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 $\epsilon$.

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.
II. Multiscale Function Approximation

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

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

In one dimension, we consider a dyadic discretization on the unit interval $\Omega = [0, 1]$. The resulting grid $\Omega_l$, itself a finite reduction of $\Omega$, 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 $\Omega$ 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 dilatation,

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

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

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

Let $y \in \mathbb{N}_d^l$, 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. $y^2 = [y_1^2, y_2^2, \ldots, y_d^2]$. 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.

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

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

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

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 $W_l^d = \text{span}\{\psi_{l,i} : i \in I_l^d\}$. Defining the family of d-dimensional nodal basis functions

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

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

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

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

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

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

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

$$
\tilde{u}_{A^d(u)}(x) = \sum_{(k,i) \in A^d(u)} w_{k,i} \cdot \psi_{k,i}(x),
$$

where the ‘active’ index set $A^d(u)$ 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 (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 \( v \in \text{extr}(S) = \{v_1, \ldots, v_n\} \),

\[
\sum_{v \in \text{extr}(S)} f_v(x) \geq 0 \, , \quad \text{positivity} \tag{4}
\]

\[
\sum_{v \in \text{extr}(S)} f_v(x) = 1 \, , \quad \text{partition of unity} \tag{5}
\]

\[
\sum_{v \in \text{extr}(S)} v f_v(x) = x \, , \quad \text{linear precision} \tag{6}
\]

Given the nodal function approximation space \( V_d^i \) and \( \Omega^d \),

for some point \( x_{i+1,j} \in \Omega^d \), \( j \in \{j \in \mathbb{N}_0 \mid 0 \leq j \leq 2^{i+1}-1\} \), \( j_k \) odd \( \forall k \in \{1, d\} \), the polytopic region \( R_j \) associated with \( x_{i+1,j} \in R_j \) can be defined

\[
R_j = \left\{ x \in \Omega^d \mid x \in \bigcap_{i \in I_{R_j}} \text{supp}(w_{i,1} \phi_{i,1}) \right\} , \tag{7}
\]

where

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

Note that \( \text{supp}(\cdot) \) denotes the support of a function, and that if the weight is zero then the support is empty. It can be seen that \( \bigcup_{j} R_j = \Omega^d \) and that \( R_j \) 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 \( \Omega^d \) and nodal function approximation space \( V_d^i \), the multivariate basis of tensor product second order interpolents defined over any \( R_j \subseteq \Omega^d \) form a barycentric interpolation over \( R_j \).

**Proof:** By definition, \( R_j \) is a polytope (hypercube) with \( 2^d \) vertices \( v \in \text{extr}(R_j) = \{v \in \Omega^d \mid i \in I_{R_j}, v = [i_1 h_1, \ldots, i_d h_d]\} \), where \( h_i \) is the width. Note that because \( R_j \) is a hypercube, the index values \( i_k \) in the \( k^{th} \) dimension take only two values, an upper \( i_k \) and a lower \( j_k \), and are separated by a value of one, i.e. \( i_k - j_k = 1 \).

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

\[
\sum_{v \in \text{extr}(R_j)} f_v(x) = \sum_{i \in I_{R_j}} \prod_{j=1}^d \phi_{i,j}(x_j),
\]

\[
= \phi_{1,j}(x_j) \sum_{i \in I_{R_{j}^{(1)}}} \prod_{j=2}^d \phi_{i,j}(x_j)
\]

\[
+ \phi_{j,j}(x_j) \sum_{i \in I_{R_{j}^{(1)}}} \prod_{j=2}^d \phi_{i,j}(x_j),
\]

\[
= \sum_{i \in I_{R_{j}^{(1)}}} \prod_{j=2}^d \phi_{i,j}(x_j)
\]

where \( I_{R_j}^{(1)} = I_{R_j} \cap \{i_1 = j_1\} = I_{R_j} \cap \{i_1 = j_1\} \). We can systematically extract \( \phi_{i,j_k}(x_k) + \phi_{j,j_k}(x_k) = 1 \) for the \( k^{th} \) dimension until we have

\[
\sum_{v \in \text{extr}(R_j)} f_v(x) = \phi_{i,j_k}(x_k) + \phi_{j,j_k}(x_k) = 1,
\]

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

Consider

\[
\sum_{v \in \text{extr}(R_j)} v f_v(x) = \sum_{i \in I_{R_j}} \prod_{j=1}^d \phi_{i,j}(x_j),
\]

\[
= \sum_{i \in I_{R_j}} i_1 h_1 \prod_{j=1}^d \phi_{i,j}(x_j),
\]

\[
\cdots , \sum_{i \in I_{R_j}} i_d h_d \prod_{j=1}^d \phi_{i,j}(x_j).
\]

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

\[
\sum_{v \in \text{extr}(R_j)} v_k f_v(x) = \sum_{i \in I_{R_j}} i_k h_i \prod_{j=1}^d \phi_{i,j}(x_j)
\]

\[
= \sum_{i \in I_{R_{j}^{(k)}}} \prod_{j \neq k} \phi_{i,j}(x_j)
\]

\[
+ \sum_{i \in I_{R_{j}^{(k)}}} i_k h_i \prod_{j \neq k} \phi_{i,j}(x_j),
\]

\[
= \sum_{i \in I_{R_{j}^{(k)}}} \prod_{j \neq k} \phi_{i,j}(x_j) + \sum_{i \in I_{R_{j}^{(k)}}} i_k h_i \prod_{j \neq k} \phi_{i,j}(x_j).
\]

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

\[
\sum_{i \in I_{R_{j}^{(k)}}} \prod_{j \neq k} \phi_{i,j}(x_j) + \sum_{i \in I_{R_{j}^{(k)}}} i_k h_i \prod_{j \neq k} \phi_{i,j}(x_j) = \sum_{i \in I_{R_{j}^{(k)}}} \prod_{j \neq k} \phi_{i,j}(x_j) + \sum_{i \in I_{R_{j}^{(k)}}} i_k h_i \prod_{j \neq k} \phi_{i,j}(x_j).
\]

Taking into account the properties \( i_k - j_k = 1 \) and \( \sum_{i \in I_{R_{j}^{(k)}}} \prod_{j \neq k} \phi_{i,j}(x_j) \leq x_k \leq \sum_{i \in I_{R_{j}^{(k)}}} \prod_{j \neq k} \phi_{i,j}(x_j) \), simple arithmetic leads us to the result

\[
\sum_{i \in I_{R_{j}^{(k)}}} \prod_{j \neq k} \phi_{i,j}(x_j) + \sum_{i \in I_{R_{j}^{(k)}}} i_k h_i \prod_{j \neq k} \phi_{i,j}(x_j) = x_k.
\]

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 \( \Lambda^d(u) \) (i.e. the indices corresponding to all non-zero detail coefficients in \( W_0^d, \ldots, W_d^d \)), the following properties hold

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

\[
R_l^d = \{R \in \Omega^d \mid R = R_j \, , \text{for some} \, j\} \tag{11}
\]
where \( R \) is defined in (7).

Proof: The first item is straightforward. We simply fill the approximate function space with \( w_{k,1} = 0 \) for all \((k, i) \in \{(k, i) \in \mathbb{N} \times \mathbb{N}^d \} \) such that \( V_l = \sum_{k \leq l} W_{k,l} \) 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:

\[
J^*(x) = \min_{\{u_0, \ldots, u_{N-1}\}} J(u_0, \ldots, u_{N-1}, x_0, \ldots, x_N)
\tag{12}
\]

\[\text{s.t. } x_{i+1} = Ax_i + Bu_i, \forall i = 0, \ldots, N-1
\]

\[\{x_i, u_i\} \in X \times U, \forall i = 0, \ldots, N-1
\]

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

and \( X, U \), and \( X_F \) are convex constraints on the states and inputs and the stage cost \( l \) is a strictly convex function. A function \( \gamma(\cdot) : \mathbb{R} \rightarrow \mathbb{R} \) 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 \( x^+ = Ax + Bu(x) \).

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

\[
\hat{u}(x) = \sum_{(k, i) \in A^l(u^*)} w_{k,i} \cdot \psi_{k,i}(x),
\tag{15}
\]

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

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

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

Lemma 17: [Hierarchical Stability] The approximate cost function \( J(\hat{u}(x)) \) satisfies the Lyapunov criteria of Theorem 20 over a region \( R \in R^d \) if \( \epsilon \) is sufficiently small.

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 \( J^*(x) \leq J(\hat{u}(x)) \leq J(x) \). 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 \( \{v, J(v)\} | v \in \text{extr}(R) \). With this key result in place, we can make use of the approximate stability theorem given in [20].

Theorem 20: Let \( J^* : \mathbb{R}^d \rightarrow \mathbb{R} \) be the cost function of the optimal control problem (12) and a Lyapunov function for the system \( x^+ = Ax + Bu(x) \). The approximate value function \( J(\hat{u}(x)) \) is a Lyapunov function for the system \( x^+ = Ax + Bu_0(x) \) if for all \( x \) in the feasible set \( R \) 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 \( \epsilon \), 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 \( R \in R^d \) if \( \epsilon \) is sufficiently small.
where \( err(R) \) is given by the convex optimization problem,
\[
err(R) = \min_{x, u} V_N(x_N) + \sum_{i=0}^{N-1} l(x_i, u_i) + \epsilon l(x_0, 0)
\]
\[
- \sum_{v \in \text{extr}(R)} J(\hat{u}(v))\lambda_v
\]
\[
s.t. \quad \lambda_v \geq 0, \sum_{v \in \text{extr}(R)} \lambda_v = 1, x_0 = \sum_{v \in \text{extr}(R)} v \lambda_v, \quad x_i \in X, u_i \in U, x_N \in X_N.
\]

Proof: Proof is based on the barycentric interpolation properties (4), (5), and (6) which result in \( \sum_{v \in \text{extr}(R)} J(\hat{u}(v))\lambda_v \) defined on the convex hull \( \text{conv}(J(\hat{u}(v_1)), \ldots, J(\hat{u}(v_2))) \). 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 \( R^* := \{ R \in R^d | err(R) \geq 0 \} \) 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 \( \Omega^d \).

**Corollary 24:** If \( J_{\min} := \min_{v \in \text{extr}(R)} J(\hat{u}(v)) \) and the conditions of Theorem 20 are satisfied, then the set
\[
I := \{ x \in R^* | J(\hat{u}(x)) \leq J_{\min} \}
\]
is invariant under the control law \( \hat{u}(x) \).

**IV. APPROXIMATE EXPLICIT CONTROL LAW**

With the results from the preceding sections, we 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

\[
\text{Algorithm 1 Adaptive Hierarchical Approximate MPC}
\]

**Require:** MPC problem (12), Cost (13), performance threshold \( \epsilon \), cost threshold \( \epsilon_J \), and minimum, maximum, and boundary level arguments \( l_0, l_{\text{max}}, \) and \( l_{\text{bdry}}. \)

**Ensure:** detail coefficients \( w \) and index set \( \Lambda \) such that the approximate control law (15) \( \hat{u}(x) \) has guaranteed feasibility, stability, and performance bound \( \epsilon. \)

1. Initialize the ‘active’ set \( \Lambda = \{(k, i) : i \in I_k, k = l_0 \} \)
2. 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 \( k = l_{\text{bdry}} \)
3. Compute the initial details \( w = \{ w_{k,i} : (k, i) \in \Lambda \} \)
4. while \( R_k \neq \emptyset \) and \( k \leq l_{\text{max}} \) do
5. \( R_{\text{update}} = \emptyset \)
6. for all \( R \in R_k \) do
7. Check Stability by Theorem (21) and Feasibility by Lemma (18) for current region \( R \)
8. Assign \( R \in R_k \) to either \( R_s, R_f, R_b, \) or \( R_{\text{update}} \)
9. end for
10. Split the hypercube regions in the update set \( R_{\text{update}} \) and assign to \( R_{k+1} \)
11. Define the set of new vertices in \( R_{k+1} \) as \( \Lambda_n \)
12. Compute
\[
w_n = \left\{ \begin{array}{ll}
w_{k,i} & (k, i) \in \Lambda_n \quad \text{and for } y = x_{k,i} \\
\hat{u}(y) \text{ infeasible, or} & J(\hat{u}(y)) - J^*(y) > \epsilon_J(y, 0) \\
\end{array} \right\}
\]
13. Let \( \Lambda_{n}^* = \{(k, i) : (k, i) \in \Lambda_n, w_{k,i} \in w_n \} \)
14. Update the ‘active’ index \( \Lambda = \Lambda \cup \Lambda_{n}^* \)
15. Update the detail set \( w = w \cup w_n \)
16. \( k = k + 1 \)
17. 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 \( l(x, u) := x^T x + 0.01 u^T u. \) 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 \( l_{\text{max}} = 7, \)
we compute a stabilizing control law using Algorithm 1 that consists of 138 detail coefficients spanning 7 levels. In Figure
2, the resulting control law and feasible regions are shown. Figure 2 also shows the performance threshold $\epsilon$ 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