INT. J. CONTROL, 1991, VOL. 53, NO. 6, 1395–1412

# Mixed coordination method for long-horizon optimal control problems

JIANXIN TANG†, P. B. LUH† and TSU-SHUAN CHANG‡

We present a new approach to solving long-horizon, discrete-time optimal control problems using the mixed coordination method. The idea is to decompose a long-horizon problem into subproblems along the time axis. The requirement that the initial state of a subproblem equal the terminal state of the preceding subproblem is relaxed by using Lagrange multipliers. The Lagrange multipliers and initial state of each subproblem are then selected as high-level variables. The equivalence of the two-level formulation and the original problem is proved for both convex and non-convex cases. The low-level subproblems are solved in parallel using extended differential dynamic programming (DDP). An efficient way to find the gradient and hessian of a low-level objective function with respect to high-level variables is developed. The high-level problem is solved using the modified Newton method. An effective procedure is developed to select initial values of multipliers based on the initial trajectory. The method can convexify the high-level problem while maintaining the separability of an originally non-convex problem. The method performs better and faster than one-level DDP for both convex and non-convex test problems.

## 1. Introduction

Large-scale optimal control problems are generally difficult to solve because of their inherent high dimensionality, problem complexity and non-linear behaviour. There are many different techniques for problems of small to medium size, but there have not been many efficient methodologies for solving large-scale problems. Some methods, such as goal coordination and model coordination, were developed along the line of state decomposition and coordination (Mesarovic *et al.* 1969, Jamshidi 1983). For many large-scale problems, the long time horizon adds another dimension of difficulty. For this reason, 'incentive coordination' (Chang *et al.* 1990, Bromberg *et al.* 1989 a, b) and 'target coordination' (Chang *et al.* 1989) were recently developed along the line of time decomposition. The idea of incentive coordination is to modify the cost function of each subproblem by adding an incentive term to it. These incentive terms are updated at a high level to insure overall optimality. In target coordination, the initial and terminal states of subproblems are selected as high-level coordination variables. Note that both the incentive and the target coordinations are primal methods. In this paper, we present another time decomposition approach for long-horizon optimal control problems using a mixed coordination method, which is a combination of the primal and the dual methods.

The mixed coordination method for static optimization was presented by Singh (1978) and Tang *et al.* (1988, 1989). When applied to time decomposition and coordination of dynamic problems, a problem is first decomposed along the time axis

Received 23 May 1989. Revised 9 August 1989.
† Department of Electrical and Systems Engineering, University of Connecticut, Storrs, CT

into smaller subproblems. The requirement that the initial state of a subproblem equal the terminal state of the preceding subproblem is relaxed by using Lagrange multipliers. The Lagrange multipliers and the initial state of each subproblem are then selected to be optimized at the high level. For a given set of high-level variables the low-level subproblems are decoupled optimal control problems with a shorter time horizon, and can therefore be solved in parallel. The high-level variables are then updated on the basis of low-level results, and the process repeats until convergence.

Although many existing techniques can be used to solve the high-level problem and low-level subproblems, fast convergence and algorithm compatibility are of crucial importance in solving a two-level problem: see the discussion in Chang *et al.* (1989). In this paper, the differential dynamic programming (DDP) of Yakowits and Rutherford (1984) is extended for low-level subproblems, and the modified Newton method is used for high-level optimization. The reasons for selecting the DDP for low-level subproblems are that it has a quadratic convergence rate near the optimal, and can provide the first- and second-order derivative information needed for high-level optimization. The modified Newton method also has fast high-level convergence near the optimum. An effective procedure is developed to select the initial values of multipliers based on the given initial trajectory.

Mixed coordination has an advantage over other coordination methods. When applied to non-convex problems the method preserves the separability of subproblems even when quadratic convexification terms are added. As a result, the method is promising for problems with non-convex cost functions or non-linear system dynamics. Although the algorithms presented in §§ 3 and 4 are only for the unconstrained case, they can be extended to problems with constraints on state and control variables. This investigation is currently under way.

In § 2 we formulate the problem and describe the time decomposition procedure. The equivalence of the two-level formulation and the original problem is shown for the convex case. The algorithms presented in §§ 3 and 4 are for convex and non-convex problems, respectively. In § 4 the equivalence of the two-level formulation and the original problem is shown for the non-convex case. In § 5 the procedure for selecting initial values of multipliers based on the given initial trajectory is presented. Numerical results show that significant speedups are achieved for both convex and non-convex test problems in a simulated parallel processing environment. There are some concluding remarks in § 6.

## 2. Mixed coordination formulation of optimal control problem

Consider the following discrete-time optimal control problem. (For simplicity, the problem considered here is without terminal cost $g_{N+1}(x_{N+1})$. A problem with terminal cost can be converted to our formulation by lumping the terminal cost with $g_N(x_N, u_N)$ using system dynamics.)

*Problem P*

$$\min_{\{u_t\}} J, \quad \text{with} \quad J \equiv \sum_{t=1}^{N} g_t(x_t, u_t) \tag{1}$$

subject to the system dynamics:

$$x_{t+1} = f_t(x_t, u_t), \quad t = 1, \ldots, N, \quad \text{with } x_1 \text{ given} \tag{2}$$



Assume that $g_t$ and $f_t$ are twice continuously differentiable in $x_t$ and $u_t$ for all $t$, and $N = MT \gg 1$, where $M$ and $T$ are positive integers. To facilitate time decomposition that will partition Problem $P$ into $M$ subproblems each with $T$ stages, we shall adopt a double subscript notation. In the new notation, the first subscript is the subproblem index, and the second subscript is the stage index within a subproblem. Problem $P$ can thus be rewritten as follows.

*Problem P′*

$$\min_{\{u_{jt}\}} J, \quad \text{with} \quad J \equiv \sum_{j=1}^{M} \sum_{t=1}^{T} g_{jt}(x_{jt}, u_{jt}) \tag{4}$$

subject to:

$$x_{j(t+1)} = f_{jt}(x_{jt}, u_{jt}), \quad j = 1, 2, \ldots, M, \quad t = 1, 2, \ldots, T \tag{5}$$

$$x_{j(T+1)} = x_{(j+1)1}, \quad j = 1, 2, \ldots, M-1, \quad t = 1, 2, \ldots, T \tag{6}$$

and

$$x_{jt} \in X_{jt} \subset R^n \quad \text{and} \quad u_{jt} \in U_{jt} \subset R^m, \quad j = 1, 2, \ldots, M, \quad t = 1, 2, \ldots, T \tag{7}$$

Note that constraint (6) requires that the initial state of a subproblem equal the terminal state of the preceding subproblem.

Using Lagrange multipliers to relax constraint (6), we have the following problem.

*Problem P″*

$$\max_{\{\lambda_j\}} \min_{\{x_{(j+1)1}\}} \min_{\{u_{jt}\}} L, \quad \text{with } L \equiv \sum_{j=1}^{M} \left[ \sum_{t=1}^{T} g_{jt}(x_{jt}, u_{jt}) + \lambda_j^T [x_{(j+1)1} - x_{j(T+1)}] \right] \tag{8}$$

with $\lambda_M \equiv x_{(M+1)1} \equiv 0$, subject to constraints (5) and (7). Selecting $\{\lambda_j\}_{j=1}^{M-1}$ and $\{x_{(j+1)1}\}_{j=1}^{M-1}$ as high-level coordination variables, Problem $P''$ can be decomposed into the following $M$ subproblems.

*Subproblems* $(P-j)$, $j = 1, 2, \ldots, M$

$$\min_{\{u_{jt}\}} L_j, \quad \text{with } L_j \equiv \sum_{t=1}^{T} g_{jt}(x_{jt}, u_{jt}) - \lambda_j^T x_{j(T+1)} \tag{9}$$

subject to relevant constraints from (5) and (7). Let $\{u_{jt}^*(\lambda_j, x_{j1})\}_{t=1}^{T}$ denote the optimal controls of Subproblem $(P-j)$ for the given high-level variables $\lambda_j$, $x_{j1}$, and $L_j^*(\lambda_j, x_{j1})$ the corresponding optimal cost. Then the high-level problem is to find the optimal $\{\lambda_j\}_{j=1}^{M-1}$ and $\{x_{(j+1)1}\}_{j=1}^{M-1}$, i.e. the following problem.

*Problem* $(P-H)$

$$\max_{\{\lambda_j\}} \min_{\{x_{(j+1)1}\}} L_H, \quad \text{where } L_H = \sum_{j=1}^{M} [L_j^*(\lambda_j, x_{j1}) + \lambda_{(j-1)}^T x_{j1}] \tag{10}$$

with $\lambda_0 \equiv \lambda_M \equiv 0$, subject to

$$x_{(j+1)1} \in X_{(j+1)1} \subset R^n, \quad j = 1, 2, \ldots, M-1 \tag{11}$$

When the cost function is convex the system dynamics is linear, and $X_t$ and $U_t$ are convex for all $t$, Problem $P'$ is a convex programming problem. Equation (5) can be rewritten as

*Theorem* 1

The solution of Problem $(P'-H)$ is a saddle point.

*Theorem* 2

Problem $(P-H)$ and Problem $P''$ are equivalent.

In Theorem 2, the two problems are equivalent in the sense that if $(\{\lambda_j^*\}, \{x_{(j+1)1}^*\})$ is the solution of Problem $P-H$ and $\{u_{jt}^*(\lambda_j^*, x_{(j+1)1}^*)\}$ is the solution of Subproblem $(P-j)$ given $(\{\lambda_j^*\}, \{x_{(j+1)1}^*\})$, then $(\{\lambda_j^*\}, \{x_{(j+1)1}^*\}, \{u_{jt}^*(\lambda_j^* x_{(j+1)1}^*)\})$ is the solution of Problem $P''$ and vice versa. The proofs of Theorems 1 and 2 are similar to those for the static problems due to Tang *et al.* (1989), and are omitted here.

## 3. Mixed coordination algorithm for convex problems

As mentioned in § 1, the algorithms presented here and in § 4 are for unconstrained problems only. That is, $X_{jt} = R^n$ and $U_{jt} = R^m$ for all $j$ and $t$ in (7). For convex problems considered in this section, the DDP of Yakowits and Rutherford (1984) is extended for low-level subproblems, and the modified Newton method is adopted for high-level optimization. Two major difficulties are associated with the modified Newton method under the mixed coordination framework (Tang *et al.* 1988, 1989). The first is how to obtain the gradient and hessian information in determining the Newton direction, since first- and second-order derivatives of the objective function with respect to all high-level variables are needed. The second is when to terminate a line search along a Newton direction, as the high level is a max–mini problem. An efficient way is developed here to find the high-level gradient and hessian information by exploiting the DDP algorithm. The line search stopping criterion, on the other hand, is based on the norm of the gradient vector (Dennis and Schnabel 1983). In this section, we shall first present the extended DDP algorithm to see how the needed information can be obtained, and then present the details of the high-level modified Newton method.

### 3.1. *Extended DDP for low-level subproblems*

DDP is a successive approximation technique for solving optimal control problems (Jacobsen and Mayne 1970, Ohno 1978, Yakowits and Rutherford 1984, Yakowits 1986, Chang *et al.* 1989). It consists of two steps: backward dynamic programming and successive policy construction. For Subproblems $(P-j)$, the backward dynamic programming procedure is first applied by making quadratic approximations of Subproblems $(P-j)$ along a nominal trajectory, and formulating at each stage a quadratic programming problem in terms of variational state and control. By solving the quadratic programming problem at each stage, coefficients of the variational feedback controls and the variational cost-to-go functions are obtained. The successive policy construction procedure then uses these coefficients and the nominal controls to construct new controls and states forward in time, and to calculate the new cost. If the cost is lower than the nominal one, the nominal trajectory is updated by the new trajectory. Otherwise the new controls are modified in a specific



The extension of Yakowits and Rutherford (1984) is needed to represent the variational feedback controls and the variational cost-to-go functions explicitly in terms of all relevant high-level variables. To see how this can be done, consider Subproblems $(P-j)$ with $\lambda_j$ and $x_{j1}$ given. Let $\{\bar{u}_{jt}, t = 1, \ldots, T\}$ be a given set of nominal controls and $\{\bar{x}_{jt}, t = 1, \ldots, T+1\}$ be the corresponding state trajectory. By taking a second-order Taylor series approximation of Problem $(P-j)$ at the last stage, the approximate quadratic programming problem in variational terms can be formulated as follows:

$$V_{jT}(\lambda_j, \delta x_{jT}) = \min_{\delta u_{jT}} QP[g_{jT}(x_{jT}, u_{jT}) - \lambda_j^T x_{j(T+1)}] \tag{13}$$

where $QP$ denotes the quadratic approximation operation, and

$$\begin{aligned} QP[g_{jT}(x_{jT}, u_{jT}) - \lambda_j^T x_{j(T+1)}] \equiv{}& \delta x_{jT}^T C_{1jT} \delta x_{jT} + \delta u_{jT}^T C_{2jT} \delta x_{jT} \\ &+ \delta u_{jT}^T C_{3jT} \delta u_{jT} + C_{4jT} \delta u_{jT} + C_{5jT} \delta x_{jT} \\ &+ \lambda_j^T C_{7jT} \delta u_{jT} + \lambda_j^T C_{8jT} \delta x_{jT} \end{aligned} \tag{14}$$

The coefficients $C_{1jT}$, $C_{2jT}$, $C_{3jT}$, $C_{4jT}$, $C_{5jT}$, $C_{7jT}$ and $C_{8jT}$ are defined as follows:

$$\left.\begin{aligned} C_{1jT} &\equiv \tfrac{1}{2}\nabla_{xx} g_{jT} \\ C_{2jT}^T &\equiv \nabla_{xu} g_{jT} \\ C_{3jT} &\equiv \tfrac{1}{2}\nabla_{uu} g_{jT} \\ C_{4jT}^T &\equiv \nabla_u g_{jT} \\ C_{5jT}^T &\equiv \nabla_x g_{jT} \\ C_{7jT} &\equiv -B_{jT} \\ C_{8jT} &\equiv -A_{jT} \end{aligned}\right\} \tag{15}$$

where $\nabla_x g_{jT}$, $\nabla_{xx} g_{jT}$ and so on denote the first- and second-order derivatives of the stagewise cost function $g_{jT}$. For convex problems, assume that matrix $C_{3jT}$ is positive definite; then problem (13) has a unique solution:

$$\begin{aligned} \delta u_{jT}^* &= -\tfrac{1}{2} C_{3jT}^{-1} (C_{4jT}^T + C_{2jT} \delta x_{jT} + C_{1jT}^T \lambda_j) \\ &\equiv \alpha_{jT} + \beta_{jT} \delta x_{jT} + \gamma_{jT} \lambda_j \end{aligned} \tag{16}$$

where

$$\left.\begin{aligned} \alpha_{jT} &\equiv -\tfrac{1}{2} C_{3jT}^{-1} C_{4jT}^T \\ \beta_{jT} &\equiv -\tfrac{1}{2} C_{3jT}^{-1} C_{2jT} \\ \gamma_{jT} &\equiv -\tfrac{1}{2} C_{3jT}^{-1} C_{7jT}^T \end{aligned}\right\} \tag{17}$$

are called control coefficients. Substituting (16) into (13), the optimal variational cost-to-go function is given by:

where

$$\left.\begin{aligned} D_{1jT} &= C_{1jT} - \beta_{jT}^T C_{3jT}\beta_{jT} \\ D_{2jT} &= C_{8jT} - 2\beta_{jT}^T C_{3jT}\gamma_{jT} \\ D_{3jT} &= -\gamma_{jT}^T C_{3jT}\gamma_{jT} \\ D_{4jT} &= C_{5jT} - 2\alpha_{jT}^T C_{3jT}\beta_{jT} \\ D_{5jT} &= -2\alpha_{jT}^T C_{3jT}\gamma_{jT} \end{aligned}\right\} \tag{19}$$

and $\Theta_{jT}$ is the sum of all other terms not containing $\lambda_j$ or $\delta x_{jT}$. Because $\partial L_H/\partial\lambda_j$ is very easy to obtain (as will be seen in (27)), coefficient $D_{5jT}$ in (19) will not be needed by the high level. Therefore $D_{5jt}$ for $1 \leqslant t < T$ will not be derived here.

By following a similar derivation the quadratic approximation of an intermediate stage $t$, $1 \leqslant t < T$, has the following form:

$$\begin{aligned} &QP[g_{jt}(x_{jt}, u_{jt}) + V_{j(t+1)}(\lambda_j, \delta x_{j(t+1)})] \\ &\quad \equiv \delta x_{jt}^T C_{1jt}\delta x_{jt} + \delta u_{jt}^T C_{2jt}\delta x_{jt} + \delta u_{jt}^T C_{3jt}\delta u_{jt} + C_{4jt}\delta u_{jt} \\ &\quad + C_{5jt}\delta x_{jt} + \lambda_j^T C_{6jt}\lambda_{jt} + \lambda_j^T C_{7jt}\delta u_{jt} + \lambda_j^T C_{8jt}\delta x_{jt} \end{aligned} \tag{20}$$

The calculations of $C_{1jt}$, $C_{2jt}$, $C_{3jt}$, $C_{4jt}$, $C_{5jt}$, $C_{6jt}$, $C_{7jt}$ and $C_{8jt}$ are as follows:

$$\left.\begin{aligned} C_{1jt} &= \tfrac{1}{2}\nabla_{xx} g_{jt} + A_{jt}^T D_{1j(t+1)} A_{jt} \\ C_{2jt} &= \nabla_{xu} g_{jt} + 2A_{jt}^T D_{1j(t+1)} B_{jt} \\ C_{3jt} &= \tfrac{1}{2}\nabla_{uu} g_{jt} + B_{jt}^T D_{1j(t+1)} B_{jt} \\ C_{4jt}^T &= \nabla_u g_{jt} + B_{jt}^T D_{4j(t+1)} \\ C_{5jt}^T &= \nabla_x g_{jt} + A_{jt}^T D_{4j(t+1)} \\ C_{6jt} &= D_{3j(t+1)} \\ C_{7jt} &= D_{2j(t+1)} B_{jt}^T \\ C_{8jt} &= D_{2j(t+1)} A_{jt}^T \end{aligned}\right\} \tag{21}$$

The optimal variational feedback control is given by:

$$\delta u_{jt}^* = \alpha_{jt} + \beta_{jt}\delta x_{jt} + \gamma_{jt}\lambda_j$$

where

$$\left.\begin{aligned} \alpha_{jt} &= -\tfrac{1}{2} C_{3jt}^{-1} C_{4jt}^T \\ \beta_{jt} &= -\tfrac{1}{2} C_{3jt}^{-1} C_{2jt} \\ \gamma_{jt} &= -\tfrac{1}{2} C_{3jt}^{-1} C_{7jt} \end{aligned}\right\} \tag{22}$$

Substituting (22) into (20), the optimal variational cost-to-go function is given by:



where

$$\left.\begin{aligned} D_{1jt} &= C_{1jt} - \beta_{jt}^T C_{3jt}\beta_{jt} \\ D_{2jt} &= C_{8jt} - 2\beta_{jt}^T C_{3jt}\gamma_{jt} \\ D_{3jt} &= C_{6jt} - \gamma_{jt}^T C_{3jt}\gamma_{jt} \\ D_{4jt} &= C_{5jt} - 2\alpha_{jt}^T C_{3jt}\beta_{jt} \end{aligned}\right\} \tag{24}$$

To let (20)–(24) also cover the $t = T$ case, we define the terminal conditions for equation (24) as follows:

$$\begin{aligned} D_{1j(T+1)} &= 0 \\ D_{2j(T+1)} &= -I \\ D_{3j(T+1)} &= 0 \\ D_{4j(T+1)} &= 0 \end{aligned}$$

where $I$ represents an identity matrix.

To ensure that the new controls we have constructed yield a cost lower than that of the nominal ones, the variational feedback control of (22) is modified as follows:

$$\delta u_{jt} = \varepsilon\alpha_{jt} + \beta_{jt}\delta x_{jt} + \varepsilon\gamma_{jt}\lambda_j \tag{25}$$

where parameter $\varepsilon$ is first set to be 1, then reduced by half if necessary until the new controls yield a cost lower than the nominal.

Since (23) represents the optimal variational cost-to-go function at stage $t$, the optimal variational cost for Subproblems $(P-j)$ is $V_{jt}$. As a result, the derivatives of $L_j^*(\lambda_j, x_{j1})$ with respect to high-level variables, e.g. $\partial L_j^*/\partial x_{(j+1)1}$, $\partial^2 L_j^*/\partial x_{(j+1)1}^2$, $\partial^2 L_j^*/\partial\lambda_j^2$ etc., are readily available from the coefficients of $V_{j1}$.

### 3.2. *High-level derivative information and modified Newton method*

The high-level variables are updated using the modified Newton method according to

$$\begin{bmatrix} \lambda_1 \\ x_{21} \\ \lambda_2 \\ x_{31} \\ \vdots \end{bmatrix}^{k+1} = \begin{bmatrix} \lambda_1 \\ x_{21} \\ \lambda_2 \\ x_{31} \\ \vdots \end{bmatrix}^{k} - \alpha^k H^{-1}\nabla L_H \tag{26}$$

where $H$ is the hessian of $L_H$, $\nabla L_H$ is the gradient of $L_H$, $k$ is the high-level iteration index, and the step size $0 \leqslant \alpha^k \leqslant 1$ is determined by an appropriate line search procedure. From (8), (10) and (23) the high-level gradients are

$$\frac{\partial L_H}{\partial\lambda_j} = x_{(j+1)1} - x_{j(T+1)} \tag{27 a}$$

The high-level hessian matrix is

$$H=\begin{bmatrix} 2D_{311} & I & 0 & 0 & \cdots \\ I & 2D_{121} & D_{221} & 0 & \cdots \\ 0 & D_{221} & 2D_{321} & I & \cdots \\ \vdots & 0 & I & 2D_{131} & \cdots \\ & \vdots & \vdots & \vdots & \end{bmatrix} \tag{28}$$

Note that this hessian matrix is with respect to high-level variables only, so its dimension will not be too large. Note also that $H$ is not positive definite nor positive semidefinite. In fact, it is an indefinite matrix since the high-level solution is a saddle point (see Theorem 1). Consequently it is difficult to use the value of $L_H$ as the criterion in determining the step size $\alpha^k$. To overcome this difficulty we use the norm of the high-level gradient as the stopping criterion, as used by Tang *et al.* (1988, 1989). Let $e$ denote the square of the euclidean norm of the high-level gradient, i.e.

$$e(\lambda_j, x_{(j+1)1}) = \sum_{j=1}^{M-1}\left[\left|\frac{\partial L_H}{\partial \lambda_j}\right|^2 + \left|\frac{\partial L_H}{\partial x_{(j+1)1}}\right|^2\right] \tag{29}$$

The goal of the high-level optimization is to reduce $e$ to zero or to minimize $e$. Therefore, if $e$ has no other local minimum, its value can be used as the stopping criterion for the high-level line search routine (Dennis and Schnabel 1983). It has been proved by Tang *et al.* (1989) that $e$ has a unique minimum for static problems. The proof of $e$ having a unique minimum for dynamic problems is similar and is omitted here. In our algorithm, if

$$e(\lambda_j^{i+1}, x_{(j+1)1}^{i+1}) < e(\lambda_j^i, x_{(j+1)1}^i) \tag{30}$$

is satisfied, where $i$ is the index within a line search, then the line search is stopped.

## 4. Mixed coordination for non-convex problems

In this section we extend the results of § 3 to non-convex problems. One basic technique in dealing with non-convex problems is to add quadratic convexification terms to the standard lagrangian function (see, for example, Bertsekas 1982, Luenberger 1984). This convexification process, however, may destroy the separability of the original problem, and prevent the decomposition of low-level subproblems. Many researchers have tried to overcome this difficulty by various approaches (see, for example, Bertsekas 1979 and Tanikawa and Mulai 1985, 1987 for static optimization). Because of the specific selection of high-level variables, our method preserves the separability of the original problem even when quadratic convexification terms are added. Further, the information needed for high-level optimization is again readily available from low-level subproblems, and the dimension of the high-level hessian matrix is the same as that for convex problems.



*Problem* $P_n$

$$\max_{\{\lambda_j\}} \min_{\{x_{(j+1)1}\}} \min_{\{u_{jt}\}} L_n, \quad \text{with } L_n \equiv \sum_{j=1}^{M}\sum_{t=1}^{T} g_{jt}(x_{jt}, u_{jt}) + \lambda_j^T[x_{(j+1)1} - x_{j(T+1)}] + \frac{c}{2}|x_{(j+1)1} - x_{j(T+1)}|^2 \tag{31}$$

subject to constraint (5). In Problem $P_n$ $\lambda_M \equiv x_{(M+1)1} \equiv 0$, $c$ is an appropriate positive constant, and subscript $n$ denotes non-convexity. The choice of $c$ will be discussed later in this section. By selecting $\{\lambda_j\}_{j=1}^{M-1}$ and $\{x_{(j+1)1}\}_{j=1}^{M-1}$ as high-level variables, Problem $P_n$ can be decomposed into the following $M$ subproblems.

*Problem* $(P-j')$ $j = 1, 2, \ldots, M$

$$\min_{\{u_{jt}\}} L_j', \quad \text{with } L_j' \equiv \sum_{t=1}^{T} g_{jt}(x_{jt}, u_{jt}) - (\lambda_j + cx_{(j+1)1})^T x_{j(T+1)} + \frac{c}{2}|x_{j(T+1)}|^2 \tag{32}$$

subject to

$$x_{j(t+1)} = f_{jt}(x_{jt}, u_{jt}), \quad t = 1, \ldots, T$$

The high-level problem, similar to Problem $(P-H)$ in § 2, is as follows.

*Problem* $P-H'$

$$\max_{\{\lambda_j\}} \min_{\{x_{(j+1)1}\}} L_H'$$

$$\text{where } L_H' = \sum_{j=1}^{M}\left(L_j'^*(\lambda_j, x_{j1}, x_{(j+1)1}) + \left[\lambda_{(j-1)}x_{j1} + \frac{c}{2}|x_{(j+1)1}|^2\right]\right) \tag{33}$$

where $\lambda_0 \equiv \lambda_M \equiv x_{(M+1)1} \equiv 0$, and $L_j'^*(\cdot)$ denotes the optimal low-level cost, given the high-level variables. From (32) and (33) we see that the separability of the original problem is preserved even with the presence of cross product terms $\{x_{j(T+1)}^T x_{(j+1)1}\}$. This is one advantage of mixed coordination over the augmented Lagrange relaxation method. To show the existence of a saddle point for Problem $(P-H')$ and the equivalence of the one-level and two-level approaches, the following notation is needed. Let

$$x_H \equiv (x_{21}^T, x_{31}^T, \ldots, x_{(j+1)1}^T, \ldots, x_{M1}^T)^T$$

$$\lambda \equiv (\lambda_1^T, \lambda_2^T, \ldots, \lambda_j^T, \ldots, \lambda_{M-1}^T)^T$$

$$z \equiv \begin{bmatrix} \lambda \\ x_H \end{bmatrix}$$

$$u_j \equiv (u_{j1}^T, u_{j2}^T, \ldots, u_{jT}^T)^T$$

$$u \equiv (u_1^T, u_2^T, \ldots, u_M^T)^T$$

We have Theorems 3 and 4 below.

*Theorem* 3

*Theorem* 4

Problem $(P-H')$ and problem $P_n$ are equivalent.

In Theorem 4 the two problems are equivalent in the sense that if $z^*$ is a local solution of Problem $(P-H')$ and $u^*(z^*)$ is a local optimal solution of Problem $(P-j')$, given $z^*$, then $(z^*, u^*(z^*))$ is a local solution of Problem $P_n$, and vice versa. The proof of Theorem 4 is similar to that for Theorem 2 and is omitted here.

4.2. *Extended DDP for low-level subproblems*

The major difference between (32) and (9) is that Subproblems $(P-j')$ contain not only the high-level variables $\lambda_j$ and $x_{j1}$, but also $x_{(j+1)1}$. Therefore, additional information from Subproblems $(P-j')$ with respect to $x_{(j+1)1}$ is needed at the high level. Consequently the forms of the optimal variational controls and variational cost-to-go functions for Subproblems $(P-j')$ have to be modified to include explicitly $x_{(j+1)1}$. By adapting a procedure similar to that in § 3 we have the following results for stage $t$ $(1 \leqslant t \leqslant T)$.

The quadratic approximation of Subproblems $(P-j')$ at stage $t$ is:

$$\begin{aligned} QP[g_{jt}(x_{jt}, u_{jt}) + V_{j(t+1)}(\lambda_j, \delta x_{j(t+1)}, x_{(j+1)1})] \\ \equiv \delta x_{jt}^T C_{1jt} \delta x_{jt} + \delta u_{jt}^T C_{2jt} \delta x_{jt} + \delta u_{jt}^T C_{3jt} \delta u_{jt} \\ + C_{4jt} \delta u_{jt} + C_{5jt} \delta x_{jt} + \lambda_j^T C_{6jt} \lambda_j + \lambda_j^T C_{7jt} \delta u_j \\ + \lambda_j^T C_{8jt} \delta x_j + x_{(j+1)1}^T C_{9jt}, x_{(j+1)1} + x_{(j+1)1}^T C_{10jt} \delta u_{jt} \\ + x_{(j+1)1}^T C_{11jt} \delta x_{jt} + x_{(j+1)1}^T C_{12jt} \lambda_j + C_{13jt} x_{(j+1)1} \end{aligned} \tag{34}$$

The calculations of $C_4$, $C_5$, $C_6$, $C_7$ and $C_8$ are the same as those in § 3 except that $A_{jt}$ is replaced by $\nabla_x f_{jt}$ and $B_{jt}$ by $\nabla_u f_{jt}$. The calculations of $C_1$, $C_2$, $C_3$, $C_9$, $C_{10}$, $C_{11}$, $C_{12}$ and $C_{13}$ are as follows:

$$\left.\begin{aligned} C_{1jt} &= \tfrac{1}{2}\Big[\nabla_{xx} g_{jt} + 2\nabla_x f_{jt}^T D_{1j(t+1)} \nabla_x f_{jt} + \sum_{i=1}^{n} (D_{4j(t+1)} + D_{2j(t+1)}^T \lambda_j \\ &\qquad + D_{7j(t+1)}^T x_{(j+1)1})_i \nabla_{xx}((f_{jt})_i)\Big] \\ C_{2jt} &= \nabla_{xu} g_{jt} + 2\nabla_x f_{jt}^T D_{1j(t+1)} \nabla_u f_{jt} + \sum_{i=1}^{n} (D_{4j(t+1)} + D_{2j(t+1)}^T \lambda_j \\ &\qquad + D_{7j(t+1)t} x_{(j+1)1})_i \nabla_{xu}((f_{jt})_i) \\ C_{3jt} &= \frac{1}{2}\Big[\nabla_{uu} g_{jt} + 2\nabla_u f_{jt}^T D_{1j(t+1)} \nabla_u f_{jt} + \sum_{i=1}^{n} (D_{4j(t+1)} + D_{2j(t+1)}^T \lambda_j \\ &\qquad + D_{7j(t+1)} x_{(j+1)1})_i \nabla_{uu}((f_{jt})_i)\Big] \\ C_{9jt} &= D_{6j(t+1)} \\ C_{10jt} &= D_{7j(t+1)} \nabla_u f_{jt} \end{aligned}\right\} \tag{35}$$



In the above equations $n$ is the dimension of $x$, the terms $\nabla_{xx}((f_{jt})_i)$ and so on are the block hessian matrices of the $i$th coordinate of $f_{jt}$. The matrices $D_6$, $D_7$, $D_8$ and $D_9$ are related to optimal cost-to-go functions, and will be presented later. Note that $C_1$, $C_2$ and $C_3$ contain $\lambda_j$ and $x_{(j+1)1}$. For simplicity, $\lambda_j$ and $x_{(j+1)1}$ are treated as constants in evaluating $C_1$, $C_2$ and $C_3$. The optimal variational control is:

$$\delta u_{jt}^* = \alpha_{jt} + \beta_{jt} \delta x_{jt} + \gamma_{jt} \lambda_j + \eta_{jt} x_{(j+1)1} \tag{36}$$

Substituting (36) into (34), the optimal variational cost-to-go is given by:

$$\begin{aligned} V_{jt}(\lambda_j, \delta x_{jt}, x_{(j+1)1}) = \delta x_{jt}^T D_{1jt} \delta x_{jt} + \lambda_j^T D_{2jt} \delta x_{jt} + \lambda_j^T D_{3jt} \lambda_j \\ + D_{4jt}^T \delta x_{jt} + D_{5jt}^T \lambda_j + x_{(j+1)1}^T D_{6jt} x_{(j+1)1} \\ + x_{(j+1)1}^T D_{7jt} \delta x_{jt} + x_{(j+1)1}^T D_{8jt} \lambda_j + D_{9jt}^T x_{(j+1)1} + \Theta_{jt} \end{aligned} \tag{37}$$

The derivations of $\alpha$, $\beta$, $\gamma$, $D_1$, $D_2$, $D_3$ and $D_4$ are as in § 3, with some of the terminal conditions modified as follows:

$$\left.\begin{aligned} D_{1j(T+1)} &= cI \\ D_{4j(T+1)} &= cx_{j(T+1)} \end{aligned}\right\} \tag{38}$$

The calculations of $\eta$, $D_6$, $D_7$, $D_8$, and $D_9$ are as follows:

$$\left.\begin{aligned} \eta_{jt} &= -\tfrac{1}{2} C_{3jt}^{-1} C_{10jt} \\ D_{6jt} &= C_{9jt} - \eta_{jt}^T C_{3jt} \eta_{jt} \\ D_{7jt} &= C_{11jt} - 2\eta_{jt}^T C_{3jt} \beta_{jt} \\ D_{8jt} &= C_{12jt} - 2\eta_{jt}^T C_{3jt} \gamma_{jt} \\ D_{9jt} &= C_{13jt} - 2\alpha_{jt}^T C_{3jt} \eta_{jt} \end{aligned}\right\} \tag{39}$$

The terminal conditions for the above iteratively defined coefficients are:

$$\left.\begin{aligned} \eta_{jT} &= -\tfrac{1}{2} C_{3jT}^{-1} C_{10jT} \\ D_{6j(T+1)} &= 0 \\ D_{7j(T+1)} &= -cI \\ D_{8j(T+1)} &= 0 \\ D_{9j(T+1)} &= 0 \end{aligned}\right\} \tag{40}$$

By similar reasoning to that which led to (25), (36) is modified as follows:

$$\delta u_{jt}^* = \varepsilon \alpha_{jt} + \beta_{jt} \delta x_{jt} + \varepsilon \gamma_{jt} \lambda_j + \varepsilon \eta_{jt} x_{(j+1)1} \tag{41}$$

where the choice of $\varepsilon$ is the same as that in (25).

4.3. *High-level derivative information*

From (31), (33) and (37) it is not difficult to see that the high-level gradients are:

$$\frac{\partial L_H}{\partial \lambda_j} = x_{(j+1)1} - x_{j(T+1)} \tag{42 a}$$

$$\frac{\partial L_H'}{}$$

The high-level hessian matrix is:

$$= \begin{bmatrix} 2D_{311} & I+D_{811} & 0 & \cdots & & \\ I+D_{811} & cI+2D_{611}+2D_{121} & D_{221} & D_{721} & 0 & \vdots \\ 0 & D_{221} & 2D_{321} & I+D_{821} & 0 & 0 \\ 0 & D_{721} & I+D_{821} & cI+2D_{621}+2D_{131} & D_{231} & D_{731} \quad \cdots \\ 0 & 0 & 0 & D_{231} & 2D_{331} & I+D_{831} \quad \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \tag{43}$$

For the choice of $c$, it should not be too small or too large. If it is too small the convexification may not be enough. If it is too large the convergence will be slowed down. The stopping criterion for high-level line search is the same as in § 3.

## 5. Numerical results

As mentioned by Yakowits and Rutherford (1984), there are no well-established, standard test problems for large-scale optimal control systems. In this section, five problems are tested. We start with a scalar problem to see how our algorithms work. We then apply our algorithms to multi-dimensional problems with uncoupled, coupled and non-linear system dynamics to see how robust the algorithms are in solving different types of problems.

As mentioned, low-level subproblems are solved by using the extended DDP algorithm, and the high-level problem is solved using the modified Newton method. For purposes of comparison, each problem is also solved using the one-level DDP algorithm without decomposition and coordination. In testing the two-level algorithms, a simple line search procedure is employed at each high-level Newton iteration. The step size is initially set to 1, and reduced by half as needed until the value of $e$ (as defined in (29)) decreases. The high-level convergence criterion is

$$\frac{|L_H^{k+1} - L_H^k|}{|L_H^k| + 1} \leqslant 0{\cdot}000001 \tag{44 a}$$

or

$$\nabla L_H \leqslant 0{\cdot}0001 \tag{44 b}$$

where $\nabla L_H$ is the high-level gradient as described in (27) or (42). The convergence criterion for the extended DDP algorithms is

$$\frac{|L_j^{k+1} - L_j^k|}{|L_j^k| + 1} \leqslant \rho \tag{44 c}$$

where $\rho = 0{\cdot}000001$ for the one-level DDP ($L_j \equiv J$ in this case), and $\rho = 0{\cdot}0001$ for the DDP in two-level algorithms.

For each test problem the initial state $x_1$ and initial nominal controls $\{\bar{u}_t\}_{t=1}^N$ are given. Nominal states $\{\bar{x}_t\}_{t=2}^N$ are then calculated according to system dynamics. To implement the two-level algorithms we also need initial conditions of Lagrange



the Lagrange multipliers for the given initial nominal trajectory using the first-order necessary conditions of optimal control. Consider Subproblem $(P-j)$ with constraint (5) relaxed by using Lagrange multipliers $\{\pi_{jt}\}_{t=1}^T$. A new lagrangian can be formed:

$$L_{j\pi} = \sum_{t=1}^{T} g_{jt}(x_{jt}, u_{jt}) - \lambda_j^T x_{j(T+1)} + \lambda_{(j-1)}^T x_{j1} + \sum_{t=1}^{T} [\pi_{jt}^T (x_{j(t+1)} - f_{jt}(x_{jt}, u_{jt}))] \tag{45}$$

By setting to zero the derivative of $L_{j\pi}$ with respect to $x_{j(T+1)}$, we have

$$-\lambda_j + \pi_{jT} = 0 \tag{46}$$

On the other hand, by setting to zero the derivative of $L_{j\pi}$ with respect to $u_{jT}$, we have

$$\frac{\partial g_{jT}}{\partial u_{jT}} - \left(\frac{\partial f_{jT}}{\partial u_{jT}}\right)^T \pi_{jT} = 0 \tag{47}$$

Combining (46) and (47), $\lambda_j$ is given by:

$$\lambda_j = \left[\left(\frac{\partial f_{jT}}{\partial u_{jT}}\right)^T\right]^+ \frac{\partial g_{jT}}{\partial u_{jT}} \tag{48}$$

where $[\,\cdot\,]^+$ denotes pseudo-inverse (Brogan 1985). When the above equation is evaluated at the initial nominal trajectory one obtains $\{\lambda_j^0\}$. In all the testing presented here, $\{\lambda_j^0\}$ are selected using this method.

*Test problem $T_1$: A scalar problem*

$$J = \sum_{t=1}^{8} [0{\cdot}5x_t^4 + (x_t - 2)^2 + (u_t - 2)^2]$$

subject to

$$x_{t+1} = x_t - u_t$$

with initial state $x_1 = 0{\cdot}5$, initial nominal control $\bar{u}_t = 0$ for all $t$, and $M = 2$, $T = 4$ (two subproblems, each with four stages). The high-level initial conditions are $x_{21}^0$ $(= x_5^0) = 0{\cdot}5$ and $\lambda_1^0 = 4$ according to (48).

*Test problem $T_2$*

This problem is taken from Chang *et al.* (1989).

$$J = \sum_{t=1}^{N} \left[ \sum_{i=1}^{n} (x_{it} - a_{it})^2 \sum_{i=1}^{m} u_{it}^2 + \sum_{i=1}^{m} u_{it}^2 + \sum_{i=1}^{m} \sum_{j>i}^{m} u_{it} u_{jt} + 100 \sum_{i=1}^{n} (x_{it} - a_{it})^2 \right]$$

subject to

$$x_{t+1} = A_t x_t + B_t u_t$$

with $N = 14$, $n = m = 2$, $a_{it} = 20$ for all $i$ and $t$, and $A_t = I_2$, $B_t = -I_2$. The initial state is $x_1 = [5 \quad 5]^T$, the initial nominal controls are $\bar{u}_{it} = -2$ for all $i$ and $t$. The

*Test problem* $T_3$

This test problem has the following $A$ and $B$ matrices:

$$A_t = I_2, \quad B_t = \begin{bmatrix} -1 & 0 \\ 1 & -1 \end{bmatrix}, \quad \text{for all } t$$

The cost function is the same as in Test problem $T_2$. The initial-state $x_1$, $N$, $M$, $T$, $n$, $m$ and $a_{it}$ are also the same as those in problem 2. The initial nominal controls are $\bar{u}_t = [-2 \quad -4]^T$ for all $t$. The high-level initial conditions are $\lambda^0 = [42 \quad 26]^T$ and $x_{21}^0 (= x_8^0) = [19 \quad 19]^T$.

*Test problem* $T_4$

This problem has a cost function similar to that in Test problem $T_2$:

$$J = \sum_{t=1}^{N} \left[ \sum_{i=1}^{n} (x_{it} - a_{it})^2 \sum_{i=1}^{m} u_{it}^2 + \sum_{i=1}^{m} u_{it}^2 + \sum_{i=1}^{m} \sum_{j>i}^{m} u_{it} u_{jt} + \sum_{i=1}^{n} (x_{it} - a_{it})^2 \right]$$

It has the following non-linear system dynamics

$$x_{t+1} = A_t x_t + B_t \omega(u_t)$$

where $A_t = I_2$, $B_t = -I_2$, $\omega(u_t) = [\sin(u_{1t}) \quad \sin(u_{2t})]^T$, and $N$, $M$, $T$ are the same as in problem 2. The low-level initial conditions are $x_1 = [5 \quad 5]^T$ and $\bar{u}_t = [-0{\cdot}3 \quad -0{\cdot}3]^T$ for all $t$. The high-level initial conditions are $\lambda_1^0 = [204 \quad 204]^T$ and $x_{21}^0 (= x_8^0) = [7{\cdot}06 \quad 7{\cdot}06]^T$. This is a non-convex problem, and the convexification parameter $c$ is chosen to be 10.

*Test problem* $T_5$

This problem is taken from Chang *et al.* (1989).

$$J = \sum_{t=1}^{N} [\exp(a_t^T x_t) + \exp(b_t^T u_t) - a_t^T x_t \exp(a_t^T x_t^*) - b_t^T u_t \exp(b_t^T u_t^*) + c_1 (x_t - x_t^*)^T (x_t - x_t^*) + c_2 (u_t - u_t^*) + d_t]$$

where

$$d_t = -\exp(a_t^T x_t^*) - \exp(b_t^T u_t^*) + a_t^T x_t^* \exp(a_t^T x_t^*) + b_t^T u_t^* \exp(b_t^T u_t^*)$$

The system dynamics is the same as that in $T_3$. In $T_5$, $n = m = 2$, $N = 42$, $M = 3$, $a_t = b_t = [0{\cdot}01 \quad 0{\cdot}01]^T$, $x_t^*$ and $u_t^*$ are user-designed optimal solutions, and $c_1 = c_2 = 0{\cdot}1$ guarantee the strict convexity of $J$. The initial state $x_1$ is $[0 \quad 0]^T$, the initial nominal controls are $\bar{u}_t = [0 \quad 0]^T$ for all $t$, and $u_t^* = [0{\cdot}5 \quad 0{\cdot}5]$ for all $t$. The high-level initial conditions are $\lambda_1^0 = \lambda_2^0 = [-0{\cdot}2 \quad -0{\cdot}1]^T$ and $x_{21}^0 = x_{31}^0 = [0 \quad 0]^T$.

Testings are performed on an IBM 3090 mainframe computer on MVS in the absence of a parallel processor. In the testings, we assume zero communication time and synchronous processing. We also assume that the number of subproblems equals the number of processors. A user-supplied subroutine (provided by UConn Computer Center) is used to time the execution of the algorithms. The low-level CPU time of an



adopted. Speedup, defined as $S_p \equiv T_s / T_p$ where $T_s$ is the one-level DDP processing time, measures the improvement in computation time using our two-level parallel algorithm for comparison with the one-level DDP. Testing results are summarized in Table 1.

From Table 1 we see that the final costs for both the one-level DDP and the two-level algorithms are either the same or very close. More importantly, the results show that significant speedups are achieved. The speedups for $T_2$ and $T_3$ are extremely high for a simulated two-processor system. Detailed examination of the execution of the two algorithms for $T_2$ and $T_3$ reveals that the parameter $\varepsilon$ (see (25)) was reduced several times during the first few iterations of the one-level DDP. This is probably due to the accumulated errors of DDP in solving long-horizon problems. This did not occur for the two-level algorithm, where each subproblem had a much shorter time horizon. The speedup measures, defined according to one of the definitions given on p. 15 of Bertsekas and Tsitsiklis (1989), is therefore very high. More tests are performed on $T_3$. The total number of stages reached 336 and the total number of subproblems 7. Other parameters, including $x_1$, $n$, $m$ and $a_{it}$, remained the same. The results are summarized in Table 2.

| | One-level DDP | | Two-level algorithm | | | | |
|---|---|---|---|---|---|---|---|
| | Exec. time, $T_s$ (s) | f.c.† | $M \times T$ | $K$‡ | Exec. time, $T_p$ (s) | $S_p$ | f.c. |
| $T_1$ | 0·010 | 41·95537 | 2 × 4 | 5 | 0·0055 | 1·82 | 41·95537 |
| $T_2$ | 0·032 | 104 163·562 | 2 × 7 | 3 | 0·015 | 2·13 | 104 163·625 |
| $T_3$ | 0·060 | 140 349·25 | 2 × 7 | 5 | 0·021 | 2·85 | 140 349·25 |
| $T_4$ | 0·022 | 5 533·28687 | 2 × 7 | 4 | 0·014 | 1·57 | 5 533·29297 |
| $T_5$ | 0·035 | 0·00004 | 3 × 14 | 3 | 0·022 | 1·58 | 0·00000 |

† f.c.: final cost.
‡ $K$: number of high-level iterations.

Table 1. Summary of test results.

| | One-level DDP | | New two-level algorithm | | | |
|---|---|---|---|---|---|---|
| Stages | Exec. time, $T_s$ (s) | f.c. | $M \times T$ | Exec. time, $T_p$ (s) | $S_p$ | f.c. |
| 42 | 0·080 | 140 349·92 | 3 × 14 | 0·036 | 2·22 | 140 349·94 |
| 84 | 0·200 | 140 349·92 | 3 × 28 | 0·068 | 2·94 | 140 349·92 |
| 168 | 0·360 | 140 349·92 | 3 × 56 | 0·116 | 3·10 | 140 349·92 |
| 336 | 0·700 | 140 349·92 | 3 × 112 | 0·24 | 2·91 | 140 349·92 |
| 42 | | | 7 × 6 | 0·04 | 2·00 | 140 349·94 |
| 84 | | | 7 × 12 | 0·058 | 3·45 | 140 349·92 |
| 168 | | | 7 × 24 | 0·092 | 3·91 | 140 349·92 |

In Table 2 the speedups increase when the number of stages and the number of subproblems increase, as expected. For a fixed size problem it may not be advantageous to divide a problem into too many subproblems. For example, the speedup for $T_3$ with $N = 42$ and $M = 3$ is 2·22. The speedup for the same problem but with $M = 7$ is reduced to 2 because of the increase in high-level computational load. On the other hand, the speedup does increase for larger problems as $M$ increases. For example, with $N = 336$ and $M = 3$ the speedup for $T_3$ is 2·91. The speedup with $M = 7$ is increased to 4·72. This indicates that large speedups can potentially be achieved for problems with large numbers of stages and subproblems. We therefore believe that the algorithms for solving long-horizon optimal control problems under a parallel processing environment will be advantageous.

## 6. Concluding remarks

This paper presents a new approach for solving long-horizon optimal control problems using the mixed coordination method. The idea is to decompose a problem along the time axis into smaller subproblems. The requirement that the initial state of a subproblem equal the terminal state of the preceding subproblem is relaxed by using Lagrange multipliers. The Lagrange multipliers and the initial states of subproblems are then selected as high-level variables. For a given set of high-level variables the low-level subproblems are decoupled and can be solved in parallel. By exploiting the DDP algorithm the high-level gradient and hessian are made available once low-level subproblems are solved. Therefore the modified Newton method is ideal for high-level optimization. Further, with specific selection of high-level variables the separability of the original problem is preserved even when quadratic convexification terms are added. As a result, our algorithm can be applied to non-convex problems. Numerical results show that our algorithm outperforms the one-level DDP under a simulated parallel processing environment and is suitable for long-horizon optimal control problems.

Compared to the incentive and target coordination schemes mentioned in § 1, the new approach offers an advantage over incentive coordination where the low-level subproblems are not completely decoupled. Additional steps have to be taken to decouple them. Compared to target coordination, the new approach offers the advantage of being able to handle non-convex problems. Further, the terminal state of each subproblem is free in the new approach in contrast to the fixed terminal state case of the target coordination. This might have special value when extended to constrained problems. This issue is currently under investigation.

### Acknowledgment

This work was supported in part by the National Science Foundation, under Grants ECS-3513163, ECS-8717167, ECS-8512815, and ECS-8717235.

## Appendix

*Proof of Theorem 3*

First examine $L'_H$ in (23) with $\lambda$ fixed. The block-diagonal component of the



for $j = 1, \ldots, M-1$. Although $\nabla^2_{x(j+1)1} L^*_j$ or $\nabla^2_{x(j+1)1} L^*_{(j+1)}$ may not be positive definite in $x_{(j+1)1}$, the term $cI$ with sufficiently large $c$ will make $\nabla^2_{x_H} L'_H$ positive definite, implying that $L'_H$ is convex in $x_H$. On the other hand, define

$$\Phi(\lambda) \equiv \min_{x_H} L'_H(\lambda, x_H)$$

and

$$h(x_H) = \begin{bmatrix} x_{21} - x_{1(T+1)} \\ x_{31} - x_{2(T+1)} \\ x_{41} - x_{3(T+1)} \\ \vdots \end{bmatrix}$$

Then the hessian of $\Phi$ is (Luenberger 1984, p. 399)

$$\nabla^2_\lambda \Phi(\lambda) = -\nabla_{x_H} h(x_H)(\nabla^2_{x_H} L'_H(\lambda, x_H))^{-1} \nabla_{x_H} h(x_H)^T$$

Since $\nabla^2_{x_H} L'_H(\lambda, x_H)$ is positive definite and $\nabla_{x_H} h(x_H)$ is full rank, $\nabla^2_\lambda \Phi(\lambda)$ is negative definite, implying that $\Phi(\lambda)$ is concave in $\lambda$. Therefore the solution of Problem $(P - H')$ is a saddle point. □

### References

Bertsekas, D. P., 1979, Convexification procedure and decomposition methods for nonconvex optimization problems. *Journal of Optimization Theory and Applications*, **29,** 169–197; 1982, *Constrained Optimization and Lagrange Multiplier Methods* (New York: Academic Press).

Bertsekas, D. P., and Tsitsiklis, J. N., 1989, *Parallel and Distributed Computation* (Englewood Cliffs, NJ: Prentice Hall).

Brogan, W. L., 1985, *Modern Control Theory*, second edition (Englewood Cliffs, NJ: Prentice Hall).

Bromberg, M., Chang, T. S., and Luh, P. B., 1989 a, Quadratic incentive coordination for non-convex optimal control problems. Part 1: Theory. *International Journal of Control*, **49,** 433–448; 1989 b, Quadratic incentive coordination for non-convex optimal control problems. Part 2: Parallel algorithms. *Ibid.*, **49,** 449–469.

Chang, S. C., Chang, T. S., and Luh, P. B., 1989, A hierarchical decomposition for large-scale optimal control problems with parallel processing structure. *Automatica*, **25,** 77–86.

Chang, T. S., Jin, X. X., Luh, P. B., and Miao, X. Y., 1990, Large-scale convex optimal control problems. Time decomposition, incentive coordination and parallel algorithm. *I.E.E.E. Transactions on Automatic Control*, **35**(1).

Dennis, J., Jr, and Schnabel, R., 1983, *Numerical Methods for Unconstrained Optimization and Numerical Equations* (Englewood Cliffs, NJ: Prentice Hall).

Jacobson, D., and Mayne, D., 1970, *Differential Dynamic Programming* (New York: Elsevier).

Jamshidi, M., 1983, *Large-Scale Systems* (Amsterdam: North-Holland).

Luenberger, D., 1984, *Linear and Nonlinear Programming*, second edition (New York: Addison-Wesley).

Mesarovic, M. D., 1969, Two coordination principles and their applications in large-scale system control. *Proceedings of the IFAC Congress*, Warsaw, Poland.

Mukai, H., and Polak, E., 1975, A quadratically convergent primal dual algorithm with global convergence properties for solving optimization problems with equality constraints. *Mathematical Programming*, **9,** 336–349.

Ohno, K., 1978, A new approach to differential dynamic programming for discrete time

TANG, J., LUH, P. B., and CHANG, T. S., 1988, Optimization with the mixed coordination method. *Proceedings of the 1988 American Control Conference*, Atlanta, Georgia, pp. 1811–1816; 1989, The mixed coordination method for nonlinear programming problems with separable structures. *International Journal of Control*, **50,** 1461–1486.

TANIKAWA, A., and MUKAI, H., 1985, A new technique for nonconvex prime dual decomposition of a large-scale separable optimization problem. *I.E.E.E. Transactions on Automatic Control*, **30,** 133–143; 1987, New lagrangian function for nonconvex primal-dual decomposition. *Computers and Mathematics, with Applications*, **13,** 661–676.

YAKOWITS, S., 1986, The stagewise Kuhn–Tucker condition and differential dynamic programming. *I.E.E.E. Transactions on Automatic Control*, **31,** 25–30.

YAKOWITS, S., and RUTHERFORD, B., 1984, Computational aspects of discrete-time optimal control. *Applied Mathematical Computing*, **15,** 29–45.