequency considered. For each frequency addressed, it uses two poles on the unit circle at this frequency which produces the error integral that demands convergence to zero tracking error at this frequency. Two additional poles inside the unit circle, one zero at the origin, and two additional zeros are automatically “generated” by this algorithm. It is shown that this design ensures stability for all sufficiently small gains. The only information required is the phase change from command to response of the feedback control system being used at each frequency being addressed (and knowledge that it is an asymptotically stable). This approach allows one to address an arbitrary number of frequencies that can be totally unrelated. Thus, the small gain stability and the convergence to zero error properties are extremely robust. Compared to the standard Repetitive Control design methods and filtered $x$-LMS approach, MBFRC is found to have many advantages. First of all, MBFRC uses one controller for each frequency addressed, while in typical RC all frequencies with the same period are addresses simultaneously. This allows one to eliminate the need for a zero-phase low-pass cutoff filter that is required in most conventional RC designs. Even when addressing many harmonics, MBFRC creates a controller whose order is potentially smaller than many of the standard RC designs, because of the elimination of the need for a cutoff filter. Finally, the MBFRC algorithm handles
multiple unrelated frequencies as effortlessly as it does frequencies that are harmonics. The price paid to obtain all of the good robustness properties listed above, is mainly related to the requirement of sufficiently small gain. Limited experience suggests that this gain limit may be influenced adversely when one needs to address many frequencies. If the limit is low, then one is forced to have slow convergence, but this can be quite practical in many applications.