Domain Decomposition for Stochastic Optimal Control

Matanya B. Horowitz   Ivan Papusha   Joel W. Burdick

***Abstract*— This work proposes a method for solving linear stochastic optimal control (SOC) problems using sum of squares and semidefinite programming. Previous work had used polynomial optimization to approximate the value function, requiring a high polynomial degree to capture local phenomena. To improve the scalability of the method to problems of interest, a domain decomposition scheme is presented. By using local approximations, lower degree polynomials become sufficient, and both local and global properties of the value function are captured. The domain of the problem is split into a non-overlapping partition, with added constraints ensuring C^1 continuity. The Alternating Direction Method of Multipliers (ADMM) is used to optimize over each domain in parallel and ensure convergence on the boundaries of the partitions. This results in improved conditioning of the problem and allows for much larger and more complex problems to be addressed with improved performance.**

I. Introduction

Motion planning in the presence of noise and dynamics remains a central issue in robotics and autonomous systems. As robots transition out of controlled factory and lab environments, the ability to move precisely in the presence of unknown environments, exterior agency, and stochastic actuators and sensors become ever more important. For a solution to be useful it must be rapid to compute, robust, and incorporate optimality criteria. The primary avenue for solving motion planning problems, and likely the most successful historically, has been that of sampling based planners [1]. Such approaches are attractive as they may be quite rapid in practice, but typically only have guarantees in the asymptotic limit, and incorporate dynamics and stochasticity in only a limited way.

Stochastic optimal control (SOC) provides an alternative, allowing for the full dynamics and various details of the problem to be incorporated into the algorithm directly. Traditionally, this has been handled through discretization, resulting in the formulation of a Markov Decision Problem (MDP), which can then be solved through methods such as value iteration [2]. These methods have met with a great deal of success in a number of communities. The caveat is that such problems in robotics may be prohibitively difficult to solve due to a number of obstacles, chiefly the *curse of dimensionality*. These techniques rely on a fine discretization of the state space when the system occupies a continuous domain, typical of many robotic and control problems. Furthermore, robotic state spaces are usually quite large, both in quantity of dimensions as well as absolute size, for all but the most academic of problems, resulting in discrete state space cardinality that may easily exceed the capabilities of current computers. Reducing the necessity for fine discretization could provide for significant gains in this area.

Recently it has been discovered that the Hamilton Jacobi Bellman (HJB) equation, a typically nonlinear partial differential equation (PDE) that arises in optimal control, may be transformed to a linear PDE given several mild assumptions. This is a large computational gain, as solving the nonlinear PDE is quite difficult [3]. Research into leveraging this computational advantage is only beginning.

One method to solve such problems lies in recent results from polynomial optimization and semidefinite programming [4]. These methods allow for optimization to be performed directly over polynomials, and have solved a number of difficult problems. Here we present a novel use of such tools to directly construct an approximate value function that satisfies the linear HJB equation. This allows for optimal control problems, including those typically found in robotic motion planning, to be solved relatively quickly and globally. In contrast to dynamic programming approaches, no direct state space discretization is required, postponing the curse of dimensionality and eliminating a potential source of approximation error.

In particular, we propose an augmentation of the algorithm first presented in [5], in which the domain is split into distinct partitions, each of which has its own local approximating polynomial. The value function may vary significantly over the domain, and thus may require a high degree polynomial if approximated over the domain's entirety. But by using a sufficiently local approximation, a similar quality of global approximation may be achieved with smaller degree on each partition. Furthermore, we demonstrate that an efficient choice of partitioning may lead to a decoupling in the optimal control problems on each partition, allowing for a degree of parallelization. The Alternating Direction Method of Multipliers (ADMM) [6] is a particularly well suited approach, providing a principled method for parallelization of certain convex problems with convergence guarantees.

A. Related Work

Linearly solvable SOC problems have recently been studied from two avenues. One is Linear MDPs [7], in which an MDP may be solved as a linear set of equations given several assumptions. By taking the continuous limit of the discretization, a linear PDE is obtained. Additionally, following the work begun by Kappen [8], the same linear PDE has been found through a particular transformation of the

I. Papusha was supported by a Department of Defense NDSEG Fellowship. The authors are with the Control and Dynamical Systems Department, California Institute of Technology, Pasadena, CA USA. The corresponding author may be reached at mhorowit@caltech.edu

HJB. The existing research has tended towards developing sampling based approaches for solving the resulting linear PDE. This is done through the use of the Feynman–Kac Lemma, that allows for a linear PDE to be solved by examining the diffusion of a stochastic process. Feynman–Kac approaches have been further developed by Theodorou et al. [9] into a path integral framework in use with dynamic motion primitives. These results have grown in a number of compelling directions, either relying on an MDP or sampling based approach [10], [9], [11].

Sampling based approaches are an alternative to the approach presented here, with several potential advantages and disadvantages. Among these, sampling based approaches such as that of Theodorou et al. may be more amenable in high dimensional state spaces. Such a comparison in part motivates the present work.

Effort has also gone towards solving the linear HJB directly, as well as exploiting its properties for computational benefit. In [12] it is shown that the property of superposition may be used to compute optimal control solutions at essentially zero computational cost, with significant implications in solving Linear Temporal Logic (LTL) specified tasks. The work of [13] leverages recent results in sparse tensor decompositions to formulate a numerical technique that scales *linearly* with dimension, allowing for the HJB to be approximately solved for a twelve dimensional system. Finally, in [14] connections are made to a broader literature, such as *navigation* functions (popular in robotics), problems of moments, and broader classes of linear PDEs.

The sum of squares approach presented here is connected via duality to problems of moments. By examining the moments of the HJB, an alternative line of work by Lasserre et al. [15], [16], [17] also reduces optimal control to a semidefinite optimization problem. In their work, the solution and the optimality conditions are integrated against monomial test functions, producing an infinite set of moment constraints. By truncating to any finite list of monomials, the optimal control problem is reduced to one of semidefinite optimization. Their method is more general, applicable to any system with polynomial nonlinearities. Our method contrasts in that we propose candidate solutions of the value function, and thus avoid the need to include the control signal in our polynomial basis, lessening the computational burden. We are also able to avoid consideration of initial and final conditions and measures. Perhaps most importantly, our use of the linear HJB allows for both upper and lower pointwise bounds to be constructed to the true solution.

Domain partitioning is an approach that has long been used in numerical methods for PDEs, from the local analysis behind the Finite Element Method to multi-scale decomposition techniques [18]. In control, these techniques have also arisen to improve local approximation to Lyapunov functions [19], and is complimentary to approaches that approximate nonlinear systems as piecewise-affine (PWA) [20]. Our work may be seen as a continuation of this research theme, seeking to extend these techniques not only to the study of stability, as is the case for Lyapunov functions, but to control as well. Furthermore, the ability to obtain general solutions to HJB has implications in regards to Control Lyapunov Functions [21], allowing for stabilization to be shown alongside near-optimality. Our method has the distinct advantage over PWA approximations in that the system itself is not approximated, and the full nonlinear dynamics are incorporated into the solution.

B. Paper Outline

The ability to perform domain decomposition for stochastic optimal control will rely on three main ideas: linear stochastic optimal control, sum of squares programming, and finally the ADMM algorithm. We begin in Section II by reviewing the linear HJB. In Section III, we develop the technique first presented in [5] to approximately solve the linear HJB using convex programming via a sum of squares relaxation. Finally, we build the domain decomposition procedure in Section IV. The need to enforce constraints on the boundaries between partitions then gives rise to our use of ADMM, which is reviewed in Section IV-A and then applied to the problem at hand, the main contribution of this paper. We illustrate each step on a simple nonlinear example in Section VI, before tackling a more sophisticated example in Section VII. Finally, we discuss some of the merits of the technique and future directions in Section VIII.

II. The Linear Hamilton Jacobi Bellman Equation

We begin by constructing the value function, which captures the *cost-to-go* from a given state. If such a quantity is known, an optimal action is chosen to follow the quantity's gradient, bringing the agent into states with lowest cost over the remaining time horizon. We define $x_t \in \mathbb{R}^n$ as the system state at time t, control input $u_t \in \mathbb{R}^m$, and dynamics that evolve according to the equation

$$dx_t = \left(f\left(x_t\right) + G\left(x_t\right) u_t\right) dt + B\left(x_t\right) d\omega_t \tag{1}$$

on a compact domain Ω, where the expressions $f(x)$, $G(x)$, $B(x)$ are assumed to be smoothly differentiable, but possibly nonlinear functions, and ω_t is a zero mean Gaussian noise process with covariance Σ_ϵ. The system has cost r_t accrued at time t according to

$$r\left(x_t, u_t\right) = q\left(x_t\right) + \frac{1}{2} u_t^T R u_t \tag{2}$$

where $q(x)$ is a state dependent cost. We require $q(x) \geq 0$ for all x in the problem domain. The goal is to minimize the expectation of the cost functional

$$J(x, u) = \phi_T\left(x_T\right) + \int_0^T r\left(x_t, u_t\right) dt, \tag{3}$$

where ϕ_T represents a state-dependent terminal cost. The solution to this minimization is obtained from the *value function*. For an initial point x_0, it is given by

$$V\left(x_0\right) = \min_{u_{[0,T]}} \mathbb{E}\left[J\left(x_0\right)\right], \tag{4}$$

where we use the shorthand $u_{[0,T]}$ to denote the trajectory of $u(t)$ over the time interval $t \in [0, T]$.

The associated Hamilton Jacobi Bellman equation, arising from dynamic programming arguments [3], is

$$-\partial_t V = \min_u \left(r + (\nabla_x V)^T f + \frac{1}{2}\operatorname{Tr}\left((\nabla_{xx} V) G \Sigma_\epsilon G^T \right) \right). \tag{5}$$

As the control effort enters quadratically into the cost function, it is a simple matter to solve for it analytically by substituting (2) into (5) and taking the gradient, yielding

$$u^* = -R^{-1} G^T (\nabla_x V). \tag{6}$$

The optimal control u^* may then be substituted into (5) to yield the following nonlinear, second order PDE

$$-\partial_t V = q + (\nabla_x V)^T f - \frac{1}{2} (\nabla_x V)^T G R^{-1} G^T (\nabla_x V) + \frac{1}{2}\operatorname{Tr}\left((\nabla_{xx} V) B \Sigma_\epsilon B^T \right). \tag{7}$$

The difficulty of solving this PDE has traditionally prevented the value function from being solved for directly. However, as has recently been found in [22], [7], if there exists a scalar $\lambda > 0$ and a control penalty cost $R \in \mathbb{R}^{n\times n}$ satisfying the noise assumption

$$\lambda G(x) R^{-1} G(x)^T = B(x) \Sigma_\epsilon B(x)^T \triangleq \Sigma_t, \tag{8}$$

then the logarithmic transformation

$$V = -\lambda \log \Psi \tag{9}$$

allows us to obtain, after substitution and simplification, the following linear PDE from equation (7),

$$-\partial_t \Psi = -\frac{1}{\lambda} q \Psi + f^T (\nabla_x \Psi) + \frac{1}{2}\operatorname{Tr}\left((\nabla_{xx}\Psi) \Sigma_t \right). \tag{10}$$

Through the transformation Ψ, which we call here the *desirability* [7], we obtain a computationally appealing method from which to compute the value function V.

Remark 1. *The noise assumption* (8) *can roughly be interpreted as a controllability-type condition: the system controls must span (or counterbalance) the effects of input noise on the system dynamics. A degree of designer input is also given up, as the constraint restricts the design of the control penalty R, requiring that control effort be highly penalized in subspaces with little noise, and lightly penalized in those with high noise. Additional discussion may be found in [7].*

The boundary conditions of (10) correspond to the exit conditions of the optimal control problem. This may correspond to colliding with an obstacle or goal region, and in the finite horizon problem there is the added boundary condition of the terminal cost at $t = T$. These final costs must then be transformed according to (9), producing added boundary conditions to (10).

Linearly solvable optimal control is not limited to the finite horizon setting. Similar analysis can be performed to obtain linear HJB PDEs for infinite horizon average cost, and first-exit settings, with the corresponding cost functionals and PDEs shown in Table I. For convenience, we define the differential operator

TABLE I

LINEAR DESIRABILITY PDE FOR VARIOUS STOCHASTIC OPTIMAL CONTROL SETTINGS, FROM [7].

	Cost Functional	Desirability PDE
Finite	$\phi_T(x_T) + \int_0^T r(x_t,u_t)dt$	$\frac{1}{\lambda}q\Psi - \frac{\partial\Psi}{\partial t} = L(\Psi)$
First-Exit	$\phi_{T_*}(x_{T_*}) + \int_0^T r(x_t,u_t)dt$	$\frac{1}{\lambda}q\Psi = L(\Psi)$
Average	$\lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\int_0^T r(x_t,u_t)dt\right]$	$\frac{1}{\lambda}q\Psi - c\Psi = L(\Psi)$

$$L(\Psi) := f^T (\nabla_x \Psi) + \frac{1}{2}\operatorname{Tr}\left((\nabla_{xx}\Psi) \Sigma_t \right). \tag{11}$$

III. THE SUM OF SQUARES RELAXATION

Building upon the results of [5], we relax the equality constraint (10), allowing for an over-approximation of the value function, and creating a linear differential inequality. This places the problem within the realm of polynomial optimization problems where tools such as the *Positivstellensatz* may be applied. Consider the relaxation

$$\frac{1}{\lambda} q\Psi \geq \partial_t \Psi + f^T (\nabla_x \Psi) + \frac{1}{2}\operatorname{Tr}\left((\nabla_{xx}\Psi) \Sigma_t \right). \tag{12}$$

Given that this is an approximation, we wish to obtain the best such approximation for a given polynomial order for Ψ, minimizing the pointwise error as our objective,

$$\begin{aligned} \text{min.} \quad & \gamma \\ \text{s.t.} \quad & \gamma - \left(\frac{1}{\lambda} q\Psi - (\partial_t \Psi + L(\Psi)) \right) \geq 0. \end{aligned}$$

Furthermore, due to the nature of the log transformation (9), we require Ψ to be positive everywhere, and we will examine this problem only on a compact, semialgebraic domain $\mathbb{S}$.

The complete (centralized) optimization problem is

$$\begin{aligned} \text{min.} \quad & \gamma && \\ \text{s.t.} \quad & \frac{1}{\lambda} q\Psi \geq \partial_t \Psi + L(\Psi) && x \in \mathbb{S} \\ & \gamma \geq \frac{1}{\lambda} q\Psi - \partial_t \Psi - L(\Psi) && x \in \mathbb{S} \\ & \Psi \geq e^{-\frac{\phi_T(x)}{\lambda}} && x \in \partial\mathbb{S} \\ & \gamma \geq \Psi - e^{-\frac{\phi_T(x)}{\lambda}} && x \in \partial\mathbb{S} \end{aligned} \tag{13}$$

The inequalities are interpreted pointwise over $x \in \mathbb{S}$. This set of polynomial inequalities motivates our need for a method to enforce non-negativity constraints over a polynomial directly.

A. Sum of Squares Review

We provide a brief review of sum of squares (SOS) programming, with additional technical details available in [23], [17]. These tools will be key in the development of approximate solutions to (13).

Formally, a *semialgebraic set* is a subset of $\mathbb{R}^n$ that is specified by a finite number of polynomial equations and inequalities. An example is the set

$$\left\{(x_1, x_2) \in \mathbb{R}^2 \mid x_1^2 + x_2^2 \leq 1, x_1^3 - x_2 \leq 0\right\}.$$

Such a set is not necessarily convex, and testing membership in the set is intractable in general [23]. As we will see, however, there exists a class of semialgebraic sets that are in fact semidefinite-representable. Key to this development is the ability to test for non-negativity of a polynomial.

A multivariate polynomial $f(x)$ is a *sum of squares* (SOS) if there exist polynomials $f_1(x), \ldots, f_m(x)$ such that

$$f(x) = \sum_{i=1}^{m} f_i^2(x).$$

A seemingly unremarkable observation is that a sum of squares is always positive. Thus, a sufficient condition for non-negativity of a polynomial is that the polynomial is SOS. Perhaps less obvious is that membership in the set of SOS polynomials may be tested as a convex problem. We denote the function $f(x)$ being SOS as $f(x) \in \Sigma(x)$.

Theorem 1. *([23]) Given a finite set of polynomials* $\{f_i\}_{i=0}^m \in \mathbb{R}[x]$ *the existence of* $\{a_i\}_{i=1}^m \in \mathbb{R}$ *such that*

$$f_0 + \sum_{i=1}^{m} a_i f_i \in \Sigma(x)$$

is a semidefinite programming feasibility problem.

Here, $\mathbb{R}[x]$ denotes the set of polynomials over x for some fixed degree. Thus, while the problem of testing non-negativity of a polynomial is intractable in general, by constraining the feasible set to SOS the problem becomes tractable. The converse question of whether a non-negative polynomial is necessarily a sum of squares is unfortunately false, indicating that this test is conservative [23]. Nonetheless, SOS feasibility is sufficiently powerful for our purposes.

B. The Positivstellensatz

At this point it is possible to determine whether a particular polynomial, possibly parameterized, is a sum of squares. The next step is to determine how to combine multiple polynomial inequalities. The answer is given by the theorem that has come to be known as Stengle's *Positivstellensatz*.

Theorem 2 (Stengle's Positivstellensatz [24])**.** *The set*

$$X = \big\{x \mid f_i(x) \geq 0,\ h_j(x) = 0 \\ \text{for all } i = 1, \ldots, m, j = 1, \ldots, p\big\}$$

is empty if and only if there exists $t_i \in \mathbb{R}[x]$*, and* $s_i, r_{ij}, \ldots \in \Sigma[x]$ *such that*

$$-1 = s_0 + \sum_i h_i t_i + \sum_i s_i f_i + \sum_{i \neq j} r_{ij} f_i f_j + \cdots$$

This powerful theorem allows for (13) to incorporate the domain requirements $x \in \mathbb{S}$ and $x \in \partial\mathbb{S}$.

IV. Domain Decomposition

We first briefly review ADMM before demonstrating its use in domain decomposition, following [6].

A. Alternating Direction Method of Multipliers

The Alternating Direction Method of Multipliers (ADMM) will serve as the basis for enforcing continuity and differentiability of $\Psi(x)$ on the boundaries of the decomposed regions. Other decomposition schemes are possible, see [25], [26] for a survey. ADMM is a "meta"-optimization scheme, where each step is carried out by solving a convex optimization problem. Consider the optimization

$$\begin{aligned} \text{min.} \quad & f(x) + g(z) \\ \text{s.t.} \quad & Ax + Bz = c \end{aligned} \tag{14}$$

over real vector variables x and z, with convex functions f and g. Define an augmented Lagrangian

$$L_\rho = f(x) + g(z) + y^T(Ax + Bz - c) + \frac{\rho}{2}\|Ax + Bz - c\|_2^2,$$

where $\rho > 0$ is an algorithm parameter, and y is the dual variable associated with the equality constraint. The constrained optimization is solved through alternately minimizing the augmented Lagrangian over the primal variables x, z, and updating the dual variable y,

$$\begin{aligned} x^{k+1} &:= \text{argmin}_x L_\rho(x, z^k, y^k) \\ z^{k+1} &:= \text{argmin}_z L_\rho(x^{k+1}, z, y^k) \\ y^{k+1} &:= y^k + \rho\left(Ax^{k+1} + Bz^{k+1} - c\right). \end{aligned}$$

The sum of squares formalism allows a general polynomial optimization problem to be converted to a sequence of SDPs, where the variables are the polynomial coefficients. ADMM extends readily to SDPs. To that end, consider

$$\begin{aligned} \text{min.} \quad & f(x) + g(z) \\ \text{s.t.} \quad & Ax + Bz = c \\ & x \in \mathcal{C}_1, \quad z \in \mathcal{C}_2, \end{aligned}$$

where $x, z \in \mathbb{R}^n$ are the variables and $\mathcal{C}_1, \mathcal{C}_2$ are SDP-representable sets. With the same form L_ρ, the ADMM iterations are quadratically penalized SDPs,

$$\begin{aligned} x^{k+1} &:= \text{argmin}_{x \in \mathcal{C}_1} L_\rho(x, z^k, y^k) \\ z^{k+1} &:= \text{argmin}_{z \in \mathcal{C}_2} L_\rho(x^{k+1}, z, y^k) \\ y^{k+1} &:= y^k + \rho\left(Ax^{k+1} + Bz^{k+1} - c\right). \end{aligned}$$

The only difference is the primal variables are now constrained to lie in the spectrahedra (the convex set of semidefinite constraints [27]) $\mathcal{C}_1$ and $\mathcal{C}_2$.

The value in this decomposition is the attendant convergence guarantees obtained with ADMM. In particular, we will make the following two assumptions, which guarantee convergence:

Assumption 1. *The (extended real valued) functions* $f : \mathbb{R}^n \to \mathbb{R} \cup +\infty$ *and* $g : \mathbb{R}^m \to \mathbb{R} \cup +\infty$ *are closed, proper, and convex.*

Assumption 2. *The unaugmented Lagrangian has a saddle point.*

If it can be demonstrated that the optimization problem obeys these assumptions, then the following general theorem becomes available:

Theorem 3. *(See [6]) Given Assumptions 1, 2 then the ADMM iterates satisfy the following:*

- ***Residual convergence****: $r^k \to 0$ as $k \to \infty$, i.e., the iterates approach feasibility*
- ***Objective convergence****: $f(x^k)+g(z^k) \to p^*$ as $k \to \infty$, i.e., the objective function of the iterates approaches the optimal value*
- ***Dual variable convergence****: $y^k \to y^*$ as $k \to \infty$, where y^* is a dual optimal point*

B. Decomposition of Stochastic Optimal Control

As the optimal control problem is assumed to take place over a compact state space, the domain of (10) may decomposed into finitely many regions $\mathcal{R}_j \subseteq \mathbb{R}^n$, $j = 1, \ldots, N_R$. Assuming the pairwise boundary between the regions may be described in terms of a semialgebraic set, we have the following result,

Theorem 4. *Given desirability function $\Psi_i(x)$ valid on region $\mathcal{R}_i$, $\Psi_j(x)$ valid on region $\mathcal{R}_j$, and shared boundary $\xi = \{x \mid h(x) = 0\}$ between $\mathcal{R}_i$ and $\mathcal{R}_j$, we have $\Psi_i(x) = \Psi_j(x)$ on ξ if there exists $c(x) \in \mathbb{R}[x]$ such that*

$$\Psi_i(x) - \Psi_j(x) + c(x)h(x) = 0$$

Proof. A straightforward result of the Positivstellensatz, see [28] for details. □

Similarly, continuity of the n-th derivative may be easily incorporated as well by imposing equality of the derivative along the boundary.

C. Two Region Explicit Example

In the following analysis, we demonstrate how this result can be used to bind together optimization problems over a decomposed domain. To obtain a useful policy, we will require the combined policy to be C^1 continuous.

For clarity, we examine a pair of bordering partitions $\mathcal{R}_1$ and $\mathcal{R}_2$, with shared boundary $h(x)$. The polynomials are assumed to be of bounded degrees, with $\deg(\Psi_i(x))$ bounded by d and $\deg(c_i(x))$ by $d-k$, for all i, j. In this case,

$$\begin{aligned}
\Psi_1(x) &= \alpha_0 + \alpha_1 x + \cdots + \alpha_d x^d \\
\Psi_2(x) &= \beta_0 + \beta_1 x + \cdots + \beta_d x^d \\
c_1(x) &= \theta_0 + \theta_1 x + \cdots + \theta_{d-k} x^{d-k} \\
c_2(x) &= \mu_0 + \mu_1 x + \cdots + \mu_{d-k} x^{d-k},
\end{aligned}$$

where $h(x) = \rho_0 + \rho_1 x + \cdots + \rho_k x^k$ defines the boundary region. The continuity constraint

$$\Psi_1(x) - \Psi_2(x) + c_1(x)h(x) = 0$$

is equivalent to the coefficient matching constraints

$$\begin{aligned}
0 &= \alpha_0 - \beta_0 + (\theta_0\rho_0) \\
0 &= \alpha_1 - \beta_1 + (\theta_0\rho_1 + \theta_1\rho_0) \\
0 &= \alpha_2 - \beta_2 + (\theta_0\rho_2 + \theta_1\rho_1 + \theta_2\rho_0) \\
&\;\;\vdots \\
0 &= \alpha_d - \beta_d + (\theta_{d-k}\rho_k).
\end{aligned}$$

Note that the coefficient matching constraints are affine in the decision variables α_i, β_i, $i = 1, \ldots, d$, and θ_j, μ_j, $j = 1, \ldots, d-k$. The derivative constraint (21) appends additional coefficient matching constraints,

$$\begin{aligned}
0 &= \alpha_1 - \beta_1 + (\mu_0\rho_0) \\
0 &= 2\alpha_2 - 2\beta_2 + (\mu_0\rho_1 + \mu_1\rho_0) \\
0 &= 3\alpha_2 - 3\beta_2 + (\mu_0\rho_2 + \mu_1\rho_1 + \mu_2\rho_0) \\
&\;\;\vdots \\
0 &= d\alpha_d - d\beta_d + (\mu_{d-k}\rho_k).
\end{aligned}$$

Continuity of higher order derivatives are incorporated similarly. The continuity and derivative coefficient matching constraints, together with the approximation error constraint (22), can be aggregated into matrix form,

$$A^{(1)}z_1 + A^{(2)}z_2 = 0,$$

where $z_1 = (\alpha_0, \ldots, \theta_{d-k}, \gamma_1)$ are the coefficients associated with $\mathcal{R}_1$, and $z_2 = (\beta_0, \ldots, \mu_{d-k}, \gamma_2)$ are the coefficients associated with $\mathcal{R}_2$. It is now straightforward to incorporate the affine matrix constraint into a dual decomposition scheme. The decomposed variant of optimization (13) is

$$\begin{aligned}
\text{min. } & \gamma_1 + \gamma_2 && (15)\\
\text{s.t. } & \frac{1}{\lambda}q\Psi_1 \geq \partial_t\Psi_1 + L(\Psi_1), \quad x \in \mathcal{R}_1 && (16)\\
& \frac{1}{\lambda}q\Psi_2 \geq \partial_t\Psi_2 + L(\Psi_2), \quad x \in \mathcal{R}_2 && (17)\\
& \gamma_1 - \left(\frac{1}{\lambda}q\Psi_1 - rhs\right) \geq 0, \quad x \in \mathcal{R}_1 && (18)\\
& \gamma_2 - \left(\frac{1}{\lambda}q\Psi_2 - rhs\right) \geq 0, \quad x \in \mathcal{R}_2 && (19)\\
& \Psi_1(x) - \Psi_2(x) + c_1(x)x = 0 && (20)\\
& \frac{\partial\Psi_1}{\partial x}(x) - \frac{\partial\Psi_2}{\partial x}(x) + c_2(x)x = 0 && (21)\\
& \gamma_1 = \gamma_2 && (22)
\end{aligned}$$

where the Positivstellensatz is used to enforce the domain restrictions (see [5] for details). The coupling constraints (20) and (21) prevent decomposition into two parallel optimizations. In addition, the objective is coupled through the equality constraint (22), which ensures that the maximum pointwise approximation error over any region is no more than $\gamma^{\max} = \gamma_1 = \gamma_2$.

To wit, define the quadratically penalized Lagrangian

$$\begin{aligned}
L_\rho(\gamma_1, z_1, \gamma_2, z_2, \lambda) = \gamma_1 + \gamma_2 + \mathcal{I}_{\mathcal{C}_1}(z_1) + \mathcal{I}_{\mathcal{C}_2}(z_2) + \\
+ \lambda^T(A^{(1)}z_1 + A^{(2)}z_2) + \frac{\rho}{2}\left\| A^{(1)}z_1 + A^{(2)}z_2 \right\|_2^2,
\end{aligned}$$

where $\mathcal{I}_{\mathcal{C}_i}(z_i)$ is the indicator function of the optimization problem over each individual partition, obtained by reduction of (13) to semidefinite program form [4]. The alternating direction iteration may then be performed as

$$(\gamma_1^{k+1}, z_1^{k+1}) := \arg\min_{\gamma_1, z_1} L_\rho(\gamma_1, z_1, \gamma_2^k, z_2^k, \lambda^k) \tag{23}$$

$$(\gamma_2^{k+1}, z_2^{k+1}) := \arg\min_{\gamma_2, z_2} L_\rho(\gamma_1^{k+1}, z_1^{k+1}, \gamma_2, z_2, \lambda^k) \tag{24}$$

$$\lambda^{k+1} := \lambda^k + \rho(A^{(1)} z_1^{k+1} + A^{(2)} z_2^{k+1}). \tag{25}$$

The above procedure may be repeated for all partitions $\mathcal{R}_i$ and $\mathcal{R}_j$ that share a common boundary. Each minimization, a semidefinite program, is taken over only those constraints associated with the specified region. This achieves a degree of decoupling, limiting the size of the polynomial optimization problem, and thus the semidefinite program, for each individual partition.

D. Parallelization

A further decoupling may be achieved through a judicious choice of domain partitions. This idea is well known in the partial differential equation community [18]. Suppose partitions $\mathcal{R}_i$ and $\mathcal{R}_j$ share no common border $h_{i,j}(x)$. As variables from disjoint partitions are only shared through the common boundary constraints (20), it is straightforward to see that z_i^{k+1} and z_j^{k+1} are independent of one another. This allows for these optimizations to be performed in parallel. One valid partition is to decompose the domain into a checkerboard pattern, separating the domain into shaded and unshaded tiles. As shaded tiles share no optimization variables with one another, they may be optimized in parallel, and similar with the unshaded. By alternating between shaded and unshaded, the correct descent direction continues to be taken, guaranteeing convergence. See [29] for a detailed discussion of parallelization ideas, and Fig. 1 for an illustration of this beneficial decomposition pattern.

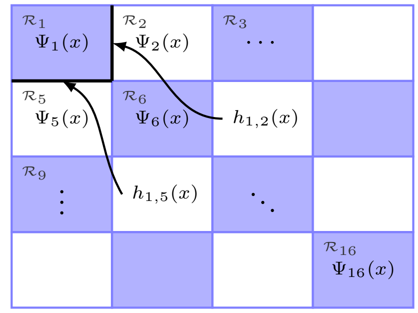


Fig. 1. A particular grid domain decomposition with the partitions grouped into shaded and unshaded sets. As the sets of the same color require no consensus over their local variables, it is possible to perform the optimization over each set in parallel while maintaining the convergence properties of ADMM.

V. Analysis

A benefit of the sum of squares relaxation approach is that the solutions produced are guaranteed to be upper and lower bounds (depending on the direction of the inequality (12)) when performed over a single partition [5]. These guarantees are retained in the domain decomposition setting.

Theorem 5. *Given a solution set $\{\Psi_i, \gamma_i\}$ to the converged optimization problem* (23) *where C^2 continuity is enforced, and if Ψ^* is the solution to* (10), *then $\Psi(x) \geq \Psi^*(x)$ for all $x \in \mathcal{R}_i$.*

Proof. (Sketch) The derivation follows the proof of Theorem 5 in [5] with little modification. The only modification arises from the fact that the elliptic and parabolic maximum principles rely on C^2 continuity of the super-solution. As the solution is polynomial on the interior of each boundary, and therefore infinitely differentiable, this requirement needs only be enforced explicitly along the partition boundaries. □

A benefit of this approach is that not only may an upper bound be computed, but in fact reversing the inequalities of the optimization results in an additional optimization problem that can be used to find a pointwise lower bound to the underlying optimal solution. As both upper and lower bounds are available, it is possible to see the maximal possible error of the solution. See [5] for details.

VI. Scalar Example

We construct the optimization for a simple scalar example for illustrative purposes. Consider the one dimensional system

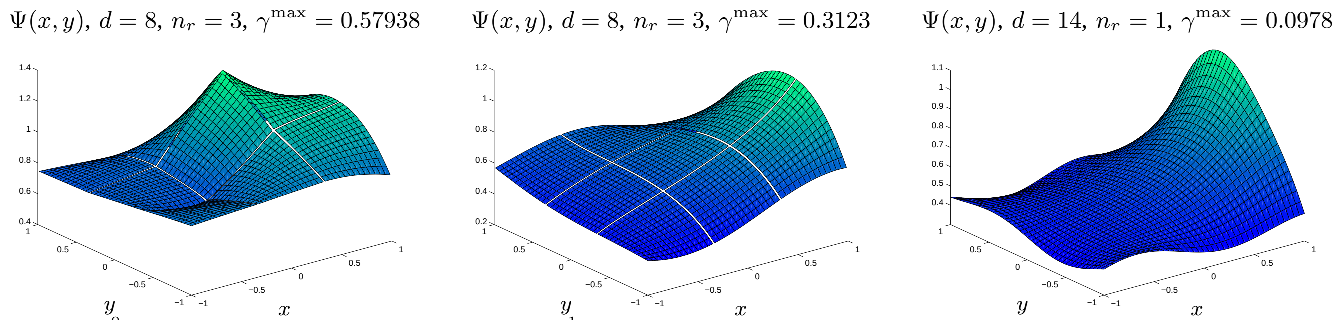
$$dx = (x^2 + u)\, dt + d\omega$$

on the domain $x \in [-1, 1]$. We have state cost $q(x) = 1$, control cost $R = 1$, and parameter $\lambda = 1$. We split the domain into regions $\mathcal{R}_1 = \{x \mid x \in [-1, 0]\}$, $\mathcal{R}_2 = \{x \mid x \in [0, 1]\}$, creating $h_{1,2}(x) = x$. For each of these problems we form the optimization (13) on $\mathcal{R}_1$, $\mathcal{R}_2$ independently. To enforce equality of both the solution and its derivative at the shared point $x = 0$ we add the coupling constraints

$$\Psi_1(x) - \Psi_2(x) + c_1(x)x = 0$$
$$\frac{\partial \Psi_1}{\partial x}(x) - \frac{\partial \Psi_2}{\partial x}(x) + c_2(x)x = 0.$$

To enforce the continuity constraint (20) for the point boundary at the origin, it suffices to match the constant coefficients of Ψ_1 and Ψ_2, i.e., we require $\Psi_1(0) = \Psi_2(0)$. This is an affine constraint when the polynomial optimization is passed to an SDP.

Numerical results for the one dimensional example are shown in Fig. 2 and Fig. 3. For simplicity, the conditioning parameter was set to $\rho = 1$, and the polynomial degree bound to 6 for each region. Fig. 2 shows that within about ten steps of ADMM, continuous differentiability at the boundary region $x = 0$ is achieved. Fig. 3 shows the evolution of the dual variables, as well as the maximum approximation gap with iteration number. The SDP optimization on each region was carried out on SDPT3 using YALMIP with the Sum of Squares module [30].



(a) C^0-continuous approximation (b) C^1-continuous approximation (c) high fidelity approximation

Fig. 4. Results of multidimensional, nonlinear example.

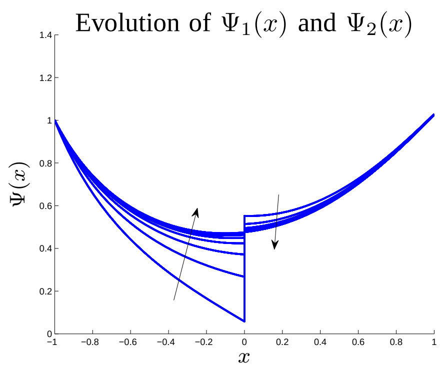


Fig. 2. Evolution of the alternative value function over 10 ADMM steps. Arrows show direction of evolution.

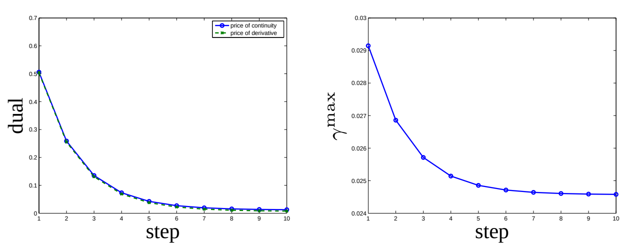


Fig. 3. Values of the dual variables (left) and maximum approximation gap (right) with iteration number.

VII. Nonlinear Cartesian System

To demonstrate the versatility of the method, a nonlinear, multidimensional problem was solved with the following dynamics,

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \left(0.1 \begin{bmatrix} -2x - x^3 - 5y - y^3 \\ 6x + x^3 - 3y - y^3 \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}\right) dt + \begin{bmatrix} d\omega_1 \\ d\omega_2 \end{bmatrix}.$$

The problem is framed as a first exit problem, with the three sides of a square domain $\mathbb{S} = [-1, 1^2]$ given a unit penalty $\phi(x,y) = 1$, while on the remaining edge at $x = 1$ a reward was given for achieving the center of the edge with $\phi(x,y) = 1-(y-1)^2$. Representative alternative value function approximations are shown in Fig. 4. In Table II we also summarize the maximum approximation gap $\gamma^{\max}$ for a checkerboard decomposition of $\mathbb{S}$ with n_r regions per dimension, and approximating polynomial degree bound d in each region.

TABLE II

SLACK VALUE $\gamma^{\max}$ AS A FUNCTION OF POLYNOMIAL DEGREE d, AND NUMBER OF REGIONS n_r PER DIMENSION.

	d					
n_r	4	6	8	10	12	14
1	6.8374	2.5085	0.6344	0.3501	0.0804	0.0978
2	6.7065	2.1561	0.6399	0.3642	0.0859	
3	6.4688	2.0579	0.5794	0.3304		
4	6.2662	2.0689	0.5591	0.3005		
5	6.6289	1.8812	0.5919	0.2917		
6	6.3017	1.7638	0.5716			
7	6.3178	1.6533	0.5403			

VIII. Conclusion

A method to perform domain decomposition on stochastic optimal control problems has been developed, allowing for local polynomial approximations to the Hamilton Jacobi Bellman equation to be generated in parallel. Of importance is the fact that the sum of squares relaxation used does not fundamentally rely on the particular structure of the HJB PDE. In fact, [5] demonstrates that the technique may be readily applied to any linear parabolic or elliptic PDE to obtain guaranteed upper and lower bounds over the domain. The domain splitting of this work extends as well, allowing for local upper and lower bounds to any linear PDE to be generated via optimization. While more involved than existing numerical techniques such as the Finite Element method, these techniques have formal guarantees that do not require an asymptotic limit in discretization mesh size.

A more direct implication lies in the generation of stabilizing controllers for nonlinear systems. Until now, there has not existed a method to generate near-optimal Control Lyapunov Functions for arbitrary nonlinear, stochastic systems [21]. These domain decomposition techniques improve the ability for optimal control policies to respond to system dynamics, enlarging the class of systems that can be handled. Furthermore, existing results on sum of squares in Lyapunov functions can be used to verify the stability of any policy produced by these decomposition methods.

A. Future Directions

It is straightforward to recognize that many domain decompositions, such as the checkerboard pattern illustrated, produce highly structured sparsity patterns in the semidefinite program's constraint matrices. Such sparsity structures have previously been used to significantly improve the computational cost of large scale semidefinite and sum of squares programs [31], [32], work that could easily be applied here as well. It is also an interesting question as to how sparse basis functions [13] might be incorporated into the domain decomposition approach.

REFERENCES

[1] S. M. LaValle, *Planning Algorithms*, 2006.

[2] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 3rd ed. Athena Scientific, 2005, vol. I.

[3] W. H. Fleming and H. M. Soner, *Controlled Markov Processes and Viscosity Solutions*. New York: Springer, July 2006, vol. 25.

[4] P. A. Parrilo, "Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization," Ph.D. dissertation, California Institute of Technology, 2000.

[5] M. B. Horowitz and J. W. Burdick, "Semidefinite relaxations for stochastic optimal control policies," in *American Control Conference (ACC)*, June 2014, pp. 3006–3012.

[6] S. P. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[7] E. Todorov, "Efficient computation of optimal actions," *Proceedings of the National Academy of Sciences*, vol. 106, no. 28, pp. 11 478–11 483, 2009.

[8] H. J. Kappen, "Path integrals and symmetry breaking for optimal control theory," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 11, p. P11011, Nov. 2005.

[9] E. A. Theodorou, J. Buchli, and S. Schaal, "A generalized path integral control approach to reinforcement learning," *Journal of Machine Learning Research*, vol. 11, pp. 3137–3181, 2010.

[10] E. A. Theodorou and E. Todorov, "Relative entropy and free energy dualities: Connections to path integral and KL control," in *Proceedings of the 51st IEEE Conference on Decision and Control (CDC)*, 2012, pp. 1466–1473.

[11] K. Dvijotham and E. Todorov, "A unified theory of linearly solvable optimal control," *Artificial Intelligence (UAI)*, p. 1, 2011.

[12] M. B. Horowitz, E. Wolff, and R. M. Murray, "A compositional approach to stochastic optimal control with co-safe temporal logic," *Intelligent Robots and Systems (IROS)*, 2014.

[13] M. B. Horowitz, A. Damle, and J. W. Burdick, "Linear Hamilton Jacobi Bellman equations in high dimensions," in *IEEE Conference on Decision and Control (CDC)*, 2014.

[14] M. B. Horowitz, "Efficient methods for stochastic optimal control," Ph.D. dissertation, California Institute of Technology, 2014.

[15] J. B. Lasserre, *Moments, Positive Polynomials and Their Applications*. World Scientific, 2009, vol. 1.

[16] J. B. Lasserre, D. Henrion, C. Prieur, and E. Trélat, "Nonlinear optimal control via occupation measures and LMI-relaxations," *SIAM Journal on Control and Optimization*, vol. 47, no. 4, pp. 1643–1666, 2008.

[17] J. B. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM Journal on Optimization*, vol. 11, no. 3, pp. 796–817, 2001.

[18] P. Bjorstad and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, 2004.

[19] M. Johansson and A. Rantzer, "Computation of piecewise quadratic Lyapunov functions for hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 555–559, 1998.

[20] P. Biswas, P. Grieder, J. Löfberg, and M. Morari, "A survey on stability analysis of discrete-time piecewise affine systems," in *Proceedings of the 16th IFAC World Congress*, 2005.

[21] E. D. Sontag, "A Lyapunov-like characterization of asymptotic controllability," *SIAM Journal on Control and Optimization*, vol. 21, no. 3, pp. 462–471, May 1983.

[22] H. Kappen, "Linear theory for control of nonlinear stochastic systems," *Physical Review Letters*, vol. 95, no. 20, p. 200201, Nov. 2005.

[23] P. A. Parrilo, "Semidefinite programming relaxations for semialgebraic problems," *Mathematical Programming*, vol. 96, no. 2, pp. 293–320, May 2003.

[24] G. Stengle, "A Nullstellensatz and a Positivstellensatz in semialgebraic geometry," *Mathematische Annalen*, vol. 207, no. 2, pp. 87–97, June 1974.

[25] D. P. Bertsekas, "Multiplier methods: a survey," *Automatica*, vol. 12, no. 2, pp. 133–145, Mar. 1976.

[26] ——, *Constrained Optimization and Lagrange Multiplier Methods*. Athena Scientific, 1996.

[27] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[28] S. Prajna and A. Papachristodoulou, "Analysis of switched and hybrid systems–beyond piecewise quadratic methods," vol. 4, pp. 2779–2784, 2003.

[29] N. Parikh and S. P. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2014.

[30] J. Löfberg, "Pre- and post-processing sum-of-squares programs in practice," *IEEE Transactions on Automatic Control*, vol. 54, no. 5, pp. 1007–1011, 2009.

[31] D. Henrion, J.-B. Lasserre, and J. Löfberg, "GloptiPoly 3: Moments, optimization and semidefinite programming," *Optimization Methods & Software*, vol. 24, no. 4-5, pp. 761–779, 2009.

[32] H. Waki, S. Kim, M. Kojima, M. Muramatsu, and H. Sugimoto, "Algorithm 883: Sparsepop—a sparse semidefinite programming relaxation of polynomial optimization problems," *ACM Transactions on Mathematical Software (TOMS)*, vol. 35, no. 2, p. 15, 2008.