Separable Dynamic Programming and Approximate Decomposition Methods

Dimitri P. Bertsekas

***Abstract*—We consider control, planning, and resource allocation problems involving several independent subsystems that are coupled through a control/decision constraint. We discuss one-step lookahead methods that use an approximate cost-to-go function derived from the solution of single subsystem problems. We propose a new method for constructing such approximations, and derive bounds on the performance of the associated suboptimal policies. We then specialize this method to problems of reachability of target tubes that have the form of a box (a Cartesian product of subsystem tubes). We thus obtain inner approximating tubes, which are the union of a finite number of boxes, each involving single subsystem calculations.**

***Index Terms*—Dynamic programming, optical control, reachability, separable problems.**

I. Introduction

We consider a stochastic optimal control problem, which involves M independent subsystems that are coupled through a control/decision constraint. The mth subsystem is described by

$$x_{k+1}^m = f_k^m\left(x_k^m, u_k^m, w_k^m\right), \qquad m = 1, \ldots, M, \quad k = 0, 1, \ldots, N-1 \tag{1.1}$$

where x_k^m is the state taking values in some set, u_k^m is the control, w_k^m is a random disturbance, f_k^m is a given function, and N is the number of stages. We assume a finite horizon, although the discussion also applies in part to infinite horizon problems where the stage costs are discounted. We assume that w_k^m is selected according to a probability distribution that may depend on x_k^m and u_k^m, but not on prior disturbances or the disturbances of other subsystems w_k^j, $j \neq m$. The cost incurred at the kth time period by the mth subsystem is $g_k^m(x_k^m, u_k^m, w_k^m)$, where g_k^m is a given real-valued function. There is also a terminal cost $g_N^m(x_N^m)$ that depends on the terminal state x_N^m. To avoid mathematical technicalities, we assume that w_k^m takes values in a finite set, but the following discussion applies qualitatively to more general cases.

We assume perfect state information, i.e., that the control vector $u_k = (u_k^1, \ldots, u_k^M)$ is chosen with knowledge of the entire current state $x_k = (x_k^1, \ldots, x_k^M)$. However, the components u_k^m of the control vector are coupled through constraints of the form

$$u_k = \left(u_k^1, \ldots, u_k^M\right) \in U_k, \qquad k = 0, 1, \ldots, N-1$$

where the sets U_k are given. We are interested in policies $\pi = \{\mu_0, \ldots, \mu_{N-1}\}$ mapping states to controls and satisfying the control constraints. Thus each μ_k must satisfy for all states x_k

$$\mu_k(x_k) \in U_k, \qquad k = 0, 1, \ldots, N-1. \tag{1.2}$$

The total cost of a policy π is the expected value of the corresponding cost

$$J_0^\pi(x_0) = E\left\{\sum_{m=1}^{M}\left(g_N^m\left(x_N^m\right) + \sum_{k=0}^{N-1} g_k^m\left(x_k^m, \mu_k^m(x_k), w_k^m\right)\right)\right\}.$$

Given an initial condition x_0, we wish to minimize $J_0^\pi(x_0)$ over all admissible policies π, i.e., over all policies that satisfy the constraints (1.2).

Dynamic programming (DP for short) is the principal method for addressing the problem, and its key concept is the cost-to-go of a policy π starting from a state x_k at time k given by

$$J_k^\pi(x_k) = E\left\{\sum_{m=1}^{M}\left(g_N^m\left(x_N^m\right) + \sum_{t=k}^{N-1} g_t^m\left(x_t^m, \mu_t^m(x_t), w_t^m\right)\right)\right\}.$$

The optimal cost-to-go starting from x_k at k is

$$J_k(x_k) = \min_\pi J_k^\pi(x_k)$$

and it is assumed that $J_k^\pi(x_k)$ and $J_k(x_k)$ are finite for all x_k, π, and k. The optimal cost-to-go functions J_k satisfy the following dynamic programming recursion:

$$J_k\left(x_k^1, \ldots, x_k^M\right) = \min_{\left(u_k^1, \ldots, u_k^M\right) \in U_k} E\left\{\sum_{m=1}^{M} g_k^m\left(x_k^m, u_k^m, w_k^m\right) + J_{k+1}\left(f_k^1\left(x_k^1, u_k^1, w_k^1\right), \ldots, f_k^M\left(x_k^M, u_k^M, w_k^M\right)\right)\right\} \tag{1.3}$$

for all $x_k = (x_k^1, \ldots, x_k^M)$ and k, with the initial condition

$$J_N(x_N) = \sum_{m=1}^{M} g_N^m\left(x_N^m\right).$$

In this note, we place special emphasis on control constraint sets that are Cartesian products of the form

$$U_k^1 \times \cdots \times U_k^M, \qquad k = 0, 1, \ldots, N-1. \tag{1.4}$$

We refer to a set of this form as a *box*. If all the control constraint sets U_k were boxes, the optimal cost-to-go functions would be separable of the form

$$J_k\left(x_k^1, \ldots, x_k^M\right) = \sum_{m=1}^{M} J_k^m\left(x_k^m\right) \tag{1.5}$$

and an optimal policy would be decomposable (i.e., the control of each subsystem would depend only on the local state of that subsystem). The reason is that for box constraints, there is no competition for control resource between the subsystems, and the minimization in the DP algorithm decomposes into a separate minimization for each subsystem. However, in the general case where there is coupling between the controls $u_k^1, \ldots, u_k^M$, imposed by the control constraints $u_k \in U_k$, the functions J_k are not separable, and this greatly complicates the DP computations. This has motivated an approximate decomposition approach, first proposed in [18] (see also [16] and [8, ex. 6.3.8]), whereby each control constraint set U_k is replaced by a box. The corresponding single subsystem subproblems are solved to obtain an approximation of the form (1.5) to the optimal cost function, which is used in turn in a one-step lookahead scheme to obtain a suboptimal control policy.

Manuscript received February 10, 2006; revised January 13, 2007. Recommended by Associate Editor I. Paschalidis.

The author is with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: dimitrib@mit.edu).

Digital Object Identifier 10.1109/TAC.2007.895901

Kimemia focused on production control problems for flexible manufacturing, and within that context, his scheme has worked well.

On the other hand, it is evident that approximation of each control constraint set U_k by a single box involves a substantial error; for example, the volume of the largest box contained within the M-dimensional unit simplex

$$\left\{\left(u_k^1,\ldots,u_k^M\right) | u_k^1+\cdots+u_k^M \le 1, u_k^m \ge 0, m=1,\ldots,M\right\}$$

is equal to M^{-M}, while the volume of the simplex is $2^{-(M-1)}$. Consequently the corresponding approximating function (1.5) can be a poor approximation of the optimal cost-to-go function J_k.

The purpose of this note is to propose suboptimal control methods that are based on more accurate approximations of the coupling constraint sets U_k, with attendant improvements in the cost-to-go approximations. In particular, the control constraint sets are approximated by *finite unions of boxes*, and the optimal cost-to-go functions J_k are approximated by upper bounding functions of the form

$$\tilde{J}_k(x_k) = \min_{n=1,\ldots,p} \sum_{m=1}^{M} \hat{J}_k^m\left(x_k^m; n\right). \tag{1.6}$$

where p is some positive integer, and each function $\hat{J}_k^m(x_k^m; n)$ is obtained through a tractable computation involving the mth subsystem only. The method that we propose bears a relation to rollout algorithms (see [8] and the survey [6] for an extensive discussion and references), whereby $\tilde{J}_k$ is computed by means of one or more suboptimal policies, called *base policies*. In our case, the base policies are derived from optimal policies for approximating subproblems involving box approximations of the control constraint sets. These base policies are *decomposable*, in the sense that the components u_k^m of the control applied to the mth subsystem depends only on the current state x_k^m of that subsystem. The approximate cost-to-go function $\tilde{J}_k$ is either the cost-to-go of a decomposable base policy or else it is computed as a function of the costs-to-go of several decomposable suboptimal policies [for example $\tilde{J}_k$ may be the pointwise minimum of the costs-to-go of several base policies; cf. (1.6)]. We refer to an extended version of this note for an account of this connection [7].

The computational savings from our methods are potentially very significant. For a crude calculation, assume that the state and control spaces are finite (perhaps after discretization), and let d_s and d_c be the numbers of points in the state and control spaces, respectively, of each subsystem. Then the computational requirements of a single stage of DP [measured in number of evaluations of the expected value in the right-hand side of (1.3)] are of the order of $O((d_s d_c)^M)$, while for our methods they are of the order of $O(pM(d_s d_c))$, where p is the number of terms in the sum (1.6). Of course, the number p can also be very large in our method if $\tilde{J}_k$ is to be a very accurate approximation of J_k. We believe, however, that for many practical problems a near-optimal policy can be obtained with a relatively small number of terms p in (1.6) (for example, in Kimemia's work [18], $p=1$ yielded a very well-performing suboptimal policy). An intuitive reason for this conjecture is that the addition of a constant to the cost-to-go function J_{k+1} in the DP (1.3) does not affect the outcome of the minimization over u_k. This suggests that it is not important for the cost-to-go differences $\tilde{J}_{k+1}(x) - \tilde{J}_{k+1}(x)$ to be small for all states x at time $k+1$, but rather that the relative differences be small, i.e., that

$$\tilde{J}_{k+1}(x) - \tilde{J}_{k+1}(x') \approx J_{k+1}(x) - J_{k+1}(x')$$

for any two states x and x' that can be generated from any given state at time k (see also the discussion in [8, p. 312]). Unfortunately, it seems to be hard to get an understanding of the beneficial effect of this phenomenon through a computational complexity analysis.

In Sections II and III, we discuss some specific approximation schemes based on decomposition. In Section IV, we specialize our approach to the problem of reachability of a target tube that has the form of a box (a Cartesian product of subsystem tubes). Our method then yields inner approximating tubes, which have the form of a union of a finite number of boxes, each involving single subsystem calculations. The target tube reachability problem, first introduced and studied by the author in [3], is central, among others, to the context of model predictive control (see the survey [6] for a discussion and references).

II. An Enforced Decomposition Scheme

Our approximation scheme is based on the computation of the functions $\hat{J}_k^m(x_k^m; n)$ in (1.6) as the optimal cost-to-go functions of optimal control problems involving the mth subsystem only. In particular, we approximate the constraint set of the problem, $U_0 \times U_1 \times \cdots \times U_{N-1}$, in multiple different ways, with subsets of the form

$$U_{0,n} \times U_{1,n} \times \cdots \times U_{N-1,n}, \qquad n=1,\ldots,p \tag{2.1}$$

where p is some positive integer. Each set $U_{k,n}$ is a *separable inner approximation* of the corresponding set U_k, i.e., a box of the form

$$U_{k,n} = U_{k,n}^1 \times \cdots \times U_{k,n}^M \subset U_k, \quad n=1,\ldots,p, \ k=0,\ldots,N-1. \tag{2.2}$$

We refer to the set sequence (2.1) as the *nth control allocation schedule*. It represents a particular way of allocating (a subset of) the control constraint set $U_0 \times U_1 \times \cdots \times U_{N-1}$ to the M subsystems. The key property is that under a given control allocation schedule, the corresponding subsystems do not compete for control resource and are decoupled, because of the Cartesian product structure (2.2). Thus, when the original constraint set is replaced by the control allocation schedule (2.2), the problem obtained is decomposed along subsystems, into a total of pM independent optimal control subproblems. In particular, for each $n=1,\ldots,p$ and $m=1,\ldots,M$, the subproblem involves the system equation

$$x_{k+1}^m = f_k^m\left(x_k^m, u_k^m, w_k^m\right) \tag{2.3}$$

the control constraint

$$u_k^m \in U_{k,n}^m \tag{2.4}$$

and the cost function

$$E\left\{g_N^m\left(x_N^m\right) + \sum_{k=0}^{N-1} g_k^m\left(x_k^m, u_k^m, w_k^m\right)\right\}. \tag{2.5}$$

Let $\hat{J}_k^m(x_k^m; n)$ be the optimal cost-to-go function of this subproblem, starting from state x_k^m at time k.

We propose the one-step-lookahead scheme where the optimal cost-to-go function at time k is approximated by

$$\tilde{J}_k(x_k) = \min_{n=1,\ldots,p} \sum_{m=1}^{M} \hat{J}_k^m\left(x_k^m; n\right). \tag{2.6}$$

In particular, having computed the functions $\hat{J}_k^m(x_k^m; n)$ for all n, m, and k, by solving the pM single subsystem problems corresponding to each subsystem m and each control allocation schedule

$\{U_{0,n},\ldots,U_{N-1,n}\}$, this scheme applies at time k and state $x_k=(x_k^1,\ldots,x_k^M)$ a control

$$\overline{\mu}_k(x_k)\in\arg\min_{u_k\in U_k}E\left\{\sum_{m=1}^{M}g_k^m\left(x_k^m,u_k^m,w_k^m\right)\right.\\ \left.+\min_{n=1,\ldots,p}\sum_{m=1}^{M}\hat{J}_{k+1}^m\left(f_k^m\left(x_k^m,u_k^m,w_k^m\right);n\right)\right\}. \quad (2.7)$$

For each n, the function

$$\sum_{m=1}^{M}\hat{J}_k^m\left(x_k^m;n\right)$$

is the cost-to-go at time k and state $x_k=(x_k^1,\ldots,x_k^M)$ of an optimal policy for the problem where the control constraint set is inner-approximated by the set $U_{k,n}$ of (2.2). It follows from general results on rollout algorithms (cf. the analysis of [8, ex. 3.2]) that the optimal cost of the suboptimal one-step-lookahead scheme, at each time k and state $x_k=(x_k^1,\ldots,x_k^M)$, satisfies

$$\overline{J}_k(x_k)\le\min_{n=1,\ldots,p}\sum_{m=1}^{M}\hat{J}_k^m\left(x_k^m;n\right). \quad (2.8)$$

This relation provides some guidelines for choosing the sets $U_{k,n}$: they should cover as much as possible the constraint set of the original problem, in the sense that

$$\cup_{n=1}^{p}(U_{0,n}\times\cdots\times U_{N-1,n})\approx U_0\times\cdots\times U_{N-1}. \quad (2.9)$$

In particular, it can be seen that if the problem is deterministic, and equality holds in the preceding relation, the suboptimal control scheme is exactly optimal. If the problem is stochastic, equality in (2.9) does not guarantee that the suboptimal control scheme is exactly optimal, but there are important classes of problems where this can be shown. In particular, for multiarmed bandit problems with discounted infinite horizon cost (see e.g., [8]), one can show equality in (2.9) under assumptions guaranteeing that an index policy exists.

To get a sense of the potential loss of optimality for stochastic problems, consider the extreme case where each set U_k is finite, and we choose the set of control allocation schedules to be the set of all feasible open-loop control sequences. Then the scheme of this section is equivalent to the *open-loop-feedback control* scheme, first proposed by Dreyfus [12] and subsequently discussed in many sources (see, e.g., [8] and [6] for recent accounts and references). Note that for deterministic problems, open-loop-feedback control is optimal, which is consistent with the preceding discussion.

In a variant of the scheme of the current section, the minimization over $u_k\in U_k$ in (2.7) may be replaced by a (potentially simpler) minimization over

$$u_k\in\cup_{n=1}^{p}U_{k,n}.$$

Then, it can be shown that the cost improvement property (2.8) is preserved, by using a modification of the analysis of [8, ex. 3.2]. A second variant arises when certainty equivalence approximations are introduced in either the subsystem subproblems used to compute the functions $\hat{J}_k^m(x_k^m;n)$, or in the minimization (2.7). However, in this second variant, it is possible that the cost improvement property (2.8) does not hold anymore.

III. Alternative Implementations

The decomposition approach of the preceding section was based on the use of multiple control allocation schedules of the form (2.1), (2.2), and the solution of the decoupled optimal control problems (2.3)–(2.5), to obtain the cost-to-go approximations (2.6). In this section, we discuss methods for solving the decoupled problems.

A. Implementation by DP

The most straightforward possibility to solve the decoupled problems (2.3)–(2.5) is by DP. Here, for each subsystem $m=1,\ldots,M$, and control allocation schedule n, we start with

$$J_N^m\left(x_N^m\right)=g_N^m\left(x_N^m\right),\qquad m=1,\ldots,M$$

and generate the functions $\hat{J}_k^m$ from the functions $\hat{J}_{k+1}^m$, by

$$\hat{J}_k^m\left(x_k^m;n\right)=\min_{u_k^m\in U_{k,n}^m}E\\ \times\left\{g_k^m\left(x_k^m,u_k^m,w_k^m\right)+\hat{J}_{k+1}^m\left(f_k^m\left(x_k^m,u_k^m,w_k^m\right);n\right)\right\}. \quad (3.1)$$

The approximate cost-to-go functions $\tilde{J}_k$ have the form

$$\tilde{J}_k\left(x_k^1,\ldots,x_k^M\right)=\min_{n=1,\ldots,p}\sum_{m=1}^{M}\hat{J}_k^m\left(x_k^m;n\right). \quad (3.2)$$

This leads to an interpretation of a special case of our method as an approximate DP method. Consider the case where the set of control allocation schedules is constructed using an independent approximation of each constraint set U_k with a union of boxes. By this, we mean that for each k, we introduce boxes of the form

$$U_{k,l}^1\times\cdots\times U_{k,l}^M\subset U_k,\qquad l=1,\ldots,r_k,\ k=0,\ldots,N-1 \quad (3.3)$$

where r_k is a positive integer, and we let the set of control allocation schedules be the set of all possible subsets of the form

$$U_{0,n}\times U_{1,n}\times\cdots\times U_{N-1,n},\qquad n=1,\ldots,p \quad (3.4)$$

where $p=r_0\cdot r_1\cdots r_{N-1}$ and each $U_{k,n}$ ranges over the collection of all boxes (3.3). Note that with this choice of the set of control allocation schedules, the minimum defining the cost-to-go approximation

$$\min_{n=1,\ldots,p}\sum_{m=1}^{M}\hat{J}_t^m\left(x_t^m;n\right)$$

does not depend on the boxes (3.3) for $k<t$.

It can now be seen that stage k of our approach is equivalent to the following: It starts with a cost-to-go for stage $k+1$ of the form

$$\tilde{J}_{k+1}(x_{k+1})=\min_{n=1,\ldots,p}\sum_{m=1}^{M}\hat{J}_{k+1}^m\left(x_{k+1}^m;n\right)$$

and executes a DP-like iteration that involves two complementary approximations.

a) Replacing the constraint set U_k with the union of the r_k boxes in (3.3)

$$\tilde{U}_k=\cup_{l=1}^{r_k}\left(U_{k,l}^1\times\cdots\times U_{k,l}^M\right).$$

b) Writing the corresponding DP algorithm iteration, which has the form of (3.5), as shown at the bottom of the next page, and modifying its right-hand side by interchanging expectation over w_k and minimization over n, so that the minimization over u_k is decoupled across subsystems. This yields

$$\tilde{J}_k\left(x_k^1,\ldots,x_k^M\right) = \min_{n=1,\ldots,p} \sum_{m=1}^{M} \min_{u_k^m \in U_{k,n}^m} E \times\Big\{g_k^m\left(x_k^m,u_k^m,w_k^m\right)+\hat{J}_{k+1}^m\left(f_k^m\left(x_k^m,u_k^m,w_k^m\right);n\right)\Big\} \tag{3.6}$$

which is equivalent to the algorithm (3.1), (3.2).

Note that these two approximations, individually and together, result in an overestimate of the true cost-to-go function, which is consistent with the analysis of the preceding section. Furthermore, in the case of a deterministic problem, the interchange of minimization and expectation, as in b), is superfluous, so the quality of the approximation obtained depends solely on the accuracy of approximation of the control constraint sets U_k by the union of boxes $\tilde{U}_k$, as in a).

B. Distributed Implementation

Separable problems and decomposition schemes are usually well-suited for distributed implementation, and this is true for our scheme as well. In particular, consider the approach of the preceding section, based on control allocation schedules, and suppose that the mth subsystem has its own processor. Upon the transition to state x_k^m, the mth subsystem can calculate for all possible values of (u_k^m, w_k^m), the quantities

$$\tilde{J}_{k+1}^m\left(f_k^m\left(x_k^m,u_k^m,w_k^m\right);n\right), \qquad n=1,\ldots,p$$

and transmit these quantities, together with its current state x_k^m, to the other subsystems. After all subsystems have done so, each subsystem will have all the information needed to execute iteration (2.7). Thus, the typically most computationally intensive portion of the computations in our scheme, the calculation of the approximate cost-to-go function needed in iteration (2.7)

$$\tilde{J}_{k+1}\left(f_k(x_k,u_k,w_k)\right) = \min_{n=1,\ldots,q} \sum_{m=1}^{M} \hat{J}_{k+1}^m\left(f_k^m\left(x_k^m,u_k^m,w_k^m\right);n\right)$$

can be fully parallelized. Similarly, in the approach of the current section, one may parallelize the calculations in (3.1), both across subsystems and across states for each subsystem.

IV. Target Tube Reachability

We will now specialize our methodology to the problem of reachability of target sets and tubes, introduced by the author in [3], and subsequent papers [1] and [4]. Reachability, under disturbances with a set-membership description, stayed outside mainstream control theory and practice for a long time, but has received renewed attention since the late 1980s, in the context of robust control and model predictive control. We refer to [11], [17], [9], and [20], which give many additional references. Section 4.6.2 of [8] provides an introduction to the subject, which is relevant to the material that follows.

We continue to consider the system of (1.1), which consists of M independent subsystems. However, we assume that the disturbance w_k^m is not random, but instead is known to belong to a given set $W_k^m(x_k^m, u_k^m)$, which may depend on the mth subsystem state and control. Our reachability problem is to find a policy $\pi = \{\mu_0,\ldots,\mu_{N-1}\}$ with $\mu_k(x_k) \in U_k$ for all x_k and k, such that for each $k = 1,2,\ldots,N$, the state x_k^m of the closed-loop mth subsystem

$$x_{k+1}^m = f_k\left(x_k^m, \mu_k^m\left(x_k^m\right), w_k^m\right)$$

belongs to a given set X_k^m, called the *target set for the mth subsystem at time k*. In other words, we want to find a policy π such that the closed-loop system state x_k belongs for each k to the set

$$X_k = X_k^1 \times \cdots \times X_k^M.$$

We may view the set sequence $\{X_1,\ldots,X_N\}$ as a "tube" within which the state must stay, even under the worst possible choice of the disturbances w_k^m from within the corresponding sets $W_k^m(x_k^m, \mu_k^m(x_k))$. This is essentially a minimax control problem, involving the cost-to-go function

$$J_k^\pi(x_k) = \max_{\substack{w_t^m \in W_t^m \\ t=k,\ldots,N-1,m=1,\ldots,M}} \times\left\{\sum_{m=1}^{M}\left(g_N^m\left(x_N^m\right)+\sum_{t=k}^{N-1} g_k^m\left(x_t^m,\mu_t^m(x_t),w_t^m\right)\right)\right\}$$

where

$$g_k^m\left(x_k^m,u_k^m,w_k^m\right) = \begin{cases}0, & \text{if } x_k^m \in X_k^m\\ 1, & \text{if } x_k^m \notin X_k^m,\end{cases} \quad k=0,\ldots,N-1$$

$$g_N^m\left(x_N^m\right) = \begin{cases}0, & \text{if } x_N^m \in X_N^m\\ 1, & \text{if } x_N^m \notin X_N^m.\end{cases}$$

Given an initial condition x_0, the problem is to find π such that $J_0^\pi(x_0) = 0$. The problem can be approached by DP, by determining the set of initial states x_0 and a policy π^* such that for all possible state sequences $\{x_1,\ldots,x_N\}$ under this policy, we have

$$J_k(x_k) = \min_\pi J_k^\pi(x_k) = J_k^{\pi^*}(x_k) = 0, \qquad k=0,\ldots,N-1.$$

We can view $J_k(x_k)$ as the optimal cost-to-go for the minimax problem. The set

$$\overline{X}_k = \{x_k | J_k(x_k) = 0\}$$

$$J_k\left(x_k^1,\ldots,x_k^M\right) = \min_{u_k \in \tilde{U}_k} E\left\{\sum_{m=1}^{M} g_k^m\left(x_k^m,u_k^m,w_k^m\right)+\tilde{J}_{k+1}\left(f_k^1\left(x_k^1,u_k^1,w_k^1\right),\ldots,f_k^M\left(x_k^M,u_k^M,w_k^M\right)\right)\right\}$$
$$= \min_{l=1,\ldots,r_k} \min_{u_k^m \in U_{k,l}^1\times\cdots\times U_{k,l}^M}\left[\sum_{m=1}^{M} E\left\{g_k^m\left(x_k^m,u_k^m,w_k^m\right)\right\}+E\left\{\min_{n=1,\ldots,p}\sum_{m=1}^{M}\hat{J}_{k+1}^m\left(f_k^m\left(x_k^m,u_k^m,w_k^m\right);n\right)\right\}\right] \tag{3.5}$$

is the set that we *must* reach at time k in order to be able to maintain the state within the subsequent target sets. Accordingly, we refer to $\overline{X}_k$ as the *effective target set at time* k.

The corresponding DP algorithm can be expressed in terms of a set recursion. Denoting

$$X_k = X_k^1 \times \cdots X_k^M$$
$$W_k(x_k, u_k) = W_k^1\left(x_k^1, u_k^1\right) \times \cdots \times W_k^M\left(x_k^M, u_k^M\right)$$

we start with

$$\overline{X}_N = X_N$$

and for $k = 0, 1, \ldots, N-1$, we have

$$\begin{aligned}\overline{X}_k = \{x_k \in X_k | &\text{there exists } u_k \in U_k \text{ such that} \\ & f_k(x_k, u_k, w_k) \in \overline{X}_{k+1}, \text{ for all } w_k \in W_k(x_k, u_k)\big\}; \end{aligned} \quad (4.1)$$

see, e.g., [8, Sec. 4.6.2].

The implementation approach of Section III, based on inner approximation of the control constraint set U_k and the approximate DP algorithm (3.1), (3.2) is well-suited for inner approximation of the effective target sets. To implement this algorithm, we start with an inner approximation of each control constraint set U_k:

$$\tilde{U}_k = \cup_{l=1}^{r_k}\left(U_{k,l}^1 \times \cdots \times U_{k,l}^M\right) \subset U_k \quad (4.2)$$

and derive an inner approximation of the corresponding effective target set $\overline{X}_k$, which is denoted by $\tilde{X}_k$ and is the union of boxes. In particular, the sets $\tilde{X}_k$ are generated sequentially, starting with

$$\tilde{X}_N = X_N = X_N^1 \times \cdots \times X_N^M.$$

Given $\tilde{X}_{k+1}$ expressed as a union of boxes,

$$\tilde{X}_{k+1} = \cup_{n=1}^{p_{k+1}}\left(\tilde{X}_{k+1,n}^1 \times \cdots \times \tilde{X}_{k+1,n}^M\right)$$

we calculate for each $m = 1, \ldots, M$, $l = 1, \ldots, r_k$, and $n = 1, \ldots, p_{k+1}$,

$$\begin{aligned}\hat{X}_{k,l,n}^m = \big\{x_k^m \in X_k^m | &\text{there exists } u_k^m \in U_{k,l}^m \text{ such that} \\ & f_k^m\left(x_k^m, u_k^m, w_k^m\right) \in \overline{X}_{k+1}^m \\ & \text{for all } w_k^m \in W_k^m\left(x_k^m, u_k^m\right)\}. \end{aligned} \quad (4.3)$$

We then set

$$\tilde{X}_k = \cup_{n=1}^{p_{k+1}} \cup_{l=1}^{r_k}\left(\hat{X}_{k,l,n}^1 \times \cdots \times \hat{X}_{k,l,n}^M\right) \quad (4.4)$$

which can also be written in the form

$$\tilde{X}_k = \cup_{n=1}^{p_k}\left(\tilde{X}_{k,n}^1 \times \cdots \times \tilde{X}_{k,n}^M\right) \quad (4.5)$$

where $\tilde{X}_{k,n}^m$ are suitable sets and $p_k = r_k p_{k+1}$. Note that the computation of the sets (4.3) involves single subsystem calculations.

The algorithm (4.3)–(4.5) can be simply explained. For each pair (l, n), and corresponding control and state boxes

$$U_{k,l}^1 \times \cdots \times U_{k,l}^M \quad (4.6)$$

and

$$\tilde{X}_{k+1,l,n}^1 \times \cdots \times \tilde{X}_{k+1,l,n}^M \quad (4.7)$$

we calculate the box

$$\hat{X}_{k,l,n}^1 \times \cdots \times \hat{X}_{k,l,n}^M$$

of states x_k from which we are guaranteed to reach the box (4.7) using control from the box (4.6). The union of these boxes is the set $\tilde{X}_k$ of (4.4), and by induction, can be shown to be a subset of the effective target set $\overline{X}_k$ of (4.1).

A drawback of the preceding algorithm is that the number p_k of boxes involved in the union (4.5) increases with each iteration. This motivates a modification of the algorithm, whereby some of the boxes in (4.5) are discarded according to some criterion. As a result the amount of computation is reduced, at the expense of additional inner approximation of the effective target tubes.

We finally note that the approximate solution to the reachability problem of this section may be used as a preliminary step towards implementing a decomposition approach to model predictive control for decoupled systems of the type considered here (see [8] and [6]). Such an approach involves the solution of single subsystem minimax control problems with state constraints derived from the inner approximations (4.5) of the effective target tubes, and control constraints derived from the inner approximations (4.2) of the control constraint sets. The details of this approach are beyond the scope of the present note.

V. Concluding Remarks

We have presented a new method for decomposition of the computations in DP problems involving independent subsystems and coupling through the control constraints. A traditional methodology for problems of this type is based on Lagrangian relaxation and duality (see e.g., [19], [13], and [14]). Our approach is very different: it is based on inner approximations of the control constraint sets by unions of boxes, and it is guided by approximate DP methodology, and rollout algorithms in particular. Our approach may be more general than the Lagrangian relaxation approach, because it can deal with more general constraint constraint sets. On the other hand, the Lagrangian relaxation approach may be more general in other cases, because it can deal with more general couplings, involving for example combined state and control constraint sets. The computations required by our methods can be formidable, and much depends on the special structure of the subsystems. In particular, it is essential that the subsystems are simple enough, so that the corresponding optimal control problems are easily solvable.

There is no computational experience with our method, except for the special case where each constraint set U_k is approximated by a single box (see [18], which motivated much subsequent work along related lines, e.g., [10] and [15]). The success obtained with this simple special case is encouraging for the prospects of our more sophisticated but also more computationally demanding approach.

Problems of target tube reachability and model predictive control, for systems that have an independent subsystem structure, are of particular practical interest. Our method is promising within this context, but the exploration of its potential is a subject for further research.

References

[1] D. P. Bertsekas and I. B. Rhodes, "On the minimax reachability of target sets and target tubes," *Automatica*, vol. 7, pp. 233–247, 1971.

[2] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 1996.

[3] D. P. Bertsekas, "Control of uncertain systems with a set-membership description of the uncertainty," Ph.D. dissertation, Mass. Inst. Technol., Cambridge, MA, 1971.

[4] D. P. Bertsekas, "Infinite time reachability of state space regions by using feedback control," *IEEE Trans. Autom. Control*, vol. AC-17, no. 5, pp. 604–613, Oct. 1972.

[5] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 1996.

[6] D. P. Bertsekas, “Dynamic programming and suboptimal control: A survey from ADP to MPC,” *Fundam. Issues Control, Eur. J. Control*, vol. 11, no. 4–5, 2005.

[7] D. P. Bertsekas, Separable dynamic programming and approximate decomposition methods Mass. Inst. Technol., Cambridge, MA, Rep. LIDS 2684, 2006.

[8] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 3rd ed. Belmont, MA: Athena Scientific, 2007, vol. II.

[9] F. Blanchini, “Set invariance in control—A survey,” *Automatica*, vol. 35, pp. 1747–1768, 1999.

[10] M. Caramanis and A. Sharifnia, “Near optimal manufacturing flow controller design,” *Int. J. Flexible Manuf. Syst.*, vol. 3, pp. 321–336, 1991.

[11] J. R. Deller, “Set membership identification in digital signal processing,” *IEEE ASSP Mag.*, pp. 4–20, Oct. 1989.

[12] S. D. Dreyfus, *Dynamic Programming and the Calculus of Variations*. New York: Academic, 1965.

[13] A. M. Geoffrion, “Elements of large-scale mathematical programming, I, II,” *Manage. Sci.*, vol. 16, pp. 652–675, 1970, 676-691.

[14] A. M. Geoffrion, “Lagrangian relaxation for integer programming,” *Math. Program. Stud.*, vol. 2, pp. 82–114, 1974.

[15] S. B. Gershwin, *Manufacturing Systems Engineering*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[16] J. Kimemia, S. B. Gershwin, and D. P. Bertsekas, “Computation of production control policies by a dynamic programming technique,” in *Analysis and Optimization of Systems*, A. Bensoussan and J. L. Lions, Eds. New York: Springer-Verlag, 1982, pp. 243–269.

[17] R. L. Kosut, M. K. Lau, and S. P. Boyd, “Set-membership identification of systems with parametric and nonparametric uncertainty,” *IEEE Trans. Autom. Control*, vol. 37, no. 7, pp. 929–941, Jul. 1992.

[18] J. Kimemia, “Hierarchical control of production in flexible manufacturing systems,” Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Mass. Inst. Technol., Cambridge, MA, 1982.

[19] L. S. Lasdon, *Optimization Theory for Large Systems*. New York: Macmillian, 1970.

[20] D. Q. Mayne, “Control of constrained dynamic systems,” *Eur. J. Control*, vol. 7, pp. 87–99, 2001.

On the Rendezvous Problem for Multiple Nonholonomic Agents

Dimos V. Dimarogonas and Kostas J. Kyriakopoulos

***Abstract*—In this note, a decentralized feedback control strategy that drives a system of multiple nonholonomic unicycles to a rendezvous point in terms of both position and orientation is introduced. The proposed nonholonomic control law is discontinuous and time-invariant and using tools from nonsmooth Lyapunov theory and graph theory the stability of the overall system is examined. Similarly to the linear case, the convergence of the multi-agent system relies on the connectivity of the communication graph that represents the inter-agent communication topology. The control law is first defined in order to guarantee connectivity maintenance for an initially connected communication graph. Moreover, the cases of static and dynamic communication topologies are treated as corollaries of the proposed framework.**

***Index Terms*—Cooperative control, decentralized control, nonholonomic agents.**

I. Introduction

In this note, the problem of rendezvous convergence for a system of multiple nonholonomic unicycles in terms of both position and orientation is considered. The rendezvous problem has been extensively approached recently, addressing the control design issue from several perspectives. Recent results include [6], [11], [12], [14], [17], and [19]. In most cases, linear models of motion are taken into account, while the information exchange topology is considered both static and dynamic, as well as bidirectional or unidirectional. A recent review of the various approaches of the rendezvous problem for linear models of motion is [21].

In this note, a decentralized control strategy that drives a system of multiple kinematic unicycles to rendezvous is presented. The proposed nonholonomic feedback law is discontinuous and time invariant, something expected, as nonholonomic systems do not satisfy Brocketts necessary smooth feedback stabilization condition [3]. These controllers have in general better convergence properties than time-varying ones. An experimental comparison between these two types of controllers that supports our preference to time-invariant strategies has appeared in [13], where the authors concluded that time-varying controllers were too slow and oscillatory for most practical cases. In contrast, time-invariant controllers achieved a significantly better performance.

A first contribution of this note is that the control law is first defined in order to guarantee connectivity maintenance. Hence, if the communication graph is initially connected, it remains connected throughout the closed loop system evolution and rendezvous is reached, under the proposed control law. Connectivity preserving for linear agents was

Manuscript received November 22, 2005; revised May 9, 2006 and November 22, 2006. Recommended by Asssociate Editor F. Bullo. This work was supported by the European Commission through contract I-SWARM (IST-2004-507006), http://www.i-swarm.org/. Preliminary versions of this work have appeared in [7] and [8].

The authors are with the Control Systems Laboratory, Mechanical Engineering Department, the National Technical University of Athens, Athens 15780, Greece (e-mail: ddimar@mail.ntua.gr; kkyria@mail.ntua.gr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2007.895897