Comments on 'Multilayer Switching Control using Generalized Sampled-Data Hold Functions'

Mark Verwoerd & Oliver Mason*
Hamilton Institute, National University of Ireland, Maynooth
{mark.verwoerd,oliver.mason}@nuim.ie

1 Summary of main results

In the paper under discussion, the authors consider the problem of designing switching controllers for plants that are subject to large levels of uncertainty or to abrupt changes in their dynamics. There has been a great deal of interest in this problem in the recent past [7, 2], and the potential applications of switching controllers are numerous. For instance, one approach to the design of fault-tolerant systems [3, 4] is to construct models and controllers for the various operating conditions corresponding to different system malfunctions. Once the correct plant has been identified, the associated controller is switched on. Of course, in order for this type of scheme to be practical, it is vitally important to be able to identify the correct plant as quickly and efficiently as possible, and to minimise any transient effects which may result from switching to incorrect controllers during the identification process.

A major issue in the design of switching control schemes is that the system can switch to destabilising controllers before finally locking onto the correct one, which leads to very poor transient behaviour. We shall refer to such switches as destabilising switches throughout this discussion. The primary contribution of the paper is to describe a novel switching control scheme, which reduces the number of undesirable, destabilising switches that occur while identifying the correct plant. In fact, under a range of assumptions, which we shall discuss in detail below, the correct plant can be identified after a finite number of switches, at most one of which is destabilising. The authors also give an upper bound on the number of switches required in order to find the correct plant.

The architecture underpinning the MLSC scheme consists of several layers or levels of controllers. In fact, given n plants, $P_1, \dots, P_n$, the proposed architecture consists of $n-2$ layers, each of which contains a number of controllers. The

*corresponding author

various control layers are constructed in such a way that if i denotes the layer number, C and C' denote controllers, and P denotes a plant model, then:

(A1) for all i, every C in layer i simultaneously stabilises exactly i plants, while destabilising all others;

(A2) for every P, there exists a C in layer $n-2$ such that C destabilises P.

(A3) for any pair (C, C'), with C in layer 1 and C' in layer $n-2$, there exists a sequence $C_1, C_2, \dots, C_{n-2}$ such that $C = C_1$, $C' = C_{n-2}$, and C_i is a parent of C_{i+1} for $i = 1, 2, \dots, n-3$.

Under conditions (A1)-(A3), and under the additional assumption that changes in the plant model are sufficiently infrequent (that is, the dwell time is sufficiently long), the authors demonstrate that a switching algorithm can be defined with the property that after at most $n-2$ switches, and with no more than one destabilising switch, the system switches to the right controller and locks on to it until the next switching event occurs.

2 Discussion

We shall next discuss in more detail the main idea behind the MLSC scheme as well as a number of key issues, which are central to its implementation. In particular, we shall focus on the existence and construction of simultaneously stabilising and destabilising controllers (which is crucial to satisfying conditions (A1)-(A3)), on the determination of system stability/instability based on (noisy) output measurements, and on how rapidly the proposed scheme can identify the correct plant.s

MLSC and Safe Switching

Multi-layer switching control (MLSC) provides an automatic procedure for selecting, from a given set of controllers, the controller that best matches a certain plant from a given family of plants. Rather than directly identifying the correct plant model before switching to the corresponding controller, this method switches between different controllers until it hits upon a controller that destabilises the loop, in which case the correct plant model can be inferred from the switching history; or until it switches to the right controller, in which case, barring a sudden change in plant dynamics, establishing loop stability is enough to identify the correct plant model.

One of the advantages of this approach, compared to e.g. the multiple model adaptive control approach with safe switching (MMACSS) that was introduced in [2], is its simplicity. The supervisory control unit of the MLSC scheme operates solely on the basis of threshold detection and logic, and does not rely on any advanced identification or estimation algorithms. However, this simplicity also encompasses a weakness. MMACSS will never switch to a destabilising controller provided the initial controller is stabilising (so called safe switching), whereas in MLSC unstable switches are instrumental in identifying the correct plant model. This would seem to limit the method's practical applicability as the auxiliary signal (defined in equation (33) in the paper), which is used as a threshold for detecting instability, does not necessarily respect the safety bounds associated with the dynamics of the plant.

Simultaneous Stabilization

It is clear from conditions (A1)-(A3) that the construction of controllers that simultaneously stabilise a given set of plants while destabilising another, disjoint, set of plants is central to the MLSC control architecture. The problem of designing a controller that simultaneously stabilises every member in a given set of LTI plants has received considerable attention in recent years. Classical results for the case of LTI controllers indicate that simultaneous stabilisation is only possible under very specific conditions. Recently, it has been shown that the problem of simultaneous stabilisation becomes more tractable when one allows for time-varying controller structures, such as the generalized sampled-data hold functions (GSHFs) used in the paper. At this point, it is worth noting that the use of GSHFs has several known disadvantages. In particular, it has been pointed out [5] that controllers based on GSHFs tend to have poor inter-sample behaviour. This problem has been addressed and approaches to minimizing the inter-sample ripple have been developed [1, 6].

The MLSC algorithm relies on being able to construct a set of controllers that not only simultaneously stabilise a given set of plants, but at the same time destabilise another, disjoint, set of plants (this assumption underlies all of the conditions (A1)-(A3)). We would like to point out that the problem of simultaneous stabilisation and simultaneous destabilisation are not, in general, symmetrical. To see this, consider the case of LTI plants and LTI controllers. In this case, the stabilisation problem amounts to finding a controller such that the closed-loop transfer function has all its poles in the left-half plane; for destabilisation, it is enough to have at least one pole in the right-half plane. The design method in [1], referred to in Example 1 in the paper does not consider the problem of destabilisation explicitly. Thus, it is not clear how this method is used to construct a controller f_1 that stabilises P_1 and *only* P_1. A similar remark applies to the construction of the GSHFs in layer 2.

Loosely speaking, for simultaneous destabilisation to be feasible, the plant models in the model set should not be too close to each other. For instance, no matter what the structure of the controller is, any one controller cannot simultaneously stabilise and destabilise the same plant. By continuity, when two plants are nearly identical, and a controller stabilises one and destabilises the other, performance is likely to be poor, since at least one of the closed-loop poles will lie close to the imaginary axis, resulting in undamped oscillations and poor transients. Likewise, simultaneous stabilisation requires that plant models in the model set are not too far apart. This tradeoff needs to be taken into account when picking the GSHFs.

Speed of Switching

In addition to the construction of simultaneously stabilising/destabilising controllers, the frequency with which changes in plant dynamics take place, and the speed of the identification process are important issues for the MLSC scheme. For the scheme to work effectively, it is important that changes in plant dynamics do not happen too frequently. If further switches in the plant dynamics occur during the identification process, this may well have a very detrimental effect on the operation of the scheme. To illustrate what can go wrong in such a scenario, consider the following situation.

Let a finite number of plant models, $P_1, \ldots, P_n$, be given and suppose that the only controller in the top level (level $n-2$ where each controller stabilises $n-2$ plants and destabilises the other 2 plants) that destabilises P_1, P_2 is $C_{\overline{1,2}}$. Further suppose that the correct plant is initially P_1, and at some time the dynamics change so that the correct plant becomes P_2. Under the scheme proposed in the paper, once the change in plant dynamics is detected, the system should switch to the controller $C_{\overline{1,2}}$ in the top layer, and then pass through a number of child-parent switchings until we encounter a destabilising switch. This identifies the 'correct' plant and the system then switches to the corresponding controller in the first layer.

The system will only detect the first change in plant dynamics when the output exceeds the bound given by the auxiliary signal (defined in equation (23)). If, after switching to $C_{\overline{1,2}}$, the plant dynamics change back so that P_1 is again the correct plant, we will conclude incorrectly that P_2 is the correct plant and engage the wrong controller! Based on this simple idea, it is straightforward to construct an example of a sequence of switches in plant dynamics for which the MLSC scheme will *never* correctly identify the correct plant and will always switch to a destabilising controller! The only way to ensure that this cannot occur is to assume that once a switch in plant dynamics occurs, no further switch can occur until such time as the identification process is completed.

If the type of scenario envisaged in the previous paragraphs is to be definitively ruled out, then either the identification process itself must be fast or else switches in the plant dynamics must take place very infrequently. This makes the speed with which the correct plant can be identified an issue of central importance to the applicability of the MLSC scheme.

Determination of System Stability

It is worth noting at this point that the speed of identification is a potential drawback for both the MMACSS scheme, mentioned above, and the MLSC scheme under discussion. In MMACSS the speed of switching is limited by the identification module, as in order to achieve steady state identification, the algorithm must wait for transients to settle down; in MLSC the speed of identification is largely determined by the time it takes to detect stability/instability.

The method for determining stability/instability proposed in the paper is to compare the output of the system to an auxiliary signal (defined in equation (23)) that serves as an upper bound for the system output. The rationale is that if the output exceeds this bound, then the system is unstable. On the other hand, system stability should be detected by verifying that the auxiliary signal is not exceeded during the so-called *safety-time*. Clearly, this approach means that some delay in identifying the correct plant is inevitable and, for some systems, this delay could be considerable. It would be interesting to see a more detailed discussion of the limits this approach puts on the speed with which the correct plant can be identified.

In the context of determining system stability/instability, some other points are worth highlighting. First, the auxiliary signal that is used to detect instability is an upper bound on the output of the system, but it is not clear how tight this bound is. This raises the question of whether it is possible for a system to be unstable without exceeding the auxiliary signal. Given the importance of this issue to the overall scheme, it is very desirable to have theoretical results proving that the output of any unstable system will eventually exceed the auxiliary signal, and perhaps giving estimates on how long it takes for the bound to be exceeded. Such results would also be useful to detect system stability and to provide guidelines on how to set the safety time t_d, which the authors suggest using to determine stability.

It is worth noting that there is a slight issue in the proof of Lemma 2. Specifically, the constant $\alpha_{3,i}$ (defined in equation (24) and used in (28) in the paper) can be zero for low values of m, where m is the number of sampling intervals for which the control input is set to zero. In order for $\alpha_{3,i}$ to be non-zero, the matrix W_i (defined below equation (23) in the paper) must be positive definite. It follows from observability that this will be true provided $m \geq N-1$, where N is the dimension of the state space. However, if $m < N-1$, then we cannot in general assume that $\alpha_{3,i} > 0$ and, if $\alpha_{3,i} = 0$, the constants $\alpha_{1,i}, \alpha_{2,i}$ in the statement of Lemma 2 will be undefined. For example, for a single-input, single-output system, the matrix W_i will be singular for any $m < N-1$.

In conclusion, the authors have described a novel switching control scheme that is interesting and has the attraction of simplicity. However, as we have outlined in our discussion, there are several theoretical and practical issues with the scheme that require further investigation.

References

[1] A.G. Aghdam and E.J. Davison. An optimization algorithm for decentralized digital control of continuous time systems which account for inter-sample ripple. In *Proc. of the American Contr. Conf.*, pages 4273–4278, 2004.

[2] B.D.O. Anderson, T. Brinsmead, D. Liberzon, and A.S. Morse. Multiple model adaptive control with safe switching. *Int. J. Adapt. Contr. Signal*

Processing, 15:445–470, 2001.

[3] J. Boskovic and R. Mehra. A multiple model-based reconfigurable flight control system design. In *proceedings of the IEEE Conference on Decision and Control*, 1998.

[4] J. Boskovic and R. Mehra. Fault accommodation using model predictive methods. In *proceedings of American Control Conference*, 2002.

[5] A. Feuer and G.C. Goodwin. Generalized sample hold functions—frequency domain analysis of robustness, sensitivity, and intersample difficulties. *IEEE Trans. Automat. Contr.*, 39:1042–1047, 1994.

[6] Y.C. Juan and P. Kabamba. Optimal hold functions for sampled-data regulation. *Automatica*, 27(1):177–181, 1991.

[7] K. S. Narendra and J. Balakrishnan. Adaptive control using multiple models. 42(3):171–187, 1997.