High-order numerical solutions to Bellman's equation of optimal control

Cesar O. Aguilar and Arthur J. Krener

***Abstract*— In this paper we develop a numerical method to compute high-order approximate solutions to Bellman's dynamic programming equation that arises in the optimal regulation of discrete-time nonlinear control systems. The method uses a patchy technique to build Taylor polynomial approximations defined on small domains which are then patched together to create a piecewise-smooth approximation. Using the values of the computed cost function as the step-size, levels of patches are constructed such that their radial boundaries are level sets of the computed cost functions and their lateral boundaries are invariants sets of the closed-loop dynamics. To minimize the computational effort, an adaptive scheme is used to determine the number of patches on each level depending on the relative error of the computed solutions.**

I. INTRODUCTION

Consider the discrete-time nonlinear control system

$$x^+ = f(x,u) \tag{1}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the control, $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ are the dynamics (assumed to be smooth), and x^+ denotes the successor state. Given a stage cost $\ell : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, the *optimal regulator problem* for (1) is to find a feedback control $\kappa : \mathbb{R}^n \to \mathbb{R}^m$ such that

$$\begin{aligned}\pi(x_0) &\triangleq \min_{u(0),u(1),\dots} \sum_{k=0}^{\infty} \ell(x(k),u(k)) \\ &= \sum_{k=0}^{\infty} \ell(x(k),\kappa(x(k)))\end{aligned}$$

for all initial conditions $x_0 = x(0)$. If they exist, the *optimal cost function* π and *optimal regulator* κ satisfy Bellman's dynamic programming equation [2]

$$\pi(x) = \pi(f(x,\kappa(x))) + \ell(x,\kappa(x)). \tag{2}$$

If the cost function π is differentiable and $u \mapsto \pi(f(x,u)) + \ell(x,u)$ is strictly convex about $(x,u) = (0,0)$, then the following first order condition for a minimum is satisfied:

$$0 = \frac{\partial \pi}{\partial x}(f(x,\kappa(x))\frac{\partial f}{\partial u}(x,\kappa(x)) + \frac{\partial \ell}{\partial u}(x,\kappa(x)). \tag{3}$$

If f is linear in x and u, say $f(x,u) = Ax + Bu$, where $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$, and ℓ is quadratic of the form $\ell(x,u) = \frac{1}{2}x'Qx + \frac{1}{2}u'Ru$, where $Q \succeq 0$ and $R \succ 0$ are symmetric matrices of appropriate dimensions, then from the classical linear quadratic regulator problem [3], there exists a unique, symmetric, and positive definite matrix P such that $\pi(x) = \frac{1}{2}x'Px$ and $\kappa(x) = Kx$, where $K = -(B'PB + R)'B'PA$, provided (A,B) is stabilizable and $(A,Q^{1/2})$ is detectable. Moreover, the closed-loop matrix $A + BK$ has eigenvalues inside the unit circle and thus the closed-loop system is (globally) asymptotically stable. The matrix P is the unique solution to the discrete-time algebraic Riccati equation (DARE)

$$P = A'PA + A'PB(B'PB + R)^{-1}B'PA + Q.$$

When f and ℓ are nonlinear and have Taylor expansions of the form

$$\begin{aligned}f(x,u) &= Ax + Bu + f^{[2]}(x,u) + \cdots \\ \ell(x,u) &= \tfrac{1}{2}x'Qx + \tfrac{1}{2}u'Ru + \ell^{[3]}(x,u) + \cdots\end{aligned}$$

where $f^{[2]}(x,u)$ are the quadratic terms of f and $\ell^{[3]}(x,u)$ are the cubic terms of ℓ, etc., a method, originating in [1], is presented in [4] for computing Taylor polynomial approximations about $x = 0$ to the solutions (π,κ) of (2)-(3). The basic idea is to Taylor expand (2)-(3) about $x = 0$ and gather terms of the same order resulting in equations for the unknown Taylor coefficients of π and κ. One assumes that the Taylor expansion of π begins with quadratic terms and that of κ with linear terms, that is,

$$\begin{aligned}\pi(x) &= \tfrac{1}{2}x'Px + \pi^{[3]}(x) + \cdots \\ \kappa(x) &= Kx + \kappa^{[2]}(x) + \cdots\end{aligned}$$

For each $d \geq 1$, the resulting equations are for the $d+1$ order coefficients of π and the d order coefficients of κ. As shown in [4], it is possible to solve for the Taylor coefficients of π and κ to any desired degree provided $A + BK$ has eigenvalues inside the unit circle. Hence, in this paper we implicitly assume the stated stabilizability and detectability properties for the linearization of the optimal regulator problem for f and ℓ.

Let π^0 and κ^0 denote the Taylor polynomial functions of π and κ to degree $d+1$ and d, respectively, at $x = 0$. In a neighborhood of the origin, the polynomials (π^0,κ^0) serve as good approximations to (π,κ). If one desires more accurate approximations to (π,κ), one can increase the degree of approximation d, but there are two main drawbacks in doing so. First, increasing d increases the accuracy of (π^0,κ^0) but on a possibly *smaller* domain because of the rapidly growing behavior of high-order polynomials away from the origin. Second, the number of Taylor coefficients of degree d in n variables is $\binom{n+d-1}{d}$ and this number grows rapidly in d. Even with current personal computers, the symbolic computations needed to execute the algorithm in [4] requires significant computational time as can be verified with even

Research performed while first author held a NRC Associateship Award at the Department of Applied Mathematics, Naval Postgraduate School, 833 Dyer Rd., Bldg. 232, Monterey, CA 93943. Research supported in part by AFOSR and NSF. coaguila@nps.edu, ajkrener@nps.edu

small state dimensions such as $n = 2$ or $n = 3$ and $d \geq 5$. For these reasons it is natural to seek alternative methods for computing approximations to (π, κ).

In this paper, we present a numerical algorithm that extends the approximations (π^0, κ^0) and produces a piecewise smooth approximations to (π, κ) by patching together local approximations to (π, κ) on disjoint domains. Our method is based on the ideas in [5] in which a patchy type algorithm is developed for the numerical computation of approximate solutions to the Hamilton–Jacobi–Bellman equation. Roughly speaking, our method can be described as follows. The initial polynomial approximations (π^0, κ^0) are accepted on a sublevel set $\mathcal{P}^0 = \{x \in \mathbb{R}^n \ : \ \pi^0(x) \leq c\}$, with $c > 0$ chosen sufficiently small so that $\mathcal{P}^0 \cong \mathbb{D}^n$, where $\mathbb{D}^n$ denotes the closed unit disk in $\mathbb{R}^n$. We partition the boundary of $\mathcal{P}^0$, select points inside each set of the partition, and compute new polynomial approximations of (π, κ) by using (2)-(3), the previously computed approximations (π^0, κ^0), and a Cauchy-Kowalevski type algorithm to compute the high-order derivatives of the new polynomial approximations. Each new approximation is accepted on a domain radiating outward from the boundary of $\mathcal{P}^0$. The domains of the new approximations are pairwise disjoint and their outer-most boundaries define a piecewise smooth hypersurface. The outer-most boundaries are chosen as level sets of the newly computed cost functions. We then repeat the whole procedure on the newly computed outer-most boundaries.

Throughout the paper, we assume that for each $u \in \mathbb{R}^m$, $x \mapsto f(x, u)$ defines a local diffeomorphism on $\mathbb{R}^n$. That is, for each $u \in \mathbb{R}^m$ and $x \in \mathbb{R}^n$ there is a neighborhood Ω of x such that $f(\cdot, u) : \Omega \to f(\Omega, u)$ is a diffeomorphism. An important example where such discrete-systems arise is in the sampling of continuous-time control systems.

II. Computation of new solutions

In this section we describe a Cauchy-Kowalevski type algorithm for computing new polynomial approximations to (π, κ) from a previously computed polynomial solution (π^0, κ^0). We assume that π^0 is of degree $d+1$ and κ^0 is of degree d, where $d \geq 1$.

The polynomial $\pi^0(x)$ begins with the quadratic term $\frac{1}{2}x'Px$ where $P \succ 0$, and therefore π^0 has $x = 0$ as a non-degenerate local minimum. Hence, by Morse's lemma, there is a neighborhood $\Omega_0 \subset \mathbb{R}^n$ of the origin such that the sublevel sets $\{x \in \Omega_0 \ : \ \pi^0(x) \leq c_1\}$ are diffeomorphic to the unit disc $\mathbb{D}^n$, provided $c_1 > 0$ is sufficiently small, and therefore the level sets $\{x \in \Omega_0 \ : \ \pi^0(x) = c \leq c_1\}$ are diffeomorphic to the sphere $\mathbb{S}^{n-1}$. Hence, we assume $c_1 > 0$ is such that

$$\mathcal{P}^0 := \{x \in \mathbb{R}^n \ : \ \pi^0(x) \leq c_1\} \cong \mathbb{D}^n$$

and we let $\mathcal{S}^1 := \partial\mathcal{P}^0 \cong \mathbb{S}^{n-1}$. By making c_1 smaller if necessary, we can assume that $\mathcal{P}^0$ is contained in the domain of attraction of the closed-loop dynamics resulting by applying the feedback κ^0, that is,

$$\begin{aligned} x^+ &= f(x, \kappa^0(x)) \\ x(0) &\in \mathcal{P}^0 \end{aligned}$$

is an asymptotically stable system, and furthermore π^0 is a Liapunov function for the closed-loop dynamics, that is,

$$\pi^0(f(x, \kappa^0(x))) - \pi^0(x) < 0$$

for $x \in \mathcal{P}^0$. In particular, for all $x \in \mathcal{S}^1$ we have

$$\pi^0(f(x, \kappa^0(x))) < \pi^0(x) = c_1$$

and thus $f(x, \kappa^0(x))$ will lie in the interior of $\mathcal{P}^0$ for all $x \in \mathcal{S}^1$.

Now let $x^* \in \mathcal{S}^1$ and we seek to augment to (π^0, κ^0) new polynomial approximations (π^*, κ^*) defined in a neighborhood of x^* and radiating outward from the boundary $\mathcal{S}^1$. To do so, we ask that near x^*, the new feedback control κ^* drive the state into the interior of the domain $\mathcal{P}^0$ after one time-step. Consequently, we ask that the pair (π^*, κ^*) satisfy

$$\pi^*(x) = \pi^0(f(x, \kappa^*(x))) + \ell(x, \kappa^*(x)) \tag{4}$$

for all x near x^*. The first order necessary condition for a minimum then becomes

$$0 = \frac{\partial \pi^0}{\partial x}(f(x, \kappa^*(x)) \frac{\partial f}{\partial u}(x, \kappa^*(x)) + \frac{\partial \ell}{\partial u}(x, \kappa^*(x)). \tag{5}$$

Using (4)-(5) and a Cauchy-Kowalevski technique, we now describe how one can compute Taylor approximations to (π^*, κ^*) centered at x^* order-by-order to degree $d+1$ and d, respectively.

To compute the zeroth order terms of (π^*, κ^*) at x^*, we first use (5) to solve for $\kappa^*(x^*)$. In practice, this can be done using Newton's method with initial guess $\kappa^0(x^*)$. Having computed $\kappa^*(x^*)$, we can determine $\pi^*(x^*)$ directly by evaluating the right-hand-side of (4) at x^*. For later use and to ease notation, let $u^* = \kappa^*(x^*)$ and $y^* = f(x^*, u^*)$.

Now we compute the first order terms of (π^*, κ^*) at x^*. To do so, differentiate (4) with respect to x_i yielding (we omit evaluation at (x, u) and $f(x, u)$ for notational simplicity)

$$\frac{\partial \pi^*}{\partial x_i} = \frac{\partial \pi^0}{\partial x_a}\frac{\partial f_a}{\partial x_i} + \frac{\partial \ell}{\partial x_i} + \left[\frac{\partial \pi^0}{\partial x_a}\frac{\partial f_a}{\partial u_\alpha} + \frac{\partial \ell}{\partial u_\alpha}\right]\frac{\partial \kappa^*_\alpha}{\partial x_i} \tag{6}$$

for $i = 1, \ldots, n$, and where we have used the summation convention. Now notice that, by construction of u^* from (5), the coefficient of $\frac{\partial \kappa^*_\alpha}{\partial x_i}$ in (6) vanishes at (x^*, u^*) and we can therefore compute $\frac{\partial \pi^*}{\partial x_i}(x^*)$ directly as

$$\frac{\partial \pi^*}{\partial x_i}(x^*) = \frac{\partial \pi^0}{\partial x_a}(y^*)\frac{\partial f_a}{\partial x_i}(x^*, u^*) + \frac{\partial \ell}{\partial x_i}(x^*, u^*). \tag{7}$$

Now consider the computation of $\frac{\partial \kappa^*_\alpha}{\partial x_j}$, for $j = 1, \ldots, m$. First, write (5) in component form

$$0 = \frac{\partial \pi^0}{\partial x_a}\frac{\partial f_a}{\partial u_\alpha} + \frac{\partial \ell}{\partial u_\alpha}, \quad \alpha = 1, \ldots, m$$

and then differentiate with respect to x_j yielding

$$0 = M_{\alpha\beta}\frac{\partial \kappa^*_\beta}{\partial x_j} + \frac{\partial^2 \pi^0}{\partial x_a \partial x_b}\frac{\partial f_b}{\partial x_j}\frac{\partial f_a}{\partial u_\alpha} + \frac{\partial \pi^0}{\partial x_a}\frac{\partial^2 f_a}{\partial u_\alpha \partial x_j} + \frac{\partial^2 \ell}{\partial u_\alpha \partial x_j} \tag{8}$$

where

$$M_{\alpha\beta} = \frac{\partial^2 \pi^0}{\partial x_a \partial x_b}\frac{\partial f_b}{\partial u_\beta}\frac{\partial f_a}{\partial u_\alpha} + \frac{\partial \pi^0}{\partial x_a}\frac{\partial^2 f_a}{\partial u_\alpha \partial u_\beta} + \frac{\partial^2 \ell}{\partial u_\alpha \partial u_\beta} \tag{9}$$

for $1 \le \alpha, \beta \le m$. The number of unknowns $\frac{\partial \kappa^*_\beta}{\partial x_j}$ is mn and (8) produces mn equations. Assuming that the symmetric matrix $M(x,u) \in \mathbb{R}^{m\times m}$ with entries $M_{\alpha\beta}(x,u)$ is invertible at (x^*,u^*), we can solve uniquely for the unknowns $\frac{\partial \kappa^*_\beta}{\partial x_j}$ from (8).

Next, to comptue the quadratic terms of (π^*,κ^*) at x^*, differentiate (6) with respect to x_j yielding

$$\frac{\partial^2 \pi^*}{\partial x_i \partial x_j} = S_{ij} + \left[\frac{\partial \pi^0}{\partial x_a}\frac{\partial f_a}{\partial u_\alpha} + \frac{\partial \ell}{\partial u_\alpha}\right]\frac{\partial^2 \kappa^*_\alpha}{\partial x_i \partial x_j} \tag{10}$$

where S_{ij} is an expression involving the derivatives of π^0 to degree 2 and derivatives of κ^* to degree 1. By construction of u^* from (5), the coefficient of $\frac{\partial^2 \kappa^*_\alpha}{\partial x_i \partial x_j}$ in (10) vanishes at (x^*,u^*) and we can therefore compute $\frac{\partial^2 \pi^*}{\partial x_i \partial x_j}(x^*)$ directly as

$$\frac{\partial^2 \pi^*}{\partial x_i \partial x_j}(x^*) = S_{ij}(x^*,u^*).$$

Next, to get equations for $\frac{\partial^2 \kappa^*_\beta}{\partial x_j \partial x_k}$, we differentiate (8) with respect to x_k yielding

$$0 = M_{\alpha\beta}\frac{\partial^2 \kappa^*_\beta}{\partial x_j \partial x_k} + T_{jk} \tag{11}$$

where T_{jk} is an expression involving the derivatives of π^0 to degree 3 and derivatives of κ^* to degree 1. The number of unknowns $\frac{\partial^2 \kappa^*_\beta}{\partial x_j \partial x_k}$ is $m\frac{n(n+1)}{2}$ and (11) produces $m\frac{n(n+1)}{2}$ equations. Assuming that the symmetric matrix $M(x,u) \in \mathbb{R}^{m\times m}$ with entries $M_{\alpha\beta}(x,u)$ is invertible at (x^*,u^*), we can solve uniquely for the unknowns $\frac{\partial^2 \kappa^*_\beta}{\partial x_j \partial x_k}$ from (11).

To compute the higher-order terms of (π^*,κ^*) at x^*, assume by induction that we have computed the derivatives of π^* and κ^* at $x = x^*$ to degree $d-1$. To compute the d order derivatives of π^*, say $\frac{\partial^d \pi^*}{\partial x_I}$, where $I = (i_1, i_2, \ldots, i_d) \in \{1,2,\ldots,n\}^d$ is a multi-index and we use the notation $\frac{\partial^d \pi^*}{\partial x_I} = \frac{\partial^d \pi^*}{\partial x_{i_1}\cdots\partial x_{i_d}}$, we apply $\frac{\partial^d}{\partial x_I}$ to (4) and obtain by induction an expression of the form

$$\frac{\partial^d \pi^*}{\partial x_I} = S_I + \left[\frac{\partial \pi^0}{\partial x_a}\frac{\partial f_a}{\partial u_\alpha} + \frac{\partial \ell}{\partial u_\alpha}\right]\frac{\partial^d \kappa^*_\alpha}{\partial x_I} \tag{12}$$

where S_I is an expression involving the derivatives of π^0 to degree d and derivatives of κ^* to degree $d-1$. By construction of u^* from (5), the coefficient of $\frac{\partial^d \kappa^*_\alpha}{\partial x_I}$ in (12) vanishes at (x^*,u^*) and we can therefore compute $\frac{\partial^d \pi^*}{\partial x_I}(x^*)$ directly as

$$\frac{\partial^d \pi^*}{\partial x_I}(x^*) = S_I(x^*,u^*). \tag{13}$$

Next, to compute $\frac{\partial^d \kappa^*}{\partial x_I}$ at x^*, we apply $\frac{\partial^d}{\partial x_I}$ to (5) and obtain by induction an expression of the form

$$0 = M_{\alpha\beta}\frac{\partial^d \kappa^*_\beta}{\partial x_I} + T_I \tag{14}$$

where T_I is an expression involving the derivatives of π^0 to degree $d+1$ and derivatives of κ^* to degree $d-1$. The number of unknowns $\frac{\partial^d \kappa^*_\beta}{\partial x_I}$ is $m\binom{n+d-1}{d}$ and (14) produces the same number of equations. Assuming that the symmetric matrix $M(x,u) \in \mathbb{R}^{m\times m}$ with entries $M_{\alpha\beta}(x,u)$ is invertible at (x^*,u^*), we can solve uniquely for $\frac{\partial^d \kappa^*_\beta}{\partial x_I}$ from (14).

Finally, for consistency with the order of π^0, we continue computing π^* to degree $d+1$. This can be done easily without the need to compute κ^* to degree $d+1$ since from (12) the $d+1$ order derivatives of κ^* will vanish from the equations for the $d+1$ order derivatives of π^*. In summary, we have proved the following.

Theorem 2.1: *Let (π^0,κ^0) be the Taylor polynomial approximations to (π,κ) at $x=0$ to degrees $d+1$ and $d \ge 1$, respectively. Suppose that $(x^*,u^*) \in \mathbb{R}^n \times \mathbb{R}^m$ satisfies*

$$0 = \frac{\partial \pi^0}{\partial x}(f(x^*,u^*))\frac{\partial f}{\partial u}(x^*,u^*) + \frac{\partial \ell}{\partial u}(x^*,u^*)$$

and that the $m \times m$ symmetric matrix M given by (9) *is invertible at (x^*,u^*). Then the Taylor coefficients at x^* of (π^*,κ^*) solving* (4)-(5) *can be computed order-by-order to degrees $d+1$ and d, respectively.*

Having computed (π^*,κ^*) to degrees $d+1$ and d, respectively, we can extend the initial approximation (π^0,κ^0) by defining a domain on which (π^*,κ^*) will be accepted and adjoining it to $\mathcal{P}^0$. The domain of (π^*,κ^*) will radiate outward from $\mathcal{S}^1$. This process can then be repeated at distinct points on $\mathcal{S}^1$ in such a way that the initial domain $\mathcal{P}^0$ is covered by the patch domains of the newly computed approximations. In the next section we give the details of this process.

III. Extending the initial polynomial approximation

A. Level one extension

In this section we describe how to extend the initial polynomial approximations (π^0,κ^0) defined on $\mathcal{P}^0$ to an extended domain $\mathcal{P}^0 \cup \mathcal{P}^1$, where $\mathcal{P}^1$ radiates outward from the boundary $\mathcal{S}^1 = \partial\mathcal{P}^0$ and surrounds $\mathcal{P}^0$ in the sense that $\mathcal{P}^0 \cap \mathcal{P}^1 = \partial\mathcal{P}^0$.

Let $\mathcal{S}^{1,1},\ldots,\mathcal{S}^{1,p_1}$ be a partition of the boundary $\mathcal{S}^1$ such that each $\mathcal{S}^{1,j}$ has a non-empty interior relative to the subspace topology on $\mathcal{S}^1 \subset \mathbb{R}^n$. Suppose that the algorithm described in §II has been executed at distinct points $x^{1,j} \in \mathcal{S}^{1,j} \cap (\partial\mathcal{S}^{1,j})^c$, for $j = 1,\ldots,p_1$, resulting in the polynomial approximations $(\pi^{1,1},\kappa^{1,1}),\ldots,(\pi^{1,p_1},\kappa^{1,p_1})$ centered at $x^{1,1},\ldots,x^{1,p_1}$ of degrees $d+1$ and d, respectively. We call the points $x^{1,j}$ *patch points*. We assume that the image of $\mathcal{S}^{1,j}$ under the corresponding closed-loop dynamics

$$x^+ = f^{1,j}(x) := f(x,\kappa^{1,j}(x))$$

is contained in the interior of $\mathcal{P}^0$.

We now describe how to construct a domain $\mathcal{P}^{1,j}$ for each new solution $(\pi^{1,j}, \kappa^{1,j})$. In words, the domain $\mathcal{P}^{1,j}$ will be the union of $\mathcal{S}^{1,j}$, lateral boundaries radiating from $\mathcal{S}^{1,j}$, and an outer-most boundary contained in the level set $\{x \mid \pi^{1,j}(x) = c_2\}$, where $c_2 > c_1$. For $z \in \mathcal{S}^{1,j}$ let

$$v^{1,j}(z) := \frac{z - f^{1,j}(z)}{\|z - f^{1,j}(z)\|},$$

that is, $-v^{1,j}(z)$ is the direction vector from z to its image under the closed-loop dynamics. To build the domain $\mathcal{P}^{1,j}$ of $(\pi^{1,j}, \kappa^{1,j})$, we take each $z \in \mathcal{S}^{1,j}$ and follow the curve

$$t \mapsto x(t; z) := (f^{1,j})^{-1}(f^{1,j}(z) + v^{1,j}(z)t) \qquad (15)$$

in positive time until reaching for the first time the level set $\{x \mid \pi^{1,j}(x) = c_2\}$. In other words,

$$\mathcal{P}^{1,j} = \bigcup_{z \in \mathcal{S}^{1,j}} \left\{x(t; z) \mid \pi^{1,j}(x(t; z)) \leq c_2,\ t \in [0, t_z]\right\}$$

where $t_z = \min\{t > 0 \,|\, \pi^{1,j}(x(t; z)) = c_2\}$. Let

$$\mathcal{S}^2 = \bigcup_{j=1}^{p_1} \left\{x(t_z; z) \mid z \in \mathcal{S}^{1,j}\right\}.$$

By construction, $\mathcal{P}^{1,j} \cap \mathcal{S}^1 = \mathcal{S}^{1,j}$, and thus we define $\mathcal{S}^{1,j}$ as the *inner boundary* of $\mathcal{P}^{1,j}$, $\mathcal{P}^{1,j} \cap \mathcal{S}^2$ as the *outer boundary* of $\mathcal{P}^{1,j}$, and the remaining component of $\partial\mathcal{P}^{1,j}$ as the *lateral boundary* of $\mathcal{P}^{1,j}$. In practice, the lateral boundaries between patches $\mathcal{P}^{1,j}$ will not generally match or will $\mathcal{S}^2$ be a smooth hypersurface. Hence, it will in general be necessary to redefine the patch domains $\mathcal{P}^{1,j}$ to avoid overlaps between adjacent patches. In any case, we can augment to the original polynomial approximation (π^0, κ^0) defined on $\mathcal{P}^0$ the domains $\mathcal{P}^{1,j}$ and the corresponding approximations $(\pi^{1,j}, \kappa^{1,j})$, for $j = 1, \ldots, p_1$, thereby obtaining a piecewise smooth approximation to π and κ defined on $\mathcal{P}^0 \cup \mathcal{P}^1$, where $\mathcal{P}^1 := \cup_{j=1}^{p_1} \mathcal{P}^{1,j}$.

Remark 3.1: The computation of the curve (15) is facilitated by the fact that it satisfies an ODE. Indeed, we have

$$\begin{aligned}\frac{\partial x}{\partial t}(t; z) &= \mathbf{D}((f^{1,j})^{-1})(f^{1,j}(z) + v^{1,j}(z)t)v^{1,j}(z) \\ &= \mathbf{D}((f^{1,j})^{-1})(f^{1,j}(x(t; z)))v^{1,j}(z) \\ &= (\mathbf{D}f^{1,j}(x(t; z)))^{-1}v^{1,j}(z).\end{aligned}$$

Hence, we can compute the curve $t \mapsto x(t; z)$ using standard high-order numerical ODE solvers that require only *evaluations* of the mapping $x \mapsto (\mathbf{D}f^{1,j}(x))^{-1}v^{1,j}(z)$, such as Runge-Kutta methods. For example, up-to first order

$$x(t; z) \approx z + (\mathbf{D}f^{1,j}(z))^{-1}v^{1,j}(z)t$$

where

$$\mathbf{D}f^{1,j}(z) = \frac{\partial f}{\partial x}(z, \kappa^{1,j}(z)) + \frac{\partial f}{\partial u}(z, \kappa^{1,j}(z))\frac{\partial \kappa^{1,j}}{\partial x}(z)$$

can be easily computed. □

Having extended the initial approximations (π^0, κ^0) to $\mathcal{P}^0 \cup \mathcal{P}^1$, in the next section we develop an iterative procedure to extend it further beyond the outer boundary of $\mathcal{P}^1$.

B. Level two and beyond extensions

Suppose that we have extended the initial polynomial approximation (π^0, κ^0) defined on $\mathcal{P}^0$ to $\mathcal{P}^0 \cup \mathcal{P}^1 \cup \cdots \cup \mathcal{P}^r$, $r \geq 1$, and we wish to extend it further in the radial direction from the outer-most boundary $\mathcal{S}^{r+1}$ of $\partial\mathcal{P}^r$. The *patch level domains* $\mathcal{P}^i$, for $i = 1, \ldots, r$, are the union of patches $\mathcal{P}^{i,j}$, $j = 1, \ldots, p_i$, with $p_i \leq p_{i+1}$. In what follows, for notational consistency we define $\mathcal{P}^{0,1} := \mathcal{P}^0$, $\kappa^{0,1} := \kappa^0$, and $\pi^{0,1} := \pi^0$, and $p_0 = 1$.

We begin by partitioning $\mathcal{S}^{r+1}$ into sets $\mathcal{S}^{r+1,1}, \ldots, \mathcal{S}^{r+1,p_{r+1}}$ and choose distinct points $x^{r+1,j} \in \mathcal{S}^{r+1,j}$ not on the boundaries of $\mathcal{S}^{r+1,j}$. More precisely, the sets $\mathcal{S}^{r+1,j}$ are the result of partitioning the outer boundaries of $\mathcal{P}^{r,1}, \ldots, \mathcal{P}^{r,p_r}$ so that each $\mathcal{S}^{r+1,j} \subset \mathcal{P}^{r,\sigma_j}$ for some unique $\sigma_j \in \{1, 2, \ldots, p_r\}$. For example, a trivial partition of $\mathcal{S}^{r+1}$ would involve taking the outer boundaries of $\mathcal{P}^{r,1}, \ldots, \mathcal{P}^{r,p_r}$ to serve as the $\mathcal{S}^{r+1,1}, \ldots, \mathcal{S}^{r+1,p_{r+1}}$, that is, $\mathcal{S}^{r,j} = \mathcal{S}^{r+1} \cap \mathcal{P}^{r,j}$, and thus $p_{r+1} = p_r$. In §IV we describe an adaptive method for partitioning $\mathcal{S}^{r+1}$ that takes into account the error of the currently computed solution. In any case, we assume that each $\mathcal{S}^{r+1,j}$ is mapped into the interior of $\mathcal{P}^0 \cup \mathcal{P}^1 \cup \cdots \cup \mathcal{P}^r$ under the corresponding closed-loop dynamics. In other words, $x \in \mathcal{S}^{r+1,j}$ implies that $f(x, \kappa^{r,\sigma_j}(x))$ is in the interior of $\mathcal{P}^0 \cup \mathcal{P}^1 \cup \cdots \cup \mathcal{P}^r$. Now, for each $x^{r+1,j}$ there exists a unique $\mathcal{P}^{\alpha_j,\beta_j}$, with $0 \leq \alpha_j \leq r$ and $1 \leq \beta_j \leq p_{\alpha_j}$, such that $y^{r+1,j} := f(x, \kappa^{r,\sigma_j}(x)) \in \mathcal{P}^{\alpha_j,\beta_j}$ and $y^{r+1,j}$ does not lie in the outer boundary of $\mathcal{P}^{\alpha_j,\beta_j}$. Hence, to compute a new polynomial approximation $(\pi^{r+1,j}, \kappa^{r+1,j})$ centered at $x^{r+1,j}$ using the algorithm in §II, we ask that

$$\pi^{r+1,j}(x) = \pi^{\alpha_j,\beta_j}(f(x, \kappa^{r+1,j}(x))) + \ell(x, \kappa^{r+1,j}(x)). \qquad (16)$$

The first order necessary condition for a minimum then becomes

$$\begin{aligned}0 = \frac{\partial \pi^{\alpha_j,\beta_j}}{\partial x}(f(x, \kappa^{r+1,j}(x)))\frac{\partial f}{\partial u}(x, \kappa^{r+1,j}(x)) \\ + \frac{\partial \ell}{\partial u}(x, \kappa^{r+1,j}(x)). \qquad (17)\end{aligned}$$

We can then use (16)-(17) and the algorithm in §II to compute polynomial approximations to $(\pi^{r+1,j}, \kappa^{r+1,j})$ of degrees $d+1$ and d, respectively.

We now construct patch domains $\mathcal{P}^{r+1,j}$ for each new solution $(\pi^{r+1,j}, \kappa^{r+1,j})$. As before, the patch $\mathcal{P}^{r+1,j}$ will be the union of $\mathcal{S}^{r+1,j}$, lateral boundaries radiating from $\mathcal{S}^{r+1,j}$, and an outer-most boundary contained in the level set $\{x \mid \pi^{r+1,j}(x) = c_{r+1}\}$, where $c_{r+1} > c_r$. Let $f^{r+1,j}(x) := f(x, \kappa^{r+1,j}(x))$ denote the closed-loop dynamics. As in the case of the level one extension, to build the domain $\mathcal{P}^{r+1,j}$ we take each $z \in \mathcal{S}^{r+1,j}$ and follow the curve

$$x(t; z) = (f^{r+1,j})^{-1}(f^{r+1,j}(z) + tv^{r+1,j}(z))$$

in positive time until reaching for the first time the level set

$\{x \mid \pi^{r+1,j}(x) = c_{r+1}\}$. In other words,

$$\mathcal{P}^{r+1,j} = \bigcup_{z \in \mathcal{S}^{r+1,j}} \{x(t;z) \mid \pi^{r+1,j}(x(t;z)) \le c_{r+1}, \quad t \in [0, t_z]\}$$

where $t_z = \min\{t > 0 \mid \pi^{r+1,j}(x(t;z)) = c_{r+1}\}$. Let

$$\mathcal{S}^{r+2} = \bigcup_{j=1}^{p_{r+1}} \left\{x(t_z;z) \mid z \in \mathcal{S}^{r+1,j}\right\}.$$

By construction, $\mathcal{P}^{r+1,j} \cap \mathcal{S}^{r+1} = \mathcal{S}^{r+1,j}$, and thus we define $\mathcal{S}^{r+1,j}$ as the inner boundary of $\mathcal{P}^{r+1,j}$, $\mathcal{P}^{r+1,j} \cap \mathcal{S}^{r+2}$ as the outer boundary of $\mathcal{P}^{r+1,j}$, and the remaining component of $\partial\mathcal{P}^{r+1,j}$ as the lateral boundary of $\mathcal{P}^{r+1,j}$. We now augment to the running approximation defined on $\mathcal{P}^0 \cup \mathcal{P}^1 \cup \cdots \cup \mathcal{P}^r$ the domains $\mathcal{P}^{r+1,j}$ and the corresponding approximations $(\pi^{r+1,j}, \kappa^{r+1,j})$, for $j = 1, \ldots, p_{r+1}$, thereby extending the approximations to $\mathcal{P}^0 \cup \mathcal{P}^1 \cup \cdots \cup \mathcal{P}^{r+1}$, where $\mathcal{P}^{r+1} := \cup_{j=1}^{p_{r+1}} \mathcal{P}^{r+1,j}$

The final piecewise-smooth approximations to (π, κ), denoted $(\pi_{\text{pch}}, \kappa_{\text{pch}})$, are given by

$$\pi_{\text{pch}}(x) = \pi^{i,j}(x), \quad \text{if } x \in \mathcal{P}^{i,j} \cap (\mathcal{S}^{i+1})^c$$

$$\kappa_{\text{pch}}(x) = \kappa^{i,j}(x), \quad \text{if } x \in \mathcal{P}^{i,j} \cap (\mathcal{S}^{i+1})^c$$

where $0 \le i \le r+1$ and $1 \le j \le p_i$.

IV. Adaptive partitioning of outer boundaries

In this section we outline an adaptive method for partitioning the outer boundaries of a newly constructed domain level $\mathcal{P}^r$. The main advantage of the method, compared to a pre-determined partitioning scheme, is to reduce the number of patch points at where the algorithm in §II is executed, and to determine the regions of the state space where the error of the computed solutions is growing more rapidly.

Suppose that the rth level domain $\mathcal{P}^r$ has been computed and we seek to extend the approximation from the outer boundary $\mathcal{S}^{r+1}$ of $\partial\mathcal{P}^r$. The boundary $\mathcal{S}^{r+1}$ is the union of the outer boundaries of $\mathcal{P}^{r,1}, \ldots, \mathcal{P}^{r,p_r}$. Therefore, to construct the $(r+1)$ level domain $\mathcal{P}^{r+1}$, we first need to partition the outer boundary of each $\mathcal{P}^{r,j}$. This can be done in a pre-determined manner. For example, we can partition each outer boundary of $\mathcal{P}^{r,j}$ into two sets so that the number of patches from level-to-level doubles. Instead, one could partition the outer boundary of $\mathcal{P}^{r,j}$ in an adaptive way by considering how well the solutions $(\pi^{r,j}, \kappa^{r,j})$ satisfy Bellman's equation (4). To this end, we define the *relative error* $\rho^{r,j} : \mathcal{P}^{r,j} \to \mathbb{R}$ by

$$\rho^{r,j}(x) = \frac{|\pi^{r,j}(x) - \pi^{\alpha_j,\beta_j}(f(x, \kappa^{r,j}(x))) - \ell(x, \kappa^{\alpha_j,\beta_j}(x))|}{\pi^{r,j}(x)}$$

where $0 \le \alpha_j \le r$, $1 \le \beta_j \le p_{\alpha_j}$, and $\mathcal{P}^{\alpha_j,\beta_j}$ is the patch domain that contains the image of the patch point $x^{r,j}$ under the closed-loop dynamics. The total relative error on the outer boundary of $\mathcal{P}^{r,j}$ is given as

$$s_{r,j} = \int_{\mathcal{P}^{r,j} \cap \mathcal{S}^{r+1}} \rho^{r,j}(x)\, dx$$

and the total relative error on the outer boundary of $\mathcal{P}^r$ is $s_r = \sum_{j=1}^{p_r} s_{r,j}$. If $\frac{s_{r,j}}{s_r} \approx \frac{1}{p_r}$, for all $j = 1, \ldots, p_r$, then the computed solutions $(\pi^{r,j}, \kappa^{r,j})$ all contribute approximately the same relative error on the outer boundaries of their domains. In this case, if the relative errors $s_{r,j}$ are within some desired tolerance level $\varepsilon_r > 0$, we can simply use the outer boundaries of the $\mathcal{P}^{r,j}$ as the inner boundaries of the patches on the next level domain $\mathcal{P}^{r+1}$, and therefore $p_r = p_{r+1}$. If, on the other hand, the coefficient of variation cv of the distribution of the relative errors $s_{r,j}$ is above some desired maximum tolerance, say $cv \ge 1$, then for those $s_{r,j}$ such that $\frac{s_{r,j}}{s_r} > \frac{1}{p_r} m_r$, for some chosen $m_r > 1$, we can partition the outer boundary of $\mathcal{P}^{r,j}$ into two sets of approximately equal size. This process can then be iterated on the newly created partitions until we have obtained a final partition of each outer boundary of $\mathcal{P}^{r,j}$, and consequently a partition $\mathcal{S}^{r+1,1}, \ldots, \mathcal{S}^{r+1,p_{r+1}}$ of the boundary $\mathcal{S}^{r+1}$. With this method, the number of sets used to partition the outer boundary of each $\mathcal{P}^{r,j}$ will vary. In particular, the outer boundaries of those patches $\mathcal{P}^{r,j}$ where the relative error is growing rapidly will be partitioned into more sets than those where the relative error is growing more slowly.

V. An example

To test the accuracy of our patchy algorithm, we can apply it to a system for which the optimal cost π and control κ are known so that we can compare the patchy approximation to the true solution. With this in mind, we test our algorithm on a nonlinear system that is equivalent to a linear one under a smooth transformation. To this end, consider the linear system

$$z^+ = \begin{bmatrix} 1 & \frac{1}{10} \\ 0 & 1 \end{bmatrix} z + \begin{bmatrix} 0 \\ \frac{1}{10} \end{bmatrix} u$$

and stage cost $\ell(z,u) = \frac{1}{20}(z_1^2 + z_2^2 + u^2)$. The optimal cost function π and optimal regulator κ are given by $\pi(z) = \frac{1}{2} z' P z$ and $\kappa(z) = Kz$, where (P, K) solve the associated DARE. Consider the change of coordinates

$$z = (\phi_1(x), \phi_2(x)) := \left(x_1, x_2 + \sin(x_1) e^{-x_1^2/100}\right).$$

In the x coordinates, the system becomes

$$x_1^+ = \phi_1(x) + \phi_2(x)$$
$$x_2^+ = \phi_2(x) - \sin(\phi_1(x) + \phi_2(x)) e^{-(\phi_1(x)+\phi_2(x))^2/100}$$

and the stage cost becomes $\ell(x,u) = \frac{1}{20}(\phi_1(x)^2 + \phi_2(x)^2 + u^2)$. The optimal cost function and optimal control in the x-coordinates are $\pi(x) = \frac{1}{2}\phi(x)' P \phi(x)$ and $\kappa(x) = K\phi(x)$, respectively.

We computed patchy approximations $(\pi_{\text{pch}}, \kappa_{\text{pch}})$ to (π, κ) using initial polynomial approximations (π^0, κ^0) of degrees 4 and 3 respectively, i.e., $d = 3$, and $N = 40$ patch levels. The cost levels c_r were chosen as $c_r = (0.3 + (r-1)0.03)^2$, for $r = 1, \ldots, N$. The method of adaptive partitioning of the outer boundaries as described in §IV was performed with the parameters $cv = 2$ for the first 20 patch levels and $cv = 3$ for the remaining levels, and $m_r = 1.5$ for $r = 1, \ldots, N$.

The number of patch points on the initial level was chosen as $p_1 = 32$ and the resulting number of patch points on the last patch level was $p_N = 324$.

In Fig. 1 we plot the error in approximation π with the polynomial π^0, and in Fig. 2 we plot the error in approximating π with the patchy approximation π_{pch}. As can be seen from Fig. 1-2, the maximum error $\pi(x) - \pi^0(x)$ is approximately 1.5, whereas the maximum error $\pi(x) - \pi_{\text{pch}}(x)$ is approximately 4×10^{-3}. In Fig. 3 we plot the exact solution π together with the polynomial approximation π^0, and in Fig. 4 we plot the exact solution π together with the patchy approximation π_{pch}.

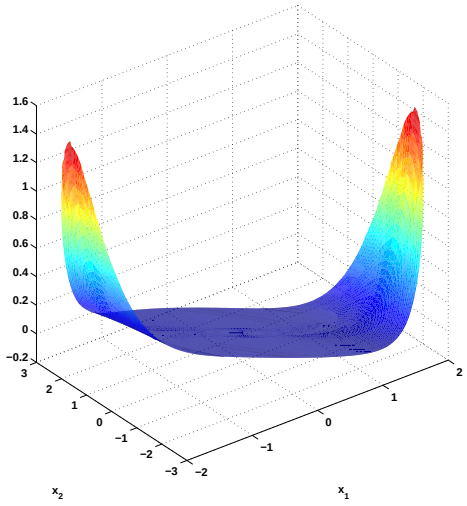

Fig. 1. Error $\pi(x) - \pi^0(x)$ with π^0 a degree four polynomial. The maximum error using π^0 is approximately 1.5.

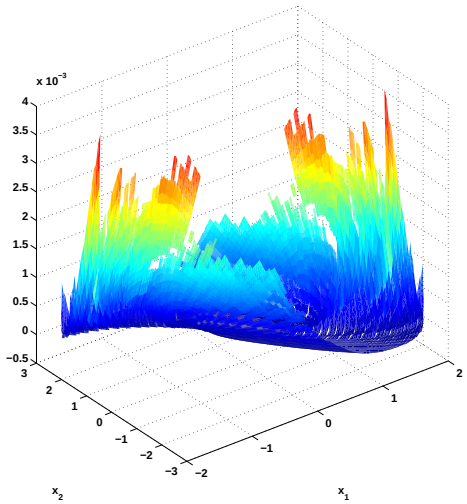

Fig. 2. Error $\pi(x) - \pi_{\text{pch}}(x)$ using $N = 40$ patch levels. The maximum error using π_{pch} is approximately 4×10^{-3}.

VI. CONCLUSION

In this paper we presented some preliminary results on a numerical method to compute high-order solutions to Bellman's dynamic programming equation of optimal control

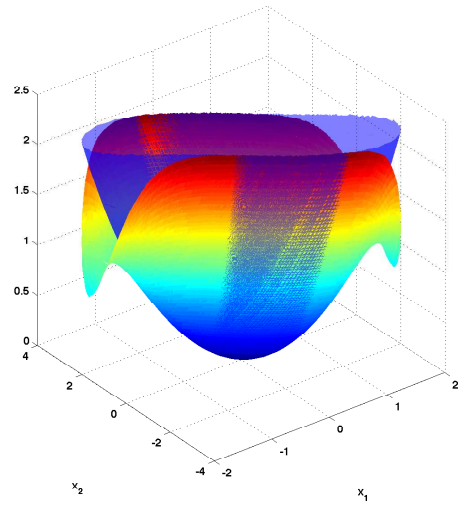

Fig. 3. Exact cost function π (transparent blue) and polynomial approximation π^0 (color gradient) of degree four.

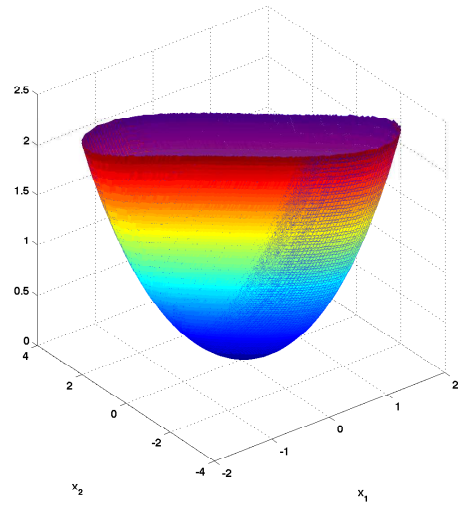

Fig. 4. Exact cost function π (transparent blue) and patchy approximation π_{pch} (color gradient) using $N = 40$ patch levels.

regulation. In a forthcoming paper, we intend to perform an error analysis of the approximation method.

REFERENCES

[1] E.G. Al'brekht *On the optimal stabilization of nonlinear systems*, Journal of Applied Mathematics and Mechanics, 25 (1961), pp. 1254-1266.

[2] R. Bellman, *Introduction to the Mathematical Theory of Control Processes*, Vol. II, Academic Press, New York, NY, 1971.

[3] F. Lewis, *Optimal Control*, John Wiley & Sons, 1986.

[4] C. Navasca, *Local solutions of the Dynamic Programming Equations and the Hamilton–Jacobi–Bellman PDEs*, PhD Thesis, University of California, Davis, 2002.

[5] C. Navasca and A.J. Krener, *Patchy Solution of the HJB PDE*, In A. Chiuso, A. Ferrante and S. Pinzoni, eds, Modeling, Estimation and Control, Lecture Notes in Control and Information Sciences, 364, pp. 251-270, 2007.