A Multiscale Approximation Scheme for Explicit Model Predictive Control with Stability, Feasibility, and Performance Guarantees

Sean Summers, Colin N. Jones, John Lygeros, and Manfred Morari
ETH Zurich, Switzerland

***Abstract*— In this paper, an algorithm is introduced based on classical wavelet multiresolution analysis that returns a low complexity explicit model predictive control law built on a hierarchy of second order interpolating wavelets. It is proven that the resulting interpolation is everywhere feasible. Further, tests to confirm stability and to compute a bound on the performance loss are introduced. Since the controller approximation is built on a gridded hierarchy, the evaluation of the control law in real-time systems is naturally fast and runs in a bounded logarithmic time. A simple example is provided which both illustrates the approach and motivates further research in this direction.**

I. INTRODUCTION

The implementation of a model predictive controller (MPC) requires the exact global solution of an optimization problem online at each sampling instant. It has become well-known that this optimization problem can be posed parametrically, with the measured state x as the parameter

$$J^*(x) := \min \{h(x,u) | g(x,u) \leq 0\}\,.$$

The offline computation of this parametric problem results in an explicit optimal control law $u^*(x)$ mapping the current state x to the optimal system input [1], [14], [18]. The result is an online computation of the optimal control law which depends on the evaluation of the explicit function $u^*(x)$ at state x rather than the solution of the optimization problem.

There are two main limitations of this approach. First, if the optimal control law can be computed, the computational complexity of the explicit controller can grow quickly with problem size. Also, as the controller complexity grows, the worst case computation time may rise above a practical value, thereby eliminating it as a viable choice in a real-time system. Therefore, it is only natural to consider approximate controllers whenever a reduction in the computational complexity and/or online computation time is needed. See [2] and [15] for a general overview of the problem.

Several authors have proposed approximation algorithms that can produce simpler piecewise affine (PWA) control laws at the cost of optimality [3], [13], [15], [17]. In almost all cases, the authors initially approximate the epigraph of the optimal cost function J^* with a polyhedron $\tilde{J}$, usually designed to ensure specific stability or performance constraints. Then, a feasible control law $\tilde{u}(x)$ is computed such that $\tilde{J}$ is a Lyapunov function for the resulting closed-loop system. Yet, the computation of a feasible control law $\tilde{u}$ which preserves the stability and performance of the cost function approximation is not straightforward, but can be done using the techniques discussed in [3], [15], [16].

In [13] an approximate scheme was proposed that has some of the same positive benefits of the method introduced in this paper. The primary similarity and strength of the approach is that the control law is defined over a hierarchical set of hypercubes, which provides an extremely fast online evaluation time that is similar to that given here. The limitation of the approach [13] is in the generation of the control law that is defined over these hypercubes, which can be discontinuous, infeasible and potentially destabilizing in the worst case. While the control law may often result in good performance, there is no systematic procedure to prove that it will satisfy the basic requirements of feasibility and stability or to bound the performance loss. It is the introduction of a constructive procedure to guarantee these additional benefits, while maintaining the fast online calculation time that distinguishes the method proposed in this paper.

In this paper, we wish to extend the class of problems that can be considered for fast online implementation. In our approach we construct an approximate control law by adaptive wavelet interpolation, and then analyze the resulting approximate cost function. Examples of wavelets are the orthogonal Daubechies wavelets [8], [9], the biorthogonal spline wavelets [5], and Interpolets [10], [11]. In this paper, we have chosen the second order Interpolet as our wavelet of choice. As it turns out, interpolating wavelets of the second order have a property which is of the utmost importance to our analysis. Namely, they result in a barycentric interpolation law, which guarantees that the resulting interpolation lies within the convex hull of the points being interpolated. As a result of this property, we are able to accurately evaluate the stability and feasibility of the control law, and even guarantee the controller to be optimal within a specified threshold ϵ.

The rest of the paper is arranged as follows. In section II we construct a d-dimension multiscale basis function, introduce an adaptive thresholding approach for sparse function approximation, and show that the multiscale basis we have constructed is barycentric. In section III we introduce the well known linear Model Predictive Control problem and the corresponding feasibility, stability, and performance guarantees for the approximate control law. In section IV we introduce the numerical implementation of the adaptive algorithm for the approximate control law. In section V we provide a numerical example of the method.

Sean Summers, Colin N. Jones, John Lygeros, and Manfred Morari are with the Automatic Control Laboratory, Department of Information Technology and Electrical Engineering, ETH Zurich, Switzerland
`summers,cjones,lygeros,morari@control.ee.ethz.ch`

II. Multiscale Function Approximation

Let the standard one-dimensional scaling function (hat function) with support $[-1, 1]$ be defined as

$$\phi(x) := \max(1 - |x|,\ 0). \tag{1}$$

In one dimension, we consider a dyadic discretization on the unit interval $\Omega = [0, 1]$. The resulting grid Ω_l, itself a finite reduction of Ω, is characterized by the level of discretization l and the index i. The level l determines the one-dimensional distance between grid points according to the dyadic formulation, i.e. the distance between points is defined $h_l = 2^{-l}$ and the number of points is $N = 2^l + 1$. The index i determines the location of the grid points $x_{l,i}$, i.e. $x_{l,i} := i \cdot h_l,\ \ 0 \leq i \leq 2^l$.

Let $\phi_{l,i}$ be a family of basis functions defined on Ω with support $[x_{l,i} - h_l, x_{l,i} + h_l]$. The function $\phi_{l,i}$ is generated from the scaling function (1) via translation and dilation,

$$\phi_{l,i}(x) = \phi\left(\frac{x - i \cdot h_l}{h_l}\right). \tag{2}$$

The family of univariate multiscale functions $\psi_{l,i}$ which make up the hierarchical basis is given as

$$\psi_{l,i} = \phi_{l,i}, \quad i \in I_l.$$

where

$$I_l = \begin{cases} \{i \in \mathbb{N}_0 | 1 \leq i \leq 2^l - 1, i \text{ odd}\} & l > l_0 \ , \\ \{i \in \mathbb{N}_0 | 0 \leq i \leq 2^l\} & l = l_0 \ . \end{cases}$$

Let $\mathbf{y} \in \mathbb{N}_0^d$, i.e. the bold font of some arbitrary variable taking values in the d-dimensional non-negative integers, denote a d-dimensional multi-index, where operations (e.g. exponentiation) and comparisons (e.g. $\leq$) hold component-wise, e.g. $\mathbf{y}^2 = [y_1^2 \ y_2^2 \ldots y_d^2]$. We now consider the construction of a multivariate multiscale basis on the unit cube $\Omega^d = [0, 1]^d$, where d is the dimension, by tensor product expansion of the one-dimensional multivariate functions $\psi_{l,i}$, i.e.

$$\psi_{l,\mathbf{i}} = \prod_{j=1}^{d} \psi_{(l,i_j)}$$

with the multi-index $\mathbf{i} \in I_l^d$ and

$$I_l^d = \begin{cases} \{\mathbf{i} \in \mathbb{N}_0^d | \mathbf{0} \leq \mathbf{i} \leq \mathbf{2}^l\} \setminus & \\ \{\mathbf{i} \in \mathbb{N}_0^d | \mathbf{0} \leq \mathbf{i} \leq \mathbf{2}^l, i_j \text{ even } \forall j \in [1, d]\} & l > l_0 \ , \\ \{\mathbf{i} \in \mathbb{N}_0^d | \mathbf{0} \leq \mathbf{i} \leq \mathbf{2}^l\} & l = l_0 \ . \end{cases}$$

Note that I_l^d is simply the full grid less those points seen at previous levels, as depicted in Figure 1. We may now define the d-dimensional hierarchical function spaces of piecewise d-linear functions as $W_l^d = \text{span}\{\psi_{l,\mathbf{i}} : \mathbf{i} \in I_l^d\}$. Defining the family of d-dimensional nodal basis functions

$$\phi_{l,\mathbf{i}}(x) = \prod_{j=1}^{d} \phi_{l,i_j}(x_j)$$

and the d-dimensional nodal function space $V_l^d = \text{span}\{\phi_{l,\mathbf{i}} : \mathbf{0} \leq \mathbf{i} \leq \mathbf{2}^l\}$, it can be shown that $V_l^d = \bigoplus_{k \leq l} W_k^d$ where $\bigoplus$ denotes the direct sum.

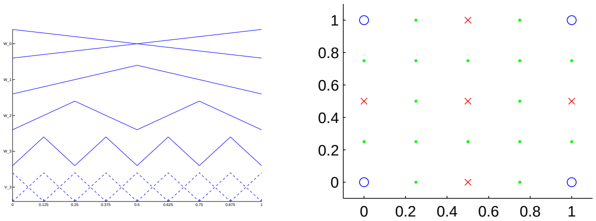


Fig. 1. (Left) Piecewise linear hierarchical basis (solid lines) vs. nodal point basis (dashed lines). (Right) Grid points for subspaces W_0^2 (blue circles), W_1^2 (red x's), and W_2^2 (green dots).

Any function $u_l \in V_l^d$ can be uniquely represented in the nodal basis by the expression

$$u_l(x) = \sum_{\mathbf{i}=\mathbf{0}}^{\mathbf{2}^l} v_{l,\mathbf{i}} \cdot \phi_{l,\mathbf{i}}(x)$$

with coefficients $v_{l,\mathbf{i}}$. Likewise, the same function $u_l \in V_l^d$ can be uniquely represented in the hierarchical basis by

$$u_l(x) = \sum_{k=\mathbf{0}}^{l} \sum_{\mathbf{i} \in I_k^d} w_{k,\mathbf{i}} \cdot \psi_{\mathbf{k},\mathbf{i}}(x)$$

with hierarchical (detail) coefficients $w_{k,\mathbf{i}} \in \mathbb{R}$.

A common thresholding argument, and the one we will employ throughout this paper, $|w_{k,\mathbf{i}}| \geq \delta$, simply reduces to taking the absolute value of the detail coefficient greater than the tolerance, and discarding the weight $w_{k,\mathbf{i}}$ if it is lower than the threshold. In this manner, an adaptive function approximation can be expressed

$$\hat{u}_{\Lambda^\delta(u)}(x) = \sum_{(k,\mathbf{i}) \in \Lambda^\delta(u)} w_{k,\mathbf{i}} \cdot \psi_{k,\mathbf{i}}(x),$$

where the 'active' index set $\Lambda^\delta(u)$ is the index set corresponding to all non-zero detail coefficients.

A. Barycentric Interpolation

In [16], the authors computed an approximate controller based on barycentric interpolation. Although this resulted in a continuous albeit nonlinear control law, the authors were able to exploit the fact that the resulting interpolation lies within the convex hull of the points being interpolated, and as a result could guarantee feasibility, stability, and performance bounds for the approximate control law. Therefore, in order to generate similar arguments for an approximate control law built from the basis functions introduced above, we must first show that the hierarchical basis functions generate hypercubes spanned by barycentric interpolation.

Let us denote the convex hull of V, i.e. the intersection of all convex sets containing V, as $\text{conv}(V)$ and the set of extreme points of V as $\text{extr}(V)$ for some $V \subset \mathbb{R}^d$. A definition of a barycentric interpolation function follows:

Definition 3 (Barycentric function): Let $S := \text{conv}(\{v_1, \ldots, v_n\}) \subset \mathbb{R}^d$ be a polytope. The function

$f_v(x)$ is called *barycentric* if three conditions hold for all $x \in S$ and $v \in \text{extr}(S) = \{v_1, \dots, v_n\}$,

$$f_v(x) \geq 0 \ , \qquad \textit{positivity} \tag{4}$$

$$\sum_{v \in \text{extr}(S)} f_v(x) = 1 \ , \qquad \textit{partition of unity} \tag{5}$$

$$\sum_{v \in \text{extr}(S)} v f_v(x) = x \ . \qquad \textit{linear precision} \tag{6}$$

Given the nodal function approximation space V_l^d and Ω^d, for some point $x_{(l+1,\mathbf{j})} \in \Omega^d$ where $\mathbf{j} \in \{\mathbf{j} \in \mathbb{N}_0^d | \mathbf{0} \leq \mathbf{j} \leq \mathbf{2}^{(l+1)}, j_k \text{ odd } \forall k \in [1,d]\}$, the polytopic region $R_\mathbf{j}$ associated with $x_{(l+1,\mathbf{j})} \in R_\mathbf{j}$ can be defined

$$R_\mathbf{j} \quad = \quad \left\{ x \in \Omega^d \left| x \in \bigcap_{\mathbf{i} \in I_{R_\mathbf{j}}} \text{supp}(w_{l,\mathbf{i}} \phi_{l,\mathbf{i}}) \right. \right\} \ , \tag{7}$$

where

$$I_{R_\mathbf{j}} := \{\mathbf{i} \in \mathbb{N}_0^d | \mathbf{0} \leq \mathbf{i} \leq \mathbf{2}^l, x_{(l+1,\mathbf{j})} \in \text{supp}(w_{l,\mathbf{i}} \phi_{l,\mathbf{i}})\}.$$

Note that supp$(\cdot)$ denotes the support of a function, and that if the weight is zero then the support is empty. It can be seen that $\bigcup_\mathbf{j} R_\mathbf{j} = \Omega^d$ and that $R_\mathbf{j}$ is a hypercube for all j. The second part holds since the basis functions are axis aligned and have hypercubic support (thus, finite intersections result in either the empty set or a hypercube).

We now arrive at a critical lemma which will lead to provable arguments regarding the feasibility, stability, and performance of the approximate control law.

Lemma 8 (Nodal Barycentric Interpolation): Given a d-dimensional space Ω^d and nodal function approximation space V_l^d, the multivariate basis of tensor product second order interpolets defined over any $R_\mathbf{j} \subseteq \Omega^d$ form a barycentric interpolation over $R_\mathbf{j}$.

Proof: By definition, $R_\mathbf{j}$ is a polytope (hypercube) with 2^d vertices $v \in \text{extr}(R_\mathbf{j}) = \{v \in \Omega_l^d | \mathbf{i} \in I_{R_\mathbf{j}}, v = [i_1 h_l, \dots, i_d h_l]\}$, where h_l is the width. Note that because $R_\mathbf{j}$ is a hypercube, the index values i_k in the k^{th} dimension take only two values, an upper $\bar{i}_k$ and a lower $\underline{i}_k$, and are separated by a value of one, i.e. $\bar{i}_k - \underline{i}_k = 1$.

We may now proceed by evaluating each of the three characteristics of barycentric interpolation. *Positivity* is straight forward by the definition of the hat function (1). For the *partition of unity* property, we can see that

$$\begin{aligned} \sum_{v \in \text{extr}(R_\mathbf{j})} f_v(x) &= \sum_{\mathbf{i} \in I_{R_\mathbf{j}}} \prod_{j=1}^d \phi_{l,i_j}(x_j), \\ &= \phi_{l,\bar{i}_1}(x_1) \sum_{\mathbf{i} \in I_{R_\mathbf{j}}^{*(1)}} \prod_{j=2}^d \phi_{l,i_j}(x_j) \\ &\quad + \phi_{l,\underline{i}_1}(x_1) \sum_{\mathbf{i} \in I_{R_\mathbf{j}}^{*(1)}} \prod_{j=2}^d \phi_{l,i_j}(x_j), \\ &= \sum_{\mathbf{i} \in I_{R_\mathbf{j}}^{*(1)}} \prod_{j=2}^d \phi_{l,i_j}(x_j) \end{aligned}$$

where $I_{R_\mathbf{j}}^{*(1)} = I_{R_\mathbf{j}} \cap \{\mathbf{i} | i_1 = \bar{i}_1\} = I_{R_\mathbf{j}} \cap \{\mathbf{i} | i_1 = \underline{i}_1\}$ .We can systematically extract $\phi_{l,\bar{i}_k}(x_k) + \phi_{l,\underline{i}_k}(x_k) = 1$ for the k^{th} dimension until we have

$$\sum_{v \in \text{extr}(R_\mathbf{j})} f_v(x) = \phi_{l,\bar{i}_d}(x_d) + \phi_{l,\underline{i}_d}(x_d) = 1,$$

which satisfies the *partition of unity*. Lastly, we prove that the multiscale basis satisfies the *linear precision* requirement. Consider

$$\begin{aligned} \sum_{v \in \text{extr}(R_\mathbf{j})} v f_v(x) &= \sum_{\mathbf{i} \in I_{R_\mathbf{j}}} [i_1 h_l, \dots, i_d h_l] \prod_{j=1}^d \phi_{l,i_j}(x), \\ &= \left[\sum_{\mathbf{i} \in I_{R_\mathbf{j}}} i_1 h_l \prod_{j=1}^d \phi_{l,i_j}(x), \dots \right. \\ &\quad \left. \dots, \sum_{\mathbf{i} \in I_{R_\mathbf{j}}} i_d h_l \prod_{j=1}^d \phi_{l,i_j}(x) \right]. \end{aligned}$$

Therefore, *linear precision* can be evaluated dimension by dimension, i.e. for each $k \in \{1,d\}$ we can evaluate that $\sum_{v \in \text{extr}(R_\mathbf{j})} v_k f_v(x) = x_k$. For arbitrary $k \in \{1,d\}$ we have

$$\begin{aligned} \sum_{v \in \text{extr}(R_\mathbf{j})} v_k f_v(x) &= \sum_{\mathbf{i} \in I_{R_\mathbf{j}}} i_k h_l \prod_{j=1}^d \phi_{l,i_j}(x) \\ &= \bar{i}_k h_l \phi_{l,\bar{i}_k}(x_k) \sum_{\mathbf{i} \in I_{R_\mathbf{j}}^{*(k)}} \prod_{j \neq k} \phi_{l,i_j}(x_j) \\ &\quad + \underline{i}_k h_l \phi_{l,\underline{i}_k}(x_k) \sum_{\mathbf{i} \in I_{R_\mathbf{j}}^{*(k)}} \prod_{j \neq k} \phi_{l,i_j}(x_j), \\ &= \bar{i}_k h_l \phi_{l,\bar{i}_k}(x_k) + \underline{i}_k h_l \phi_{l,\underline{i}_k}(x_k). \end{aligned} \tag{9}$$

By the definition of the interpolating wavelets, (1) and (2), we can expand (9) such that

$$\begin{aligned} &\bar{i}_k h_l \phi_{l,\bar{i}_k}(x_k) + \underline{i}_k h_l \phi_{l,\underline{i}_k}(x_k) = \\ &\bar{i}_k h_l \left(1 - \left|\frac{x_k - \bar{i}_k h_l}{h_l}\right|\right) + \underline{i}_k h_l \left(1 - \left|\frac{x_k - \underline{i}_k h_l}{h_l}\right|\right). \end{aligned}$$

Taking into account the properties $\bar{i}_k - \underline{i}_k = 1$ and $\underline{i}_k h_l \leq x_k \leq \bar{i}_k h_l$, simple arithmetic leads us to the result

$$\bar{i}_k h_l \left(1 - \left|\frac{x_k - \bar{i}_k h_l}{h_l}\right|\right) + \underline{i}_k h_l \left(1 - \left|\frac{x_k - \underline{i}_k h_l}{h_l}\right|\right) = x_k.$$

Repeating this process for all dimensions $k \in \{1,d\}$ leads to the property of *linear precision*. ■

Corollary 10 (Hierarchical Barycentric Interpolation): Given an active multiscale index set $\Lambda^\delta(u)$ (i.e. the indices corresponding to all non-zero detail coefficients in $W_0^d, \dots, W_l^d$), the following properties hold

1) $V_l^{d,\delta}$ can be decomposed into hypercubes spanned by barycentric interpolation, where $V_l^{d,\delta} = \bigoplus_{k \leq l} W_k^{d,\delta}$ and $W_l^{d,\delta} = \text{span}\{\psi_{l,\mathbf{i}} : (l,\mathbf{i}) \in (l, I_l^d) \cap \Lambda^\delta(u)\}$.
2) Given $V_l^{d,\delta}$, the set required for barycentric interpolation to hold is given by

$$R^\delta = \{R \in \Omega^d | R = R_\mathbf{j} \ \text{, for some } \mathbf{j}\} \tag{11}$$

where $R_{\mathbf{j}}$ is defined in (7).

Proof: The first item is straight forward. We simply fill the approximate function space with $w_{k,\mathbf{i}} = 0$ for all $(k,\mathbf{i}) \in \{(k,\mathbf{i}) \in (\mathbb{N},\mathbb{N}^d) | (k,\mathbf{i}) \notin \Lambda^\delta(u), k \leq l\}$ such that $V_l^{d,\delta} = \bigoplus_{k\leq l} W_k^{d,\delta}$ becomes full, and then we apply Lemma 8. The second item is due to (7). ■

III. Application to Model Predictive Control

In this paper we are specifically interested in the following semi-infinite horizon optimal control problem:

$$\begin{aligned} J^*(x) = \quad & \min_{\{u_0,\ldots,u_{N-1}\}} J(u_0,\ldots,u_{N-1},x_0,\ldots,x_N) \quad (12) \\ \text{s.t.} \quad & x_{i+1} = Ax_i + Bu_i, \ \forall i = 0,\ldots,N-1 \\ & (x_i,u_i) \in \mathcal{X} \times \mathcal{U}, \ \forall i = 0,\ldots,N-1 \\ & x_N \in \mathcal{X}_F, \ \ x_0 = x, \end{aligned}$$

where

$$J(u_0,\ldots,u_{N-1},x_0,\ldots,x_N) := V_N(x_N) + \sum_{i=0}^{N-1} l(x_i,u_i) \ , \tag{13}$$

and $\mathcal{X}$, $\mathcal{U}$, and $\mathcal{X}_F$ are convex constraints on the states and inputs and the stage cost l is a strictly convex function. A function $\gamma(\cdot) : \mathbb{R} \to \mathbb{R}$ is assumed to exist that is continuous, strictly increasing and has $\gamma(0) = 0$ such that $\gamma(||x||) \leq l(x,0)$ for all x. Problem (12) can be re-written as a parametric optimization problem:

$$u^*(x) := \arg\min_u \{h(x,u) | g(x,u) \leq 0\}, \tag{14}$$

where u is a vector containing the sequence of inputs $u_0,\ldots,u_{N-1}$ and appropriate auxiliary variables and the functions h and g are convex. The system input is then given in a receding horizon fashion [19] by $u_0^*(x)$, which is the first input in the optimal control sequence of (12).

A. Stability Guarantees of Approximate Controllers

Given that in II-A we have shown that the approximate controller built from the adaptive multiscale basis functions can be separated into polytopic regions (hypercubes) of barycentric interpolation, we may now introduce the following results which enable us to evaluate the feasibility, stability, and performance of the individual regions. This will be done by constructing a Lyapunov function for the approximate closed-loop system $x^+ = Ax + B\hat{u}(x)$.

Consider a hierarchical approximate control law defined on $\Omega^d \subset \mathbb{R}^d$ with maximal level l

$$\hat{u}(x) = \sum_{(k,\mathbf{i})\in\Lambda^\delta(u^*)} w_{k,\mathbf{i}} \cdot \psi_{k,\mathbf{i}}(x). \tag{15}$$

where u^* is given by (14). By Corollary 10, we see that the approximate control law within $R \in R^\delta$ can be expressed by the barycentric interpolation law

$$\hat{u}(x) = \sum_{v\in \text{extr}(R)} \hat{u}(v) f_v(x), \qquad \text{if } x \in R. \tag{16}$$

Note that $\hat{u}(v)$ is not necessarily the optimal control law, but the approximate control law.

Lemma 17: [16] If $R = \text{conv}(v_1,\ldots,v_{2^d}) \in \Omega^d$ (where $v_1 \ldots v_{2^d}$ are the vertices of the hypercube R), $\hat{u}(v_j)$ is the approximate control law for the vertex v_j and $\hat{u}(x)$ is defined as in (16), then for all $x \in R$,

$$\begin{pmatrix} x \\ \hat{u}(x) \end{pmatrix} \in \text{conv}\left(\begin{pmatrix} v_1 \\ \hat{u}(v_1) \end{pmatrix} \cdots \begin{pmatrix} v_{2^d} \\ \hat{u}(v_{2^d}) \end{pmatrix} \right).$$

Lemma 17 leads us to the following result.

Lemma 18: [Barycentric Feasibility] The approximate control law $\hat{u}(x)$ is a feasible solution of (12), i.e. $g(x,\hat{u}(x)) \leq 0, \ \forall x \in R$, if and only if $\hat{u}(v)$ is feasible for all $v \in \text{extr}(R)$.

Proof: Follows directly from Lemma 17 and the convexity of g. ■

We will now show that the cost function (13) evaluated for the approximate control law (16) over $R \in R^\delta$ can be upper bounded by an approximate cost defined by barycentric interpolation. In doing so, we can show that the cost generated by the approximate control law (which is non-convex) is no more sub-optimal than the cost generated by barycentric interpolation, for which stability and performance can be computed. Thus, enabling us to prove stability and performance of the approximate control law.

Lemma 19: If $\hat{u}(x)$ is the barycentric control law defined in (16) and $\hat{u}(v)$ is feasible for all $v \in \text{extr}(R)$, then $J^*(x) \leq J(\hat{u}(x)) \leq \tilde{J}(x), \ \forall x \in R$ where

$$\tilde{J}(x) = \sum_{v\in\text{extr}(R)} J(\hat{u}(v)) f_v(x), \quad \forall x \in R.$$

Proof: The control law $\hat{u}(x)$ is feasible for all $x \in R$ (Lemma 18), thus the cost function $J(\hat{u}(x))$ must be sub-optimal, which gives us the lower bound $J^*(x) \leq J(\hat{u}(x))$. The second portion of the proof can be found in [16], which gives the upper bound. ■

Lemma 19 proves that for each $x \in R$, the true approximate cost function $J(\hat{u}(x))$ will lie within the convex hull of the extreme points $\left\{(v,\tilde{J}(v)) | v \in \text{extr}(R)\right\}$. With this key result in place, we can make use of the approximate stability theorem given in [20].

Theorem 20: Let $J^* : \mathbb{R}^d \to \mathbb{R}$ be the cost function of the optimal control problem (12) and a Lyapunov function for the system $x^+ = Ax + Bu_0^*(x)$. The approximate value function $J(\hat{u}(x))$ is a Lyapunov function for the system $x^+ = Ax + B\hat{u}_0(x)$ if for all x in the feasible set $\mathcal{R}$ the condition $J^*(x) \leq J(\hat{u}(x)) \leq J^*(x) + \gamma(||x||)$ holds.

Theorem 20 and Lemma 19 lead us to our main stability and performance guarantee. Specifically, the following theorem states that the suboptimality of a polytopic region, in terms of threshold ϵ, can be computed simply by solving a convex optimization problem.

Theorem 21: [Hierarchical Stability] The approximate cost function $J(\hat{u}(x))$ satisfies the Lyapunov criteria of Theorem 20 over a region $R \in R^\delta \in \mathcal{R}$ if $err(R) \geq 0$, where $err(R)$ is given by the convex optimization problem,

$$\begin{aligned} err(R) \quad &= \quad \min_{x,\lambda,u} V_N(x_N) + \sum_{i=0}^{N-1} l(x_i,u_i) + \epsilon l(x_0,0) \\ &\qquad - \sum_{v\in\text{extr}(R)} J(\hat{u}(v))\lambda_v \qquad (22) \\ s.t. \quad & \lambda_v \geq 0, \sum_{v\in\text{extr}(R)} \lambda_v = 1, x_0 = \sum_{v\in\text{extr}(R)} v\lambda_v, \\ & x_i \in \mathcal{X} \ , u_i \in \mathcal{U} \ , x_N \in \mathcal{X}_N. \end{aligned}$$

Proof: Proof is based on the barycentric interpolation properties (4), (5), and (6) which results in $\sum_{v\in\text{extr}(R)} J(\hat{u}(v))\lambda_v$ defined on the convex hull $\text{conv}(J(\hat{u}(v_1)), \ldots, J(\hat{u}(v_{2^d})))$. Therefore, by construction, if the worst case error between the optimal cost function and the convex hull of the approximate cost function is less than $\epsilon l(x_0,0)$, then the entire region spanned by the barycentric interpolation satisfies the same bounds. ■

Corollary 23: The approximate control law $J(\hat{u}(x))$ defined over the set $R^* := \{R \in R^\delta | err(R) \geq 0\}$ is a Lyapunov function over R^*.
Since the system must also be invariant or feasible for all time, a Lyapunov function alone is insufficient to prove stability for a constrained system. As discussed in [15], since level sets of Lyapunov functions are invariant [4], it is possible to determine an invariant subset of Ω^d.

Corollary 24: If $J_{\text{min}} := \min\{J(\hat{u}(v)) | v \in \text{extr}(R^*)\}$ and the conditions of Theorem 20 are satisfied, then the set

$$I := \{x \in R^* | J(\hat{u}(x)) \leq J_{\text{min}}\}$$

is invariant under the control law $\hat{u}(x)$.

IV. Approximate Explicit Control Law

With the results from the preceding sections, we can now develop a recursive algorithm for the approximate model predictive control law. The algorithm initializes at a user defined coarse uniform grid, and then proceeds with a dyadic refinement strategy, saving only the points which violate a user defined thresholding law and/or feasibility condition. Pseudocode for the process is given in Algorithm 1.

A. Complexity of Real-Time Implementation

The online implementation of the approximate control law consists of evaluating (15) at state x at each step. With this in mind, we introduce two data structures for the computation of (15) which trade-off speed and storage complexity.

In the first approach, we store the non-zero detail coefficients in a perfect hash function [6], [7], [12], [21] and compute (15) using only the basis functions which are non-zero at state x. With this approach, the required storage is minimal and the online computation of the control value at x becomes independent of the number of details stored. For example, at a depth of $l = 7$ and assuming a processor speed of 1 GHz , we can evaluate the control law at 2 MHz, 1 MHz, and 500 KHz for $d = 2$, $d = 3$, and $d = 4$.

In the second approach, similar to [13] and enabled by Corollary 10, we create a minimal search tree comprised of hypercubes aligned to the axis, where each hypercube in the tree has 2^d values associated to it representing (15) at the vertices. With this approach, while there is a slight increase in the necessary storage, the online computation time of (15) is minimal and independent of the number of details stored. For example, at a tree depth of $l = 7$ and assuming a processor speed of 1 GHz, we can evaluate the control law at 20 MHz, 10 MHz, and 5 MHz for $d = 2$, $d = 3$, and $d = 4$.

Algorithm 1 Adaptive Hierarchical Approximate MPC

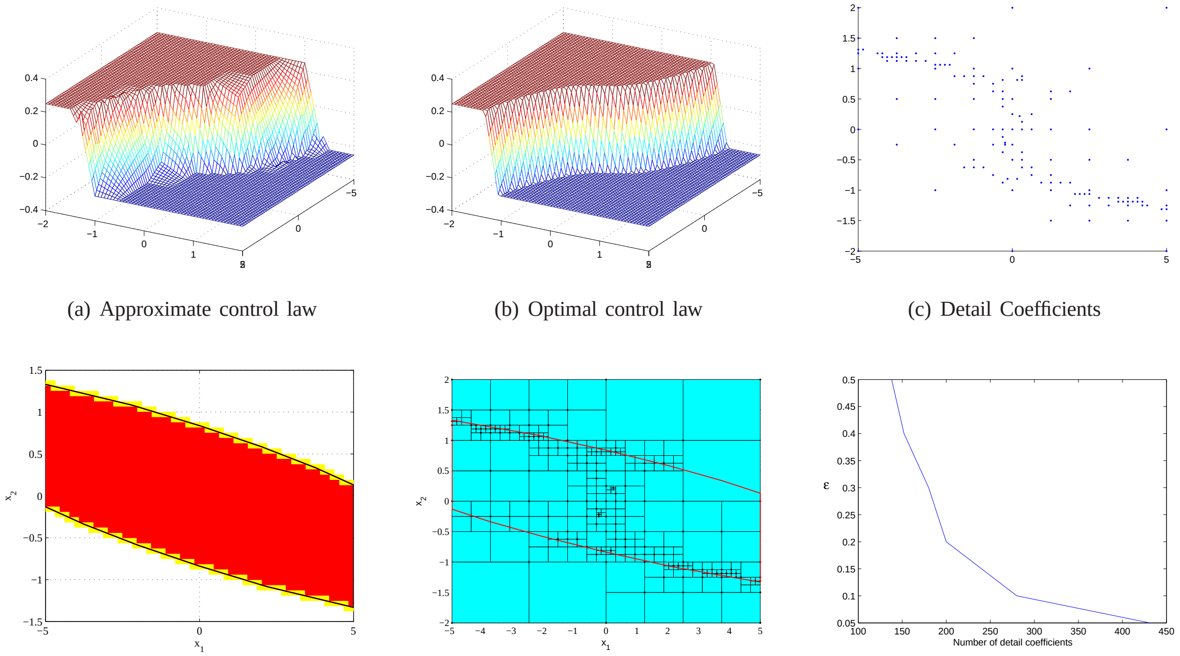
```
Require: MPC problem (12), Cost (13), performance threshold $\epsilon$, cost threshold $\epsilon_J$, and minimum, maximum, and boundary level arguments $l_0$, $l_{\max}$, and $l_{\text{bdry}}$.
Ensure: detail coefficients $\mathbf{w}$ and index set $\Lambda$ such that the approximate control law (15) $\hat{u}(x)$ has guaranteed feasibility, stability, and performance bound $\epsilon$.
Initialize the 'active' set $\Lambda = \{(k,i) : i \in I_k, k = l_0\}$
Initialize the evaluation set $R_k$ for $k = l_0$, the set of feasible regions $R_s = \emptyset$, the set of infeasible regions $R_f = \emptyset$, and the set of regions intersecting with the boundary $R_b = \emptyset$ at $k = l_{\text{bdry}}$
Compute the initial details $\mathbf{w} = \{w_{k,i} : (k,i) \in \Lambda\}$
while $R_k \neq \emptyset$ and $k \leq l_{\max}$ do
    $R_{\text{update}} = \emptyset$
    for all $R \in R_k$ do
        Check Stability by Theorem (21) and Feasibility by Lemma (18) for current region $R$
        Assign $R \in R_k$ to either $R_s$, $R_f$, $R_b$, or $R_{\text{update}}$
    end for
    Split the hypercube regions in the update set $R_{\text{update}}$ and assign to $R_{k+1}$
    Define the set of new vertices in $R_{k+1}$ as $\Lambda_n$
    Compute
        $\mathbf{w}_n = \left\{ \begin{array}{ll} w_{k,\mathbf{i}} : & (k,\mathbf{i}) \in \Lambda_n \text{ ,and for } y = x_{k,\mathbf{i}} \\ & \hat{u}(y) \text{ infeasible, or} \\ & J(\hat{u}(y)) - J^*(y) > \epsilon_J l(y,0) \end{array} \right\}$
    Let $\Lambda_n^* = \{(k,\mathbf{i}) : (k,\mathbf{i}) \in \Lambda_n, w_{k,\mathbf{i}} \in \mathbf{w}_n\}$
    Update the 'active' index $\Lambda = \Lambda \cup \Lambda_n^*$
    Update the detail set $\mathbf{w} = \mathbf{w} \cup \mathbf{w}_n$
    $k = k + 1$
end while
```

V. Numerical Example

Consider the simple two-dimensional example:

$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u,$$

with the input and state constraints $|u| \leq 0.25$, $||x||_\infty \leq 5$, and a horizon of $N = 10$. The stage cost is taken as $l(x,u) := x'x + 0.01u'u$. The optimal control law in this case requires 221 regions and can be seen in Figure 2. With $\epsilon = 0.5$, $\epsilon_J = 0.25$, $l_0 = 1$, $l_{\text{bdry}} = 5$, and $l_{\max} = 7$, we compute a stabilizing control law using Algorithm 1 that consists of 138 detail coefficients spanning 7 levels. In Figure



(a) Approximate control law

(b) Optimal control law

(c) Detail Coefficients

(d) Feasible Region

(e) Hierarchical Tree Structure

(f) Optimality vs. Complexity

Fig. 2. The approximate control law and optimal control law are shown in (a) and (b) respectively. Notice the sparsity of the required detail coefficients as shown in (c). In (d), the red region denotes a feasible and stable region with bounded performance, the yellow region denotes the regions which intersect with the boundary, and the black line gives the boundary of the optimal feasible set. In (e), we see the resulting hierarchical tree structure if speed is chosen over storage space. In (f) we illustrate optimality (ϵ) versus approximation complexity (i.e. number of non-zero detail coefficients).

2, the resulting control law and feasible regions are shown. Figure 2 also shows the performance threshold (ϵ) versus the required non-zero detail coefficients, the hierarchical tree structure, and the map of detail coefficients.

VI. Conclusion

The approximate explicit MPC method we have presented consists of a simple hierarchical gridding scheme which is easy to implement. The approach approximates the optimal control law directly, and because of the basis functions used to build the function approximation, has guaranteed stability, feasibility, and bounds on the performance. The ability to guarantee a level of accuracy between grid points enables an adaptive approach based on thresholding which can lead to a sparse representation of the explicit control law.

References

[1] V. Dua A. Bemporad, M. Morari and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, January 2002.

[2] A. Alessio and A. Bemporad. A surver on explicit model predictive control. In *Assessment and Future Directions of NMPC*, Sept 2008.

[3] Alberto Bemporad and Carlo Filippi. An algorithm for approximate multiparametric convex programming. *Comput. Optim. Appl.*, 35(1):87–108, 2006.

[4] F. Blanchini. Set invariance in control - a survey. *Automatica*, 35.

[5] Feauveau J.-C. Cohen A., Daubechies I. Biorthogonal bases of compactly supported wavelets. *Comm. pure and Appl. Math.*, 45:485–560, 1992.

[6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Second Edition*. The MIT Press, September 2001.

[7] Zbigniew J. Czech, George Havas, and Bohdan S. Majewski. An optimal algorithm for generating minimal perfect hash functions. *Information Processing Letters*, 43:257–264, 1992.

[8] I. Daubechies. Orthonormal bases of compactly supported bases. *Communications On Pure and Applied Mathematics*, 41:909–996, 1988.

[9] Ingrid Daubechies. *Ten lectures on wavelets*, volume 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992.

[10] G. Deslauriers and S. Dubuc. Symmetric iterative interpolation processes. *Constructive Approximation*, 5:49–68, 1989.

[11] David L. Donoho. Interpolating wavelet transforms.

[12] M. L. Fredman, J. Komlós, and E. Szemerédi. Storing a sparse table with 0(1) worst case access time. *J. ACM*, 31(3):538–544, 1984.

[13] T. A. Johansen and R. Grancharova. Approximate explicit constrained linear model predictive control via orthogonal search tree. *IEEE Trans. Automatic Control*, 48:810–815, 2003.

[14] Tor A. Johansen and Idar Petersen. On explicit suboptimal LQR with state and input constraints. In *in Proc. IEEE Conf. Decision and Control*, pages 05–6, 2000.

[15] C. N. Jones, M. Barić, and M. Morari. Multiparametric linear programming with applications to control. *European Journal of Control*, 13(2-3):152–170, 2007.

[16] C. N. Jones and M. Morari. Polytopic approximation of explicit model predictive controllers. 2009. Submitted.

[17] B. Lincoln and A. Rantzer. Relaxing dynamic programming. *Automatic Control, IEEE Transactions on*, 51(8):1249–1260, 2006.

[18] G. C. Goodwin M. M. Seron and J. A. De Dona. Geometry of model predictive control including bounded and stochastic disturbances under state and input constraints. Technical report, University of Newcastle, 2000.

[19] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, June 2000.

[20] P. O. M. Scokaert, D. Q. Mayne, and J. B. Rawlings. Suboptimal model predictive control (feasibility implies stability). *Automatic Control, IEEE Transactions on*, 44(3):648–654, 1999.

[21] Renzo Sprugnoli. Perfect hashing functions: a single probe retrieving method for static sets. *Commun. ACM*, 20(11):841–850, 1977.