Designing Dynamic Temporal Controls for Critical Systems *

Seonho Choi Ashok K. Agrawala
Institute for Advanced Computer Studies
Department of Computer Science
University of Maryland
College Park, MD 20742
{seonho, agrawala}@cs.umd.edu

Leyuan Shi
Department of Industrial Engineering
University of Wisconsin
Madison, WI 53706
leyuan@ie.engr.wisc.edu

May 28, 1997

Abstract

Traditional control systems have been designed to exercise control at regularly spaced time instants. When a discrete version of the system dynamics is used, a constant sampling interval is assumed and a new control value is calculated and exercised at each time instant. In this paper, we propose a new control scheme, *dynamic temporal control*, in which we not only calculate the control value but also dynamically decide the time instants when the new control computations have to be calculated. Taking a discrete, linear, time-invariant system, and a cost function which reflects a cost for computation of the control values, as an example, we show the feasibility of using this scheme. We implement the dynamic temporal control scheme in a rigid body satellite control example and demonstrate the significant reduction in cost. The scheme proposed here can be implemented using real-time operating system, such as *Maruti*, which schedules activities along the time axis. The reduced computations for control permit the use of the same processor for higher level functions resulting in a significant improvement in the performance of the overall system.

*This work is supported in part by ONR and ARPA under contract N66001-95-C-8619 to the Computer Science Department at the University of Maryland. The views, opinions, and/or findings contained in this report are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the Advanced Research Projects Agency, ONR or the U.S. Government.

1 Introduction

Control systems have been used for the control of dynamic systems by generating and exercising control signals. Traditional approach for feedback controls has been to define the control signals, $u(t)$, as a function of the current state of the system, $x(t)$. As the state of the system changes continuously the controls change continuously, i.e. they are defined as functions of time, t, such that time is treated as a continuous variable. When computers are used for implementing the control systems, due to the discrete nature of computations, time is treated as a discrete variable obtained by regularly spaced sampling of the time axis at Δ seconds. Many standard control formulations are defined for the discrete version of the system, with system dynamics expressed at discrete time instants. In these formulations the system dynamics and the control are expressed as sequences, $x(k)$ and $u(k)$.

Most of the traditional control systems were designed for dedicated controllers which had only one function, to accept the state values, $x(k)$ and generate the control, $u(k)$. However, when a general purpose computer is used as a controller, it has the capabilities, and may, therefore, be used for other functions. Thus, it may be desirable to take into account the cost of computations and consider control laws which do not compute the new value of the control at every instant. When no control is to be exercised, the computer may be used for other functions. In this paper we formulate such a control law and show how it can be used for control of systems, achieving the same degree of control as traditional control systems while reducing computation costs by changing the control at a few, specific time instants. We term this *dynamic temporal control*.

To the best of our knowledge this approach to the design and implementation of controls has not been studied in the past. However, taking computation time delay into consideration for real-time computer control has been studied in several research papers [1, 6, 7, 10, 12, 15]. But, all of these papers concentrated on examining computation time delay effects and compensating them while maintaining the assumption of exercising controls at regularly spaced time instants.

The basic idea of temporal control is to dynamically determine not only the values for u but also the time instants at which new controls are to be calculated. The control values are assumed to remain constant between changes. By doing so the designer has an additional degree of freedom for optimization. In this paper we present the idea and demonstrate its feasibility through an example using a discrete, linear, and time invariant system. Clearly, the same idea can be extended to continuous time as well as non-linear system.

In order to implement the dynamic temporal control scheme proposed here, the ability to carry out computations at dynamically decided time instants is required. Note that traditional real-time systems [14], which either operate as cyclic executives or manage resources based on static or dynamic priorities, may not be able to implement the temporal control scheme. But, operating systems such as Maruti [4, 11, 9, 8], which manage resources by using a dynamic time based scheduling scheme, can easily implement the scheme. Clearly, the reduction in overall computations for control results in the CPU being made available for other functions including higher level control and planning functions.

The paper is organized as follows. In Section 2, we formulate the dynamic temporal control problem and introduce computation cost into performance index function. The solution approach for dynamic temporal control scheme is discussed in Section 3. In Section 4, implementation issues are addressed. We provide an example of controlling rigid body satellite in Section 5 . In this example, a dynamic temporal controller is designed. Results show that the dynamic temporal control approach performs better than the traditional sampled data control approach with the same number of control exercises. Section 6 discusses the issues arising from the application of dynamic temporal controls to the design of real-time control systems. Finally, Section 7, we present our conclusion.

2 Problem Formulation

In dynamic temporal control, the control changing time instants are chosen such that a cost function is minimized which incorporates computational costs as well as state, input costs. We consider a steady state control problem on a finite time line $[0, T_f]$. To formulate the dynamic temporal control problem for a discrete, linear time-invariant system, we first discretize the time interval $[0, T_f]$ into M subintervals of length $\Delta = T_f/M$. Let $D_M = \{0, \Delta, 2\Delta, \ldots (M-1)\Delta\}$ denote M time instants that are regularly spaced. Here, control exercising time instants are restricted within D_M for the purpose of simplicity. The linear time-invariant controlled process is described by the difference equation:

$$x(k+1) \quad = \quad Ax(k) + Bu(k) \tag{1}$$

where k is the time index. One unit of time represents the subinterval Δ, whereas $x \in \mathcal{R}^n$ and $u \in \mathcal{R}^l$ are the state and input vectors, respectively.

It is well known that there exists a steady state optimal control law [5]

$$u^o(i) = f_i[x(i)] \qquad i = 0, 1, ..., M-1 \tag{2}$$

that minimizes the quadratic performance index function (Cost)

$$J_M = \sum_{k=0}^{M-1} [x^T(k)Qx(k) + u^T(k)Ru(k)] + x^T(M)Qx(M) \tag{3}$$

where $Q \in \mathcal{R}^{n\times n}$ is positive semi-definite and $R \in \mathcal{R}^{l\times l}$ is positive definite.

As we can see, traditional controller exercises control at every time instant in D. However, in temporal control, we are no longer constrained to exercise control at every time instant in D. In *dynamic temporal control* we require that the control be exercised with the following steps: At time t_i, $t_i \in D_M$ and $1 \le i$,

1. Compute a current state $x(t_i)$
2. Compute $\delta(t_i)$
3. Compute and apply $u(t_i)$ to the system
4. Repeat the process at $t_{i+1} = t_i + \delta(t_i)$

Note that t_i, $1 \le i$, denote control changing time instants, and $\delta(t_i)$ denotes the time interval between i-th control exercise and $(i+1)$-th control exercise.

For the purpose of simplicity, dual mode dynamic temporal control is considered. That is, $\delta(t_i)$ may take one of the following two values:

- $a\Delta$
- $b\Delta$

a and b are positive integers($a < b$) such that b is an integer multiple of a. Also, it is assumed that b divides M without any remainder. $b\Delta$ is called a *base sampling period* and $a\Delta$ is called a *rapid sampling period*. Let $M = \beta b$ where β is a positive integer.

In addition to the above assumption, we further assume that at all time instants in $\{0,\ b\Delta,\ 2b\Delta, \ldots, (\beta-1)b\Delta\}$ new controls are computed. Let each time interval $[(i-1)b\Delta, ib\Delta]$ of size $b\Delta$ be called a *frame* for $1 \le i \le \beta$. The sampling period decision function δ is evaluated at only time instants that are start times of frames, and once $\delta(ib\Delta)$ is decided it will be enforced during the next time frame $[ib\Delta, (i+1)b\Delta]$. In other words, if $\delta(ib\Delta) = a\Delta$ the control computations will be done at $ib\Delta$, $(ib+a)\Delta$, ..., $(ib+b-a)\Delta$. And, if $\delta(ib\Delta) = b\Delta$ the control computations will be done only at $ib\Delta$ in $[ib\Delta, (i+1)b\Delta)$. Under these assumptions the steps performed by a dynamic temporal controller can be summarized as follows: At time $ib\Delta$, $0 \le i \le \beta - 1$,

1. Compute a current state $x(ib\Delta)$
2. Compute $\delta(ib\Delta) = g_i(x(ib\Delta))$

   (a) If $\delta(ib\Delta) = a\Delta$

- At $t_j = (ib + ja)\Delta$ for $0 \leq j \leq (b/a - 1)$, compute and apply $u(t_j) = h_{i,j}(x(t_j))$

(b) If $\delta(ib\Delta) = b\Delta$

- compute and apply $u(ib\Delta) = h_i(x(ib\Delta))$

3. Repeat the process at $(i+1)b\Delta$ if $i < \beta - 1$.

This new formulation of dynamic temporal control makes it possible to find a good approximation approach to optimal control laws as can be seen in later sections of this paper.

We want to find a feedback control law, g_i, h_i, and $h_{i,j}$ for $i = 0, 1, 2, ..., \beta - 1$ and $j = 0, 1, \ldots, (b/a - 1)$, that minimizes a new performance index function

$$J_M' = J_M + \mu \cdot \gamma \tag{4}$$

Here, μ is the computation cost of exercising control with a rapid sampling period instead of a base sampling period in one frame, and γ denotes the number of frames in $[0, M\Delta]$ done with a rapid sampling period. Hence, exercising controls with a rapid sampling period increases the cost term $\mu \cdot \gamma$. So, if exercising control with a rapid sampling period doesn't reduce the term J_M by more than this increase, exercising control with a base sampling period is likely to be a better choice. This is a key idea of the solution approach given in the next section.

This new cost function is different from J_M in two aspects. First, the concept of computational cost is introduced in J_M' as $\mu \cdot \gamma$ term to regulate the number of frames with rapid sampling periods. If we do not take this computation cost into consideration γ is likely to become β. If computation cost is high (i.e., μ has a large value) then γ is likely to be small in order to minimize the total cost function. Second, in dynamic temporal control, not only do we seek control law $u(x(t))$, but also the control exercising time instants and the number of control changes. In the next section, we present in detail specific techniques for finding a dynamic temporal control law with performances close to optimal solutions.

3 Temporal Control with Fixed Control Changing Times

Let $\mathcal{T} = \{t_0, t_1, t_2, \ldots, t_{\nu-1}\}$ denote a set of control changing time instants where $t_0 = 0$, $t_1 = n_1\Delta$, ..., $t_{\nu-1} = n_{\nu-1}\Delta$. That is n_0, n_1, ..., $n_{\nu-1}$ are the indices for control changing time instants. In this section, an optimal control law is derived when $\mathcal{T}$ is given which minimizes the cost function J_M. In the next section, the results developed in this section will be used in devising good heuristics for deciding δ values minimizing J_M'.

Assume that $\mathcal{T}$ is given. Then a new control input calculated at t_i will be applied to the actuator for the next time interval from t_i to t_{i+1}. Our objective here is to determine the optimal control law

$$u^o(n_i) = f_i[x(n_i)] \;\; i = 0, 1, ..., \nu - 1 \tag{5}$$

that minimizes the quadratic performance index function (Cost) J_M which is defined in (4).

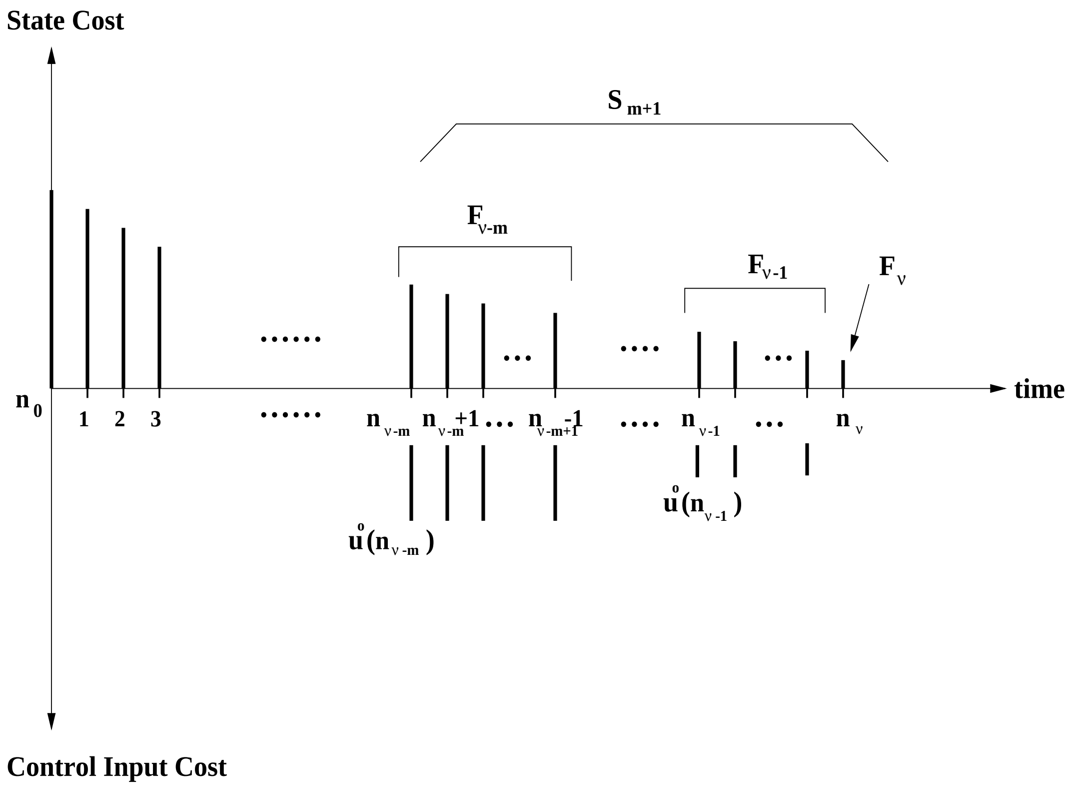

Figure 1: Decomposition of J_M into F_i.

The principle of optimality, developed by Richard Bellman[2, 3] is the approach used here. That is, if a closed loop control $u^o(n_i) = f_i[x(n_i)]$ is optimal over the interval $t_0 \leq t \leq t_\nu$, then it is also optimal over any sub-interval $t_m \leq t \leq t_\nu$, where $0 \leq m \leq \nu$. As it can be seen from Figure 1, the total cost J_M can be decomposed into F_is for $0 \leq i \leq \nu$ where

$$\begin{aligned} F_i &= x^T(n_i)Qx(n_i) + x^T(n_i+1)Qx(n_i+1) \\ &+ x^T(n_i+2)Qx(n_i+2) + ... + x^T(n_{i+1}-1)Qx(n_{i+1}-1) \\ &+ (n_{i+1}-n_i)u^T(n_i)Ru(n_i) \end{aligned} \tag{6}$$

That is, from (1),

$$F_i = x^T(n_i)Qx(n_i) + (Ax(n_i) + Bu(n_i))^T Q(Ax(n_i) + Bu(n_i)) \tag{7}$$

$$
\begin{aligned}
&+ \quad (A^2x(n_i) + ABu(n_i) + Bu(n_i))^T Q (A^2x(n_i) + ABu(n_i) + Bu(n_i)) \\
&+ \quad ... + (A^{n_{i+1}-n_i-1}x(n_i) + A^{n_{i+1}-n_i-2}Bu(n_i) + ... + ABu(n_i) + Bu(n_i))^T Q \\
&\quad\quad (A^{n_{i+1}-n_i-1}x(n_i) + A^{n_{i+1}-n_i-2}Bu(n_i) + ... + ABu(n_i) + Bu(n_i)) \\
&+ \quad (n_{i+1} - n_i)u^T(n_i)Ru(n_i)
\end{aligned}
$$

This can be rewritten as

$$
\begin{aligned}
F_i &= x^T(n_i)Qx(n_i) + \sum_{j=1}^{n_{i+1}-n_i-1} [A_j x(n_i) + B_j u(n_i)]^T Q [A_j x(n_i) + B_j u(n_i)] \\
&+ (n_{i+1} - n_i)u^T(n_i)Ru(n_i)
\end{aligned} \tag{8}
$$

where $A_j = A^j$ and $B_j = \sum_{k=0}^{j-1} A^k B$.

Then J_M can be expressed as

$$
J_M = F_0 + F_1 + F_2 + ... + F_\nu. \tag{9}
$$

Let S_m be the cost from $i = \nu - m + 1$ to $i = \nu$:

$$
S_m = F_{\nu-m+1} + F_{\nu-m+2} + ... + F_{\nu-1} + F_\nu, \quad 1 \leq m \leq \nu + 1. \tag{10}
$$

These cost terms are well illustrated in the above Figure 1.

Therefore, by applying the principle of optimality, we can first minimize $S_1 = F_\nu$, then choose $F_{\nu-1}$ to minimize $S_2 = F_{\nu-1} + F_\nu = S_1^o + F_{\nu-1}$ where S_1^o is the optimal cost occurred at t_ν. We can continue choosing $F_{\nu-2}$ to minimize $S_3 = F_{\nu-2} + F_{\nu-1} + F_\nu = F_{\nu-2} + S_2^o$ and so on until $S_{\nu+1} = J_M$ is minimized. Note that $S_1 = F_\nu = x^T(n_\nu)Qx(n_\nu)$ is determined only from $x(n_\nu)$ which is independent of any other control inputs.

3.1 Inductive Construction of an Optimal Control Law with $\mathcal{T}$ Given

We inductively derive an optimal controller which changes its control at ν time instants $t_0, t_1, \ldots, t_{\nu-1}$. As we showed in the previous section, the inductive procedure goes backwards in time from S_1^o to $S_{\nu+1}^o$. Since $S_1 = F_\nu = x^T(n_\nu)Qx(n_\nu) + u^T(n_\nu)Ru(n_\nu)$ and $x(n_\nu)$ is independent of $u(n_\nu)$, we can let $u^o(n_\nu) = u^o(M) = 0$ and $S_1^o = x^T(n_\nu)Qx(n_\nu)$ where Q is symmetric and positive semi-definite.

Induction Basis: $S_1^o = x^T(n_\nu)Qx(n_\nu)$ where Q is symmetric.

Inductive Assumption: Suppose that

$$S_m^o = x^T(n_{\nu-m+1})P(\nu - m + 1)x(n_{\nu-m+1})$$

holds for some m where $1 \leq m \leq \nu$ and $P(\nu - m + 1)$ is symmetric.

We can write S_m^o as

$$\begin{aligned} S_m^o &= [A_{(n_{\nu-m+1}-n_{\nu-m})}x(n_{\nu-m}) + B_{(n_{\nu-m+1}-n_{\nu-m})}u(n_{\nu-m})]^T P(\nu - m + 1) \\ &\quad [A_{(n_{\nu-m+1}-n_{\nu-m})}x(n_{\nu-m}) + B_{(n_{\nu-m+1}-n_{\nu-m})}u(n_{\nu-m})] \end{aligned} \tag{11}$$

From the definition of S_m and (8),

$$\begin{aligned} S_{m+1} &= S_m^o + F_{\nu-m} \\ &= S_m^o + x^T(n_{\nu-m})Qx(n_{\nu-m}) \\ &+ \sum_{j=1}^{n_{\nu-m+1}-n_{\nu-m}-1} [A_j x(n_{\nu-m}) + B_j u(n_{\nu-m})]^T Q[A_j x(n_{\nu-m}) + B_j u(n_{\nu-m})] \\ &+ (n_{\nu-m+1} - n_{\nu-m})u^T(n_{\nu-m})Ru(n_{\nu-m}) \end{aligned} \tag{12}$$

And the above equation becomes

$$\begin{aligned} S_{m+1} &= [A_{n_{\nu-m+1}-n_{\nu-m}}x(n_{\nu-m}) + B_{n_{\nu-m+1}-n_{\nu-m}}u(n_{\nu-m})]^T P(\nu - m + 1) \\ &\quad [A_{n_{\nu-m+1}-n_{\nu-m}}x(n_{\nu-m}) + B_{n_{\nu-m+1}-n_{\nu-m}}u(n_{\nu-m})] \\ &+ x^T(n_{\nu-m})Qx(n_{\nu-m}) \\ &+ \sum_{j=1}^{n_{\nu-m+1}-n_{\nu-m}-1} [A_j x(n_{\nu-m}) + B_j u(n_{\nu-m})]^T Q[A_j x(n_{\nu-m}) + B_j u(n_{\nu-m})] \\ &+ (n_{\nu-m+1} - n_{\nu-m})u^T(n_{\nu-m})Ru(n_{\nu-m}) \end{aligned} \tag{13}$$

If we differentiate S_{m+1} with respect to $u(n_{\nu-m})$, then

$$\begin{aligned} \frac{\partial S_{m+1}}{\partial u(n_{\nu-m})} &= B^T_{n_{\nu-m+1}-n_{\nu-m}} P(\nu - m + 1)A_{n_{\nu-m+1}-n_{\nu-m}}x(n_{\nu-m}) \\ &+ (A^T_{n_{\nu-m+1}-n_{\nu-m}} P(\nu - m + 1)B_{n_{\nu-m+1}-n_{\nu-m}})^T x(n_{\nu-m}) \\ &+ 2B^T_{n_{\nu-m+1}-n_{\nu-m}} P(\nu - m + 1)B_{n_{\nu-m+1}-n_{\nu-m}}u(n_{\nu-m}) \\ &+ \sum_{j=1}^{n_{\nu-m+1}-n_{\nu-m}-1} [2B_j^T QA_j x(n_{\nu-m}) + 2B_j^T QB_j u(n_{\nu-m})] \\ &+ 2(n_{\nu-m+1} - n_{\nu-m})Ru(n_{\nu-m}) \end{aligned} \tag{14}$$

$$= \quad 2\{B^T_{n_{\nu-m+1}-n_{\nu-m}} P(\nu - m + 1)A_{n_{\nu-m+1}-n_{\nu-m}} \tag{15}$$

$$
\begin{aligned}
&+ \sum_{j=1}^{n_{\nu-m+1}-n_{\nu-m}-1} B_j^T Q A_j\} x(n_{\nu-m}) \\
+ \quad & 2\{B^T_{n_{\nu-m+1}-n_{\nu-m}} P(\nu-m+1) B_{n_{\nu-m+1}-n_{\nu-m}} \\
&+ \sum_{j=1}^{n_{\nu-m+1}-n_{\nu-m}-1} B_j^T Q B_j + (n_{\nu-m+1}-n_{\nu-m})R\} u(n_{\nu-m})
\end{aligned}
$$

Note that $P(\nu - m + 1)$ is symmetric and the following three rules are applied to differentiate S_{m+1} above.

$$
\begin{aligned}
\frac{\partial}{\partial x}(x^T Q x) &= 2Qx \\
\frac{\partial}{\partial x}(x^T Q y) &= Qy \\
\frac{\partial}{\partial y}(x^T Q y) &= Q^T x
\end{aligned}
$$

Let $\frac{\partial S_{m+1}}{\partial u(n_{\nu-m})} = 0$, from Lemma 1 and Lemma 2 given later we can obtain $u^o(n_{\nu-m})$ which minimizes S_{m+1} and thus obtain S^o_{m+1}.

$$
\begin{aligned}
u^o(n_{\nu-m}) &= -\{B^T_{n_{\nu-m+1}-n_{\nu-m}} P(\nu-m+1) B_{n_{\nu-m+1}-n_{\nu-m}} \\
&\quad + \sum_{j=1}^{n_{\nu-m+1}-n_{\nu-m}-1} B_j^T Q B_j + (n_{\nu-m+1}-n_{\nu-m})R\}^{-1} \\
&\quad \{B^T_{n_{\nu-m+1}-n_{\nu-m}} P(\nu-m+1) A_{n_{\nu-m+1}-n_{\nu-m}} + \sum_{j=1}^{n_{\nu-m+1}-n_{\nu-m}-1} B_j^T Q A_j\} x(n_{\nu-m}) \\
&= -K(\nu-m) x(n_{\nu-m})
\end{aligned} \tag{16}
$$

where $K(\nu - m)$ is defined in (16).

Therefore, we can write

$$
\begin{aligned}
A_{n_{\nu-m+1}-n_{\nu-m}} x(n_{\nu-m}) \quad + \quad & B_{n_{\nu-m+1}-n_{\nu-m}} u^o(n_{\nu-m}) = \\
& [A_{n_{\nu-m+1}-n_{\nu-m}} - B_{n_{\nu-m+1}-n_{\nu-m}} K(\nu-m)] x(n_{\nu-m})
\end{aligned} \tag{17}
$$

If we use (16) and (17), we have

$$
\begin{aligned}
S^o_{m+1} &= \{[A_{n_{\nu-m+1}-n_{\nu-m}} - B_{n_{\nu-m+1}-n_{\nu-m}} K(\nu-m)] x(n_{\nu-m})\}^T P(\nu-m+1) \\
&\quad \{[A_{n_{\nu-m+1}-n_{\nu-m}} - B_{n_{\nu-m+1}-n_{\nu-m}} K(\nu-m)] x(n_{\nu-m})\} \\
&+ x^T(n_{\nu-m}) Q x(n_{\nu-m}) \\
&+ \sum_{j=1}^{n_{\nu-m+1}-n_{\nu-m}-1} \{[A_j - B_j K(\nu-m)] x(n_{\nu-m})\}^T Q \{[A_j - B_j K(\nu-m)] x(n_{\nu-m})\} \\
&+ (n_{\nu-m+1}-n_{\nu-m}) [K(\nu-m) x(n_{\nu-m})]^T R [K(\nu-m) x(n_{\nu-m})]
\end{aligned} \tag{18}
$$

This equation can be rewritten as

$$
\begin{aligned}
S^o_{m+1} &= x^T(n_{\nu-m})\{[A_{n_{\nu-m+1}-n_{\nu-m}} - B_{n_{\nu-m+1}-n_{\nu-m}}K(\nu-m)]^T P(\nu-m+1) \\
& \quad [A_{n_{\nu-m+1}-n_{\nu-m}} - B_{n_{\nu-m+1}-n_{\nu-m}}K(\nu-m)] \\
& + Q \\
& + \sum_{j=1}^{n_{\nu-m+1}-n_{\nu-m}-1} [A_j - B_jK(\nu-m)]^T Q [A_j - B_jK(\nu-m)] \\
& + (n_{\nu-m+1}-n_{\nu-m})K^T(n_{\nu-m})RK(\nu-m)\}x(n_{\nu-m}). \\
&= x^T(n_{\nu-m})P(\nu-m)x(n_{\nu-m})
\end{aligned} \tag{19}
$$

where $P(\nu-m)$ is obtained from $K(\nu-m)$ and $P(\nu-m+1)$ as in (19). Also note that knowing $P(\nu-m+1)$ is enough to compute $K(\nu-m)$ because other terms of (16) are known a priori.

Therefore, we find a symmetric matrix $P(\nu-m)$ satisfying $S^o_{m+1} = x^T(n_{\nu-m})P(\nu-m)x(n_{\nu-m})$. From (16) and (19), we have the following recursive equations for obtaining $P(\nu-m)$ from $P(\nu-m+1)$ where $m = 1, 2, ..., \nu$.

$$
\begin{aligned}
K(\nu-m) &= \{B^T_{n_{\nu-m+1}-n_{\nu-m}} P(\nu-m+1) B_{n_{\nu-m+1}-n_{\nu-m}} \\
& + \sum_{j=1}^{n_{\nu-m+1}-n_{\nu-m}-1} B_j^T Q B_j + (n_{\nu-m+1}-n_{\nu-m})R\}^{-1} \\
& \quad \{B^T_{n_{\nu-m+1}-n_{\nu-m}} P(\nu-m+1) A_{n_{\nu-m+1}-n_{\nu-m}} + \sum_{j=1}^{n_{\nu-m+1}-n_{\nu-m}-1} B_j^T Q A_j\}
\end{aligned} \tag{20}
$$

$$
\begin{aligned}
P(\nu-m) &= [A_{n_{\nu-m+1}-n_{\nu-m}} - B_{n_{\nu-m+1}-n_{\nu-m}}K(\nu-m)]^T P(\nu-m+1) \\
& \quad [A_{n_{\nu-m+1}-n_{\nu-m}} - B_{n_{\nu-m+1}-n_{\nu-m}}K(\nu-m)] \\
& + Q \\
& + \sum_{j=1}^{n_{\nu-m+1}-n_{\nu-m}-1} [A_j - B_jK(\nu-m)]^T Q [A_j - B_jK(\nu-m)] \\
& + (n_{\nu-m+1}-n_{\nu-m})K^T(\nu-m)RK(\nu-m)
\end{aligned} \tag{21}
$$

Also, we know that at each time instant $n_{\nu-m}\Delta$

$$
u^o(n_{\nu-m}) = -K(\nu-m)x(n_{\nu-m}) \tag{22}
$$

Hence, with $P(\nu) = Q$, we can obtain $K(i)$ and $P(i)$ for $i = \nu - 1, \nu - 2, ..., 0$ recursively using (20) and (21). At each time instant $n_i\Delta,\ i = 0, 1, 2, ..., \nu - 1$ the new control input value will be obtained using (22) by multiplying $K(i)$ by $x(n_i)$ where $x(n_i)$ is the estimate of the system state at $n_i\Delta$. Also, note that the optimal control cost is $J_M^o = S_{\nu+1}^o = x^T(0)P(0)x(0)$ where $P(0)$ is found from the above procedure.

To prove the optimality of this control law we need the following lemmas.

Lemma 1 *If Q is positive semi-definite and R is positive definite, then $P(i),\ i = \nu, \nu-1, \nu-2, ..., 0$, matrices are positive semi-definite. Hence, $P(i)$s are symmetric from the definition of a positive semi-definite matrix.*

Proof Since $P(\nu) = Q$, from assumption $P(\nu)$ is positive semi-definite. Assume that for $k = i + 1$, $P(k)$ is positive semi-definite. We use induction to prove that $P(i)$ is semi-definite. Note that Q is positive semi-definite and R is positive definite. From (21) we have

$$\begin{aligned} P(i) &= [A_{n_{i+1}-n_i} - B_{n_{i+1}-n_i}K(i)]^T P(i+1) \\ &\quad [A_{n_{i+1}-n_i} - B_{n_{i+1}-n_i}K(i)] \\ &+ Q \\ &+ \sum_{j=1}^{n_{i+1}-n_i-1} [A_j - B_jK(i)]^T Q [A_j - B_jK(i)] \\ &+ (n_{i+1} - n_i)K^T(i)RK(i) \end{aligned} \tag{23}$$

Since $P(i + 1)$ and Q are positive semi-definite, R is positive definite, and $(n_{i+1} - n_i) > 0$, it is easy to verify that for $\forall y \in R^m\ :\ y^T P(i)y \geq 0$. This means that $P(i)$ is positive semi-definite. This inductive procedure proves the lemma.

Lemma 2 *Given $\mathcal{T}$, the inverse matrix in (20) always exists.*

Proof Let $V = B_{n_{\nu-m+1}-n_{\nu-m}}^T P(\nu - m + 1)B_{n_{\nu-m+1}-n_{\nu-m}} + \sum_{j=1}^{n_{\nu-m+1}-n_{\nu-m}-1} B_j^T Q B_j + (n_{\nu-m+1} - n_{\nu-m})R$. From Lemma 1, $P(\nu - m + 1)$ is positive semi-definite. Therefore, $\forall y \in R^m$: $y^T V y > 0$ because Q is positive semi-definite, R is positive definite and $n_{\nu-m+1} - n_{\nu-m} > 0$. This implies that V is positive definite. Hence the inverse matrix exists.

Theorem 1 *Given $\mathcal{T}$, $K(i)$ $(i = 0, 1, 2, ..., \nu - 1)$ obtained from the above procedure are the optimal feedback gains which minimize the cost function J_M (and J_M') on $[0, M\Delta]$.*

Proof Note that given $\mathcal{T}$, J_M is a convex function of $u(n_i), i = 0, 1, ..., \nu - 1$. Thus the above feedback control law is optimal.

Suppose that $\mathcal{T}_1$ and $\mathcal{T}_2$ denote two sets of control changing time instants.

Lemma 3 *If $\mathcal{T}_1 \subseteq \mathcal{T}_2$, then $J^o_{M,1} \geq J^o_{M,2}$ where $J^o_{M,1}$ and $J^o_{M,2}$ are the optimal costs of controls which change controls at time instants in $\mathcal{T}_1$ and $\mathcal{T}_2$ respectively.*

Proof Suppose that $J^o_{M,1} < J^o_{M,2}$, then, in controlling the system with $\mathcal{T}_2$, if we do not change controls at time instants in $\mathcal{T}_2 - \mathcal{T}_1$ and change controls at time instants in $\mathcal{T}_1$ to the same control inputs that were exercised to get $J^o_{M,1}$ with $\mathcal{T}_1$, we obtain $\hat{J}_{M,2}$ which is equal to $J^o_{M,1}$. This contradicts the fact that $J^o_{M,2}$ is the minimum cost obtainable with D_q since we have found $\hat{J}_{M,2}$ which is equal to $J^o_{M,1}$ and therefore less than $J^o_{M,2}$. Hence, $J^o_{M,1} \geq J^o_{M,2}$.

This lemma implies that if we do not take computation cost, μ, into consideration, then the more control exercising points, the better the controller is (less cost). With the computation cost being included in the cost function, the statement above is no longer true. Therefore we need to search for an optimal $\mathcal{T}$ which minimizes the cost function J'_M. The following sections provide a detailed discussion on searching for such an optimal solution. Note that if we let $\mathcal{T} = D_M$ then the optimal temporal control law is the same as the traditional linear feedback optimal control law.

3.2 Dynamic Temporal Control

In this section, we design a dynamic temporal controller by introducing a heuristic for $\delta(ib\Delta)$ function. The heuristic tries to estimate how much performance gain(reduction of J_M term in J'_M) and how much performance loss (increase of $\mu\gamma$ term) will incur if a rapid sampling period is used in the next frame. If the performance gain is greater than or equal to a given threshold θ, then $\delta(i) = a\Delta$, otherwise $\delta(i) = b\Delta$.

By making use of the results developed in the previous section, we can obtain an optimal control law for $\mathcal{T}^1_i = \{ib\Delta, (i+1)b\Delta, \ldots, (\beta-1)b\Delta\}$ on a time interval $[ib\Delta, \beta b\Delta]$ where $0 \leq i \leq \beta - 1$. Let $\mathcal{K}_1(i)$ and $\mathcal{P}_1(i)$ denote two matrices found from $\mathcal{T}^1_i$ by applying the algorithm given in the previous section.

Consider another control changing time instants set $\mathcal{T}^2_i = \{ib\Delta, (ib+a)\Delta, \ldots, (ib+b-a)\Delta, (i+1)b\Delta, \ldots, (\beta-1)b\Delta\}$ where $0 \leq i \leq \beta - 1$. Also, let $\mathcal{K}_2(i)$ and $\mathcal{P}_2(i)$ denote two matrices found from $\mathcal{T}_i^2$ by applying the algorithm given in the previous section. Also, let $\mathcal{K}_2(i,j), 0 \le j \le (b/a-1)$, denote a gain matrix obtained for time instant $(ib+ja)\Delta$.

Two control changing time sets, $\mathcal{T}_i^1$ and $\mathcal{T}_i^2$, are depicted in Figure 2.

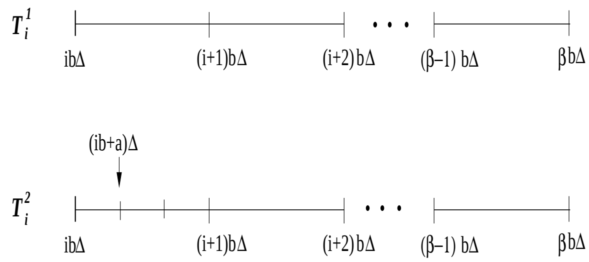

Figure 2: Two control changing time sets $\mathcal{T}_i^1$ and $\mathcal{T}_i^2$.

From Lemma 3 we know that $x^T(ib\Delta)\mathcal{P}_1(i)x(ib\Delta)$ is less than or equal to $x^T(ib\Delta)\mathcal{P}_2(i)x(ib\Delta)$. Furthermore, $x^T(ib\Delta)\mathcal{P}_1(i)x(ib\Delta)$ is less than or equal to $x^T(ib\Delta)\mathcal{P}_a(i)x(ib\Delta)$ where $\mathcal{P}_a(i)$ is a matrix found from any arbitrary control changing time instant set on $[ib\Delta, \beta b\Delta]$ conforming to the assumptions given in the problem formulation section, i.e., the same sampling period is enforced during one frame.

In addition, the cost $x^T(ib\Delta)\mathcal{P}_2(i)x(ib\Delta)$ is less than or equal to $x^T(ib\Delta)\mathcal{P}_b(i)x(ib\Delta)$ where $\mathcal{P}_b(i)$ is a matrix found from any arbitrary control changing time instant set on $[ib\Delta, \beta b\Delta]$ that contains time instants $ib\Delta, (ib+a)\Delta, \ldots, (ib+b-a)\Delta$, i.e., a rapid sampling period is used in the first frame $[ib\Delta, (i+1)b\Delta]$.

From these facts, it can be said that a cost $x^T(ib\Delta)\mathcal{P}_1(i)x(ib\Delta)$ is a lower bound of the costs found from any control changing time instant sets on $[ib\Delta, \beta b\Delta]$ that conform to the assumptions, and a cost $x^T(ib\Delta)\mathcal{P}_2(i)x(ib\Delta)$ is a lower bound of the costs found from any control changing time instant sets that enforce rapid sampling period in the first frame $[ib\Delta, (i+1)b\Delta)$.

In our solution approach, the above costs are used at time $ib\Delta$ to estimate the performance gain of using a rapid sampling period in the next frame $[ib\Delta, (i+1)b\Delta]$. This is a heuristic approach, and the effectiveness of this approach is validated through an example in a later section.

We present a heuristic dynamic temporal control law which performs the following steps at each frame start time:

1. Compute a current state $x(ib\Delta)$
2. If $x^T(ib\Delta)(\mathcal{P}_1(i) - \mathcal{P}_2(i))x(ib\Delta) < \theta$, let $\delta(i) = b\Delta$.
   Otherwise, let $\delta(i) = a\Delta$.
   (a) If $\delta(i) = a\Delta$,

- At each time instant $t_j = ib\Delta + ja\Delta$, $0 \le j \le (b/a - 1)$, apply $u(t_j) = -\mathcal{K}_2(i,j)x(t_j)$

(b) If $\delta(i) = b\Delta$,

- $u(ib\Delta) = -\mathcal{K}_1(i)x(ib\Delta)$

3. Repeat the process at $(i+1)(b\Delta)$

The following theorem proves that the dynamic temporal control using the above control law guarantees the cost term J_M of $J_M^{'}$ to be less than or equal to $x^T(0)\mathcal{P}_1(0)x(0)$ which is a cost for $\mathcal{T}_0^1$ with only a base sampling period enforced on the entire interval $[0, T_f]$.

Theorem 2 *If the above dynamic temporal control law is used, the cost J_M of $J_M^{'}$ is less than or equal to $x^T(0)\mathcal{P}_1(0)x(0)$ where $\mathcal{P}_1(0)$ is obtained from $\mathcal{T}_0^1$.*

Proof Suppose that $\mathcal{T}^d(x_0)$ denotes a set of time instants at which new controls are exercised according to the above dynamic temporal control law for a given initial state x_0. Let $I^d(x_0) = \{i \mid 1 \le i \le \beta\}$ denote a set of frame indices at which a rapid sampling period is used. Also, let $i_1 \in I^d(x_0)$ denote a smallest index in $I^d(x_0)$, and $i_2 \in I^d(x_0)$ denote a second smallest index, and so on. Consider two control changing time sets, $\mathcal{T}_0^1$ and $\mathcal{T}_0^{'}$, where in $\mathcal{T}_0^{'}$ only i_1-th frame uses a rapid sampling period. $\mathcal{T}_0^1$ is a set of control changing time instants shown in Figure 2. Also, suppose that for these two control changing time sets, $\mathcal{K}_1(l)$ is used if l-th frame uses a base sampling period, and $\mathcal{K}_2(l,j)$ is used if l-th frame uses a rapid sampling period. Under these assumptions, it is clear that the control cost (without computation cost) for $\mathcal{T}_0^1$ is greater than or equal to that for $\mathcal{T}_0^{'}$, when the same initial state x_0 is used. This is clear from Lemma 3.

Consider two control changing time sets, $\mathcal{T}_0^{'}$ and $\mathcal{T}_0^{''}$, where in $\mathcal{T}_0^{''}$ i_1-th and i_2-th frames use a rapid sampling period. Also, suppose that for these two control changing time sets, $\mathcal{K}_1(l)$ is used if l-th frame uses a base sampling period, and $\mathcal{K}_2(l,j)$ is used if l-th frame uses a rapid sampling period. Under these assumptions, it is clear that the control cost (without computation cost) for $\mathcal{T}_0^{'}$ is greater than or equal to that for $\mathcal{T}_0^{''}$, when the same initial state x_0 is used.

If we transitively and inductively apply this process, we can conclude that, for the same initial state x_0, the control cost (without computation cost) for $\mathcal{T}_0^1$ is greater than or equal to that obtained by applying the dynamic temporal control law. This proves the theorem.

4 Implementation

To implement dynamic temporal control, we need to calculate and store $\mathcal{K}_1(i)$ and $\mathcal{K}_2(i,j)$ matrices, and use them when controlling the system. The number of matrices that need to be stored is

$O(\beta + (b/a)\beta)$, which is $O((b/a)\beta)$. Note that in traditional optimal linear control a similar matrix is obtained and used at every time instant in D_M to generate control input value.

In dynamic temporal control, there is a CPU time overhead for calculating $x^T(ib\Delta)(\mathcal{P}_1(i) - \mathcal{P}_2(i))x(ib\Delta)$ at the start of each frame. This calculation can be done within $O(n^2)$ time. This calculation has to be done once each frame. More discussion is presented in a discussion section on this overhead.

In order to implement temporal control we require an operating system that supports scheduling control computations at specific time instants, and allows dynamic selection of sampling periods. The Maruti system developed at the University of Maryland is a suitable host for the implementation of dynamic temporal control [11, 9, 8]. In Maruti, all executions are scheduled in time and the time of execution can be modified dynamically, if so desired. This is in contrast with traditional cyclic executives often used in real-time systems, which have a fixed, cyclic operation and which are well suited only for the sampled data control systems operating in a static environment. It is the availability of the system such as Maruti that allows us to consider the notion of dynamic temporal control, in which time becomes an emergent property of the system.

5 Example

To illustrate the advantages of a dynamic temporal control scheme let us consider a simple example of rigid body satellite control problem [13]. The system state equations are as follows:

$$
\begin{aligned}
x(k+1) &= \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 0.00125 \end{bmatrix} u(k) \\
y(k) &= \begin{bmatrix} 1 & 1 \end{bmatrix} x(k)
\end{aligned}
$$

where k represents the time index and one unit of time is the discretized subinterval of length $\Delta = 0.05$. The linear quadratic performance index J_M in (4) is used here with the following parameters.

$$
\begin{aligned}
Q &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\
R &= 0.0001 \\
M &= 40 \\
\Delta &= 0.05 \\
a &= 1
\end{aligned}
$$

$$b = 4 \tag{24}$$

The objective of the control is to drive the satellite to the zero position and the desired goal state is $x_f = [0,\ \ 0]^T$.

We applied the dynamic temporal control law with an initial state space $\{(x_1, x_2) \mid \ 0.2 \leq x_1, x_2 \leq 0.8\}$ with the following parameter:

$$\theta = 0.01 \tag{25}$$

The performance of the dynamic temporal controller is compared to that of traditional optimal control with a sampling period 0.05. In Figure 3 the cost differences between dynamic temporal controller and a traditional optimal controller are depicted for each initial state (x_1, x_2). Note that the maximum cost difference is less than 0.03. In Figure 4 the number of control computation performed by a dynamic temporal controller is shown for each initial state. Note that the maximum number of control computation is less than 20, and for many of initial states they are less than 18.

To estimate how much cost reduction is achieved through dynamic temporal control, we compare its performance with that of traditional optimal controller with 0.1 sampling period, i.e., sampling is done at 20 regular spaced time instants. In Figure 5 the cost differences between optimal controller with 0.05 sampling period and an optimal controller with 0.1 sampling period are depicted for each initial state (x_1, x_2). Note that the maximum cost difference is almost 0.5. The cost differences shown in Figure 3 and Figure 5 are compared together in Figure 6. Note that with almost all initial states the dynamic temporal controller outperforms traditional optimal controller with sampling period 0.1, even though the number of control computations done by a dynamic temporal controller is smaller than that for optimal controller.

If we normalize the costs from dynamic temporal controller and from optimal controller with sampling period 0.1 by dividing by the cost from optimal controller with sampling period 0.05, we obtain graphs shown in Figure 7.

The Figure 7 shows two graphs, one for normalized costs from dynamic temporal controller and the other for normalized costs from optimal controller with a sampling period 0.1. Note that for some initial states the optimal controller outperforms dynamic temporal controller. However, this is from using uniform threshold value θ for the entire initial state space. As a result of using one threshold value, the number of control computations over initial state space shows non-uniformity as can be seen in Figure 4. By adjusting threshold values for some initial state, we can obtain more uniform graphs. This is seen from Figure 8 which is found after using different(smaller) threshold values for the initial states that results in higher normalized costs in Figure 7.

The differences between normalized costs shown in Figure 8 is not so big, less than 0.01. However, the advantage of dynamic temporal scheme is more clearly seen from the following experiment.

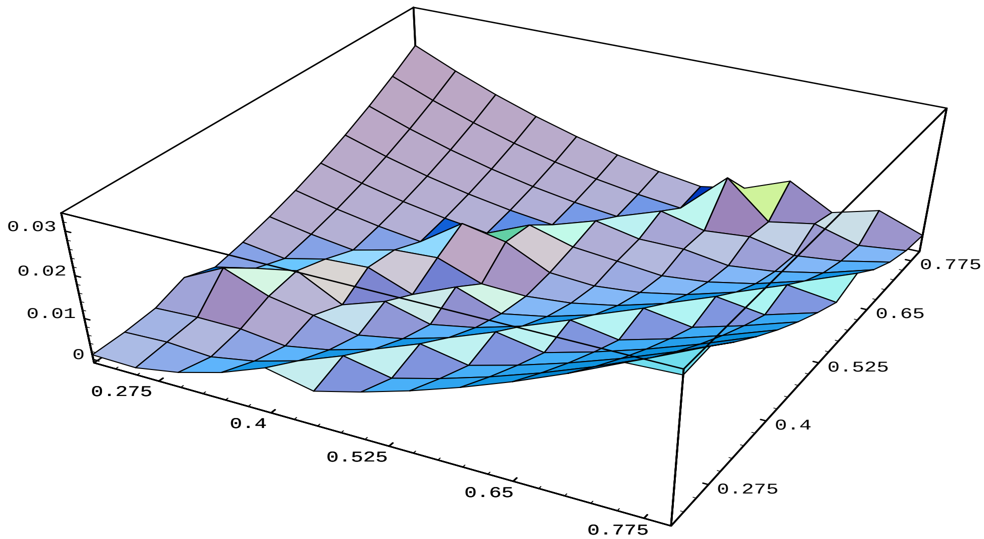

Figure 3: Cost differences between dynamic temporal controller and traditional controller with 0.05 sampling period.

Usually, in concurrent real-time systems, the actual control update time instants for one periodic control task varies in consecutive periods. This is from the variations of task execution times and also from the resource contention between different tasks. The delay of control update from the ideal control updating time instant is called *computational delay*. Computational delay has an adverse effect on control algorithm's performance. Figure 9 shows the differences of worst case normalized costs between a dynamic temporal control law with $\theta = 0.01$ and an optimal controller with a sampling period 0.1. The computational delays are randomly generated with a normal distribution in $[0, a\Delta]$, and they are injected into the simulation. For each initial state, the control trajectories are found 100 times, and the maximum cost among them is recorded. The figure shows that the benefits we can get by using dynamic temporal controller is bigger in the presence of computational

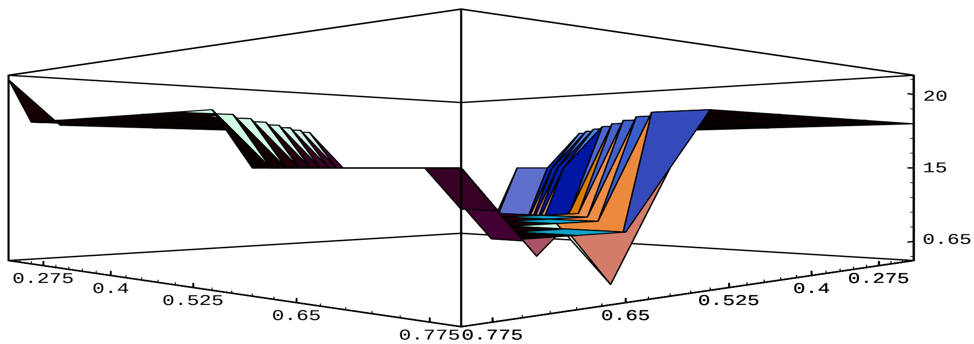

Figure 4: Number of control changes.

delays.

6 Discussion

In the previous section, we showed by using an example that the number of control computations can be dramatically reduced by using dynamic temporal control law, while not sacrificing the quality of control. Employing the dynamic temporal control methodology in concurrent real-time embedded systems will have a significant impact on the way computational resources are utilized by control tasks. A minimal amount of control computations can be obtained for a given regulator by which we can achieve almost the same control performance compared to that of traditional controller with equal sampling period. This significantly reduces the CPU times for each controlling task and thus increases the number of real-time control functions which can be accommodated concurrently in

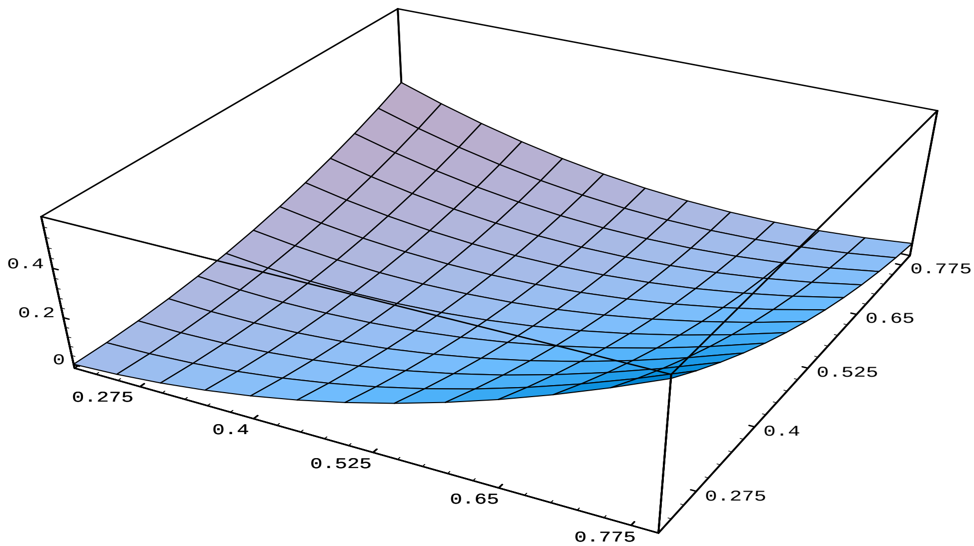

Figure 5: Cost differences between two traditional controllers.

one embedded system. Particularly, in a hierarchical control system if dynamic temporal controllers can be employed for lower level controllers the higher level controllers will have a great degree of flexibility in managing resource usages by adjusting computational requirements of each lower level controller. For example, in emergency situations the higher level controller may force the lower level controller to run as infrequently as they possibly can (thus freeing computational resources for handling the emergency). In contrast, during normal operations the temporal control tasks may run as necessary, and the additional computation time can be used for higher level functions such as monitoring and planning, etc.

As is mentioned in Section 4, there is an associated CPU overhead with dynamic temporal controller. At start of each frame the sampling period decision has to be done, which requires $O(n^2)$ execution time. However, n is generally not big. Moreover, this computation is required

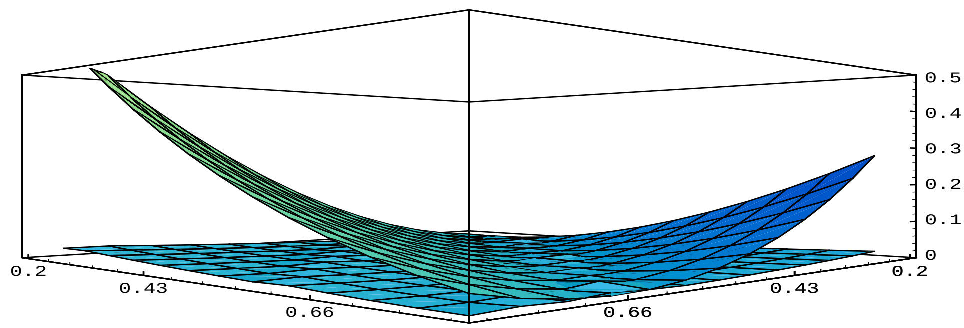

Figure 6: Comparison of costs

once every frame, and we can get benefits by reducing the number of context switches in concurrent real-time systems. Even if this overhead is large, then we can construct tables for this purpose, and they can be used at run-time.

More work needs to be done on the effects of computational delays(or jitters) on control systems performance when dynamic temporal controls are used in concurrent real-time systems.

7 Conclusion

In this paper we proposed a *dynamic temporal control* technique based on a new cost function which takes into account computational cost as well as state and input cost. In this scheme new control input values are defined at time instants which are not necessarily regularly spaced. For the linear control problem we showed that almost the same quality of control can be achieved while much less

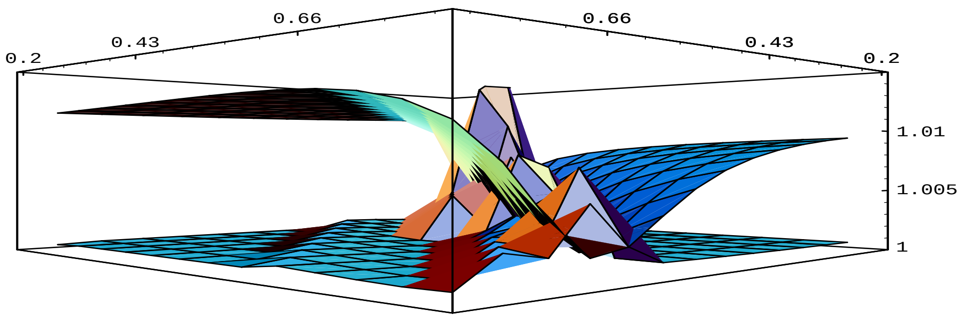

Figure 7: Normalized costs from dynamic temporal controller and from traditional controller with a sampling period 0.1.

computations are used than in a traditional controller.

The proposed formulation of dynamic temporal control is likely to have a significant impact on the way concurrent embedded real-time systems are designed. In hierarchical control environment, this approach is likely to result in designs which are significantly more efficient and flexible than traditional control schemes. As it uses less computational resources, the lower level temporal controllers will make the resources available to the higher level controllers without compromising the quality of control.

The key idea of dynamic temporal control, i.e., using dynamic sampling periods for different system states, may be applied to other time-critical applications, such as multimedia systems, as well as pure control applications.

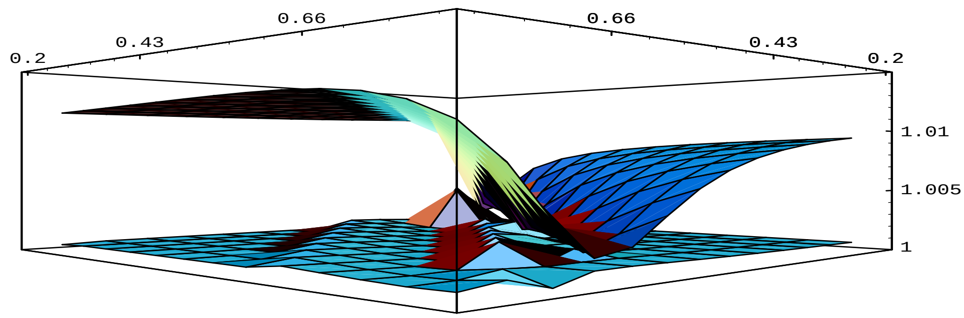

Figure 8: Normalized costs with adjusted threshold values.

References

[1] A. Belleisle. "Stability of systems with nonlinear feedback through randomly time-varying delays". *IEEE Transactions on Automatic Control*, AC-20:67–75, February 1975.

[2] R. Bellman. *Adaptive Control Process: A Guided Tour.* Princeton,NJ: Princeton University Press, 1961.

[3] R. Bellman. Bellman special issue. *IEEE Transactions on Automatic Control*, AC-26, October 1981.

[4] Seonho Choi and Ashok K. Agrawala. Dynamic dispatching of cyclic real-time tasks with relative constraints. Technical Report CS-TR-3770, UMIACS-TR-97-30, Department of Computer Science, University of Maryland, March 1997.

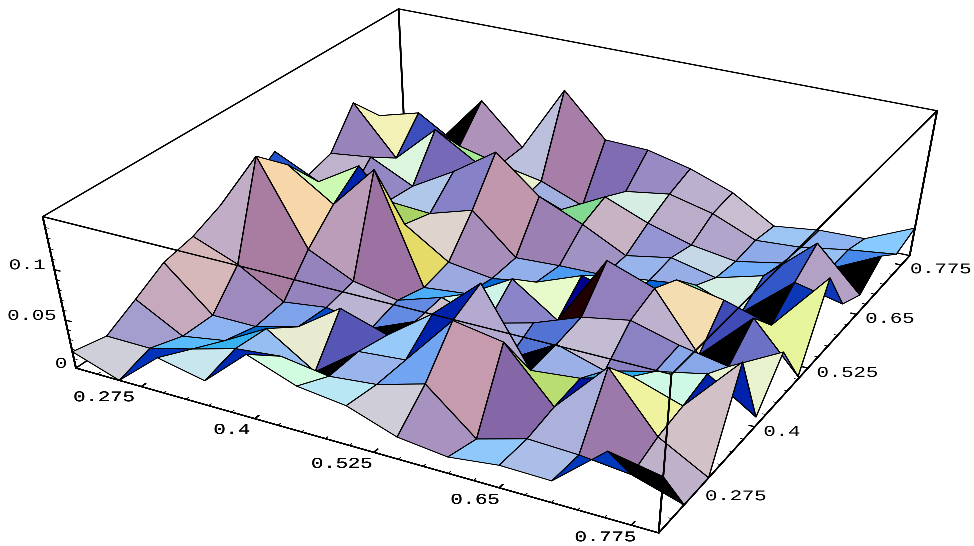

Figure 9: The difference of normalized costs in the presence of computational delay effects.

[5] P. Dorato and A. Levis. “Optimal linear regulators: The discrete time case”. *IEEE Transactions on Automatic Control*, AC-16:613–620, December 1971.

[6] A. Gosiewski and A. Olbrot. “The effect of feedback delays on the performance of multivariable linear control systems”. *IEEE Transactions on Automatic Control*, AC-25(4):729–734, August 1980.

[7] K. Hirai and Y. Satoh. “Stability of a system with variable time delay”. *IEEE Transactions on Automatic Control*, AC-25(3):552–554, June 1980.

[8] S. T. Levi, Satish K. Tripathi, Scott Carson, and Ashok K. Agrawala. “The MARUTI hard real-time operating system”. *ACM Symp. on Op. Syst. Principles, Op. Syst. Review*, 23(3), July 1989.

[9] Shem-Tov Levi and Ashok K. Agrawala. *Real Time System Design.* McGraw Hill, 1990.

[10] Z. Rekasius. "Stability of digital control with computer interruptions". *IEEE Transactions on Automatic Control*, AC-31:356–359, April 1986.

[11] Manas Saksena, James da Silva, and Ashok K. Agrawala. *"Design and implementation of Maruti-II"*, chapter 4. Prentice Hall, 1995. In *Advances in Real-Time Systems*, edited by Sang H. Son.

[12] K. G. Shin and H. Kim. "Derivation and application of hard deadlines for real-time control systems". *IEEE Transactions on Systems, Man and Cybernetics*, 22(6):1403–1413, November 1992.

[13] G.S. Virk. *Digital Computer Control Systems*, chapter 4. McGraw Hill, 1991.

[14] J. Xu and D. Parnas. "On satisfying timing constraints in hard real-time systems". In *ACM SIGSOFT '91 Conference on Software for Critical Systems*, pages 132–146, December 1991.

[15] K. Zahr and C. Slivinsky. "Delay in multivariable computer controlled linear systems". *IEEE Transactions on Automatic Control*, pages 442–443, August 1974.