Int. J. Appl. Math. Comput. Sci., 2003, Vol. 13, No. 1, 101–111

ON THE INTERACTION BETWEEN THEORY, EXPERIMENTS, AND SIMULATION IN DEVELOPING PRACTICAL LEARNING CONTROL ALGORITHMS

RICHARD W. LONGMAN*

* Columbia University
New York, New York 10027, USA
e-mail: RWL4@columbia.edu

Iterative learning control (ILC) develops controllers that iteratively adjust the command to a feedback control system in order to converge to zero tracking error following a specific desired trajectory. Unlike optimal control and other control methods, the iterations are made using the real world in place of a computer model. If desired, the learning process can be conducted both in the time domain during each iteration and in repetitions, making ILC a 2D system. Because ILC iterates with the real world, and aims for zero error, the field pushes the limits of theory, modeling, and simulation, to predict the behavior when applied in the real world. It is the thesis of this paper that in order to make significant progress in this field it is essential that the research effort employ a coordinated simultaneous synergistic effort involving theory, experiments, and serious simulations. Otherwise, one very easily expends effort on something that seems fundamental from the theoretical perspective, but in fact has very little relevance to the performance in real world applications.

Keywords: iterative learning control, ILC, 2D systems, learning transients

1. Introduction

Starting in 1984 there has been considerable activity in the field of iterative learning control (ILC) initially motivated by robots performing repetitive tracking tasks. Learning controllers aim to improve their performance while repeatedly executing a task (Arimoto *et al.*, 1984; Bien and Xu, 1998; Casalino and Bartolini, 1984; Craig, 1984; Moore, 1993; Moore and Xu, 2000; Middleton *et al.*, 1985; Owens, 1977; Rogers and Owens, 1992; Uchiyama, 1978). For typical classical control systems executing the same tracking command repeatedly, there are repeating deterministic errors in following the desired trajectory. Often there are also repeating disturbance histories, such as the history of gravity torque on a robot link along the path. In ILC the robot or system is returned to the same initial condition before each repetition of the task. Hence, there are two independent variables, time and repetitions, making a 2D control problem. Often all of the learning is done in repetitions, and simple feedback control applies in the time domain, but a true 2D approach is logical and is applied by various researchers, for example (Amann *et al.*, 1998; Owens *et al.*, 2000). A closely related field is repetitive control (RC)—control systems that learn to improve their performance while executing a periodic command, or ones executing a constant command but subject to a periodic disturbance (Hara and Yamamoto, 1985; Hara *et al.*, 1985; Inoue *et al*, 1981; Longman, 1998; 2000; Middleton *et al.*, 1985; Nakano and Hara, 1986; Omata *et al.*, 1984; Tomizuka *et al.*, 1989). In RC the time and repetition domains get replaced by time and time shifted backward by one or more periods.

The learning process in both ILC and RC can take many forms (Bien and Xu, 1998; Longman, 1998; 2000; Moore, 1993; Moore and Xu, 2000; Rogers and Owens, 1992). For example, it can be based on integral control concepts from classical control theory, but applied in repetitions. It can be based on contraction mappings in either the time or frequency domains. It can be based on indirect adaptive control theory or model reference adaptive control theory operating in time or in repetitions or both. It can be based on numerical methods for minimising of a function, or numerical methods for root finding. Or one can try to model the system and invert it to find the input needed for the desired output. The word learning in the ILC name suggests to many the use of neural networks, which can in fact be applied to the problem, but the majority of the field makes use of control theoretic concepts as described above.

2. Approaches to Solving Engineering Problems

In the evolution of any engineering field, there is an interplay between three basic ingredients, three approaches to addressing new problems, i.e. three different points of view (Fig. 1):

1. the development of a mathematical theory,
2. the use of modeling and simulation,
3. experiments performed in the real world.

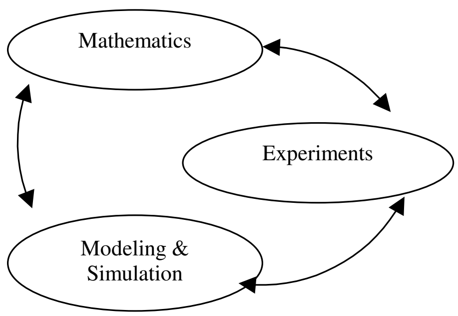

Fig. 1. Interplay between theory, experiment and simulation.

It is very tempting to specialize in one approach and ignore the others. A mathematician may have little interest in realistic simulations and even less interest in conducting experiments. There are engineers that solve all of their problems in the laboratory, simply finding a way in hardware to make it work, without the knowledge of the underlying theory.

It is claimed here that if you want to make major contributions to a new field, and especially to ILC, then you must make use of all three approaches. Using only one point of view may very easily lead to wrong conclusions, very easily make one spend effort at research objectives that do not actually make sense, in spite of the fact that they seem to make sense from the single point of view chosen. This happens even when one works hard to focus on problems that seem fundamental from this one point of view.

The field of ILC is unusual within engineering as a result of asking for convergence to zero tracking error—and doing so in the real world rather than with a mathematical model. Below we will see that this pushes the predictive limits of the approaches, and hence, ILC dramatizes the need of research to make use of all three approaches. This paper illustrates this need by documenting the sequence of research objectives of one group of researchers in the field, the author and co-workers, who repeatedly aimed for research objectives that missed the point, and repeatedly found that the experiments define what the theory should be addressing.

3. Possible Structures of ILC

This paper concentrates on linear discrete time ILC. For simplicity, single-input, single-output systems are considered. The system can be modeled in state variable form

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + v(k), \\ y(k) &= Cx(k), \end{aligned} \tag{1}$$

where A can be a closed loop system matrix if one learns only in repetitions, or it can be open loop. In these two cases, u can be the signal added to the desired trajectory for learning, or it can be the complete input signal including learning. Then $v(k)$ represents the repeating disturbance plus the desired trajectory forcing function, or just the repeating disturbance, respectively. The desired trajectory has p time steps, always starting from the same initial condition. The column matrix giving the input history at repetition j is written below in terms of input and error histories of the current and previous repetitions:

$$\begin{aligned} \underline{u}_j = {} & R\underline{u}_j + S_1\underline{u}_{j-1} + S_2\underline{u}_{j-2} \\ & + G\underline{e}_j + L_1\underline{e}_{j-1} + L_2\underline{e}_{j-2}. \end{aligned} \tag{2}$$

Here, just two previous repetitions are included for simplicity, but one can employ as many repetitions as desired. The underbar in each case indicates a vector containing the whole history of the variable during the j-th repetition. The R and G, if present, must be lower triangular matrices of gains due to causality, and are called current cycle feedback. The remaining matrices can be full. See (Phan *et al.*, 2000) for a discussion of the equivalence of such general forms to forms using only the previous repetition input and error. Most all ILC laws are a special case of this equation, although one can make gains that vary with repetitions as well. The integral control based learning is obtained by setting all matrices to zero except for S_1, which is the identity matrix, and L_1 which is a scalar learning gain times the identity matrix. It corresponds to the following algorithm: if the robot link was 2 degrees too low in the previous repetition, add the gain on the diagonal times 2 degrees to the command in the next repetition.

4. Two Experimental Testbeds

The experiments cited here were performed on two testbeds. One is the Robotics Research Corporation robot shown in Fig. 2. ILC is implemented in a decentralized manner, using one ILC for each link operating independently. The objective is to have each of the seven links of the robot perform simultaneously a cycloidal 90 degree turn, followed by a cycloidal 90 degree return. The maneuver time is chosen so that the base joints reach the maximum velocity allowed by the manufacturer. This maximizes nonlinear coupling effects such as centrifugal force and Coriolis force. The experimental RMS error of a good ILC law as a function of repetitions of the task is given in Fig. 3 (Elci *et al.*, 1994a; 1994b; 1994c; Lee-Glauser *et al.*, 1996). In only 12 repetitions of the task for learning, the tracking accuracy along the high speed trajectory was decreased by nearly a factor of 1000. To give some interpretation to this number, tests were conducted that simply asked the robot to perform the same maneuver repeatedly, and took statistics as to how much variation there was in the angle encoder counts of the actual trajectory executed by the robot. This factor of 1000 is necessarily a small factor of 2 or 3 above this repeatability level when the trajectory is repeated in succession on the same day. However, if one takes statistics running the trajectory one time each day, then this factor of 1000 is substantially below the repeatability level from day to day. Hence, to maintain this level, one would need to keep the learning process on. The learning process is correcting for changes in the system from one day to the next. It would take considerable time and effort to identify the source of these variations—changes in temperature, in humidity, etc.

The second experimental testbed is a timing belt drive system shown in Fig. 4. It consists of a motor rotating the input gear, which is connected to an idler shaft by a timing belt with teeth, and the other end of the idler shaft is connected to the output shaft by a second timing belt (Hsin *et al.*, 1997a; 1997b). This produces a gear reduction of a factor of 8 in a compact set of hardware. The objective is to get a constant velocity of the output shaft. The error sources are variations in velocity from inaccuracies in machining and mounting of the shafts and gears, as well as tooth meshing. These produce errors that are periodic with the period of one rotation of each shaft and each belt, including the fundamental and many harmonics. The velocity error spectrum with a large number of error frequencies is given in Fig. 5. This problem constitutes a repetitive control problem. The best performance obtained in our experiments is shown in Fig. 6. As discussed below, this error level is actually too good to be used in practice.

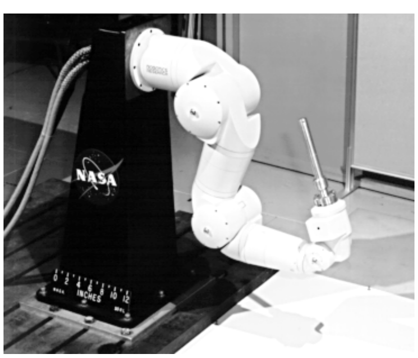

Fig. 2. Robotics Research Corporation robot.

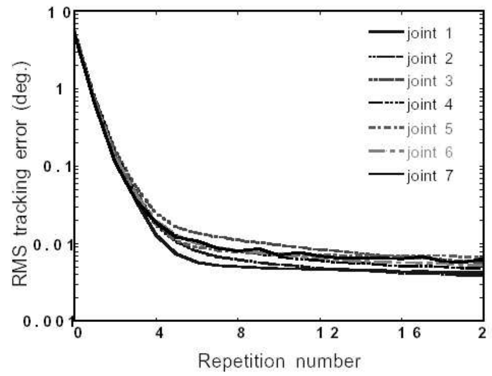

Fig. 3. Robot RMS error vs. repetitions for each joint using integral control based learning with a compensator.

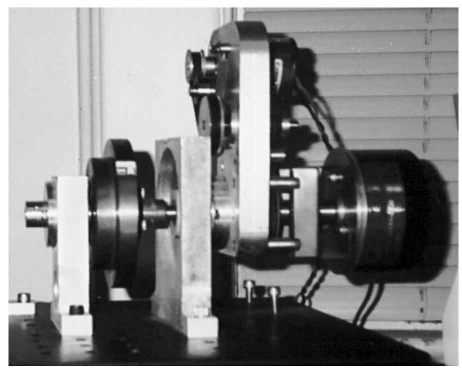

Fig. 4. Timing belt drive system.

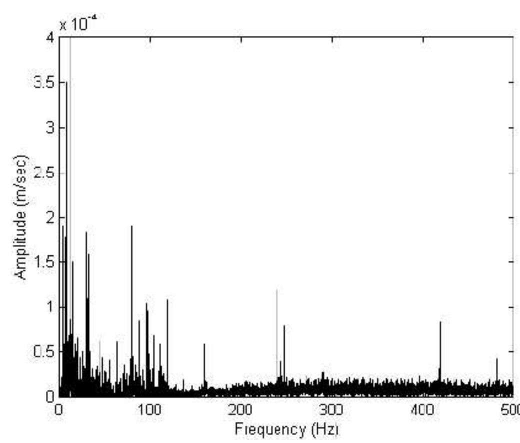

Fig. 5. Output velocity error spectrum using a well-designed feedback controller (amplitude scale is 10^{-4}).

5. Some Logical Objectives from the Mathematical Viewpoint

The following statements sound very logical when viewed from the point of view of mathematics:

1. My main research objective should be to prove that my ILC law converges to zero tracking error.

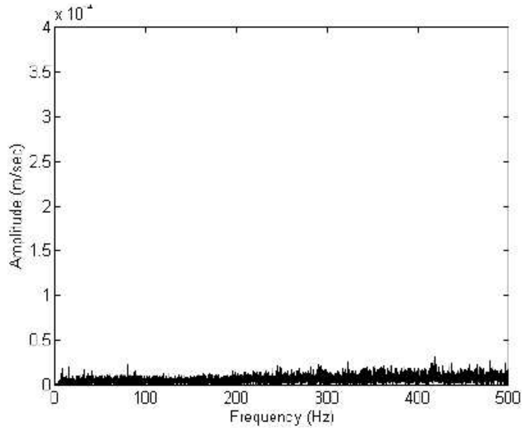

Fig. 6. Best output error spectrum using repetitive control.

2. For an RC law to be useful I must ensure that it is stable.
3. To make the most useful possible control law, I should create a law that is guaranteed to converge to zero tracking error for the largest possible class of systems. (If it is general enough, we can call it a universal learning controller).
4. Designing an ILC law that is a contraction mapping should produce a good learning control law.
5. There is no point in iterating in ILC, so that one should just model the system and invert the model to find the command needed to get zero tracking error.
6. I should aim to eliminate all tracking error.

In the sections that follow, it will be shown that in one way or another, each of these statements is wrong.

6. Amazing Success/Failure of the All-Purpose, Universal ILC

Following Statement 3 above, this section documents the efforts to obtain ILC that converges to zero tracking error for the largest possible class of systems.

Amazing success: Consider system (1) and the simplest ILC based on integral control applied in repetitions:

$$u_{j+1}(k) = u_j(k) + \phi e_j(k+1). \tag{3}$$

The solution of this difference equation is the sum of all previous errors times the learning gain ϕ, and hence it is a discrete equivalent of integral control applied in repetitions to each time step. The learning control looks one time step ahead to examine the error that is first affected by a change in control at the present time step. One can prove that this learning law is asymptotically stable, converging to the zero tracking error (Phan and Longman, 1988):

1. for all sufficiently small ϕ,
2. of the right sign,
3. provided $CB \neq 0$.

Note that this result is completely independent of the system matrix A in (1). In other words, it is completely independent of the system dynamics in discrete time!

You might wonder if Item 1 above is a catch, but as the sample time tends to zero, the upper limit on the gain tends to infinity. So the range of gains producing convergence to zero error can be made arbitrarily large by picking a sufficiently fast sample time. For normal feedback control systems, a reasonable value of the control gain would seem to be around unity or somewhat less. This corresponds to the statement: if the robot link is 2 degrees too low during a certain time step the last repetition, add 2 degrees or something a bit less, to the command in the next repetition. In the case of the robot, the stability boundary was at a gain of over 90, corresponding to trying to correct the 2 degree error with a correcting command of 180 degrees. Perhaps we should be impressed that with such an extreme action, the ILC still converges to zero error.

Item 3 is not usually a constraint either. If system (1) comes from a continuous time differential equation fed by a zero order hold, then provided a reasonable sample frequency is used, the first Markov parameter will not be zero (i.e. it is difficult to have the response of the system to a unit step input be zero at the first time step, starting from zero initial conditions). When (1) models a typical digital control system, similar arguments can be made.

Item 2 asks that we know the sign of CB. Normally one does know this. But since we wanted to have the ILC law work for the largest possible class of systems, efforts were expended to eliminate this condition. This can be accomplished by generalizing the learning control law (3) to the alternating sign ILC (Chang *et al.*, 1992; Longman *et al.*, 1992), which uses $+\phi, -\phi, +\phi, -\phi, \dots$ as the learning gain for successive time steps in the first repetition, then changes all the signs in the next repetition, and changes back the third, etc.

Also, one might argue that the system model in (1) is somewhat restrictive—it is linear. The reference (Longman *et al.*, 1992) applies (3) to a system of the form

$$\dot{x} = f(x, u) \tag{4}$$

through a zero order hold, and aims for zero tracking error at the sample times. It is shown that a very similar general stability condition applies. The main requirement is that (4) must satisfy a Lipschitz condition (at least piecewise). This is a very large class of nonlinear systems. One way of looking at the nonlinear problem is that as far as convergence is concerned, only the nonlinearity observed over one time step influences the result. And for reasonable time steps, there will be very little of such nonlinearity.

We conclude that the very simple ILC law (3) or its alternating sign variant can produce zero tracking error in the vast majority of single-input, single-output systems in the world. This appears to be a very powerful result—much more powerful than one would expect for such a simple law. It is almost a universal learning controller. Just connect the wires, turn it on, and it will converge to the zero error—so, simple, and guaranteed to work for all but rather pathological systems!

Simulations 1: When producing publications discussing control laws, one naturally makes some simulations to show how the law works. We made the usual simulations of simple systems, e.g. second order systems such as for the rotation of a robot link with a PD feedback controller (Chang *et al.*, 1992). For simplicity, we picked trajectories of 50 or 100 time steps. With a learning gain in the range of 2/3, reasonably fast convergence was demonstrated. There was some concern about the transients during learning being somewhat large. The alternating sign law helped keep the transients well behaved.

Experiments and failure: ILC law (3) was applied to the robot in Fig. 2. The RMS tracking error decreased by a factor of 35 dB in about 9 repetitions, or a factor of 50. That sounds like a success. A factor of 50 improvements in the tracking error just by running the robot 9 times could be very significant. But theory says that we are supposed to be able to converge to zero tracking error. When the repetitions were repeated past 9 going up to 15, the RMS error got worse. By repetition 15, the robot was making so much noise, we were afraid to continue the experiment for fear of damaging the hardware. This was frustrating—if only I could run the experiment 5 more repetitions, maybe the error would come down, and I could continue the path toward zero tracking error.

Simulations 2: Since we could not continue in hardware, we turned to a simulation, and used a linear third order model of the input-output relationship of each robot link. The model fits the robot performance rather well, and consists of a break frequency at 1.4 Hz, defining the bandwidth of the feedback controllers, and a vibration mode at an undamped natural frequency of 5.9 Hz with a damping ratio of 0.5. The simulation resulted in exponential overflow (Longman and Songchon, 1999)!

But the theory says it converges. The theory is right, the exponential overflow is just a bad learning transient (see (Huang and Longman, 1996) for explanations of this phenomenon from several points of view). To make the transient small enough so that the computer could handle it, we decreased the length of the desired trajectory to 1 second instead of 6 seconds. We changed the sample rate from the 400 Hz of the robot feedback controller, to 100 Hz, and we simplified the trajectory slightly. And success was obtained—the computer could simulate the learning process, cf. Fig. 7 (Longman and Songchon, 1999). The initial RMS error decreased as shown in the inset, from 0.4330 to 0.1402 radians in repetition 7. Then the error increased to a maximum of 1.1991×10^{51} at repetition 62,132. Of course, we know that the error converges to zero, and the simulation agrees, giving a numerical zero of 1.3145×10^{-48} at repetition 3×10^5.

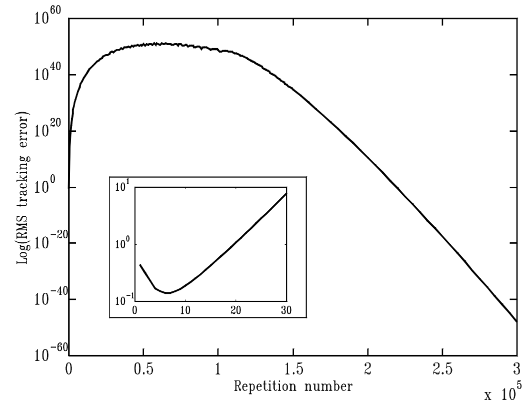

Fig. 7. Simulation of the learning transients of integral control based learning applied to a robot link.

This makes the universal learning controller impractical in a couple of respects. First, it is rare in physical problems that one can run 3×10^5 repetitions—the robot is worn out long before. Second, it is a rare robot that can allow the links to rotate by 1.1991×10^{51} radians without hitting limit switches (or breaking the electrical wires). There might also be a problem that the actuators are not strong enough to rotate to this angle within one second.

We conclude that convergence is not really the point in ILC, and not even much of an accomplishment. What is needed is guaranteed good learning transients. And that *is* an accomplishment. Furthermore, it is dreaming to aim for a universal ILC. If universal controllers made sense, we, control engineers, would have been out of business long ago. Although Statements 1 and 3 of Section 5 seemed very reasonable research objectives from the mathematical point of view, this experiment showed that they were not addressing the real issues.

7. Who Needs Stability?

The interplay between the mathematical thinking and the experiments during our ILC research produced a number of surprises related to stability:

1. The guaranteed stable "universal" controller of the previous section that proved to be useless in practice.

2. Or is it useless? This ILC with "unstable" behavior (going to exponential overflow on the way to convergence) can actually be a very practical ILC.

3. And then there is a case of ILC designs that are supposed to be unstable according to the theory, but are stable in the real world.

“Unstable” but practical: The previous section discussed Statements 1 and 2 of Section 5. Concerning Statements 1 and 2 we can make a further comment. We are all taught that infinite sums are useless if they diverge. Then later we may learn about asymptotic expansions, where a finite number of terms in a divergent sum actually gives a very good approximation of the expanded function, and can in fact make the divergent sum very useful. When the ILC law (3) was applied to the real world robot, the error decreased by a factor of 50 in 9 repetitions. That is a rather impressive improvement. So just stop the learning at that point, and this makes a very useful and simple way to improve the tracking accuracy of a feedback control system (Longman and Huang, 1998). Although technically the ILC is stable but with bad transients, the corresponding RC is actually unstable (Longman, 2000; Huang and Longman, 1996), and of course the same possibility exists for improving the performance of repetitive controllers. Hence, we conclude that an unstable repetitive control law can actually be practical.

When an experimentalist “tests” stability: The experimentalist determines what he calls stability in a different manner than the mathematician. He would certainly call the exponential overflow situation unstable, after seeing the error increase sufficiently so that he had to stop the learning process, and never observing the ultimate convergence. The reverse can also easily happen. Figure 8 shows

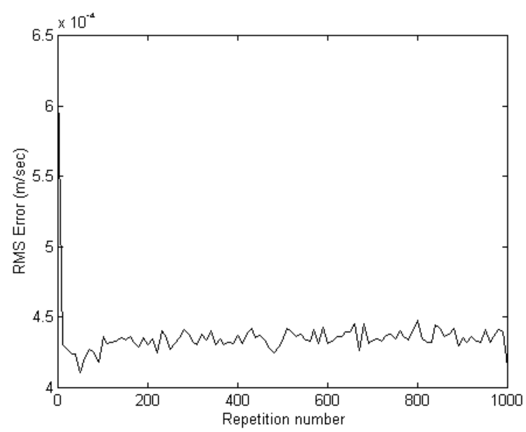

Fig. 8. Timing belt drive system RMS errors for 1,000 repetitions, linear phase lead repetitive control with a 12-th order Butterworth filter.

the RMS error versus repetitions using an RC with a linear phase lead and a 12-th order causal Butterworth low pass filter (Hsin *et al.*, 1998). The initial error is near the top of the graph, although it is not discernable in the plot. The error decreases in a few repetitions and then bounces around at the noise level. Any normal experimentalist would be quite satisfied with 1000 repetitions of good performance, and pronounce the system stable. In this particular case, the experimenter decided to run it longer, and found that around 2,650 repetitions an instability started to appear. So he decided to try to fix the problem, and applied an 18-th order Butterworth to get a better cutoff. Then he ran it for 10,000 repetitions and everything behaved well. Then he was satisfied.

Unstable in math, but stable in practice: The theoretician complains, I don’t care if you ran it for 10,000 repetitions, I know it is unstable. Surely if you ran it still longer you would see that it is unstable. Nevertheless, the experiment started us thinking. Maybe the size of the word used in the digital to analog converter and the analog to digital converter was able to produce stability. Computations showed that this was indeed the case, that the 18-th order Butterworth was a sufficiently ideal cutoff that the error remaining above the cutoff frequency was smaller than the last digit in the converter and therefore could not accumulate (about 8 digits for the hardware). This observation then led to a new type of ILC and a new set of theory—for ILC stabilization by quantization. Figure 9 shows an ex-

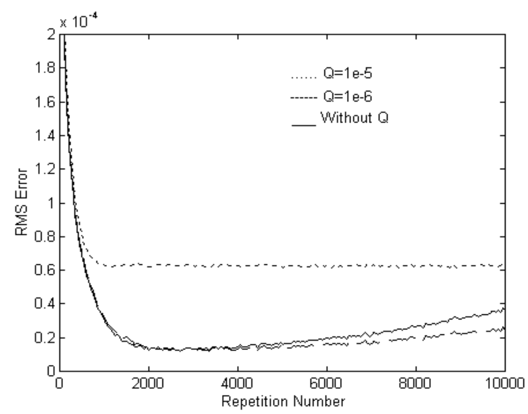

Fig. 9. Stabilization by finite word length in integral control based learning.

ample (Hsin *et al.*, 1998). Without quantization the RMS error in this example goes through a minimum and then starts up again. With a quantization level of 10^{-6} it still goes through a minimum and then starts to increase again, but more slowly. Then with a level of 10^{-5} this particular ILC is stabilized. So, the experiments spawned a new theory, and a new class of ILC.

8. Did You Say You Wanted Zero Tracking Error?

From the mathematical point of view it is natural to ask for zero tracking error in the limit as the repetitions progress, i.e. to ask for asymptotic stability. We have already seen some reasons not to ask for this, and there are more.

The asymptotic expansion property: The previous section produced an improvement in the RMS error of a factor of 50 in 9 repetitions. Maybe this significant improvement is quite sufficient, and there is no need to work on further learning.

The finite word length: Also above, we saw that stabilization by quantization can be an effective design method for ILC or RC. This obtains stability at the sacrifice of no longer achieving zero tracking error.

Be kind to the hardware: We commented about Fig. 6 that this performance was too good. Although the output shaft was giving a nearly perfect velocity performance, the motor was working very hard. Again, the hardware was making a lot of noise. We concluded that trying to control the substantial peak at 240 Hz in Fig. 5 was not advisable, otherwise we would be wearing out the equipment. This peak is associated with the dynamics of tooth meshing on the faster of the two timing belts. It is very far above the bandwidth of the controller, and any corrective action at this frequency requires an amplitude 11 times as big as the error being corrected. This situation is rather typical of hardware. In most situations one would want a frequency cutoff of the learning for similar reasons.

Obtaining good learning transients: Perhaps the most useful method of ensuring good learning transients, avoiding the problem of Fig. 7, is to ask for a monotonic decay of the steady state frequency components of the error (Longman, 2000; Elci *et al.*, 1994a). Violating this condition is very hard to avoid at high frequencies, and hence supplying a cutoff for frequencies that cause such trouble makes it much easier to design a learning controller. The cutoff must be done without disturbing the phase, using a zero phase filter as in (Elci *et al.*, 1994a; 1994b; Longman, 2000; Plotnik and Longman, 1999). Hence, we can obtain good learning transients at the expense of not asking the zero error at high frequencies.

Robustness to the problem of parasitic poles, singular perturbations: Unlike many other fields of control, it is not stability robustness that is important (the universal controller has more such robustness than one can normally hope for). It is the robustness of the good learning transients that is important in ILC. This robustness requires that the phase lag of the system be known reasonably accurately. In this kind of knowledge an error of 90 degrees will normally violate the frequency based monotonic decay condition for good transients. Most systems when pushed hard enough exhibit additional dynamics, sometimes called parasitic poles. What earlier behaved like a rigid body, when pushed for zero tracking error is seen to exhibit some flexibility. An amplifier that might have been considered simply a gain, exhibits some low pass characteristics, etc. All it takes is one additional pole to produce an additional 90 degree phase lag. Hence, the good learning transients condition is not robust to such singular perturbations. The simplest way to address the issue is to cut off the learning for high frequencies where one loses confidence in the phase model. Or one can let the experiments show you what cutoff you need, by observing the frequency content of the error if it happens to start to grow, and then pick a cutoff below this value. One can even make this robustification into a self-tuning process (Wirkander and Longmann, 1999).

Thus, there are many reasons not to ask for zero tracking error, and Statement 5 of Section 5 was misguided.

9. What are Good Models, Anyway?

Essentially, all of engineering depends on models to get the desired results. ILC pushes the limits of one's ability to model. In normal feedback control it would be just fine to leave out some dynamics at high frequencies (far above the bandwidth of the controller) of a very low amplitude, and not even identifiable in normal input-output data. The gain and phase margins that determine stability are determined at much lower frequencies, so the small amplitude high frequency behavior is not an issue in stability. In ILC, such dynamics could easily destabilize the controller. It might take a long time, maybe 2,650 repetitions. In that following, some comments are given on modeling issues in ILC:

- In the early days of ILC, a large percentage of the literature worked very hard to make algorithms that were guaranteed to converge for robots modeled as multiple rigid bodies, i.e. modeled as the nonlinear dynamics of multibody systems. These works made substantial mathematical achievements. But, Figs. 3 and 11 show that using uncoupled ILC controllers, independent ones for each joint, and using simple linear theory, produces errors very close to the reproducibility level of the robot. So all those nonlinear dynamics equations will not be of any help in getting better performance. In this case all the theoretical effort has no practical payoff. The complex and more accurate model does not help.

- It is very natural to think that there is no real need for ILC. Instead, we should simply make a model, and invert it to get the input needed to produce the desired output. For the robot, we went through this exercise using system identification methods. The only dynamics that were visible were one pole at 1.4 Hz for each robot link, i.e. the break frequency of the feedback controllers. So we invert this model, and apply the result to the robot. This reduces the RMS tracking error by roughly a factor of 10 (Elci *et al.*, 1994b). Using a good ILC algorithm continues to reduce the error by about another factor of 100. So what does all the work of making a model and inverting do for you? It eliminates the need for the first 1 or 2 repetitions of the ILC. It is not clear that there is any point in going through the effort. It is much easier and faster to simply perform one or two more repetitions.

- But there is more to say about the suggestion of making a model and inverting it. When controlling a system governed by a differential equation fed by a zero order hold, as in nearly all digital control systems, and when the differential equation of the system has three more poles than zeros, then (Åström *et al.*, 1980) shows that when the sampling is fast enough, the discrete equivalent system model will have one or more zeros outside the unit circle. This makes the inversion a solution of an unstable difference equation. The result is that in practice for most physical systems it is in fact impossible to invert the system to find the control (Statement 5 of Section 5).

- One way of looking at the modeling issue is to say that no finite order system model is good enough for ILC. As has been mentioned above, when using typical robot data, for the robot doing substantial movements, we were only able to identify the break frequency at 1.4 Hz, making a first order model for each link. Applying a few steps of learning control, however, cuts out much of the low frequency error quickly. Suddenly, the data show clearly the first resonance. In fact, if we keep the learning control on for a while and then take data, the only thing that we can identify is the resonance, and we can no longer see the break frequency. All error in that frequency range is gone. Note that there is a very natural connection between ILC and optimal experiment design. So now we use data at some intermediate repetition in order to make a third order model, and use it to design a phase cancellation controller (Elci *et al.*, 1994c), and then apply the ILC. The resulting RMS error (Fig. 10) decreased by about two and a half orders of magnitude, and then started to diverge. Looking at the data shows that the divergence was related to the second vibration mode of the robot. Using this data to make a new 5-th order model in place of the old 3-rd order model, allowed the error to decay to nearly 3 orders of magnitude (Fig. 11). The question is: Does this process simply keep continuing? Does it ever stop? So far the process has found only two of the seven vibration modes that should exist in the seven-degree-of-freedom robot with flexible harmonic drives between each link. Or perhaps each link has some structural flexibility, and can exhibit vibration modes. The learning process could keep finding more and more parasitic poles or residual modes, more and more dynamics missing from our current model. The implication is that no finite order model is good enough for ILC. Of course, things like the finite word length will actually put a limit on how far the process can proceed. One reaction to this thinking is to create ILC laws that do not rely on finite order models. The author suggests the use of experimental frequency response plots (Longman, 2000; Longman and Songchon, 1999). The objective is to take care to make experimental Bode plots of the system that are accurate to the Nyquist frequency. And if they prove to be inaccurate above some frequency as demonstrated by the growth of the error, then look at the frequency content of the error and create a frequency cut-off below that frequency (or try to improve the frequency response information in that frequency range).

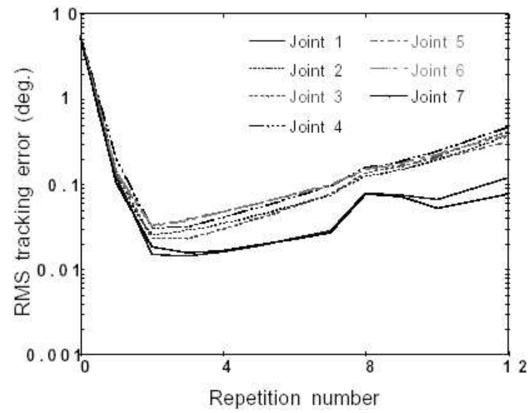

Fig. 10. Phase cancellation RMS error for all joints without model updating.

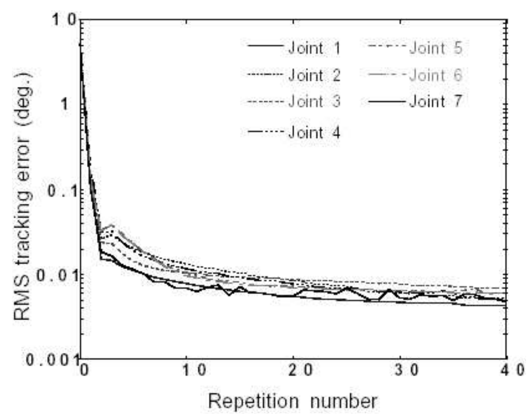

Fig. 11. Phase cancellation RMS error for all joints with model updating.

10. Can Simulations Predict the Performance?

Simulations are supposed to be a substitute for much more time consuming and expensive experiments. Could we have made the same conclusions about what is important in ILC simply by use of simulations, without having performed the experiments cited?

- Simulations 1 of Section 6 were typical textbook size problems, and they did not uncover the magnitude of the difficulty. So the simulations must be more sophisticated.
- Would the simulations have shown us the difficulties of long term instabilities, requiring 2,650 repetitions before we start to see the instability? No, if we do not think to look for it. In this respect simulations are like experiments.
- Could we have ever learned about stabilization due to the finite word length in the D/A converter? Not very likely. We would have to have been smart enough to decide that it is important to include finite word size effects in the simulation. This is like having to know the answer before you can ask the question.
- There is a “catch 22” in making a simulation. As has been discussed above, when we make a model and invert it to find the input to produce zero error according to the model, and then turn on a learning controller, the error decreases further by two orders of magnitude. ILC learns to eliminate errors that are beyond the errors predicted by our model. Hence, the simulation needs to know a model that contains physical phenomena that we have not modeled. In the case of the robot, we know that there will be another mode somewhere, even if we are unable to identify it with any normal identification routine. We might be able to make a realistic simulation for the robot including more vibration modes, but in a less structured situation, there is no way of including small dynamic effects that we have not even thought about. So, simulation is not going to give us the right answer.
- One of the important issues in ILC is what final error level candidate laws will produce. One would like to find the ILC law that gives the smallest possible final error level. The contraction mapping algorithm of (Jang and Longman, 1994) seemed very promising on paper. Experiments gave the results shown in Fig. 12 (Lee-Glauser *et al.*, 1996). The final error levels are worse than those of other learning algorithms in Figs. 11 and 3. Hence, Statement 4 of Section 5 was shown to be wrong in some sense (use of a design based on a partial isometry (Jang and Longman, 1996) or on phase cancellation (Elci *et al.*, 1994c) can produce contraction mappings without this problem). Could we have learned this by simulation? It would be hard. It would be best to have a real understanding of what the effects are that produce variations in the system response, i.e. to have modeled the irreproduciblity of the system. Typical mathematical thinking is to substitute white noise. We might learn something from this, but I suspect it would not give us the answer. It seems likely that it is the finite word length effect again that produces the poor final error level, because of the slow learning rate at high frequencies. Again, getting the right answer by simulation requires too much understanding of the problem before you have seen the phenomenon.

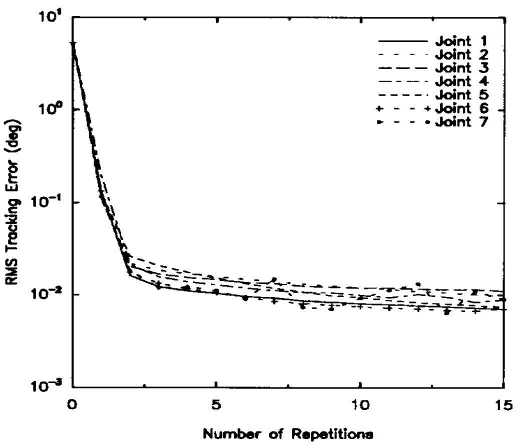

Fig. 12. RMS error using contraction mapping ILC after 2 repetitions of inverse model learning.

But there are other issues. Examine the error history in Fig. 7 – is there anything wrong in the latter part of the curve? It says that the error reaches 10^{-48}. This is a prediction from programming a mathematically correct equation that gives the error in the next repetition based on the error in the previous repetition. If instead one uses more basic formulas, computing the control action each repetition, computing the result of applying it to the system, and then computing the error, the final error level will appear in a range like 10^{-11} to 10^{-14} when programmed in MATLAB with its double precision computations. These results do not agree, although mathematically the equations programmed are totally equivalent. And certainly both are wrong. The final error level will be very highly influenced by the number of digits carried in the digital-to-analog and analog-to-digital converters, and this might be 8 digits. But even if this is included in the computations, it appears that there is no way to know enough about modeling errors and noise to be able to predict the final error levels. And hence, there is no way to know in advance by simulation what learning control law gives the best error levels. This situation makes ILC somewhat unusual. In most fields of engineering simulations can give reasonably good predictions of performance, at least if one works hard enough.

- And then it seems unlikely that we would realize in simulations that we need to stop the learning in order to be kind to the hardware.

The conclusion is that extreme care and a lot of insight would have been required to get the simulations to predict the outputs correctly. And to have this amount of insight, most likely one needs to run experiments.

11. Conclusions

This paper has presented a case history of the interplay between mathematical theory, modeling and simulation, and experiment in the development of a field. This particular case history is unusual in the sense that it pushes some of these to their limit or beyond. ILC tries to learn everything needed for zero tracking error, going beyond any model one can develop. Both theory and simulation need models. Simulations are unusually inefficient, because they rely on models that are being pushed beyond their range of validity. In addition, simulations suffer routinely from ill conditioning and experiments can be pushed beyond hardware limits.

It was shown that limiting oneself to purely mathematical thinking, even with the aid of simulation, can very easily suggest research directions that appear to be important, but in the light of experiments are seen to be rather irrelevant. Section 5 listed several research objectives that the author and co-workers considered at one time or another to be very appropriate, perhaps central to the field. Experiments and simulations showed them all to be wrong in one way or another, or showed them to simply be misguided. Experiments were needed to learn these lessons, and to refocus the effort in a direction that addressed the real issues of the field. Simulations were seen to be rather ineffective for most purposes, requiring that one know too much about the problem before being able to get a result that predicts the experiments.

Experiments told us what types of models to use to avoid as much as possible singular perturbations issues. They told us not to expend substantial efforts on specialized nonlinear models for robots. The linear based thinking produces experimental results approaching the reproducibility level of the robot, so no more complex modeling can do significantly better. Experiments suggested new forms of ILC stabilized by quantization. But we also learned not to trust what seemed like stability in experiments—instability could lie dormant for 2650 repetitions. It was only the fact that the theory said it was unstable, that made us run that many repetitions. For ILC, the interplay between experiments, simulations, and theory is unusually lively, full of surprises, and full of apparent paradoxes.

References

Amann N., Owens D.H. and Rogers E. (1998): *Predictive optimal iterative learning control*. — Int. J. Contr., Vol. 69, No. 2, pp. 203–226.

Arimoto S., Kawamura S. and Miyazaki F. (1984): *Bettering Operation of Robots by Learning*. — J. Robot. Syst., Vol. 1, No. 2, pp. 123–140.

Åström K., Hagander P. and Strenby J. (1980): *Zeros of sampled systems*. — Proc. 19th IEEE Conf. *Decision and Control*, pp. 1077–1081.

Bien Z. and Xu J.-X. (1998): *Iterative Learning Control: Analysis, Design, Integration, and Applications*. — Boston: Kluwer.

Casalino G. and Bartolini G. (1984): *A learning procedure for the control of movements of robotic manipulators*. — Proc. IASTED Symp. *Robotics and Automation*, Amsterdam, pp. 108–111.

Chang C.K., Longman R.W. and Phan M. (1992): *Techniques for improving transients in learning control systems*. — Adv. Astronaut. Sci., Vol. 76, pp. 2035–2052.

Craig J.J. (1984): *Adaptive control of manipulators through repeated trials*. — Proc. *American Control Conference*, San Diego, pp. 1566–1573.

Elci H., Longman R.W., Phan M., Juang J.-N. and Ugoletti R. (1994a): *Discrete frequency based learning control for precision motion control*. — Proc. IEEE Int. Conf. *Systems, Man, and Cybernetics*, San Antonio, TX, Vol. 3, pp. 2767–2773.

Elci H., Phan M., Longman R.W., Juang J.-N. and Ugoletti R. (1994b): *Experiments in the use of learning control for maximum precision robot trajectory tracking*. — Proc. Conf. *Information Science and Systems*, Princeton, NJ, pp. 951–958.

Elci H., Longman R.W., Phan M., Juang J.-N. and Ugoletti R. (1994c): *Automated learning control through model updating for precision motion control*. — Adapt. Struct. Comp. Mat.: Anal. Appl., ASME, AD-Vol. 45/MD-Vol. 54, pp. 299–314.

Hara S. and Yamamoto Y. (1985): *Synthesis of repetitive control systems and its applications*. — Proc. 24th IEEE Conf. *Decision and Control*, Fort Lauderdale, Florida, pp. 326–327.

Hara S., Omata T. and Nakano M. (1985): *Stability of repetitive control systems*. — Proc. 24th IEEE Conf. *Decision and Control*, Fort Lauderdale, Florida, pp. 1387-1392.

Hsin Y.P., Longman R.W., Solcz E.J. and de Jong J. (1997a): *Experiments bridging learning and repetitive control*. — Adv. Astronaut. Sci., Vol. 95, pp. 671–690.

Hsin Y.P., Longman R.W., Solcz E.J. and de Jong J. (1997b): *Experimental comparisons of four repetitive control algorithms*. — Proc. 31st Ann. Conf. *Information Sciences and Systems*, Baltimore, Maryland, pp. 854–860.

Hsin Y.P., Longman R.W., Solcz E.J. and de Jong J. (1998): *Stabilization due to finite word length in repetitive and learning control*. — Adv. Astronaut. Sci., Vol. 97, pp. 817–836.

Huang Y.-C. and Longman R.W. (1996): *The source of the often observed property of initial convergence followed by divergence in learning and repetitive control*. — Adv. Astronaut. Sci., Vol. 90, pp. 555–572. A modified version appears in Intell. Automat. Soft Comput., Special Issue on Learning and Repetitive Control, Vol. 8, No. 2, pp. 107–128.

Inoue T., Nakano M. and Iwai S. (1981): *High accuracy control of a proton synchrotron magnet power supply*. — Proc. 8th IFAC World Congress, Kyoto, Japan, Vol. 20, pp. 216–221.

Jang H.S. and Longman R.W. (1994): *A new learning control law with monotonic decay of the tracking error norm*. — Proc. 32nd Ann. Allerton Conf. *Communication, Control, and Computing*, Monticello, Illinois, pp. 314–323.

Jang H.S. and Longman R.W. (1996): *Design of digital learning controllers using a partial isometry*. — Adv. Astronaut. Sci., Vol. 93, pp. 137–152.

Lee-Glauser G., Juang J.-N. and Longman R.W. (1996): *Comparison and combination of learning controllers: Computational enhancement and experiments*. — J. Guid., Contr. Dynam., Vol. 19, No. 5, pp. 1116–1123.

Longman R.W. (1998): *Designing iterative learning and repetitive controllers*, In: Iterative Learning Control: Analysis, Design, Integration, and Applications, Chapter 7 (Z. Bien and J.-X. Xu, Eds.). — Kluwer Academic Publishing, Norwell, MA, pp. 107–146.

Longman R.W. (2000): *Iterative learning control and repetitive control for engineering practice*. — Int. J. Control, Special Issue on Iterative Learning Control, Vol. 73, No. 10, pp. 930–954.

Longman R.W. and Huang Y.-C. (1998): *Use of unstable repetitive control for improved tracking accuracy*. — J. Chinese Soc. Mech. Eng., Special Issue on Dynamics and Control, Vol. 19, No. 1, pp. 81–93.

Longman R.W. and Songchon T. (1999): *Trade-offs in designing learning/repetitive controllers using zero-phase filtering for long term stability*. — Adv. Astronaut. Sci., Vol. 102, pp. 243–263.

Longman R.W., Chang C.-K. and Phan M. (1992): *Discrete time learning control in nonlinear systems*. — A Collection of Technical Papers, *1992 AIAA/AAS Astrodynam. Spec. Conf.*, Hilton Head, South Carolina, pp. 501–511.

Middleton R.H., Goodwin G.C. and Longman R.W. (1985): *A method for improving the dynamic accuracy of a robot performing a repetitive task*. — Int. J. Robot. Res., Vol. 8, No. 5, pp. 67–74.

Moore K. (1993): *Iterative Learning Control for Deterministic Systems*. — London: Springer.

Moore K. and Xu J.-X. (Eds.) (2000): *Special Issue on Iterative Learning Control*. — Int. J. Contr., Vol. 73, No. 10.

Nakano M. and Hara S. (1986): *Microprocessor-based repetitive control*, In: *Microprocessor-Based Control Systems*. — Amsterdam: D. Reidel Publ. Comp., pp. 279–296.

Omata T., Nakano M. and Inoue T. (1984): *Applications of repetitive control method to multivariable systems*. — Trans. SICE, Vol. 20, No. 9, pp. 795–800.

Owens D.H. (1977): *Stability of multipass processes*. — Proc. IEE, Vol. 124, No. 11, pp. 1079–1082.

Owens D.H., Amann N., Rogers E. and French M. (2000): *Analysis of linear iterative learning control schemes—A 2D systems/repetitive processes approach*. — Multidim. Syst. Signal Process., Vol. 11, No. 1–2, pp. 125–177.

Phan M.Q. and Longman R.W. (1988): *A mathematical theory of learning control for linear discrete multivariable systems*. — Proc. *AIAA/AAS Astrodynam. Spec. Conf.*, Minneapolis, Minnesota, pp. 740–746.

Phan M.Q., Longman R.W. and Moore K.L. (2000): *A unified formulation of linear iterative learning control*. — Adv. Astronaut. Sci., Vol. 105, pp. 93-111.

Plotnik A.M. and Longman R.W. (1999): *Subtleties in the use of zero-phase low-pass filtering and cliff filtering in learning control*. — Adv. Astronaut. Sci., Vol. 103, pp. 673–692.

Rogers E. and Owens D.H. (1992): *Stability Analysis for Linear Repetitive Processes*. — Berlin: Springer.

Tomizuka M., Tsao T.-C. and Chew K.-K. (1989): *Analysis and synthesis of discrete time repetitive controllers*. — J. Dynam. Syst. Meas. Contr., Vol. 111, No. 3, pp. 353–358.

Uchiyama M. (1978): *Formulation of High-Speed Motion Pattern of a Mechanical Arm by Trial*. — Trans. SocInstrum. Contr. Eng., Vol. 14, pp. 706–712 (in Japanese).

Wirkander S.-L. and Longman R.W. (1999): *Limit cycles for improved performance in self-tuning learning control*. — Adv. Astronaut. Sci., Vol. 102, pp. 763–781.