Transients in Reconfigurable Control Loops

Gyula Simon, Tamás Kovácsházy, Gábor Péceli

Department of Measurement and Information Systems
Technical University of Budapest, H-1521 Budapest, Hungary
Phone: + 36 1 463 2057, Fax: +36 1 463 4112
email: {simon,khazy,peceli}@mit.bme.hu

Abstract

In this paper transient management techniques are investigated in a scenario where the reconfiguration or replacement of a forward-loop controller is required due to changes in the environment or in the plant. Such a change in the closed control loop may have undesirable transient effects, which may degrade the performance of the controlled system. Since the transient cancellation and reduction schemes used in open-loop systems can not be used here, new solutions are proposed for run-time transient handling.

1. Introduction

The study of reconfigurable dynamic systems is related mainly to larger scale, distributed intelligence monitoring and control systems. To use reconfiguration techniques in monitoring and control has real meaning if drastic changes occur in the operation of a physical system. Changes due to faults evolving into system degradation are typical examples. In such cases the supervisory system should observe the changes and turn to another operation mode. With other words the models applied within the computer program or within the controller are also to be changed to correctly represent the physical system. Model changes can be performed using different techniques [1]. For conventional system models the typical solution is the adaptation or direct change of the coefficients and/or the (signal processing) structure. Such changes, however, can cause undesirable transient effects, which may (temporarily) degrade the performance of the overall system [2]. The transient management of such systems has two main fields: the proper choice of the structure with advantageous transient properties [2], and the run-time transient suppression. In this paper the latter field, the reconfiguration transients and their run-time handling in control loops are investigated. Note that although the control-loop scenario is considered, the emphasis is laid on the transient effects and the possible suppression methods in the closed loop, rather than on the overall control performance.

In the signal-processing framework successful techniques have been proposed to cancel or suppress transients caused by the abrupt changes of filter coefficients (see [3] for an overview). In a frequently used model the coefficients (and possibly the order of the filter as well) are changed abruptly. In this context the transient is defined as the difference between the actual output and the ideal steady-state output of the new filter. The so-called *output-switching* method uses the above definition and runs filters in parallel, as shown in Fig. 1. This solution provides transient-free output but the overhead is high. More sophisticated methods use (usually simpler) transient eliminator filters to collect information and to aid the initialization of the new filter. At the time instant of the change not only the filter parameters, but also the filter states are updated using the information gathered by the transient eliminator filter [3, 4]. This elegant solution is part of a more general framework where the state variables are updated so that the transient effects are minimized or cancelled.

In closed control loops the use of parallel controllers or parallel transient elimination filters is meaningless. The general framework of state variable update is fortunately still applicable, the proposed algorithm uses the same idea.

In Section 2 the problem will be highlighted, with special care of the closed-loop nature of the problem. The transient management techniques will be surveyed in Section 3, and the new algorithm will be proposed. Technical details will be discussed in Section 4. In Section 5 an example will be given to illustrate the efficiency of the novel method.

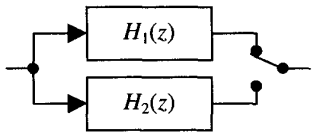

Figure 1. Output switching method.

0-7803-5890-2/00/$10.00 © 2000 IEEE

2. Reconfiguration Transients in Control Loops

In this paper the classical series compensation control loop shown in Fig. 2 is considered. The discrete time controller gets the error signal $e(k)$, (which is the difference of the reference signal $r(k)$ and the plant output $y(k)$) as input, and produces the control input signal $v(k)$. The reconfiguration is performed at time instant k_R, so the control input is defined as follows:

$$v(k)=\begin{cases} v_1(k), & \text{for } k < k_R \\ v_2(k), & \text{for } k \geq k_R \end{cases} \quad (1)$$

where v_1 and v_2 are the outputs of the old and new controller, respectively.

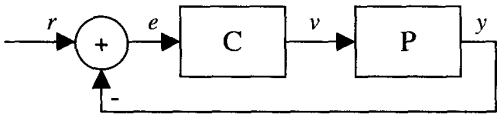

Figure 2. The feedback control loop system. C: controller, P: plant

The following assumptions are also made:

(a) The controller is a linear system of which the state variables can be read and arbitrarily initialized.
(b) The system is in steady state (or close to it) when the controller is reconfigured.
(c) The feedback error signal is small before reconfiguration.
(d) The controller is changed abruptly, in one time instant.
(e) The reconfiguration is independent of the reference signal (the reference is not changed at reconfiguration).

The plant can be a nonlinear dynamic system, no other restrictions are assumed. Assumption (a) makes the reconfiguration possible, and (b) enables the clear detection of the transients, the separation of transient errors from tracking errors. Assumptions (c)-(e) will be used in Section 3 to construct new transient management algorithms.

In the case of a closed loop control system the definition of the transient is similar to that of the open loop systems. The transient caused by the reconfiguration is the difference between the actual plant output and the hypothetical steady-state plant output (i.e. assuming the new controller was used for a long time).

3. Transient management in control loops

Reconfiguration transients depend significantly on the actual internal energy conditions of the controllers. With a state-variable interpretation the internal energy of the dynamic controllers is distributed among their state variables. At reconfiguration these state variable values serve as initial conditions causing transients. These state variables also give the possibility to suppress reconfiguration transients through proper initialization.

As it was already mentioned, the output-switching methods and its more sophisticated versions cannot be used in closed loops, because parallel processing is not possible. In practice two widely used techniques are used if reconfiguration is needed: state zeroing and state preserving. Both methods can be considered as very simple state updating algorithms.

State Zeroing Method: When the controller's parameters are changed, the state variables are set to zero. This is a usual and 'safe' way of reconfiguration, since the new controller starts from a zero-energy state, thus its behavior is predictable. Another advantageous property of this simple method is that the structural representation has no effect on the transient. However, apart from the rare case when the state variables are all zeros, this method always produces transients.

State Preserving Method: This method preserves the old controller's state variables after the parameter change. This method may be useful when the successive controllers are similar (i.e. the successive parameter sets are similar), and the old states are also 'meaningful' for the new controller. The application of this method is troublesome if the structure is also changed. A drawback of this solution is that the transient behavior of the new controller depends on the last state of the old controller, thus it is unpredictable in design time.

Both state zeroing and state preserving methods may have their field of application, but generally neither of them provides satisfactory results, as it will be illustrated in Section 5. Hereinafter a more adequate solution will be proposed.

Output Fitting Method. If assumption (e) holds, and the control error was zero before reconfiguration, then the new controller should produce the same output as the old one in order to avoid transients. If the error signal is not exactly zero, but small according to assumption (c), then the output of the new controller should still be close to that of the old one. Based on this fact the transient management algorithm can be constructed, which tries to keep the control input signal 'smooth' around the reconfiguration. A reasonable definition of smoothness is based on Taylor-series. Let

$$\begin{aligned} v_1(k_R) &= v_2(k_R), \\ \dot{v}_1(k_R) &= \dot{v}_2(k_R), \\ v_1^{(2)}(k_R) &= v_2^{(2)}(k_R), \\ &\vdots \\ v_1^{(N-1)}(k_R) &= v_2^{(N-1)}(k_R), \end{aligned} \quad (2)$$

where $v_1^{(l)}(k_R)$ is the lth (left-hand side) derivative of the old controller's output at the reconfiguration time instant, and $v_2^{(l)}(k_R)$ is the lth (right-hand side) derivative of that

of the new controller. This method is illustrated in Fig. 3. *a*.

Assuming for a while, that the new controller operates in parallel with the old controller, and thus producing output values before the reconfiguration, the right-hand side derivatives of the new control input signal can be replaced by the left-hand side derivatives. Having discrete samples, the equality of the left-hand side derivatives in Equation 2 implies the equality of the samples preceding the reconfiguration:

$$v_1(k_R - i) = v_2(k_R - i), \text{ for } i = 0...N-1. \quad (3)$$

Thus a possible way to ensure the smoothness of the control input is to run the old and new controllers *virtually* in parallel so that both controllers produce the same outputs before reconfiguration, as illustrated in Fig. 3. *b*. In this case the new controller can continue the operation 'smoothly', or, with other words, its state variables contain values which enable the smooth reconfiguration. Of course, instead of running the controllers in parallel, which is impossible in one loop, the state variables are computed from the constraint that the past N output samples of the new controller is the same as that of the old controller, provided their inputs are the same (i.e. the past N error signal samples).

Note that this method cannot ensure the *complete* rejection of transients, since the initial state variables enforced to the new controller are not really produced by it. These initial values change, and a new stationary state is reached, which may also cause small transients. But the level of these secondary transients is much smaller than that of the transients without proper management, according to practical experiments. The effectiveness of the method will be illustrated in Section 5.

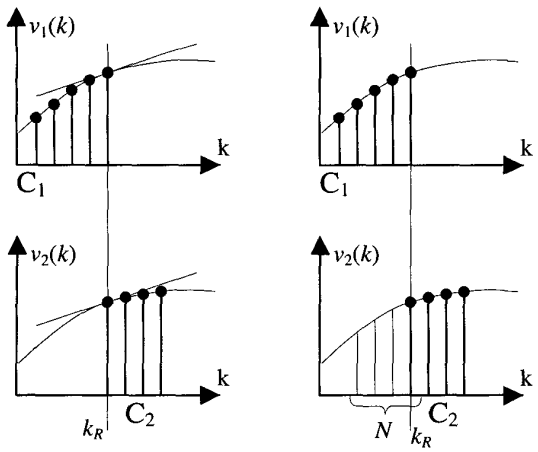

Figure 3. Output Fitting strategies. The reconfiguration is made at time instant k_R. *a.* The new controller (C_2) is initialized to continue the output of the old controller (C_1) smoothly, by fitting the derivatives. *b.* The new controller is initialized as if it produced the past few samples of the old controller.

4. Computation of the initial values

There are multiple ways for the calculation of the state variables, depending on the time and hardware resource constraints. If the reconfiguration must be performed immediately, then the recalculation of the state variables must be made in one time instant. In this case a linear equation system must be solved with N unknown variables (the state variables) and N equations (the relationship between the input and output in the past N samples). The form of the equations strongly depends on the controller's structure, and can be solved using different well-known techniques, but in general for higher N it has high computational complexity (order N^2).

If the reconfiguration can be delayed by N samples, or the reconfiguration command is known at last N samples before the required reconfiguration time instant, or it is possible to run an auxiliary filter in parallel with the controller, then it is possible to solve the equations in a recursive manner. This recursive method will briefly be described here.

Let us consider the state-variable representation of the controller:

$$\begin{aligned} x_{k+1} &= Ax_k + be_k, \\ v_k &= c^T x_k + c_0 e_k, \end{aligned} \quad (4)$$

where x is the state vector, e and v are the input and output of the controller, respectively, A is the state transition matrix, and b, c are the input and output coupling matrices, respectively. For this controller a dead-beat observer [6] can be designed with the following structure:

$$z_{k+1} = Az_k + be_k + g(v_k - c^T z_k) - gc_0 e_k, \quad (5)$$

where z is the observed state vector and g is a parameter vector which has to be chosen adequately to provide the dead-beat behavior. The constraint is that all roots of the polynomial $\det(\lambda I - A + gc^T)$ are 0. From this constraint parameter vector g can be calculated off-line. The complexity of the above observer is the same as that of the controller. When operated, the observer must be switched on N samples before the reconfiguration.

It is possible, of course, that the order of the controller is higher than the required number of fitted derivatives (N). In this case the initialization can be made by different sets of parameters, all of them satisfying the constraint in Equation 3, but they may be different from the viewpoint of transient suppression. In these cases, unfortunately, no general solution exists, the structure of the controller must be taken into account to decide which variables to update, and how. As a general guideline, it can be stated, that in

those structures, where the state variables has clear physical meaning, certain state variables can be chosen, for which the state update is advantageous, and there can be states for which the preservation of the old values is the best strategy. E.g. in a PID controller structure, where the actual states of the integrator and he differentiator are the state variables, it is much better idea to update the integrator's state than that of the differentiator. If the controller's representation conceals the physical meaning of the states (e.g. a rational function form of the previous PID controller), the decision is much harder.

5. Illustration

The following example illustrates the reconfiguration effect on a system in a feedback loop, and also compares the discussed solutions. A two-link planar robot arm (see Fig. 4), which is a strongly nonlinear mechanical system [5], was controlled by simple digital controllers (separate controllers on each joint). The robot was to move the tool (the end point of the second link) on a straight line (see the upper right plot of Fig. 5). Each joint was controlled separately by a simple PID-like controller, using only the joint position error e_k (the difference between the desired and measured joint angle) and the joint speed error $\dot{e}_k$ (the difference between the desired and measured joint angular speed), as inputs. The output of each controller is calculated by

$$v_k = Pe_k + Ix_k + De_k$$

where P, I, and D are the parameters of the controller. The state variable x_k contains the integrated angle error, and is updated by

$$x_k = x_{k-1} + T_S e_{k-1},$$

where T_S is the sampling interval. In the example two controller sets were used: at the beginning and at the end of the line controller set #1 (in time intervals [0s...1s] and [3s...4s]), and in the middle of the line controller set #2 was used (in time interval [1s...3s]). The change of the controller means the application of different parameter set and possibly the update of the controller's state variable, which is done in one sampling interval. The reconfiguration of the controllers was performed using three different strategies. In the upper row of Fig. 5 the desired tool trajectory (right side), and the joint angles (left side), which were designed from it, are shown. The joint angle errors, together with the tool's position error trajectory, for different reconfiguration strategies, are shown in the next three rows. Note that the joint angle errors are due to two phenomena: the controllers' tracking error (always present unless the robot arm is motionless) and the transients caused by the reconfiguration. The aim is to keep the second effect as low as possible. The tool-position error, in a 100 ms time interval after the reconfiguration, is highlighted by thick lines to show the effect of the reconfiguration transients.

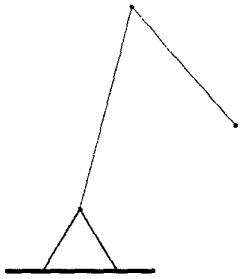

Fig. 4. The simulated two-link planar robot used in the illustration. The links are 0.6m and 0.8 m long, their mass is 1kg and 0.5 kg (concentrated in the joints), and the friction coefficients are 3 kgm^2/s.

The second row of Fig. 5 shows the reconfiguration using the state zeroing method. The large transients are clearly visible both on the joint angle and the tool position error plots. In this case the ideal state variables were far from the zero value, so this strategy was not successful. The state preserving method caused also high, although little smaller transients. The reason is that the consecutive controllers had very different parameters, so the old state variables caused large transients, as it is shown in the 3rd row of Fig. 5. The output fitting method produced much less transient, as shown in the 4th row of Fig. 5.

6. Conclusion

In this paper the problem of transient effects in reconfigurable control loops was highlighted. It was shown that the reconfiguration transients cause serious performance degradations. These errors can be much higher than the normally occurring ones in control loops, if no extra effort is taken to keep their level as low as possible. It was also emphasized that the reason of the transients is the state variable mismatch of the old and the new controllers. Unfortunately, the well-known transient reduction methods used in open loop systems cannot be applied for systems operating in closed loop.

The state variables of the controllers are usually available for the system designer thus the reconfiguration transients can be reduced, if their proper initialization is solved. Two simple and widely used reconfiguration techniques, the state zeroing and state preserving methods were surveyed, and the novel output fitting method was proposed. The state zeroing and state preserving methods usually have bad transient properties, since the energy distribution of state variables in the subsequent systems may be very different. The new output fitting method updates the state variable so that the transition between subsequent controllers is smooth. This technique is much

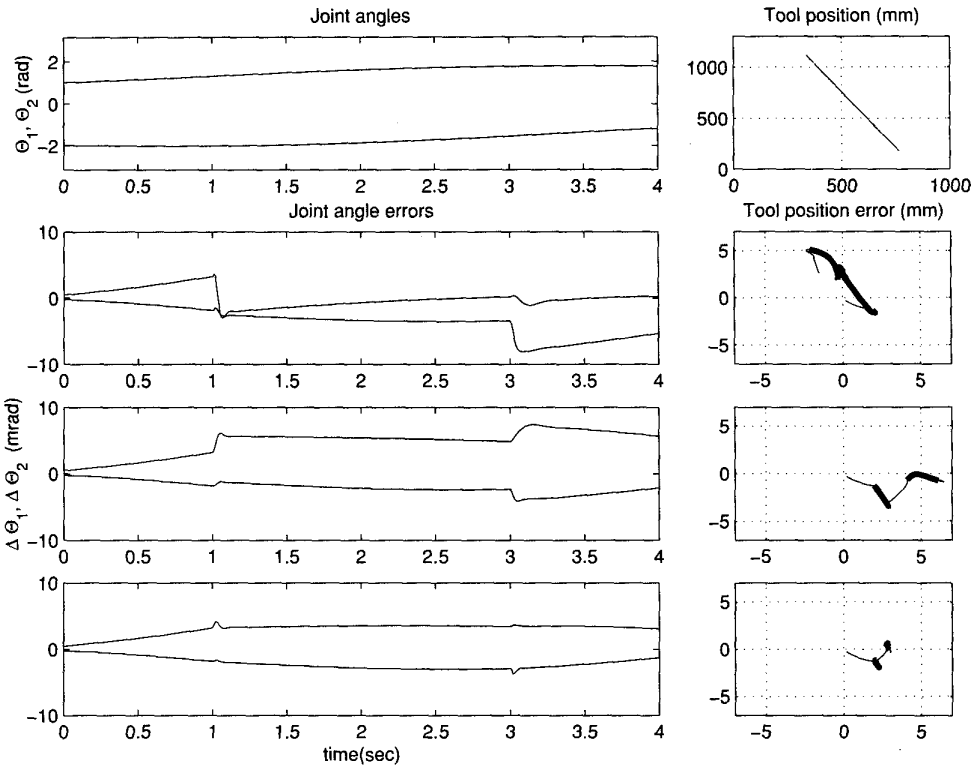

Fig. 5. Reconfiguration transients of the two-joint planar robot arm. The reconfiguration of the joint controllers was done in time instants 1s and 3s. *1st row*: ideal joint angles and tool trajectory. *2nd row-4th row*: joint angle errors and tool position error trajectories using different reconfiguration strategies. On the tool position error trajectories thick lines show the effect of the reconfiguration transients.

less sensitive to the actual parameters, and provides smaller transients than the simple methods.

The smoothness of the transition can be adjusted through the number of fitted derivatives. For the calculation of the initial values two methods was suggested: the one-step reconfiguration with intensive computational load at the reconfiguration time instant, and a recursive method, which makes the calculations in parallel with the old controller. The efficiency of the novel method was also illustrated by a simulated control example of a nonlinear system (a two-link robot arm).

Acknowledgements

The research was sponsored, in part, by the Defense Advanced Research Projects Agency (DARPA) (US) under agreement number F33615-99-C-3611.

References

[1] Sztipanovits, J., D.M. Wilkes, G. Karsai, Cs. Biegl, L.E. Lynd, "The Multigraph and structural adaptivity," IEEE Transactions on Signal Processing, Vol.41, No.8, pp. 2695-2716, Aug. 1993.

[2] Péceli, G., T. Kovácsházy, "Transients in Reconfigurable DSP Systems," IEEE Trans. on Instrumentation and Measurement, Vol. 48, No.5, pp.986-989, Oct. 1999.

[3] Välimäki, V., T.I. Laakso, "Suppression of Transients in Variable Recursive Digital Filters with a Novel and Efficient Cancellation Method," IEEE Transactions on Signal Processing, Vol.46, No.12, pp. 3408-3414, Dec. 1998.

[4] Zetterber, L.H., Q. Zang, "Elimination of transients in Adaptive Filters with Application to Speech Coding," Signal Processing, Vol.15, No.4, pp. 419-428, 1988.

[5] John. J. Craig, Introduction to robotics: mechanics and control. Addison-Wesley Publishing Company, New York, 1989.

[6] Luenberger, D.G., "An Introduction to Observers," IEEE Trans. on Automatic Control, Vol. AC-16, No.6, pp.596-602, Dec. 1971