Robust Learning Control Method in Saturating Actuators

Yeong Soon Ryu*
(Received July 24, 1998)

Anti-reset windup is an important tool in the practical implementation of integral controllers. Integral control can have poor performance due to windup of the integral during transient periods when the actuator is operating at its saturation limit. Anti-reset windup speeds up the departure from the saturation limit by stopping the build-up of the integral during saturation. The most basic form of learning and repetitive control makes use of integral control concepts applied in the repetition domain. Therefore, this paper studies the use of anti-reset windup concepts in learning control. The integral is operating in repetitions, but the system dynamics are in time, making the application nonstandard. Various forms of anti-reset windup are developed for use in learning and repetitive control, and shown to improve performance of the learning process, especially when one does not know enough about the system to obtain well-behaved learning transients. Anti-reset windup is also shown to be helpful in situations where the desired trajectory is not feasible, and in situations where the initial conditions are systematically in error, such as in a robot subject to gravity disturbance.

Key Words : Anti-reset Windup, Saturating Actuators, Saturation Limits, Integral Control, Learning Control, Robustness

1. Introduction

Proportional control systems exhibit a constant steady-state error in response to a constant command, or to a constant disturbance. In order to eliminate this error, integral control is often used. It will not tolerate a constant error since it would result in a linearly increasing corrective control action. In practical implementation, integral control can exhibit poor performance when actuator saturation limits are encountered. The integral keeps asking for larger control action, but the hardware stops increasing its output when the saturation limit is reached. During periods of saturation, the integral of the error can keep increasing without having any corresponding influence on the control applied. Before the control can leave the saturation regime, the integral must see errors of the opposite sign for an amount of time needed to negate the accumulated integral. This phenomenon is called windup, and various methods have been introduced to eliminate the sluggish response that results. These methods are called anti-reset windup (Astrom, 1984, Wittenmark, 1989). In heuristic terms the interaction of saturation limits with the integral control action can cause the controller to get hung up at the saturation limit, and produce poor performance.

There are a number of approaches to producing learning and repetitive controllers that learn from previous experience executing a command, in order to converge on zero tracking error as the repetitions of the command progress. The simplest form of these algorithms is based on integral control concepts applied in the repetition domain (Phan, and Longman, 1988). When a feedback controller executes a tracking command it usually produces tracking errors, since it is only rarely that the particular solution of a differential equation is equal to the command that determines the forcing function. When the same command is given repeatedly to the feedback system, it then produces the same errors at the same time steps,

* Andong National University, Andong, Kyungbuk, Korea

except for random disturbance effects. This appears as a constant error at that time step when viewed in repetitions. Thus, this form of learning control introduces the equivalent of an integral at each time step, applied in the repetition domain in order to get rid of tracking errors. This type of learning and repetitive control is very simple to implement, and requires very little on-line computation.

Since this basic form of learning control relies on integral control concepts, it is natural to ask whether in practice the use on anti-reset windup concepts in learning control has advantages. It is the purpose of this work to investigate this question.

Integral control-based learning control can have poorly behaved transients during the learning process, particularly when one does not know enough about the system to set the gain properly. Some ad hoc methods of improving the transients are given in (Chang, Longman, and Phan, 1992). It is also possible that no learning gain exists that gives good transient behavior (Elci, Longman, Phan, Juang, and Ugoletti, 1994). And it may be possible to get improved tracking accuracy from learning control without having a stable learning process, by learning for a limited period of time, as described in (Longman, and Huang, 1994).

Here we study the possible benefits of using anti-reset windup (ARW) ideas for the following purposes in learning and repetitive control:

- Limiting poor transient behavior during learning, particularly when the gain is poorly set.
- Learning control assumes that the initial condition is on the desired trajectory. In some cases, such as robots under gravity, one does not know how to obtain the desired initial condition, and may start with whatever initial condition the robot supplies given the desired position as command to the robot. ARW may be helpful under these conditions.
- When the desired trajectory is executable without saturation, but the transients induce saturation, ARW may speed the recovery following a time interval of saturation.
- Sometimes the desired trajectory is not physically executable, such as a unit step command, because it requires control actions that go beyond the saturation limit. We study whether ARW may have advantages in such situations.

2. Anti-Reset Windup in Classical Control

The basic idea behind ARW is to avoid updating the integral in the feedback controller when the actuator is saturated. Different approaches alter the exact conditions for suspending and resuming integration, and how the integral part is treated when it is suspended. One method (Astrom, 1984, Wittenmark, 1989) is shown in Fig. 1 for a proportional plus integral controller as written in the z-transform domain. The governing equations are

$$\begin{aligned} &u_s(kT) = sat(u(kT)); \\ &u(kT) = u_i(kT) + u_p(kT) \\ &u_p(kT) = K_p e(kT); \\ &e(kT) = y^*(kT) - y(kT) \\ &u_i(kT+T) = u_i(kT) + K_i T e(kT) \\ &\qquad + (u_s(kT) - u(kT))\, T/\alpha \quad (1) \end{aligned}$$

where K_p denotes the proportional gain, K_i the integral gain, and α a "time constant" for the anti-reset windup. When the parameter α is chosen equal to the sampling time T, and the actuator is not saturated so that $u_s(kT) = u(kT)$, then the last term in the last of these equations becomes zero. Then the equation becomes a recursive computation of the sum of all previous errors, multiplied by a gain, representing the discrete form of an integral. If instead the actuator is saturated, then

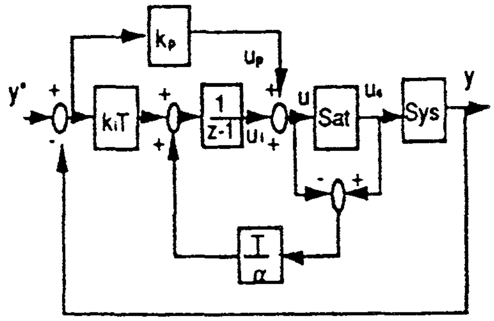

Fig. 1 Block diagram for sampled data PI controller with anti-reset windup.

$$u_i(kT+T)=[u_s(kT)-u_p(kT)] + K_i Te(kT) \quad (2)$$

The first term on the right gives the saturation limit minus the contribution from the proportional control, and the last term on the right can be thought of as negligible since it is multiplied by the sample time which is usually small. Thus the integral action, u_i, is approximately bounded by the value that would produce saturation, and is not allowed to grow beyond that value. The formula shows that at the first time step for which the error changes sign, the integral can leave the saturation limit. This is in contrast to pure integral control where the integral has been allowed to grow during saturation, and one may need many steps with an error of the opposite sign in order to eliminate the accumulated integral. The extra feedback path introduced in Fig. 1 does nothing when the actuator is not saturated, and when it is saturated, it limits the integral action to the value producing saturation.

When α is not chosen as T, then (2) becomes

$$u_i(kT+T)=(1-T/\alpha)\,u_i(kT) + (T/\alpha)\,u_s(kT)+(K_i-K_p/\alpha)\,Te(kT) \quad (3)$$

Ignoring the last term, and recognizing the middle term on the right as a constant during saturation, one sees that the first term represents the root of the associated homogeneous difference equation, and hence the coefficient determined by α acts like a time constant, or in this case a forgetting factor for the contribution of old information in the sum. When there is considerable noise, and when derivative action is included in the controller, the use of an α larger than T can be beneficial.

3. Integral Control Based Learning Control

Reference (Phan, and Longman, 1988) develops a mathematical framework for linear learning control including as a special case integral control based learning control. The development is in a discrete modern control formulation, which is natural for hardware implementation.

$$x(k+1)=Ax(k)+Bu(k)+w(k)$$

$$y(k)=Cx(k) \quad (4)$$

Here $u(k)$ is the learning control action, A is the closed loop system matrix, and $w(k)$ contains the command and any disturbance or forcing function that repeats each repetition. The same initial condition applies at the start of each repetition in the learning control problem. The integral control -based learning control is given by the following sum over the errors observed at the appropriate time step in all previous repetitions, and can be calculated in recursive form

$$u_{j+1}(k)=\Phi\sum_{i=1}^{j} e_i(k+1)$$
$$u_{j+1}(k)=u_j(k)+\Phi e_j(k+1) \quad (5)$$

Such a sum is computed for each time step of the desired trajectory. The subscript indicates the repetition number, which starts at 0 for the first run using feedback control only, and then with the first repetition the learning control adjustments begin. Φ is the learning gain. Note that the errors involved are one step ahead of the control signal to account for the one time step delay of an input affecting the output in (4). The condition required for convergence to zero tracking error as the repetitions progress is that $|\lambda_i(I-CB\Phi)|<1$ for all i, where the λ_i are the eigenvalues (Phan, and Longman, 1988). In scalar-input, scalar-output problems, the product CB is a scalar whose sign & magnituse one usually knows (at least roughly), so that one can pick a learning control gain that satisfies this condition easily. Note that this condition depends only on the input and output matrices in discrete time, and is independent of the system dynamics. Reference (Elci, Longman, Phan, Juang, and Ugoletti, 1994) discusses a more restrictive condition that involves knowledge of the system dynamics, which when satisfied insures well behaved transients.

4. The System Model Used in ARW Simulations

In this paper we test ARW methods on a mathematical model of a single robot joint. The joint angle is $\theta(t)$ and the desired joint angle specified at sample times is $\theta^*(kT)$ where T is the sample time. We use a proportional controller with rate feedback to execute a rotation in the horizontal plane, so the governing equation can be written as

$$I\ddot{\theta} = k_p u_c - k_p \theta - k_d \dot{\theta} \tag{6}$$

u_c is the command to the feedback system, so that during the first repetition, number 0, it is the desired trajectory using a zero order hold. For later repetitions it is adjusted according to learning law (5) in order to converge to that command that causes the feedback control system to produce the desired trajectory as output.

The inertia chosen corresponds to the inertia seen by the base joint of the Robotics Research Corporation K-series 807iHp manipulator shown in Fig. 2, with the arm fully extended, and carrying a point mass load equal to the maximum load rating of 20 lbs. The feedback controller gains are chosen to make the robot motion overdamped in this configuration in order to prevent overshoot and possible collision. The eigenvalues are chosen as -9 and -10. The associated coefficients in (6) are $I = 27.3\text{Nms}^2$, $k_p = 2457$, $k_d = 518.7$. The saturation limit for the motor for this joint of the robot is $M_1 = 247\text{Nm}$, and the sample time is chosen as 0.02 sec. From the modern state variable representation of (6), the stability condition for convergence to zero tracking error given in the previous section indicates that the learning gain must be within the range $0 < \Phi < 126$.

The desired trajectory is given in Fig. 3, together with the error produced by the feedback controller alone in repetition 0. It is the error between these two curves, which reaches nearly 9 degrees, that the learning control must eliminate. The desired trajectory is a smooth cycloidal path for a 90 degree turn followed by a return to the starting position. The total time of the desired trajectory is taken as 7 seconds (the number of sample times p is equal to 350) which causes the robot link to reach its maximum rated speed of 55 deg per second.

With a learning gain of $\Phi = 1$, this learning control converges very quickly and smoothly as shown in Fig. 4; to the resolution of the figure, zero error is reached in four repetitions, and the torque commands do not approach the torque saturation limit. However, several situations may apply. One may not know enough about the system to be able to set the gain a priori for such good behavior. Or one may not be given the opportunity to search experimentally for such a gain. Furthermore, there may not exist a gain that produces such good behavior as discussed in (Elci, Longman, Phan, Juang, and Ugoletti, 1994).

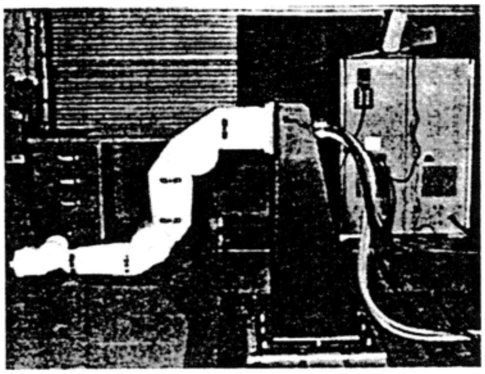

Fig. 2 A seven degree-of-freedom Robotics Research robot.

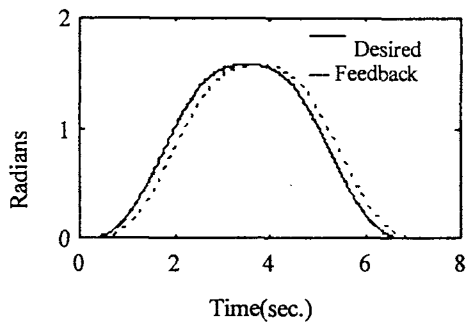

Fig. 3 Desired trajectory and feedback position output.

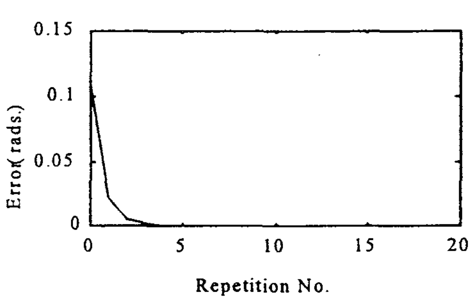

Fig. 4 RMS error when learning gain is 1.

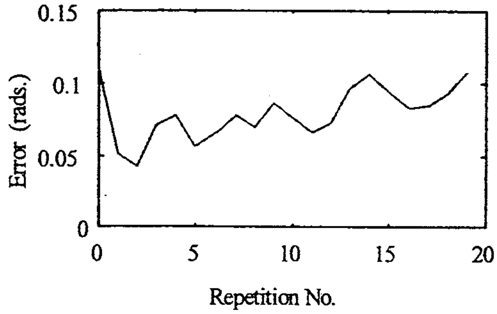

Fig. 5 RMS error with true hardware actuator saturation when learning gain is 2.

If instead a learning gain of $\Phi=2$ is used, which is still well within the limit of 126 needed for convergence, the saturation limits are hit many times during the learning process, and the resulting RMS tracking errors for all points in a repetition as a function of repetition are shown in Fig. 5.

Assuming that one of the above situations applies, and we are using the learning gain of $\Phi=2$, we now study three possible modes of applying anti-reset windup ideas to this learning process with the aim of improving the learning performance.

5. Imposing a Chosen Saturation Limit on The Learning Control Signal Alone

If we know that large torques are not required to perform the trajectory, then it is reasonable to create artificial limits M^+ and M^- to impose on the learning control signal to prevent it from applying large torques. It is then reasonable to include anti-reset windup. The system equations become

$$
\begin{aligned}
&I\ddot{\theta}_j + k_d\dot{\theta}_j + k_p\theta_j = k_p u_{cs,j} \\
&u_{c,j+1}(kT) = u_{cs,j}(kT) + \Phi e_j((k+1)T) \qquad (7) \\
&u_{cs,j}(kT) = Sat(u_{c,j}(kT)) = \begin{cases} M^+ \; if u_{c,j}(kT) > M^+ \\ -M^- \; if u_{c,j}(kT) < -M^- \\ u_{c,j}(kT) \; otherwise \end{cases}
\end{aligned}
$$

Figure 6 shows the learning control signal history for the 4th, 9th, and 19th repetitions with $M^+=M^-=2$. The result is unaffected by the saturation limit since it never reaches this value. Decreasing the limit to $M^+=1.6$ and $M^-=-0.1$ produces Fig. 7. The limits successfully influence the peak errors both positive and negative. Although the figure is somewhat difficult to read, it is clear that introducing the limits also causes more violent changes in the portion of the trajectory to the right of the peak where it is no longer saturated. This suggests that a more sophisticated rule is needed.

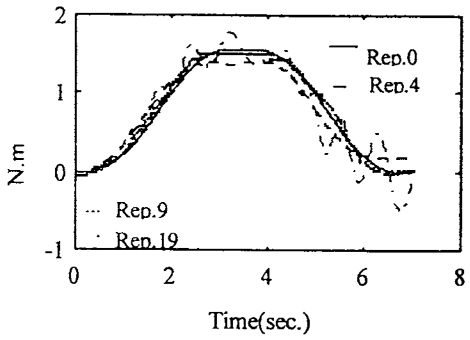

Fig. 6 Learning signal for pseudo-saturation limit $M^+=M^-=2$.

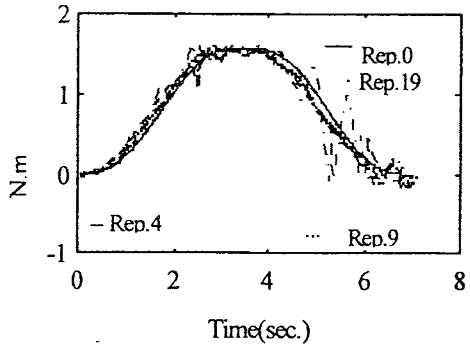

Fig. 7 Learning signal for pseudo-saturation limit $M^+=1.6$ and $M^-=0.1$.

6. Applying ARW on the True DC Motor Saturation Limit

We now apply the full concepts of anti-reset windup underlying Eq. (1). The integral action is operating in repetitions, and the feedback signal is operating in time. At any given time step of a given repetition it is the sum of these that must be applied by the actuator. To obtain anti-reset windup to prevent the windup of the integral action in repetitions, we need to subtract off the contribution of the feedback signal. The result is in the same form as Eq. (7), except that the limits

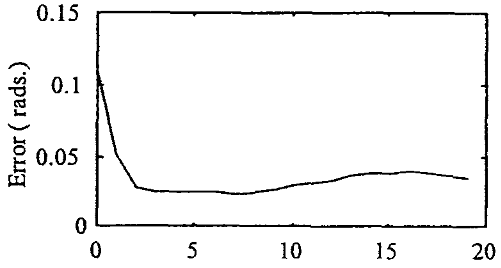

Fig. 8 RMS error for the DC motor limit saturation.

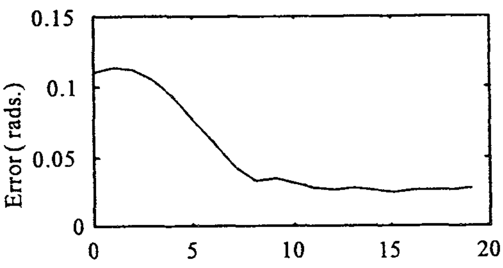

Fig. 9 RMS error for the adaptive saturation limit.

M^+ and M^- now become time-step dependent based on the feedback signal

$$M^+(kT) = \frac{M_1}{k_p} + \theta_j(kT) + \frac{k_d}{k_p}\dot{\theta}_j(kT)$$
$$M^-(kT) = -\frac{M_1}{k_p} + \theta_j(kT) + \frac{k_d}{k_p}\dot{\theta}_j(kT) \quad (8)$$

Figure 8 presents the results. This is to be compared to Fig. 5 which does not have ARW, and we see that there is a significant improvement in learning behavior. We know that the learning process is convergent to zero tracking error. But as discussed in (Longman, and Huang, 1994) we may wish to freeze the learning signal when the RMS error has reached a local minimum that represents substantial improvement as in the region from repetitions 4 to 7. It is clear from the comparison of Figs. 5 and 8 that the use of ARW in learning control can be of help in improving the transients during the learning process.

7. Using an Adaptive Saturation Limit in ARW

In many applications it may be desirable to keep the transients under very tight control, possibly at the expense of taking a significantly longer time to learn. For example, it could be very expensive to shut down an assembly line in order to conduct learning control repetitions to improve some part of the process. However, if one could learn during assembly line operation, and guarantee that the learning process would not cause the production of products that must be scrapped because they do not meet the required specifications, then one would be happy to use learning control to improve the product, and the learning control need not learn quickly.

In most applications employing a feedback controller, this controller will do a reasonably good job, although it leaves errors that a learning controller can fix. This suggests that during the learning process there is no need for the torque history to make large excursions away from a pure feedback torque history. Therefore, we record the torque input history produced on the first repetition when there is no learning control signal, and then during the repetitions that follow, the torque at each time step k is not allowed to deviate more than some prescribed amount from the recorded feedback torque at that time step. We can start with the limits being very tight, and then relax them as needed. If we are on the desired trajectory when we reach the chosen saturation limits, then we know we must extend the limit. One can generate many schemes of this general type. The following one is designed to operate rather quickly, by using the current feedback control signal

$$M_j^+(kT) = \frac{u_{FB,j}(kT) + M_{2,j}(k)}{k_p} + \theta_j(kT) + \frac{k_d}{k_p}\dot{\theta}_j(kT)$$
$$M_j^-(kT) = \frac{u_{FB,j}(kT) - M_{2,j}(k)}{k_p} + \theta_j(kT) + \frac{k_d}{k_p}\dot{\theta}_j(kT) \quad (9)$$

It starts with limits $M_2 = 20$ around the feedback signal for repetition zero, and when the value is increased at a certain time step because the trajectory hits the limit, the limit is not allowed to increase again until 10 time steps have passed (Chang, 1994). Figure 9 shows the resulting RMS errors as a function of repetition. The learning process has been slowed down, and better behavior is obtained for later repetitions than in Fig. 8.

There are other versions of this general approach. It is interesting to note that we started with the intent of seeing the usefulness of anti-reset windup ideas in learning control. Such thinking has led us to methods in this section that appear very similar to methods obtained in (Chang, Longman, and Phan, 1992), and in more detail in (Chang, 1994), obtained by ad hoc thinking.

In the present section and the two previous sections, we have addressed the use of ARW ideas in learning control, when the desired trajectory is executable by the system within the hardware saturation limit, but transients may cause saturation. We have also seen that ARW can be effective in facilitating smooth learning. In learning control there are two other situations in which ARW might be of help: handling repetitive errors in the initial conditions, and recovery from saturation in a nonfeasible trajectory.

8. ARW Applied with an Initial Condition Mismatch

A basic assumption of learning control is that the system returns to the same desired initial condition after every repetition of the command, before the start of the next repetition. Presumably this initial condition is on the desired trajectory. The difficulties of having incorrect initial conditions in learning control are discussed in (Heinzinger, Fenwick, Paden, and Miyasaki, 1989, Arimoto, 1990, Arimoto, Naniwa, and Suzuki, 1991, Bien, and Lee, 1991). We can have initial condition errors of two kinds. There may be random effects that cause different initial conditions on different repetitions. There can also be systematic errors that produce the wrong initial conditions every time (Elci, Longman, Phan, Juang, and Ugoletti, 1994). The latter occurs in robots subject to gravity, which disturbs the feedback control system. Commanding the desired starting point to the feedback control system

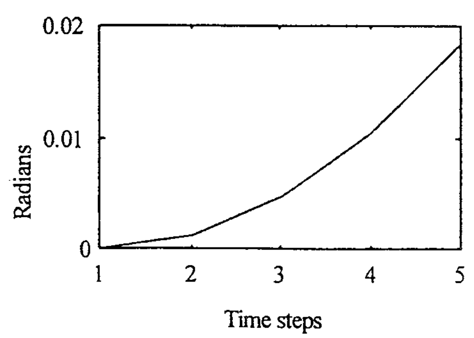

Fig. 10 Desired trajectory for 5 step trajectory.

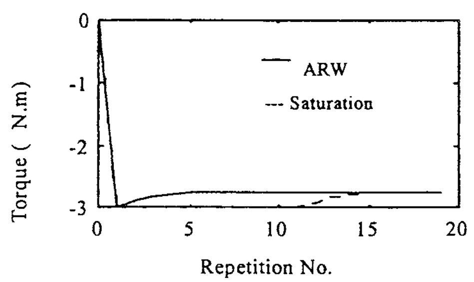

Fig. 11 Input torque for step 2 in initial condition mismatch case.

results in going to the wrong position. If one does not do some kind of learning control to find out how to reach the desired initial condition, then systematic errors exist that can saturate the actuator during the first time step or steps.

We study the use of ARW for this situation. For simplicity of understanding we pick a different problem to simulate consisting of learning during the first few steps with a system given by

$$y_j(k+1) = 0.99 y_j(k) + 0.01 u_{cs,j}(k) \quad (10)$$

The initial condition is $y_j(1) = 0.1$, but we wish it to be zero. The integral control based learning control will converge with learning gains in the range $0 < \Phi < 200$. We pick $\Phi = 50$. We apply learning control with ARW in the form of Eq. (5) with $M^+ = M^- = 3$. This value is chosen so that it is impossible for the system to correct the error in the first time step, but by the second time step it is capable of having zero tracking error.

Figure 10 shows the 5 time step desired trajectory, and Fig. 11 shows the resulting torque histories for time step 2 as a function of repetition number, for the ARW case and for the case of

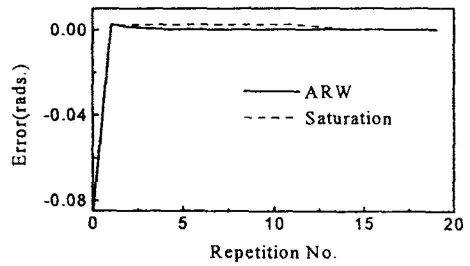

Fig. 12 Error for step 2 in initial condition mismatch case.

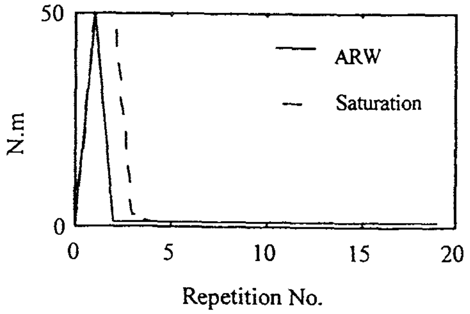

Fig. 13 Input torque for step 12 in nonfeasible desired trajectory case.

actuator saturation only. This figure shows that the ARW case leaves saturation after repetition 1, but without ARW, the learning control action remains saturated until after repetition 11. Figure 12 shows the corresponding error for time step 2 as a function of repetition number, for the ARW case and for the case of actuator saturation only. Thus, ARW is shown to be very effective.

In (Heinzinger, Fenwick, Paden, and Miyasaki, 1989, Arimoto, 1990, Arimoto, Naniwa, and Suzuki, 1991) the use of a forgetting factor is suggested as a means of improving the performance with initial condition mismatch. The learning control suggested takes the form

$$u_j(k) = (1-\alpha)\, u_{j-1}(k) + \alpha u_b(k) + \Phi e_{j-1}(k+1) \tag{11}$$

where $u_b(k)$ is the learning history from the past having the best error, and the forgetting factor lies in the range $0<\alpha<1$. The rule does not consider saturation limits. Applying this rule to the example under discussion did not produce any improvement of the learning. It resulted in the same error histories as with learning control alone, staying at saturation until after repetition 11. The best learning history was always the latest, so (11) reduces to the standard learning control.

9. ARW in Learning Control with a Nonfeasible Desired Trajectory

It is a common situation that the only trajectory that one can state as being the desired trajectory is one that no physical system can perform, e. g. performing a perfect right angle turn at constant velocity in welding with a robot. Learning control will not be able to produce zero tracking errors in such situations due to the actuator saturation limits. However, from the results of the previous section we expect that ARW can improve the tracking performance of a learning controller immediately after the nonfeasible portion of the trajectory. During the necessarily saturated portion of the trajectory, there can be no difference between pure learning control and ARW. ARW will stop the windup, but this has no physical implications since the windup is occurring only in the controller (unless it causes overflow in a digital controller). Just as in the response to initial condition errors, ARW improves the recovery behavior for subsequent time steps in which the system can again follow the commanded trajectory.

We consider a unit step command, of zero for time steps up through 10 and 1 for time steps thereafter, applied to (10). This time we pick the learning gain $\Phi = 100$ and the saturation value $M = 50$. The desired trajectory requires a torque history that is zero at all time steps except time step 10, at which the torque must be 100.

Steps 10 and 11 remain saturated throughout the repetitions. Figure 13 shows the input torque for time step 12. On the initial rise the dashed curve coincides with the solid curve, and for the next repetition, the dashed curve stays at 50 before coming down over the next two repetitions. Again, we see that ARW gives faster recovery in time steps after the end of a time period in which the actuators must be saturated, compared to pure learning control applied to the system actuator

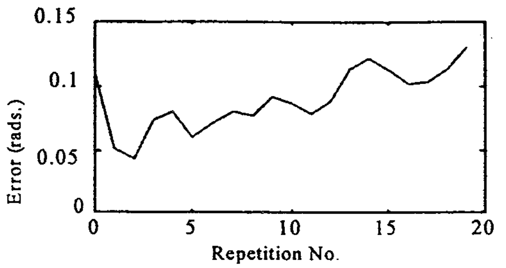

Fig. 14 RMS error for repetitive control with hardware saturation and learning gain 2.

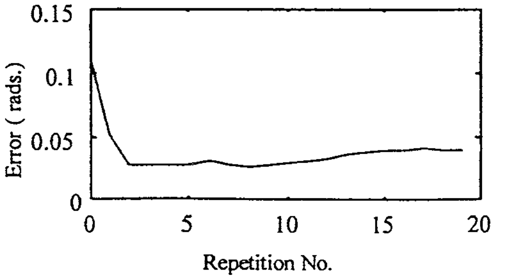

Fig. 15 RMS error for repetitive control with anti-reset windup and learning gain 2.

saturation limits.

10. ARW in Integral Control Based Repetitive Control

In repetitive control the command is a periodic function of time. It differs from learning control in the sense that the system need not return to the same initial condition before the next repetition starts. Transients can propagate from one period into the next, and this produces a more difficult stability problem for the learning process. The extension of the results above to this case is immediate. The ARW repetitive control analogous to (7) for a process of period p steps is

$$I\ddot{\theta}(t)+k_d\dot{\theta}(t)+k_p\theta(t)=k_p u_{cs}(t)$$
$$u_c(kT+(j+1)pT)=u_{cs}(kT+jpT)+\Phi e((k+1)T+jpT)$$
$$u_{cs}(kT+jp)=Sat(u_c(kT+jp))=\begin{cases} M^+ & if u_c(kT+jp)>M^+ \\ -M^- & if u_c(kT+jp)<-M^- \\ u_c(kT+jp) & otherwise\end{cases} \quad (12)$$

where $u_{cs}(t)$ employs a zero order hold. Figure 14 shows the RMS error plot for the repetitive control problem analogous to Fig. 5, using hardware saturation limits and a learning gain of 2. Although the learning process for the learning control problem is known to be convergent, it is more difficult to test for stability of repetitive control problems, and they are more likely not to be convergent. However, it is very often the case that the error decreased significantly before it increases, and therefore, one can use the method to decrease the error to a minimum, and then freeze the repetitive control signal (Longman, and Huang, 1994). Use of anti-reset windup on this problem produces Fig. 15 exhibiting much improved performance.

11. Conclusions

In this paper we have studied the use of anti-reset windup ideas to improve the performance of integral control-based learning control. An issue that has received some attention in the literature is initial condition mismatch in learning control. It is shown here that use of anti-reset windup is particularly helpful in speeding up the convergence of the learning operation in such cases. Similarly, anti-reset windup is shown to be effective in situations where the desired trajectory is not feasible and requires actuator output beyond the saturation limits. Anti-reset windup was also investigated to improve the learning transients when the desired trajectory does not require inputs above the actuator saturation limit, but the transients request actuator outputs above this limit. The approach is of more limited use in this situation. It is shown that anti-reset windup can improve the transients, and this can be useful when one uses learning control for a limited number of repetitions, and then freezes the learned signal at one that produces good results.

Acknowledgments

This work was supported by grant NO. 981-1001-003-1 from the Basic Research Program of the KOSEF.

References

Arimoto, S., 1990, "Robustness of Learning Control for Robot Manipulators," *Proceedings of the IEEE International Conference on Robotics and Automation*, Cincinnati, Ohio, pp. 1528～1533.

Arimoto, S., Naniwa, T., and Suzuki, H., 1991, "Selective Learning with a Forgetting Factor for Robotic Motion Control," *Proceedings of the IEEE International Conference on Robotics and Automation*, Sacramento, CA, pp. 728～733.

Astrom, K. J., and Wittenmark, B., 1984, *Computer Controlled Systems*, Prentice Hall, Englewood cliffs, NJ.

Bien, Z., and Lee, K. H., 1991, "Initial Condition Problem of Learning Control," *Proceedings of the IEE*, Vol. 138, No. 6, pp. 525～528.

Chang, C. -K., Longman, R. W., and Phan, M., 1992, "Techniques for Improving Transients in Learning Control Systems," *Advances in the Astronautical Sciences*, American Astronautical Society, Vol. 76.

Chang, C. K., 1994, *Learning Control in Linear and Nonlinear Systems*, Ph. D. Dissertation, Columbia University, New York, NY.

Elci, H., Longman, R. W., Phan, M., Juang, J. -N., and Ugoletti, R., 1994, "Discrete Frequency Based Learning Control for Precision Motion Control," *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, San Antonio, TX.

Heinzinger, G., Fenwick, D., Paden, B., and Miyasaki, F., 1989, "Robust Learning Control," *Proceedings of the 28th IEEE Conference on Decision and Control*, Tampa, Florida

Longman, R. W., and Huang, Y. -C., 1994, "Use of Unstable Repetitive Control for Improved Tracking Accuracy," ASME Winter Annual Meeting, *Adaptive Structures and Material Systems Symposium*.

Phan, M., and Longman, R. W., 1988, "A Mathematical Theory of Learning Control for Linear Discrete Multivariable Systems," *Proceedings of the AIAA/AAS Astrodynamics Conference*, Minneapolis, Minnesota, pp. 740～746.

Ryu, Y. S., and Longman, R. W., 1994, "Use of Anti-Reset Windup in Integral Control Based Learning Control," *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, San Antonio, TX, Oct.

Solcz, E. J. and Longman, R. W., 1992, "Robustness of Repetitive Controllers to Frequency Uncertainties," *Proceedings of the Conference on Information Sciences and Systems*

Wittenmark, B., 1989, "Integrators, Nonlinearities and Anti-Reset Windup for Different Control Structures," *Proceedings of the American Control Conference, Pittsburgh*, PA, pp. 1679～1683.