2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

Adaptive Time Horizon Optimization in Model Predictive Control

Greg Droge and Magnus Egerstedt

Abstract— **Whenever the control task involves the tracking of a reference signal the performance is typically improved if one knows the future behavior of this reference. However, in many applications, this is typically not the case, e.g., when the reference signal is generated by a human operator, and a remedy to this can be to try and model the reference signal over a short time horizon. In this paper, we address the problem of selecting this horizon in an adaptive fashion by minimizing a cost that takes into account the performance of the underlying control problem (that prefers longer time horizons) and the effectiveness of the reference signal model (that prefers shorter time horizons). The result is an adaptive time horizon controller that operates in a manner reminiscent of Model Predictive Control (MPC).**

I. Introduction

The basic idea behind Model Predictive Control (MPC) is to select the control signal at the current time in such a way that it optimizes a cost function over some time horizon. This process is repeated at each time step and, as it is based on optimal control, it inevitably involves a forward simulation of the system states based on a model of the system. This has proved to be a highly useful control design methodology in a number of applications (see for example [1], [2], [3], [4], [5], [6], [7]). However, if the cost involves an unknown reference signal, this signal must be estimated in order to be able to evaluate the cost through forward simulation. But, if the reference is for example generated by a human operator it may not be so easy to obtain a reliable estimate over the nominal time horizon over which the control design task is defined. In fact, a standard cost function to be minimized for such an optimal control (e.g., [8]) may take on the form

$$\min_u \int_t^{t+\Delta} L(x(s),u(s),r(s))ds, \tag{1}$$

$$\text{subject to } \frac{dx(s)}{ds} = f(x(s),u(s)),$$

where t is the current time, $x(s)$ is the simulated state trajectory at a future time $s \geq t$, which is assumed to be equal to the actual state $x(t)$ at the current time, $u(s)$ is the corresponding control input, and $r(s)$ is the reference signal that has been observed for $s \leq t$ and is assumed to be unknown for $s > t$. Moreover, Δ is the time horizon over which the problem is considered.

But, since we do not actually know $r(s)$ for $s > t$, we cannot actually solve (1) since this is not a casual problem. However, we can use some method to predict for future values of $r(s)$ in order to make the problem well-posed. We denote this predicted signal by $\hat{r}(\tau,s)$ where $\tau \leq t$ is the time at which the prediction of $r(s)$ was made. Replacing $r(s)$ with $\hat{r}(t,s)$ in (1) we thus get the following well-posed problem

$$\min_u \int_t^{t+\Delta} L(x(s),u(s),\hat{r}(t,s))ds. \tag{2}$$

The relationship between the solutions to (2) and (1) depends on the quality of the estimate of the reference signal. If the estimate was perfect then one could in fact solve the original problem by directly using an MPC method where, in order to improve performance further, Δ is chosen as large as possible subject to factors such as computation speed, convergence, stability, and satisfaction of terminal contraints (see for example [7], [9], [10], [11], [12], [13]). However, when the estimate is not perfect, a longer time horizon may in fact be detrimental in that the effect of the poor estimate is amplified over a longer time period. In this paper we make this trade-off explicit by using an MPC approach to selecting Δ itself in an adaptive manner.

As the MPC approach to optimal control uses the system model as a generator of estimates of future state values, the problem when the reference signal r is unknown for future times can naturally be addressed using a similar approach. At each time step, one could make a prediction for r and then use the quality of that prediction as a way of determining how far off into the future one can effectively define the original MPC problem. As such, there are two main contribution of this paper. The first contribution is the underlying idea of choosing an optimal time horizon in an adaptive fashion and the second contribution is the introduction of three different candidate quality measures which are used to find the optimal time horizon. The reason why we introduce three different such quality measures is that the problem of determining the quality of an estimate is also an inherently ill-posed problem. The estimate is obtain at the current time and its performance must be understood against future values of the entity that it is trying to estimate. But these values are not available at the current time, which calls for some novel causal quality measures of a non-causal entity.

The remainder of this paper will proceed as follows: In the next section we will formulate the general time horizon optimization problem and introduce the three candidate quality measures. In section III, we will find the gradients to each of these quality measures. These gradients are then used in a steepest descent setting to solve for the time horizon optimization problem. In section IV we will illustrate the proposed approach on a reference tracking example and show that it outperforms any constant time horizon solution.

Email: {gregdroge,magnus}@ece.gatech.edu.
School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA.

II. Problem Setup

As explained in the introduction, we would like to choose the best time horizon at each time step. Ideally, we would like to have the time horizon as large as possible, but we want to avoid the detrimental effects from poor predictions of the reference signals. The method we propose is to change the time horizon based on how well we are doing at predicting the reference trajectory based on our past performance. This makes the philosophic assumption that the immediate future is not too different from the immediate past, i.e., that our ability to predict r in the past will reflect on our future ability to predict r. In other words, we choose to make the time horizon as large as possible if $\hat{r}$ has been a good prediction of r and scale it back when $\hat{r}$ has not done so well. Note that this may not always be the case but it is an assumption that allows us to formulate a relevant and well-posed problem. In fact, in this section we will introduce three different cost functions that act as quality measures for the performance of the estimated reference signal, which will be used for finding the optimal time horizon.

As there is an inherent trade-off between the quality of the underlying MPC controller (that favors larger time horizons) and the accuracy of the prediction (that favors shorter horizons), this trade-off must be present in the quality measures. We let $G(\Delta)$ be a point cost on the time horizon that penalizes smaller time horizons, and let $F(r,\hat{r})$ be an instantaneous cost that measures the quality of the estimated reference signal. If $\hat{r}$ does a poor job of predicting r, it will cause F to have a large impact on the cost and cause the resulting Δ to shrink. On the other hand, if $\hat{r}$ does well at predicting r, F will have little influence and G will cause Δ to grow. These two functions will work against each other to incorporate the underlying trade-off.

A. Back to the Future: Parts 1, 2, and 3

Ideally, we would like to evaluate how well the current estimate is doing at predicting the future, but since we do not know the reference for future times, we are forced to evaluate how our system identification has performed in the past. The three three cost functions we propose differ in how they evaluate the performance of our capability of predicting the input. Neither is fundamentally correct in that they are all causal approximations of a non-causal problem.

The first cost function evaluates how well the past estimate of the reference signal managed to predict the actual reference. For this, we initialize the prediction at time $\tau = t-\Delta$ using an estimate obtained at time τ, with $\hat{r}(\tau,\tau) = r(\tau)$. This prediction is integrated (e.g., based on system identification techniques) forward in time from time τ to the current time, t, in order to accumulate a cost for the difference in r and $\hat{r}$. In other words, we let the first cost function be given by

$$J_1(t,\Delta) = \int_{t-\Delta}^{t} F(r(s),\hat{r}(t-\Delta,s))ds + G(\Delta) \qquad (3)$$

and the basic idea behind this cost function is to look at the past to see how well our prediction process has been performing, and to use this as a guide for the performance of future predictions.

Although J_1 captures the actual performance of the estimate with a time lag of Δ, it does not explicitly evaluate the performance of the *current* prediction. As such, we let the second cost function use the current estimate of the reference signal and move that estimate back in time Δ time units. This construction forces us to make $\hat{r}$ a function of three rather than two time indices, namely $\hat{r}(\tau_1,\tau_2,s)$, where τ_1 is the time at which the estimate was constructed (typically current time t), τ_2 is the time at which the estimate was initialized (typically $t-\Delta$), and s is the time at which the quality of the prediction is evaluated. In fact, as we initialize $\hat{r}$ at time τ_2 we have that $\hat{r}(\tau_1,\tau_2,\tau_2) = r(\tau_2)$ for any τ_1. The second proposed cost function is thus

$$J_2(t,\Delta) = \int_{t-\Delta}^{t} F(r(s),\hat{r}(t,t-\Delta,s))ds + G(\Delta). \qquad (4)$$

The idea is that we want to see how far back in time our current system identification can be placed and still produce a good prediction.

We can note that J_2 deals with the current prediction of the reference signal and, as such, it may be more relevant than J_1. However, it suffers from the drawback that it is compared against an actual reference signal that it was not designed to mimic, i.e., the actual signal Δ time units ago. One way around this is to use the current estimate of the reference signal, but let the comparison run backwards in time, from the current time t to time $t-\Delta$. We can again use the initialization and notation for $\hat{r}$ as we did in (3) since there is no longer any need to distinguish between the time at which we initialize the prediction and the time we create the prediction. We place a negative sign in front of the integral so that we can use the same function for F as we did in the previous two cost functions and still get a positive cost. The third cost function is given by

$$J_3(t,\Delta) = -\int_{t}^{t-\Delta} F(r(s),\hat{r}(t,s))ds + G(\Delta). \qquad (5)$$

Each one of these three cost functions is fundamentally different, evaluating slightly different aspects of how well our method of predicting the reference signal is performing. The first one evaluates how well we have done in the past when predicting the unknown signal, the second evaluates how well our current prediction would have done in the past, and the final cost function evaluates how far back in time our current prediction will be able to capture the previous values of the actual reference signal. It will be seen in Section IV that each of these different cost functions give a slightly different value for the optimal time horizon.

III. Optimizing the Prediction Costs

In this section we will derive expressions for the gradients for each of the three cost functions with respect to the time horizon Δ. These gradients are important because they allow us to employ descent methods for solving for the optimal control time horizons. We will go through the derivation of the gradient for $J_1(t,\Delta)$ in some detail, including a discussion

of how the problem simplifies if the prediction is generated by the output of a linear system, obtained using standard system identification techniques.

A. Gradient of $J_1(t,\Delta)$

Since Δ appears in both the limits of integration in $J_1(t,\Delta)$ as well as in the integrand, we need to use the generalized Leibniz rule. If

$$H(\Delta) = \int_{a(\Delta)}^{b(\Delta)} h(\Delta, s)ds \tag{6}$$

then

$$\frac{dH}{d\Delta}(\Delta) = h(\Delta, b(\Delta))\frac{db}{d\Delta}(\Delta) - \tag{7}$$
$$h(\Delta, a(\Delta))\frac{da}{d\Delta}(\Delta) + \int_{a(\Delta)}^{b(\Delta)} \frac{dh}{d\Delta}(\Delta, s)ds$$

Comparing $J_1(t,\Delta)$ to (6) and (8) we have that

$$\frac{\partial J_1}{\partial \Delta}(t,\Delta) = \frac{dG}{d\Delta}(\Delta) + F(r(t-\Delta), \hat{r}(t-\Delta, t-\Delta)) - \tag{8}$$
$$\int_{t-\Delta}^{t} \frac{\partial F}{\partial \hat{r}}(r(s), \hat{r}(t-\Delta, s))\frac{\partial \hat{r}}{\partial \tau}(\tau, s)|_{\tau=t-\Delta}ds$$

Furthermore, if we assume that F is designed such that $F(x,x) = 0$ (which is not a bad assumption since F should be designed to penalize the difference between its two arguments), we can further simplify (8) because we initialize $\hat{r}(\tau,\tau)$ to be $r(\tau)$, and we obtain the following expresion for the gradient of J_1

$$\frac{\partial J_1}{\partial \Delta}(t,\Delta) = \frac{dG}{d\Delta}(\Delta) - \tag{9}$$
$$\int_{t-\Delta}^{t} \frac{\partial F}{\partial \hat{r}}(r(s), \hat{r}(t-\Delta, s))\frac{\partial \hat{r}}{\partial \tau}(\tau, s)|_{\tau=t-\Delta}ds$$

1) Assuming Linear System Identification: The most difficult part of using (9) is the calculation of $\frac{\partial \hat{r}}{\partial \tau}(\tau, s)$ because $\hat{r}(\tau, s)$ is not necessarily an explicit function of τ, rather there is a distinct $\hat{r}$ for any given τ. However, assuming that we used linear system identification to predict r we can derive an analytical expression for $\frac{\partial \hat{r}}{\partial \tau}(\tau, s)$.

We are going to first make the following assumptions about the method used to obtain the linear system used to generate the prediction:

1) $\frac{\partial \hat{R}}{\partial \tau}(\tau, s) = A_\tau \hat{R}(\tau, s)$; $\hat{R}(\tau,\tau) = R(\tau)$
   - $R(s)$ is some function of $r(s)$ (and possibly its derivatives)
   - A_τ is the system matrix obtained at time τ
2) $\hat{r}(\tau, s) = C\hat{R}(\tau, s)$ for some matrix C

With these assumptions we know that $\hat{R}(\tau, s)$ can be expressed directly in terms of the matrix exponential of A_τ and $R(\tau)$ (e.g., [14]) as

$$\hat{R}(\tau, s) = e^{A_\tau(s-\tau)}R(\tau) \tag{10}$$

i.e.,

$$\hat{r}(\tau, s) = Ce^{A_\tau(s-\tau)}R(\tau) \tag{11}$$

Let us also assume that $R(\tau)$ is sufficiently smooth (at least twice continuously differentiable) such that, for small enough ε, $R(\tau+\varepsilon)$ can be expressed using a Taylor expansion as

$$R(\tau+\varepsilon) = R(\tau) + \varepsilon Q(\tau) + o(\varepsilon), \tag{12}$$

where $Q(\tau)$ is some matrix representing how R is varying at time τ and $o(\varepsilon)$ represents higher order terms of ε. Similarly, assuming A_τ is sufficiently smooth we let

$$A_{\tau+\varepsilon} = A_\tau + \varepsilon B(\tau) + o(\varepsilon) \tag{13}$$

which in turn gives

$$e^{A_{\tau+\varepsilon}s} = e^{(A_\tau + \varepsilon B(\tau))s} + o(\varepsilon) \tag{14}$$

where $B(\tau)$ is some matrix representing how A_τ is varying at time τ.

Using these two smoothness assumptions, (11) and (14), we can express $\hat{r}(\tau+\varepsilon, s)$ in terms of C, A_τ, $R(\tau)$, $B(\tau)$, and $Q(\tau)$.

$$\hat{r}(\tau+\varepsilon, s) = Ce^{(A_\tau+\varepsilon B(\tau))(s-\tau-\varepsilon)}(R(\tau) + \varepsilon Q(\tau)) + o(\varepsilon). \tag{15}$$

Now, we can find $\frac{\partial \hat{r}}{\partial \tau}(\tau, s)$ through

$$\begin{aligned}\frac{\partial \hat{r}}{\partial \tau}(\tau, s) &= \lim_{\varepsilon\to 0} \frac{\hat{r}(\tau+\varepsilon, s) - \hat{r}(\tau, s)}{\varepsilon} \\ &= \lim_{\varepsilon\to 0} \frac{1}{\varepsilon} C((e^{(A_\tau+\varepsilon B(\tau))(s-\tau-\varepsilon)} - e^{A_\tau(s-\tau)})R(\tau) + \\ &\quad \varepsilon e^{(A_\tau+\varepsilon B(\tau))(s-\tau-\varepsilon)}Q(\tau) + o(\varepsilon)),\end{aligned} \tag{16}$$

which in turn gives

$$\frac{\partial \hat{r}}{\partial \tau}(\tau, s) = Ce^{A_\tau(s-\tau)}Q(\tau). \tag{17}$$

This tells us that if the prediction is given by the output of a linear system the gradient to (3) can be written as

$$\frac{\partial J_1}{\partial \Delta}(t,\Delta) = \frac{dG}{d\Delta}(\Delta) - \tag{18}$$
$$\int_{t-\Delta}^{t} \frac{\partial F}{\partial \hat{r}}(r(s), \hat{r}(t-\Delta, s))Ce^{A_{(t-\Delta)}(s-(t-\Delta))}Q(t-\Delta)ds.$$

B. Gradient of $J_2(t,\Delta)$

The derivation of the gradient of $J_2(t,\Delta)$ in (4) is only slightly different than that of $J_1(t,\Delta)$ in (3) because it uses the current prediction placed into the past to see how well it would have performed. Making the same assumptions we used to obtain (9), we obtain an analogous expression in (19), which only differs in the integral term where $\hat{r}$ depends on the current system identification instead of the past system identification.

$$\frac{\partial J_2}{\partial \Delta}(t,\Delta) = \frac{dG}{d\Delta}(\Delta) - \tag{19}$$
$$\int_{t-\Delta}^{t} \frac{\partial F}{\partial \hat{r}}(r(s), \hat{r}(t, t-\Delta, s))\frac{\partial \hat{r}}{\partial \tau}(t, \tau, s)|_{(\tau=t-\Delta)}ds$$

1) Assuming Linear System Identification: The derivation of the simplifications one obtained when the output of a linear system is used to generate the predition is also quite similar. We use the same assumptions for the method of forming the system identification and for smoothness of $R(\tau)$ as in (12), but no longer have the requirment on smoothness for A_τ because we always use the current system matrix. This is a significant improvement over the previous cost function because we can expect $R(\tau)$ to be smooth, but we have no gaurentee in general that the method of system identification will produce smooth changes in the matrix over time. Under these assumptions, we again get a very similar result for the gradient.

$$\frac{\partial J_2}{\partial \Delta}(t,\Delta) = \frac{dG}{d\Delta}(\Delta) - \int_{t-\Delta}^{t} \frac{\partial F}{\partial \hat{r}}(r(s),\hat{r}(t,t-\Delta,s)) C e^{A_t(s-(t-\Delta))} Q(t-\Delta) ds \tag{20}$$

Note that the only difference between (18) and (20) is that (20) uses the current system identification in the integral term.

C. Gradient of $J_3(t,\Delta)$

The gradient of $J_3(t,\Delta)$ in (5) differs from that of $J_1(t,\Delta)$ and $J_2(t,\Delta)$ because there is no dependence on Δ in the integrand, only in the upper limit of integration. This allows us to obtain a more simple gradient.

$$\frac{\partial J_3}{\partial \Delta}(t,\Delta) = \frac{dG}{d\Delta}(\Delta) + F(r(t-\Delta),\hat{r}(t,t-\Delta)) \tag{21}$$

1) Assuming Linear System Identification: Because (21) does not have a term that deals with the derivative of $\hat{r}$, there is not much of a simplification of (21) for the linear case except for writing $\hat{r}(\tau,s)$ in terms of r. We can make the same assumptions that we did before to get the desired relationship. This allows us to immediately write the solution to the linear case in (22) which is much more simple than the previous two cases because there is no need for a costate. This gradient is less restrictive than both of the previous gradients because it makes no assumptions on the smoothness of A_τ or $R(\tau)$.

$$\frac{\partial J_3}{\partial \Delta}(t,\Delta) = \frac{dG}{d\Delta}(\Delta) + F(r(t-\Delta), C e^{A_t(-\Delta)} R(t)) \tag{22}$$

D. Kuhn Tucker Conditions

It may be the case that there needs to be upper and lower bounds on the time horizon window. For example, if r was first observed at time t_0 then when solving for Δ at the current time, t, the furthest the integrals in (3), (4), and (5) can go back is to time t_0. This would put a natural upper bound on $\Delta \leq t - t_0$. Likewise, when solving these problems, it does not make sense to have nonpositive values for Δ (which would correspond to a non-causal predictor) so a lower bound would be $\Delta \geq 0$. To solve for the general case we are going to give the bounds $\Delta_{min} \leq \Delta^* \leq \Delta_{max}$, where Δ^* denotes the (local) optimal solution for Δ. For ease of notation, we will drop the subscript from J_i , $i = 1,2,3$, because the Kuhn Tucker Conditions are the same for each J_i. The problem of minimization becomes

$$\begin{aligned} &\min_{\Delta} J(t,\Delta) \\ &\text{s.t. } g_1(\Delta) = \Delta_{min} - \Delta^* \leq 0 \\ &\quad\; g_2(\Delta) = \Delta^* - \Delta_{max} \leq 0 \end{aligned} \tag{23}$$

To impose the inequality constraints, we now introduce the Lagrangians μ_1 and μ_2 which satisfy the Kuhn-Tucker conditions. We can write the cost function, $\bar{J}$, in terms of the original J and the constraints which we can minimize with respect to Δ.

$$\begin{aligned} &\min_{\Delta} \bar{J}(t,\Delta) = J(t,\Delta) + \mu_1 g_1(\Delta) + \mu_2 g_2(\Delta) \\ &\text{s.t. } \mu_i g_i(\Delta) = 0;\ i = 1,2 \\ &\quad\; \mu_i \geq 0;\ i = 1,2 \end{aligned} \tag{24}$$

Equations (23) and (24) give all of the necessary information to solve for μ_1 and μ_2 using techniques found in many optimal control texts (e.g. [8]). The solution is simple and tells us that we can solve for Δ as if neither constraint were active and if the solution tells us that Δ is less than Δ_{min} we know that $\Delta^* = \Delta_{min}$. Likewise, if the solution tells us that Δ is greater than Δ_{max}, we know that $\Delta^* = \Delta_{max}$.

IV. Example: Reference Tracking

To illustrate the usefulness of the adaptive time horizon method proposed in this paper, we chose to track the output of a linear system. Doing this in simulation allows us to compare the results of solving the problem with a variable time horizon to an optimal solution. Using a Linear Quadratic Regulator (LQR) we can compare the results of the solution to the problem when the final time is fixed and given with the solutions we get when using variable time horizons. This will allow us to compare the results of the adaptive time horizon solutions and the optimal fixed time solution with the optimal solution obtained with the LQR.

A. Problem Setup

To setup the problem, we chose to minimize the cost function given in (25) where $r(s)$ is the reference signal we would like to track with the output of the system, $y(s)$, $u(s)$ is the system input, and q is a weight that gives the relative importance of the tracking versus the control effort.

$$\begin{aligned} &\min_{u} \int_0^T (q(y(s) - r(s))^2 + u(s)^2) ds \\ &\text{s.t. } \dot{x} = Ax + bu; x(0) \text{ given} \\ &\qquad y = cx \end{aligned} \tag{25}$$

where

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix},\ b = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ c = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

As stated in the introduction, we are not able to solve this problem directly when $r(s)$ is unknown for s greater than the current time. We instead use the two step process outlined and try to minimize the following cost function at each time step while finding Δ from minimizing (3) through (5).

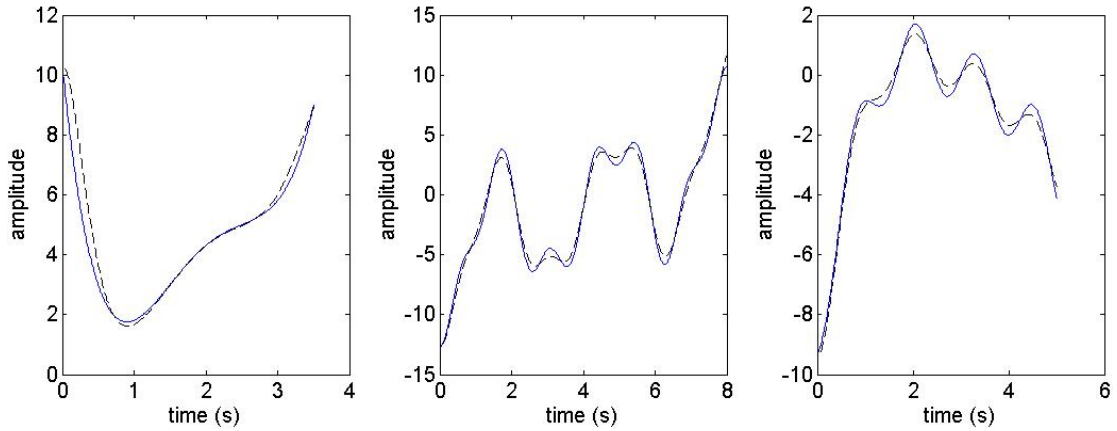

Fig. 1. The three reference signals used for comparison of variable time horizon (shown with the solid line) plotted against their respective optimal LQR solutions (dashed line). The LQR solution was obtained assuming $r(s)$ is known for all s. The left is a polynomial, the middle is a summation of sinusoids, and the right is a mix of polynomials and sinusoids

$$\min_u \int_t^{t+\Delta} (q(y(s)-\hat{r}(t,s))^2 + u(s)^2) ds \tag{26}$$

B. Least Squares System Identification

We used a 3^{rd} degree continuous-time least-squares system process to generate the prediction. Let the system matrix at time τ, A_τ, be defined as a system of integrators as shown below with the last row, θ_τ, being what we must calculate. We will let $R(s)$ be a vector of derivatives of $r(s)$ and let the relationship between $\hat{r}$ and r be defined as in (11). The vector $Q(\tau)$ required to minimize (3) and (4) simply becomes the element-wise derivative of $R(\tau)$. In other words,

$$A_\tau = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \theta_{\tau 1} & \theta_{\tau 2} & \theta_{\tau 3} \end{bmatrix} \tag{27}$$

$$R(\tau) = \begin{bmatrix} r(\tau) \\ \dot{r}(\tau) \\ \ddot{r}(\tau) \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, Q(\tau) = \dot{R}(\tau) = \begin{bmatrix} \dot{r}(\tau) \\ \ddot{r}(\tau) \\ r^{(3)}(\tau) \end{bmatrix}$$

In order to get a least squares estimate for θ_τ, we can solve for θ_τ from the equation $R = X\theta_\tau$, where X is a matrix of derivitives of r as shown below. To get the least squares estimate we solve for θ_τ that minimizes the squared error (ie $\min_{\theta_\tau} J = \frac{1}{2}(R - X\theta_\tau)^T(R - X\theta_\tau)$). This gives the least squares solution $\theta_\tau = (X^T X)^{-1} X^T R$, where

$$X = \begin{bmatrix} \dot{r}(\tau) & \ddot{r}(\tau) & r^{(3)}(\tau) \\ \ddot{r}(\tau) & r^{(3)}(\tau) & r^{(4)}(\tau) \\ r^{(3)}(\tau) & r^{(4)}(\tau) & r^{(5)}(\tau) \end{bmatrix} \tag{28}$$

C. Results

For the reference signal $r(s)$, we chose three different signals to test our proposed methodology, as shown in Figure 1. The first signal was a polynomial, the second was a summation of different sinusoids, and the third was a mix between polynomials and sinusoids. For each of these reference signals we did the following:

1) Solved the LQR problem with $r(s)$ known in (25) to get the optimal solution as a performance baseline.
2) Solved (2) using the optimal constant time horizon to allow for a comparison between the adaptively varying time horizon and the constant time horizon.
3) Ran the simulation three more times, assuming $r(s)$ was unknown for future times, and used (18), (20), and (22) at each time step to find the optimal time horizon.
   - We found that the optimal varying time horizon is disadvantaged at first because the largest it can make the time horizon is $t - t_0$, where t is the current time and t_0 is the initial time. To allow for a fair comparison we used the best constant time horizon for our horizon until $t - t_0$ was greater than the constant time horizon and we then switched to the adaptive time horizon method.
   - In solving for Δ in (18), (20), and (22), we used $F(r(s), \hat{r}(\tau, s)) = \frac{1}{2}(r(s) - \hat{r}(\tau, s))^2$ and $G(\Delta) = \frac{\rho}{\Delta}$
   - We iterated through different values of ρ to find the value that created the lowest squared error in comparison with the optimal solution found using LQR.

Table I shows the total squared error associated with the different methods. We can see that each method, except for one case for J_2, always outperformed the optimal constant time horizon case. We can also see that the cost functions for J_3, which corresponds to running the current system identification backward in time always outperformed every other method.

Figures 2 and 3 correspond to the results shown in the last column of Table I where the reference signal was a mixture of sinusoids and polynomials. In this case, each of the cost functions performed better than the optimal constant time horizon and so it is apparent that an adaptable time horizon can perform better than a fixed time horizon. It is also apparent that both J_1 and J_2 gave jagged results while J_3 varied much more smoothly with time. This was typical

Cost Function	Squared Error		
	Polynomial	Sinusoid	Polynomial Sinusoid
J_1	118.84	238.07	42.82
J_2	123.82	243.69	40.21
J_3	15.51	222.92	36.88
Constant Δ	122.26	270.55	42.91

TABLE I

This table lists the total squared error for each cost function and for the best constant time horizon with the reference trajectory being a summation of sinusoids in time. The squared error is the error of the output compared to the optimal output obtained from (25).

of the results from the other reference signals as well. The observations about J_3 both providing the lowest error as well as providing a smoother result leaves us to believe that it would be the best of the three cost functions for evaluating the variable time horizon. However, a more rigorous analysis would be needed to be able to make this claim conclusively as well as to study the stability characteristics of such an approach.

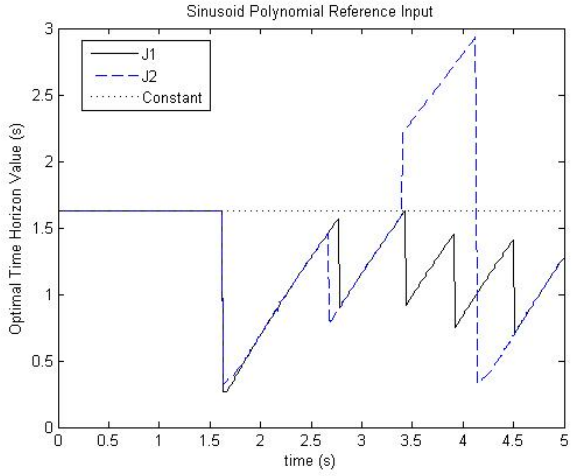

Fig. 2. The results for optimal time horizons for an input of both sinusoids and polynomials using cost functions J_1, J_2, from (3) and (5) respectively, as well as the optimal constant time horizon.

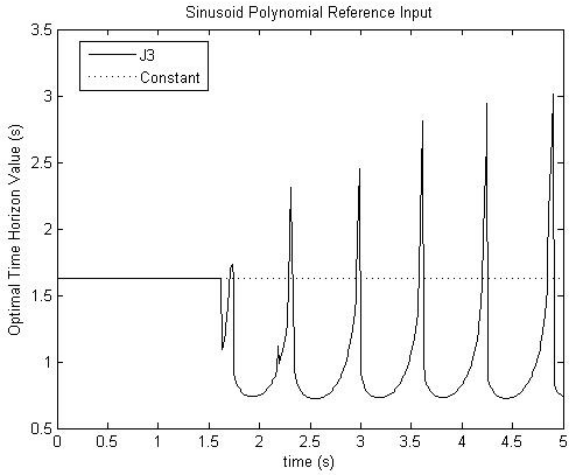

Fig. 3. The results for optimal time horizons for an input of both sinusoids and polynomials using cost function J_3 from (5) as well as the optimal constant time horizon.

REFERENCES

[1] S. Joe Qin, Thomas A. Badgwell, "A survey of industrial model predictive control technology", *Control Engineering Practice*, Volume 11, Issue 7, July 2003, Pages 733-764.

[2] Wenlin Wang, Daniel E. Rivera, Karl G. Kempf, "Model predictive control strategies for supply chain management in semiconductor manufacturing", *International Journal of Production Economics*, Volume 107, Issue 1, Special Section on Building Core-Competence through Operational Excellence, May 2007, Pages 56-77.

[3] Haslinda Zabiri, Yudi Samyudia, "A hybrid formulation and design of model predictive control for systems under actuator saturation and backlash", *Journal of Process Control*, Volume 16, Issue 7, August 2006, Pages 693-709.

[4] Gongsheng Huang, Shengwei Wang, Xinhua Xu, "A robust model predictive control strategy for improving the control performance of air-conditioning systems", *Energy Conversion and Management*, Volume 50, Issue 10, October 2009, Pages 2650-2658, ISSN 0196-8904, DOI: 10.1016/j.enconman.2009.06.014.

[5] Xiang Li, Thomas E. Marlin, "Robust supply chain performance via Model Predictive Control", *Computers & Chemical Engineering*, Volume 33, Issue 12, FOCAPO 2008 - Selected Papers from the Fifth International Conference on Foundations of Computer-Aided Process Operations, 10 December 2009, Pages 2134-2143.

[6] Raluca Roman, Zoltan K. Nagy, Mircea V. Cristea, Serban P. Agachi, "Dynamic modelling and nonlinear model predictive control of a Fluid Catalytic Cracking Unit", *Computers & Chemical Engineering*, Volume 33, Issue 3, Selected Papers from the 17th European Symposium on Computer Aided Process Engineering held in Bucharest, Romania, May 2007, 20 March 2009, Pages 605-617.

[7] Camacho, E. F, and C Bordons. *Model Predictive Control, second edition*. London: Springer-Verlag, 2004.

[8] Locatelli, Arturo. *Optimal Control: an Introduction*. Basel: Birkhauser Verlag, 2001.

[9] D. Q. Mayne, J. B. Rawlings, C. V. Rao, P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality", *Automatica*, Volume 36, Issue 6, June 2000, Pages 789-814.

[10] Rawlings, J.B.; Bonne, D.; Jorgensen, J.B.; Venkat, A.N.; Jorgensen, S.B.; , "Unreachable Setpoints in Model Predictive Control," *Automatic Control, IEEE Transactions on* , vol.53, no.9, pp.2209-2215, Oct. 2008

[11] Michalska, H.; Mayne, D.Q.; , "Robust receding horizon control of constrained nonlinear systems", *Automatic Control, IEEE Transactions on* , vol.38, no.11, pp.1623-1633, Nov 1993.

[12] Jadbabaie, A.; Yu, J.; Hauser, J.; , "Unconstrained receding-horizon control of nonlinear systems," Automatic Control, IEEE Transactions on , vol.46, no.5, pp.776-783, May 2001.

[13] Marco A. Rodrigues, Darci Odloak, "MPC for stable linear systems with model uncertainty", *Automatica*, Volume 39, Issue 4, April 2003, Pages 569-583.

[14] Brogan, William L. *Modern Control Theory. 3rd ed*. Englewood Cliffs, N.J.: Prentice Hall, 1991.