Pareto Optimality and Multiobjective Trajectory Planning for a 7-DOF Redundant Manipulator

Alexis Guigue, Mojtaba Ahmadi, *Member, IEEE*, Rob Langlois, and M. John D. Hayes

***Abstract*—This paper presents a novel approach to solve multiobjective robotic trajectory planning problems. It proposes to find the Pareto optimal set, rather than a single solution usually obtained through scalarization, e.g., weighting the objective functions. Using the trajectory planning problem for a redundant manipulator as part of a captive trajectory simulation system, the general discrete dynamic programming (DDP) approximation method presented in our previous work is shown to be a promising approach to obtain a close representation of the Pareto optimal set.When compared with the set obtained by varying the weights, the results confirm that the DDP approximation method can find approximate Pareto objective vectors, where the weighting method fails, and can generally provide a closer representation of the actual Pareto optimal set.**

***Index Terms*—Dynamic programming, multiobjective trajectory planning, Pareto optimality, redundant robotic manipulator.**

I. Introduction

Motion planning for redundant robotic manipulators involves an optimization process that leads to generation of feasible joint trajectories. The local [2] and global [3]–[6] approaches to redundancy resolution have been extensively studied in the past. The global approaches, which are also the focus of this paper, formulate the redundancy resolution problem either as a constrained problem in calculus of variations [4], [5] or as a constrained optimal control problem [3], [6]. Constraints typically include joint constraints, such as mechanical, speed, and torque limits [3], [5], or task space constraints, such as obstacles [3], [5] and motion constraints at the end effector [4], [6]. Constrained variational and optimal control problems are, in general, difficult to solve numerically [7]; however, several resolution methods have been proposed for trajectory planning problems [3], [5], [6].

In many applications, one may find it desirable to optimize two or more objective functions (or *performance criteria*). Trajectory planning problems with multiple objective functions have not been studied beyond utilizing the weighting method, which belongs to the class of scalarization methods. The general principle for these methods is to transform the original multiobjective optimization problem into a parameterized single objective optimization problem and solve this resulting problem. For the weighting method, the parameters are the weights. As such, the majority of the previous work on redundancy resolution, either local [8], [9] or global [3]–[5] has generated only a single joint trajectory, to which corresponds a single Pareto optimal element in the objective space, which is defined as a *Pareto objective vector*. In this paper, it is proposed instead to provide the entire set of Pareto objective vectors or a close representation of this set. Once the Pareto optimal set is identified, the user can more freely choose a Pareto objective vector and the associated joint trajectory based on secondary criteria or other application requirements, which may not be easily modeled with a performance criterion.

By varying the weights, it is possible with the weighting method to obtain several Pareto objective vectors and, therefore, a representation of the Pareto optimal set. However, it will be illustrated in this paper, using a specific robotic problem, how a large subset of the Pareto optimal set may not be obtained in such a way. Alternatively, for the same robotic problem, it is proposed to use the numerical approximation method presented in [1]. This numerical approximation method, which uses discrete dynamic programming (DDP), applies to the general class of multiobjective deterministic finite-horizon optimal control problem to which trajectory planning problems, such as the one presented in this paper, can be shown to belong. The convergence results obtained in [1] suggest that the DDP approximation method will provide a close representation of the Pareto optimal set. Although dynamic programming has previously been used to solve trajectory planning problems [6], [10], [11], to our knowledge, this is the first time it is applied to determine the Pareto optimal set for a trajectory planning problem for a redundant manipulator. A preliminary and simplified implementation of this paper, without the collision constraints, was presented in [12] before the formal formulation of the DDP approximation method in [1].

The robotic system used in this paper is part of a captive trajectory simulation (CTS) system [13] operating inside the trisonic blowdown wind tunnel facilities at the National Research Council Canada. The main objective of such a system is to predict the trajectory of a *store* (an object released from a parent aircraft) to ensure its safe clearance from the aircraft. During a wind tunnel run, an 8-degree-of-freedom (DOF) CTS manipulator [13] moves the store through various positions and orientations with respect to the parent aircraft model to collect the aerodynamic loads acting on the store. The trajectory planning problem arising from the CTS system involves two performance criteria that includes the joint speed norm and the aerodynamic interference, which will be discussed later. For this problem, it is important to obtain a good representation of the Pareto optimal set as the choice of the final joint trajectory will critically depend on the operating conditions of the wind tunnel (Mach number) and the required measurement precision.

This paper is organized as follows. In Section II, the trajectory planning problem arising from the CTS system is formulated. It is first solved in Section III with the weighting method and then in Section IV with the DDP approximation method. The results show that a closer representation of the Pareto optimal set is obtained with the DDP approximation method. Finally, conclusions and future work are discussed in Section V.

II. Problem Formulation

The main type of experiment conducted with a CTS system is a *grid survey* in which the store has to follow a sequence of straight-line paths between grid points, which are defined by positions and orientations. During the motion, the aerodynamic loads acting on the store are measured by a high-precision force sensor and collected in a look-up table, which later serves as input to an equation of motion solver to simulate the store release trajectory. However, the presence of the manipulator in the test section can degrade the accuracy of the data collected by causing flow interference, i.e., it changes the flow angularity around the parent aircraft. To minimize this effect, all exposed links of the manipulator are aerodynamically shaped; the seventh link is particularly made in the form of a *gooseneck*. Alternatively, the redundancy of the manipulator can be exploited to reduce the flow interference; preference can be given to joint trajectories with gooseneck configurations involving less flow interference. Moreover, as grid surveys can

Manuscript received May 1, 2010; revised; accepted August 13, 2010. Date of publication September 23, 2010; date of current version December 8, 2010. This paper was recommended for publication by Associate Editor T. Simeon and Editor K. Lynch upon evaluation of the reviewers' comments.

The authors are with the Department of Aerospace and Mechanical Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada (e-mail: aguigue@connect.carleton.ca; mahmadi@mae.carleton.ca; rlangloi@mae.carleton.ca; jhayes@mae.carleton.ca).

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/TRO.2010.2068650

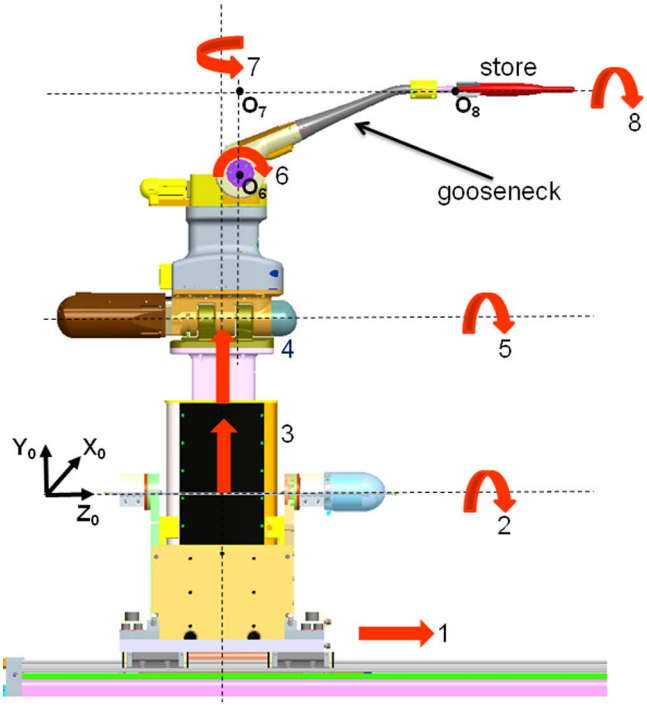

Fig. 1. Joint motions for the CTS manipulator.

be performed over a large range of speeds (Mach 0.7 to 1.3), actuator loading requirements can be very different. In general, joint trajectories with low joint speeds are preferred; however, this preference becomes stronger at higher wind speeds.

A complex iterative design process [13] resulted in the 8-DOF redundant manipulator designed, as illustrated in Fig. 1. This manipulator features a main translational joint, a boom with two revolute and two prismatic joints (telescopic arrangement), a wrist with pitch and yaw joints, and a final joint providing the roll. No singularities exist within the operating envelope, which is a box around the airplane model. Because of the particular joint arrangement of this manipulator: First, the two prismatic joints 3 and 4 can be treated as a single joint, and second, the inverse kinematics can be found in a closed form, which is omitted here for the sake of brevity. The inverse kinematics equation for the CTS manipulator can be stated as follows:

$$\mathbf{q} = \mathbf{g}(\mathbf{p}, v) \tag{1}$$

where $\mathbf{q} \in \mathbf{R}^7$ denotes the joint configuration, $\mathbf{p} \in \mathbf{R}^6$ denotes the position, *Z–Y–X* denotes Euler angles of the end effector, and the scalar $v = \mathbf{q}_2 + \mathbf{q}_5$ is defined as the *redundancy parameter*. For a given $\mathbf{p}$, the set $\mathcal{A}(\mathbf{p})$ of values that v can be taken from is a function of $\mathbf{p}$ and is included in the bounded interval $[v_{\min}, v_{\max}] = [\mathbf{q}_{\min,2} + \mathbf{q}_{\min,5}, \mathbf{q}_{\max,2} + \mathbf{q}_{\max,5}]$, where $\mathbf{q}_{\min}$ and $\mathbf{q}_{\max}$ are the joint mechanical limits.

In a grid survey, the joint trajectories $\mathbf{q}(t)$ must satisfy several constraints. First, the end effector must follow the prescribed store trajectory $\mathbf{p}(t)$:

$$\mathbf{f}(\mathbf{q}(t)) = \mathbf{p}(t) \tag{2}$$

where $\mathbf{f}(\cdot)$ denotes the forward kinematics. Meanwhile, the joint configuration is limited by the joint mechanical limits

$$\mathbf{q}_{\min} \leq \mathbf{q}(t) \leq \mathbf{q}_{\max} \tag{3}$$

the self-collision between the manipulator links, and the collision between the manipulator and its environment. Collision avoidance can be treated as the one-dimensional (1-D) constraint

$$d_{\text{safe}} - c(\mathbf{q}(t)) \leq 0 \tag{4}$$

where $c(\cdot)$ is a function that returns the minimum distance between any pair of geometries composing the manipulator and its environment [14], and d_{safe} is a safety distance. The joint speeds along the joint trajectory are also limited

$$-\dot{\mathbf{q}}_{\max} \leq \dot{\mathbf{q}}(t) \leq \dot{\mathbf{q}}_{\max}. \tag{5}$$

Finally, the initial joint configuration $\mathbf{q_0}$ satisfying $\mathbf{p}(t_0) = \mathbf{f}(\mathbf{q_0})$ is supposed to be known. To obtain good tracking performance from the manipulator, the joint trajectories must be sufficiently smooth and are, therefore, taken in the set $C^1([t_0, t_f], \mathbf{R}^7)$. The set of feasible joint trajectories $\mathcal{T}(\mathbf{q_0})$ is defined as the subset of $C^1([t_0, t_f], \mathbf{R}^7)$ with initial joint configuration $\mathbf{q_0}$ such that the constraints (2)–(5) are satisfied.

Because of the manipulator redundancy, the number of joint trajectories in $\mathcal{T}(\mathbf{q_0})$ is infinite in most cases. For the joint trajectory selection, two performance criteria are considered. The first criterion selected

$$f_1(\mathbf{q}(t)) = \frac{1}{2}\dot{\mathbf{q}}^T(t)\dot{\mathbf{q}}(t)$$

aims to minimize the joint speed norm. The second criterion selected aims to reduce the flow interference caused by the manipulator, which can be achieved by maintaining the gooseneck as close as possible to the vertical plane with normal $\mathbf{X_0}$. The angle α between this plane and the plane of the gooseneck, which is defined by the three origins O_6, O_7, and O_8 obtained with the Denavit–Hartenberg convention, is as follows:

$$\alpha = \arccos\left(\left\langle \mathbf{X_0}, \frac{\overrightarrow{O_7O_8} \times \overrightarrow{O_7O_6}}{\|\overrightarrow{O_7O_8} \times \overrightarrow{O_7O_6}\|} \right\rangle\right).$$

Therefore, the second criterion is $f_2(\mathbf{q}(t)) = \alpha$, where the coordinates of O_6, O_7, and O_8 can be obtained with the forward kinematics. Finally, it is desirable to minimize the performance criteria f_1 and f_2 over the entire motion. Therefore, the vector-valued function $\mathbf{F}(\cdot): \mathbf{q}(\cdot) \in C^1([t_0, t_f], \mathbf{R}^7) \rightarrow \mathbf{R}^2$ is introduced as

$$\mathbf{F}(\mathbf{q}(\cdot)) = \left(\int_{t_0}^{t_f} f_1(\mathbf{q}(t))\mathrm{d}t, \int_{t_0}^{t_f} f_2(\mathbf{q}(t))\mathrm{d}t\right). \tag{6}$$

As mentioned earlier, the trajectory generation problem in a grid survey, which is denoted by (P), is therefore a multiobjective problem in calculus of variations. The objective space for (P) is $\mathbf{F}(\mathcal{T}(\mathbf{q_0}))$. To solve (P), we propose to find the Pareto optimal set $\mathcal{E}(\mathrm{cl}(\mathbf{F}(\mathcal{T}(\mathbf{q_0}))), \mathbf{R}^2_+)$ and the corresponding optimal joint trajectories $\mathbf{q}^*(\cdot) \in \mathcal{T}(\mathbf{q_0})$, if they exist. The closure of the objective space $\mathrm{cl}(\mathbf{F}(\mathcal{T}(\mathbf{q_0})))$ is taken to guarantee the existence of Pareto objective vectors [15, p. 51, Th. 3.2.2]. In the context of single objective optimization, this amounts to using the infimum instead of the minimum in the formulation of the problem, when the existence of a minimizing joint trajectory cannot be guaranteed.

III. Solving (P) With the Weighting Method

The motivation for using the weighting method for obtaining a representation of the Pareto optimal set is that the objective vector(s) corresponding to the optimal solution(s) of the scalarized problem can be shown to be Pareto optimal [15, p. 72, Th. 3.4.3].Hence, by varying the weights, a subset of the Pareto optimal set can be obtained. The question is then whether all the Pareto objective vectors can be obtained this way. Unless the objective space is convex, the answer to

this question, in general, is no. Moreover, even in the most favorable case of a convex objective space, it is, in general, difficult to obtain a uniform distribution of Pareto objective vectors [16]. We illustrate in the following these two weaknesses with (P).

For simplicity of the numerical resolution, all the constraints described in Section II except the end-effector trajectory constraint (2) are removed. Therefore, after normalization of the weights, the *scalarized problem* $(\mathrm{P_s})$ is to find

$$\inf_{\mathbf{q}(\cdot)\in C_1([t_0,t_f],\mathbf{R}^7)} \int_{t_0}^{t_f} (f_1(\mathbf{q}(t)) + w f_2(\mathbf{q}(t)))\mathrm{d}t$$

subject to

$$\mathbf{f}(\mathbf{q}(t)) = \mathbf{p}(t)$$

with initial joint configuration $\mathbf{q_0}$, and where $w > 0$ is the *weight*. The numerical resolution of $(\mathrm{P_s})$ follows the exact same steps as in [4].

For the numerical experiments as following, two positions $\mathbf{p_0} = [55, -6, -31, 0, 0, 0]$ and $\mathbf{p_f} = [55, -16, -27, 0, 0, 0]$ in the task space of the manipulator are considered (units are in inches). A sufficiently smooth straight-line trajectory $\mathbf{p}(t)$ with duration 6.11 s, i.e., the time needed to travel from $\mathbf{p_0}$ to $\mathbf{p_f}$ at the maximum allowed Cartesian velocity, is generated between these two positions. The scalarized problem $(\mathrm{P_s})$ is solved for $w = 0, 1, \ldots, 30$. For $w = 0$, where only the first objective function is considered, the optimal solution is unique. Therefore, the corresponding objective vector can be shown to be Pareto optimal. The Pareto objective vectors obtained for each value of w are displayed in Fig. 2(a). A closer look at the Pareto objective vectors obtained for $w > 0$ is provided in Fig. 2(b).

Fig. 2(a) and (b) illustrates that a small variation in w might result in a large variation in the Pareto objective vector and conversely that a large variation in w might result in a small variation in the Pareto objective vector. Indeed, let $\mathbf{z_1} = (0.0770, 0.3053)$ and $\mathbf{z_3} = (0.3608, 0.0044)$ be the Pareto objective vectors obtained, respectively, for $w = 0$ and 1. It can be seen in Fig. 2(a) that there is a considerable jump between $\mathbf{z_1}$ and $\mathbf{z_3}$, whereas, as shown in Fig. 2(b), the change in the Pareto objective vector is not significant between $w = 1$ and 30. Therefore, it is difficult to predict a set of weights, which will result in a uniform distribution of Pareto objective vectors.

By illustrating that not all the Pareto objective vectors can be obtained with the weighting method is more difficult as it requires the knowledge of the Pareto optimal set, which is not available analytically. To obtain an approximation of the objective space, and hence of the Pareto optimal set for (P), we use the joint trajectories generated during the numerical resolution of $(\mathrm{P_s})$. The objective vectors corresponding to these trajectories are plotted with dots in Fig. 3. It is clear that the objective space Z for (P) is not convex. Let $\mathbf{z_2}$ be the intersection between the vertical line passing through $\mathbf{z_3}$ and the boundary of the objective space. The objective vectors along the boundary of the objective space between $\mathbf{z_1}$ and $\mathbf{z_2}$ are Pareto optimal. However, it is interesting to note that there is a large subset of these objective vectors that cannot be obtained with the weighting method. To see this [16], we can rewrite $(\mathrm{P_s})$ as follows:

$$\inf_{(\mathbf{F}_1,\mathbf{F}_2)\in Z} \mathbf{F}_1 + w\mathbf{F}_2.$$

For example, taking $w = 2.5$, the line with slope $-1/w = -1/2.5$ matches the tangent to the objective space at $\mathbf{z_4}$. This line also intersects the objective space at $\mathbf{z_5}$ and, therefore, can slide further down, which decreases the value of $\mathbf{F}_1 + w\mathbf{F}_2$. This proves that $\mathbf{z_4}$ cannot be obtained with the weighting method. Note that $\mathbf{z_4}$ is, in fact, a Pareto objective vector.

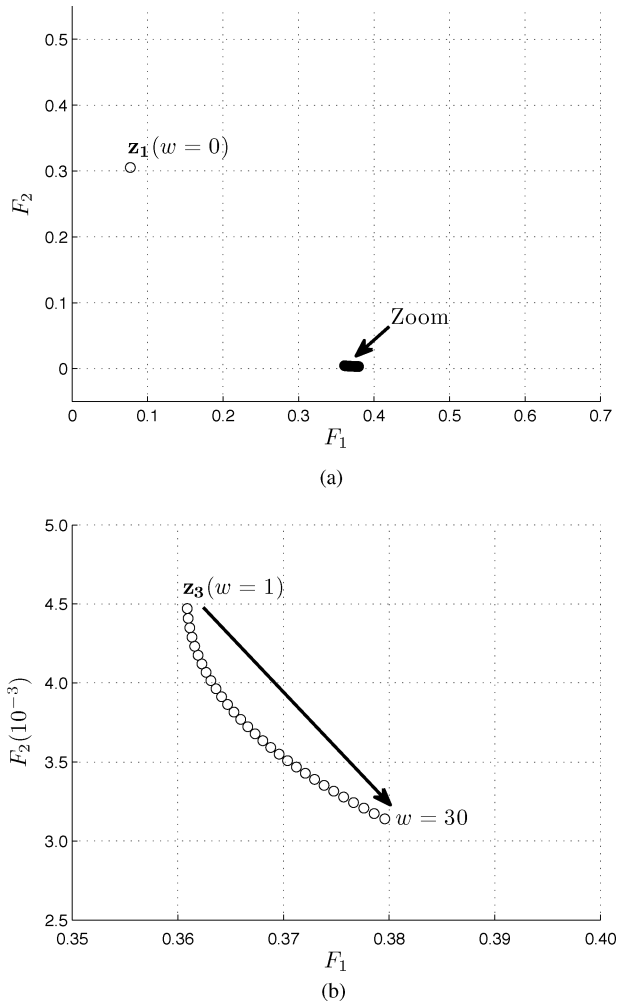

Fig. 2. Pareto objective vectors obtained with the weighting method (○). (a) $w = 0, 1, \ldots, 30$. (b) Zoom for $w > 0$.

IV. Solving (P) With the Discrete Dynamic Programing Approximation Method

In previous work [1], a DDP approximation method for a general class of multiobjective deterministic finite-horizon optimal control problem was proposed, where the objective space was assumed to be partially ordered by a cone. Here, the same method is applied to find an approximate Pareto optimal set for (P). The DDP approximation method proposed in [1] consists of a two-step discretization in time and state space. Following the first-order time discretization, the dynamic programming principle is used to find the multiobjective dynamic programming equation that is equivalent to the resulting discrete problem. The multiobjective dynamic programming equation is finally discretized in the state space.

The application of the DDP approximation first requires the reformulation of (P) in terms of the redundancy parameter v using the inverse kinematics equation (1). As a result, (P) becomes the problem to find the Pareto optimal set $\mathcal{E}(\mathrm{cl}(\widetilde{\mathbf{F}}(\mathcal{T}(v_0))), \mathbf{R}^2_+)$, where $\widetilde{\mathbf{F}}(\cdot):\ v(\cdot) \in C^1([t_0, t_f], \mathbf{R}) \to \mathbf{R}^2$ is defined as the vector-valued objective function $\mathbf{F}$ when expressed in terms of v

$$\widetilde{\mathbf{F}}(v(\cdot)) = \left(\int_{t_0}^{t_f} \widetilde{f}_1(t, v(t), \dot{v}(t))\mathrm{d}t, \int_{t_0}^{t_f} \widetilde{f}_2(t, v(t))\mathrm{d}t \right)$$

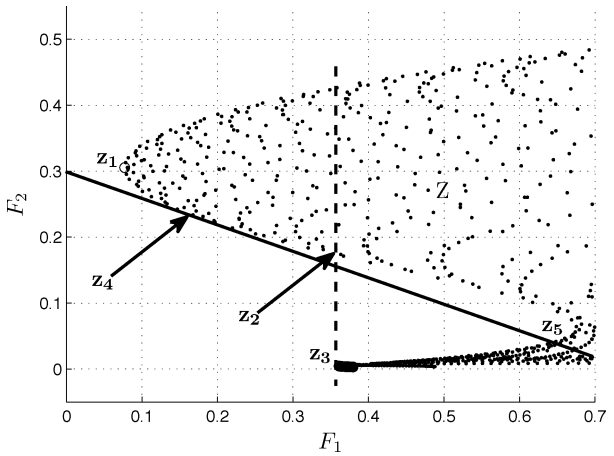

Fig. 3. Objective space Z. The Pareto objective vector $\mathbf{z_4}$ cannot be obtained with the weighting method.

and $\mathcal{T}(v_0)$ is the subset of $C^1([t_0, t_f], \mathbf{R})$ such that the following constraints are satisfied:

$$v(t) \in \mathcal{A}(t) \tag{7}$$

$$\dot{v}(t) \in \mathcal{B}(t, v(t)) \tag{8}$$

with initial redundancy parameter v_0. The explicit dependance on the time for the functions $\widetilde{f}_1$ and $\widetilde{f}_2$ comes from the term $\mathbf{p}(t)$. In the reformulation in terms of the redundancy parameter v, since all the equations are expressed in terms of v, the end-effector trajectory constraint (2) disappears. The set $\mathcal{A}(t)$ introduced in Section II represents the set of redundancy parameter values at a given time t along $\mathbf{p}(\cdot)$ such that the joint mechanical limits (3) are respected. From now on, this set also takes into account the collision constraint (4). Finally, the set $\mathcal{B}(t, v)$ represents the set of values at a given time t and redundancy parameter v that can be taken by the derivative of the redundancy parameter such that the maximum joint speed (5) is respected. From the definition of the redundancy parameter, it follows that $\mathcal{B}(t, v(t))$ is bounded.

A. First-Order Discretization in Time

The first step in the DDP approximation method is to proceed to a first-order discretization in the time of (7), (8), and $\widetilde{\mathbf{F}}(\cdot)$. The time step h is $(t_f - t_0)/N_T$, where N_T is the number of discretization steps. Let $v_i = v(t_i),\ i = 0, \ldots, N_T$, and $\dot{v}_i = \dot{v}(t_i),\ i = 0, \ldots, N_T - 1$. The derivative of the redundancy parameter $\dot{v}_i$ at time t_i is approximated using the forward Euler scheme

$$\dot{v}_i = \frac{v_{i+1} - v_i}{h}.$$

The integrals in $\widetilde{\mathbf{F}}(\cdot)$ are approximated with the rectangle formula. Defining $\mathcal{A}_i^h = \mathcal{A}(t_i)$ and $\mathcal{B}_i^h(v_i) = \mathcal{B}(t_i, v_i)$, the constraints (7) and (8) become

$$v_i \in \mathcal{A}_i^h \tag{9}$$

$$\dot{v}_i \in \mathcal{B}_i^h(v_i). \tag{10}$$

At step k, if $\mathcal{T}_k^h(v_k)$ denotes the set of discrete trajectories $\{v_i,\ i = k+1, \ldots, N_T\}$ with initial condition v_k such that (9) and (10) are satisfied, then the discrete problem (P^h) is defined as the problem of finding the Pareto optimal set $\mathcal{E}(\mathrm{cl}(\widetilde{\mathbf{F}}^h(\mathcal{T}_0^h(v_0))), \mathbf{R}_+^2)$, where $\widetilde{\mathbf{F}}^h(\cdot):\ \{v_i\} = \{v_i,\ i = 1, \ldots, N_T\} \in \mathcal{T}_0^h(v_0) \to \mathbf{R}^2$ is the vector-valued objective function defined by

$$\widetilde{\mathbf{F}}^h(\{v_i\}) = \left(h \sum_{i=0}^{N_T - 1} \widetilde{f}_1(i, v_i, \dot{v}_i), h \sum_{i=0}^{N_T - 1} \widetilde{f}_2(i, v_i) \right)$$

with initial redundancy parameter v_0.

B. Multiobjective Dynamic Programming Equation

Let $J_k^h(\cdot)$ be the set-valued return function that associates, with each $v_k \in \mathcal{A}_k^h$, the Pareto optimal set $\mathcal{E}(\mathrm{cl}(\widetilde{\mathbf{F}}_h(\mathcal{T}_k^h(v_k))), \mathbf{R}_+^2)$

$$J_k^h(v_k) = \mathcal{E}(\mathrm{cl}(\widetilde{\mathbf{F}}^h(\mathcal{T}_k^h(v_k))), \mathbf{R}_+^2). \tag{11}$$

Note that setting $k = 0$ in (11) yields exactly (P^h). The set-valued return function $J_k^h(\cdot)$ can be shown to satisfy the multiobjective dynamic programming equation

$$J_k^h(v_k) = \mathcal{E}(\mathrm{cl}(\{h(\widetilde{f}_1(k, v_k, \dot{v}_k), \widetilde{f}_2(k, v_k)) + J_{k+1}^h(v_k + h\dot{v}_k),\ \dot{v}_k \in \mathcal{B}_k^h(v_k)\}), \mathbf{R}_+^2) \tag{12}$$

with terminal data condition

$$J_{N_T}^h(v_{N_T}) = \{(0, 0)\}. \tag{13}$$

C. Discretization in the Redundancy Parameter

In the second approximation step, the set-valued return function $J_0^h(\cdot)$ is approximated by performing a discretization in the redundancy parameter. Let N_X be the number of discretization steps, $d = (v_{\max} - v_{\min})/N_X$ be the discretization step, and $\mathcal{A}_k^{h,d}$ be the set resulting from the discretization of $\mathcal{A}_k^h$ in the redundancy parameter. The approximate set-valued return function $J_k^{h,d}(\cdot)$ is defined as the solution to the multiobjective dynamic programming equation (12) with terminal data condition (13), where the redundancy parameter is restricted to take on values only from $\mathcal{A}_k^{h,d}$, i.e., $\forall v_k \in \mathcal{A}_k^{h,d}$

$$J_k^{h,d}(v_k) = \mathcal{E}(\{h(\widetilde{f}_1(k, v_k, \dot{v}_k), \widetilde{f}_2(k, v_k)) + J_{k+1}^{h,d}(v_k + h\dot{v}_k),\ \dot{v}_k \in \mathcal{B}_k^{h,d}(v_k)\}, \mathbf{R}_+^2) \tag{14}$$

with terminal data condition

$$\forall v_{N_T} \in \mathcal{A}_{N_T}^{h,d},\ J_{N_T}^{h,d}(v_{N_T}) = \{(0, 0)\}. \tag{15}$$

The set $\mathcal{B}_k^{h,d}(v_k) \subset \mathcal{B}_k^h(v_k)$ is finite. The values that can be taken by $\dot{v}_k \in \mathcal{B}_k^{h,d}(v_k)$ are such that $\dot{v}_k = (v_{k+1} - v_k)/h$, where $v_k \in \mathcal{A}_k^{h,d}$ and $v_{k+1} \in \mathcal{A}_{k+1}^{h,d}$. Note that the closure is no longer in the approximate multiobjective dynamic programming equation (14), as all the sets involved are finite.

Solving the approximate multiobjective dynamic programming equation (14) with terminal data condition (15) is straightforward. Indeed, let $v_k \in \mathcal{A}_k^{h,d}$, and assume that the approximate set-valued return function $J_{k+1}^{h,d}(\cdot)$ is known. Knowing $\dot{v}_k \in \mathcal{B}_k^{h,d}(v_k)$, the term $h(\widetilde{f}_1(k, v_k, \dot{v}_k), \widetilde{f}_2(k, v_k))$ can be calculated, from which the set $\{h(\widetilde{f}_1(k, v_k, \dot{v}_k), \widetilde{f}_2(k, v_k)) + J_{k+1}^{h,d}(v_k + h\dot{v}_k)\}$ can be determined. The set $\mathcal{B}_k^{h,d}(v_k)$ being finite, the set $\{h(\widetilde{f}_1(k, v_k, \dot{v}_k), \widetilde{f}_2(k, v_k)) + J_{k+1}^{h,d}(v_k + h\dot{v}_k),\ \dot{v}_k \in \mathcal{B}_k^{h,d}(v_k)\}$ is also finite. Therefore, to determine the Pareto optimal set of $\{h(\widetilde{f}_1(k, v_k, \dot{v}_k), \widetilde{f}_2(k, v_k)) + J_{k+1}^{h,d}(v_k + h\dot{v}_k),\ \dot{v}_k \in \mathcal{B}_k^{h,d}(v_k)\}$, which is precisely $J_k^{h,d}(v_k)$, a finite number of comparisons are needed. The repetition of this procedure for every $v_k \in \mathcal{A}_k^{h,d}$ yields the approximate set-valued return function

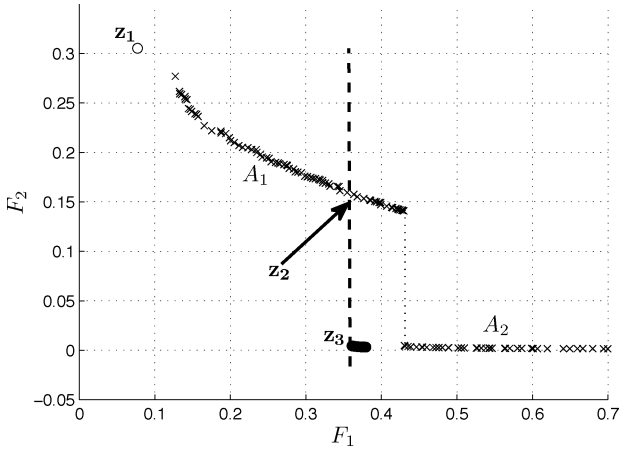

Fig. 4. Pareto objective vectors (○) obtained with weighting method, and $J_0^{h,d}(v_0)$ (×). $J_0^{h,d}(v_0)$ is the union of the sets A_1 and A_2.

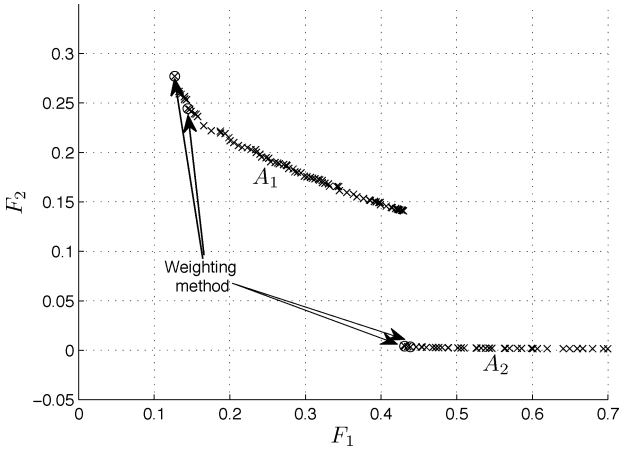

Fig. 5. Approximate Pareto objective vectors (○) obtained with the weighting method and $J_0^{h,d}(v_0)$ (×).

$J_k^{h,d}(\cdot)$. Therefore, starting from the terminal data condition (15), the approximate set-valued return function $J_0^{h,d}(\cdot)$ can be recursively obtained. Finally, $J_0^{h,d}(v_0)$ is the proposed approximation to the Pareto optimal set $\mathcal{E}(\mathrm{cl}(\mathbf{F}(\mathcal{T}(\mathbf{q_0}))), \mathbf{R}_+^2)$ of (P). Note that the corresponding optimal discrete joint trajectories $\{v_i,\ i = 0, \ldots, N_T\}$ are generated during this resolution process.

D. Results

First, consider how the approximate Pareto optimal set $J_0^{h,d}(v_0)$ can be practically obtained. Let the end-effector trajectory $\mathbf{p}(\cdot)$, and h and d be given. The discretization in time and in the redundancy parameter yields a grid $\{(k, v_{k,i}),\ k = 0, \ldots, N_T,\ i = 0, \ldots, N_X\}$. At each point $(k, v_{k,i})$ of this grid, the joint position $\mathbf{q}$ is calculated using the inverse kinematics equation (1) with $v_{k,i}$ and $\mathbf{p}(t_k)$. If $\mathbf{q}$ does not satisfy the joint mechanical limits (3) or the collision constraint (4), then the point is removed from the grid. The repeatition of this procedure yields the sets $\mathcal{A}_k^{h,d}$. For each remaining node $(k, v_{k,i})$ of the grid, the sets $\mathcal{B}_k^{h,d}(v_k)$ are determined by differentiating the inverse kinematics equation (1) and using the joint speed limits (5). The values of the function $\widetilde{f}_1$ and $\widetilde{f}_2$ are then calculated. From the earlier discussion, observe that it would be straightforward to include more constraints on (P). Including more constraints would have the effect of reducing the cardinality of the sets $\mathcal{A}_k^{h,d}$ and $\mathcal{B}_k^{h,d}(v_k)$, which would improve the computational efficiency of the resolution of the approximate multiobjective dynamic programming equation (14), as discussed in Section IV-C.

For the numerical experiments in the following, we use the same positions $\mathbf{p_0}$ and $\mathbf{p_f}$ and, accordingly, the same store trajectory as in Section III. A good compromise between accuracy and computational efficiency was found to be obtained with $h = 0.5$ s and $d = 1^\circ$, yielding a grid size of $(N_T, N_X) = (13, 290)$. The computation time was 24 628 s. As briefly discussed in Section V, for application purposes, this computation time can be greatly reduced.

In Fig. 4, the Pareto objective vectors obtained with the weighting method in Section III are plotted together with the approximate Pareto optimal set $J_0^{h,d}(v_0)$, whose cardinality is 138. In the following discussion, it is important to keep in mind that first, the elements of $J_0^{h,d}(v_0)$ are only approximate Pareto objective vectors, whereas the Pareto objective vectors obtained with the weighting method are "exact," and second, when solving (P) with the weighting method, the constraints (3)–(5) were not included.

From Fig. 4, two important observations can be made.

1) The set $J_0^{h,d}(v_0)$ is composed of two subsets A_1 and A_2. The first subset A_1 can be seen to approximate the *entire* subset of the Pareto optimal set between $\mathbf{z_1}$ and $\mathbf{z_2}$, where, as shown in Section III, the weighting method failed to generate Pareto objective vectors. The second subset A_2 is approximating the subset of the Pareto optimal set, as displayed in Fig. 2(b).
2) The approximate Pareto objective vectors are evenly distributed.

Note that the quality of the approximation degrades for A_2. This can be explained by the fact that first, as the joint speed norm is higher, the approximation of the redundancy parameter derivative with the forward Euler scheme is less accurate, and second, not all the minimizing joint trajectories obtained with the weighting method satisfy the joint mechanical limits (3) and the collision constraint (4) (whereas this is the case for A_1).

Another interesting experiment that can be made to compare the weighting method to the DDP approximation method is to solve $(\mathrm{P_s})$ using the grid obtained from the DDP approximation method. The approximate Pareto objective vectors obtained for the same set of weights, as in Section III, i.e, $w = 0, 1, \ldots, 30$, are plotted in Fig. 5, together with $J_0^{h,d}(v_0)$. Only four different approximate Pareto objective vectors result, which belong to $J_0^{h,d}(v_0)$. For the same reasons, as discussed in Section III, there is large subset of A_1 that the weighting method cannot generate.

Once (P) is solved and $J_0^{h,d}(v_0)$ is obtained, the final joint trajectory to be executed by the manipulator is chosen based on the operating conditions, which, as mentioned earlier, can be significantly different. At larger Mach numbers, having low joint speeds is more critical to make more joint torques available to sustain higher aerodynamic loads and allow for a shorter stopping time in case of emergencies. Therefore, an approximate Pareto objective vector with a lower value for $\mathbf{F}_1$ will be chosen. At lower Mach numbers, having low joint speeds is less critical; therefore, it is possible to choose an approximate Pareto objective vector with a lower value for $\mathbf{F}_2$ to increase the accuracy of the data collected. To limit the number of choices, only a subset of $J_0^{h,d}(v_0)$, obtained with *clustering*, can be considered [17].

V. Conclusion

For the resolution of trajectory planning problems with multiple criteria, it has been proposed to find the Pareto optimal set. It has also been shown that the DDP approximation method presented in [1]

was applicable to the trajectory planning problem arising from a CTS system. It has finally been illustrated qualitatively that this method provided a closer representation of the Pareto optimal set than the one obtained with the weighting method. Quantitative comparison using the Hausdorff distance could also be performed [17] but was beyond the scope of this paper.

In general, the main challenge faced for the practical applicability of the DDP approximation method is to address the *curse of dimensionality*. For (P), a study is under way to use clustering [18] of the sets $J_k^{h,d}(v_k)$ in the approximate dynamic programming equation (14). With clustering, the complexity of the resolution of (14) can be shown to reduce to polynomial in the grid size without an excessive degradation of the quality of the approximation of the Pareto optimal set. The computation time for the experiment presented in Section IV-D could be reduced from 24 628 to 159 s [17]. Finally, terminal costs can simply be accommodated within the DDP approximation method [17].

REFERENCES

[1] A. Guigue, M. Ahmadi, M. J. D. Hayes, and R. G. Langlois, "A discrete dynamic programming approximation to the multiobjective deterministic finite horizon optimal control problem," *SIAM J. Control Optim.*, vol. 48, pp. 2581–2599, 2009.

[2] B. Siciliano, "Kinematic control of redundant robot manipulators: A tutorial," *J. Intell. Robot. Syst.*, vol. 3, no. 3, pp. 201–212, 1990.

[3] M. Galicki, "The planning of robotic optimal motions in the presence of obstacles," *Int. J. Robot. Res.*, vol. 17, no. 3, pp. 248–259, 1998.

[4] D. P. Martin, J. Baillieul, and J. M. Hollerbach, "Resolution of kinematic redundancy using optimization techniques," *IEEE Trans. Robot. Autom.*, vol. 5, no. 4, pp. 529–533, Aug. 1989.

[5] Y. Shen and K. Huper, "Optimal trajectory planning of manipulators subject to motion constraints," in *Proc. 12th Int. Conf. Adv. Robot.*, Seattle, WA, Jul. 2005, pp. 9–16.

[6] A. J. Cahill, M. R. James, J. C. Kieffer, and D. Williamson, "Remarks on the application of dynamic programming to the optimal path timing of robot manipulators," *Int. J. Robust Nonlinear Control*, vol. 8, pp. 463–482, 1998.

[7] K. D. Malanowski, "Convergence of the lagrange-newton method for optimal control problems," *Int. J. Appl. Math. Comput. Sci.*, vol. 14, no. 4, pp. 531–540, 2004.

[8] F.-T. Cheng, M.-S. Shih, F.-C. Kung, and Y.-Y. Sun, "The improved parallel scheme for multiple-goal priority considerations of redundant manipulators," in *Proc. IEEE Int. Conf. Robot. Autom.*, Albuquerque, NM, Apr. 1997, pp. 2409–2414.

[9] C. Kapoor, M. Cetin, and D. Tesar, "Performance based redundancy resolution with multiple criteria," presented at the Amer. Assoc. Mech. Eng. Design Eng. Tech. Conf., Atlanta, GA, Sep. 1998.

[10] K. G. Shin and N. D. McKay, "A dynamic programming approach to trajectory planning of robotic manipulators," *IEEE Trans. Autom. Control*, vol. AC-31, no. 6, pp. 491–500, Jun. 1986.

[11] S. Singh and M. C. Leu, "Optimal trajectory generation for robotic manipulators using dynamic programming," *J. Dyn. Syst., Meas. Control*, vol. 109, pp. 88–96, Jun. 1987.

[12] A. Guigue, M. Ahmadi, M. J. D. Hayes, R. G. Langlois, and F. C. Tang, "A dynamic programming approach to redundancy resolution with multiple criteria," in *Proc. IEEE Int. Conf. Robot. Autom.*, Rome, Italy, Apr. 2007, pp. 1375–1380.

[13] M. Ahmadi, A. Guigue, M. Gibeault, and F. C. Tang, "Mechatronic design of redundant robotic systems for captive trajectory trajectory simulation applications," in *Proc. 10th IASTED Int. Conf. Control Appl.*, Quebec City, Canada, May 2008, pp. 24–29.

[14] M. Ahmadi, M. Jaber, and F. C. Tang, "Real-time multi-body collision detection for a captive trajectory simulation system," in *Proc. IEEE Int. Conf. Mechatron. Robot.*, Aachen, Germany, Sep. 2004, pp. 720–725.

[15] Y. Sawaragi, H. Nakayama, and T. Tanino, *Theory of Multiobjective Optimization*. Orlando, FL: Academic, 1985.

[16] I. Das and J. E. Dennis, "A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems," *Struct. Optim.*, vol. 14, pp. 63–69, Sep. 1997.

[17] A. Guigue, "Multiobjective kinematic trajectory planning: An application to the captive trajectory simulation (CTS) system," Ph.D. dissertation, Dept. Mech. Aero. Eng., Carleton Univ., Ottawa, ON, Canada, 2010.

[18] D. S. Hochbaum, *Approximation Algorithms for NP-Hard Problems*. Boston, MA: PWS, 1997.

Decentralized Navigation of Groups of Wheeled Mobile Robots With Limited Communication

Andrey V. Savkin
and Hamid Teimoori

***Abstract*—In this paper, we consider a group of wheeled mobile robots, where each robot has very limited information on other robots in the group. We propose a simple bio-inspired decentralized navigation law, which guarantees that all robots will eventually move in the same direction and with the same speed.**

***Index Terms*—Bio-inspired robot control, decentralized control, flocking motion, multiagent coordination, multirobot networks, robot navigation, wheeled robots.**

I. INTRODUCTION

The study of decentralized control laws for groups of mobile autonomous robots has emerged as a challenging new research area in recent years (see, e.g., [2], [4]–[7], [11], [16], [22], and references therein). Broadly speaking, this problem falls within the domain of decentralized control, but the unique aspect of it is that groups of mobile robots are dynamically decoupled, meaning that the motion of one robot does not directly affect that of the others. Researchers in this new emerging area are finding much inspiration from biology, where the problem of animal aggregation is central in both ecological and evolutionary theory. Animal aggregations, such as schools of fish, flocks of birds, groups of bees, or swarms of social bacteria, are believed to use simple, local motion coordination rules at the individual level that result in remarkable and complex intelligent behavior at the group level (see, e.g., [3], [12], [18]). Such intelligent behavior is expected from very large scale robotic systems. The "very large scale robotic system" was introduced in [14] for a system consisting of autonomous robots numbering from hundreds to tens of thousands or even more. Because of decreasing costs of robots, interest in very-large-scale robotic systems is growing rapidly. The possible applications include underwater exploration, military surveillance, and many others. In such systems, robots should exhibit some forms of cooperative behavior.

In 1995, Vicsek *et al.* proposed a simple, but interesting discrete-time model of a system consisting of several autonomous agents, e.g., particles, moving in the plane [21]. Each agent's motion is updated using a local rule based on its own state and the state of its "neighbors." This model can be viewed as a special case of a computer model

Manuscript received April 5, 2009; revised June 17, 2010; accepted September 21, 2010. Date of current version December 8, 2010. This paper was recommended for publication by Associate Editor I-M. Chen and Editor L. Parker upon evaluation of the reviewers' comments. This work was supported by the Australian Research Council.

The authors are with the School of Electrical Engineering and Telecommunications, The University of New South Wales, Sydney 2052, Australia (e-mail: a.savkin@unsw.edu.au; h.teimoori@unsw.edu.au).

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/TRO.2010.2081430