Comparison of Alternate Methods for Distributed Motion Planning of Robot Collectives within a Potential Field Framework*

Leng-Feng Lee, Rajankumar Bhatt and Venkat Krovi
Mechanical and Aerospace Engineering, State University of New York at Buffalo, 318 Jarvis Hall, Buffalo, NY 14221, USA
{llee3, rmbhatt, vkrovi}@eng.buffalo.edu

***Abstract* – In this paper, we evaluate the performance of two candidate formulations for distributed motion planning of robot collectives within an Artificial Potential Field (APF) framework. We exploit the parallel between the formulation of motion planning for group of robots coupled by constraints and the forward dynamics simulation of constrained multibody systems to develop the candidate approaches. We compare and contrast these approaches on the basis of ease of formulation, distribution of computation and overall computational accuracy. Traditionally penalty formulations have enjoyed a prominent position in motion planning of robot collectives due to their ease of formulation, decentralization and scalability. However, the instabilities introduced in the form of "formulation stiffness" at the algorithm development stage have the potential to hinder the subsequent control. Representative results from the distributed motion planning for a group of 3 point-mass robots moving in formation to a desired target location are used to highlight the differences.**

Index Terms - Artificial potential field, cooperative robot collectives, distributed computing, mobile robots, motion planning.

I. INTRODUCTION

In the past decade, ongoing revolutions in computing effectiveness and miniaturization of processors/sensors/ actuators have facilitated the transition from an individual mobile robot to networked distributed teams of mobile robots. The development of effective motion-planning methods for such collectives is critical to realizing the full potential of the group in numerous applications from reconnaissance, foraging, herding to cooperative payload transport.

While considerable literature exists for motion planning of individual mobile agents, the renewed challenge lies in creating motion plans for the entire team while incorporating notions such as cooperation. The "formation" paradigm has emerged as a convenient mechanism for abstraction and coordination with approaches ranging from leader-following [1, 2], virtual structures [3, 4] and virtual leaders [5, 6]. The group control problem now reduces to a well-known single-agent control problem from which the other agents derive their control laws but requires communication of some coordination information. Early implementations involved the kinematic specification of the followers' motion-plans as a "prescribed motions" relative to a team-leader without the ability to affect the dynamics of the leader. Subsequent approaches have incorporated some form of "formation-feedback" from the members to the overall group using natural or artificially introduced dynamics within the constraints. The formation paradigm has evolved to allow prescription of parameterized formation maneuvers [6, 7] and group feedback [6-9]. From these seemingly disparate approaches, a dynamic system-theoretic perspective has emerged for examining the decentralized multi-agent "behavioral control" in the context of "formations" [5-11]. "Behavioral" control laws, derived implicitly as gradients of limited-range artificial potentials, can be implemented in a decentralized manner while permitting a Lyapunov-based analysis of formation maintenance.

Various variants of the Artificial Potential Field (APF) framework have been leveraged in implementing such behavioral motion-planning/control of robot collectives due to their seeming ease of formulation, decentralization and scalability. However, we note that while stability guarantees (typically asymptotic) may be obtained, APF approaches are unable to guarantee strict formation maintenance. Such strict formation maintenance is critical in applications such as cooperative payload transport by collectives [12] or in distributed sensor deployment applications where the robots are to form some geometric pattern and maintain it while moving about in the world [7].

We note that the group of independent mobile robots moving together in formation and coupled together by constraint dynamics can alternatively be viewed as a constrained mechanical system. The computation of motion plans for such collectives in a potential field may also be viewed as simulating the forward dynamics of a constrained multi-body mechanical system. By doing so, we would like to link (and leverage) the extensive literature on formulation and implementation of computational simulation of multibody systems [13-17] to the problem of motion planning of robot collectives.

In this paper, we evaluate the formation maintenance performance of several formulations developed by analogy to the approaches used for constrained mechanical systems. These include: (i) a direct Lagrangian multipler elimination approach (to serve as the benchmark); (ii) a penalty-formulation approach which is the most popular implementation; and (iii) a constraint manifold projection approach. We note that the instabilities introduced in the form of the "formulation stiffness" at the algorithm

* This work is partially supported by Research Foundation of State University of New York and National Science Foundation CAREER Award (IIS-0347653).

development stage have the potential to hinder the subsequent control and requires a careful quantitative examination [18]. Hence, we compare and contrast the various approaches on the basis of modular formulation, distributed computation and relative computational efficiency and accuracy. These aspects will be studied in the context of the motion-planning of a group of 3 point-mass mobile robots which are constrained together by means of rheonomous holonomic constraints.

The rest of the paper is organized as follows: Section II presents a brief discussion of various candidate formulations of forward dynamics approaches for constrained multibody systems. In Section III, the dynamic model of the system of 3 point-mass robots moving in plane is introduced and the candidate methods are evaluated from viewpoint of distribution of computation. Section IV discusses the standardized test arena and the performance evaluation metric which is then used in Section V to compare and contrast the methods. Section VI presents a brief discussion and concluding remarks.

II. Forward Dynamics Formulations for Constrained Mechanical Systems

In this section we briefly review some of the available alternative formulations for developing the forward dynamics simulations in constrained mechanical systems. At the outset, we note that suitable selection of a set of *configuration coordinates* is of particular importance due to its impact both on the ease of formulation and the subsequent computational efficiency. We make use of expanded sets of *dependent Cartesian coordinates* linked together by holonomic *constraints as being most* appropriate for modular composition and general-purpose analysis.

The overall dynamics can be formulated as a system of ODEs whose solutions are required to satisfy additional holonomic (algebraic) constraint equations as Lagrangian equations of the first kind [19]:

$$\dot{\mathbf{q}} = \mathbf{v} \quad (1)$$

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{v}} = \mathbf{f}(\mathbf{q},\mathbf{v},t,\mathbf{u}) - \mathbf{A}(\mathbf{q})^T\boldsymbol{\lambda} \quad (2)$$

$$\mathbf{C}(\mathbf{q},t) = \mathbf{0} \quad (3)$$

where

$\mathbf{q}$ is the n-dimensional vector of generalized coordinates.

$\mathbf{v}$ is the n-dimensional vector of generalized velocities.

$\mathbf{M}(\mathbf{q})$ is the $n \times n$ dimensional inertia matrix.

$\mathbf{f}(\mathbf{q},\mathbf{v},t,\mathbf{u})$ is the n-dimensional vector of external forces.

$\mathbf{u}$ is the vector of actuator forces/torques.

$\mathbf{C}(\mathbf{q},t)$ is a m-dimensional vector of holonomic constraints.

$\mathbf{A}(\mathbf{q}) = \partial\mathbf{C}/\partial\mathbf{q}$ is the $m \times n$ dimensional constraint Jacobian matrix.

$\boldsymbol{\lambda}$ is the m-dimensional vector of Lagrange multipliers.

The solution of resulting system of index-3 Differential Algebraic Equations (DAEs) by direct finite difference discretization is not possible using explicit discretization methods. We adopt a *converted ODE approach,* wherein all the algebraic position and velocity level constraints are differentiated and represented at the acceleration level to obtain an augmented index-1 DAE (in terms of both the unknown accelerations and the unknown multipliers). Differentiating the position constraints in (3), with respect to time, yields the velocity-level constraints:

$$\dot{\mathbf{C}} = \mathbf{A}(\mathbf{q})\mathbf{v} = 0 \quad (4)$$

Further differentiation with respect to time yields the acceleration level constraints as:

$$\ddot{\mathbf{C}} = \mathbf{A}(\mathbf{q})\dot{\mathbf{v}} + \dot{\mathbf{A}}(\mathbf{q})\mathbf{v} = 0 \quad (5)$$

Thus, (2) can then be written together with Equation (5) as an index-1 DAE as:

$$\begin{bmatrix} \mathbf{M} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix}_{(n+m)\times(n+m)} \begin{bmatrix} \dot{\mathbf{v}} \\ \boldsymbol{\lambda} \end{bmatrix}_{(n+m)\times 1} = \begin{bmatrix} \mathbf{f} \\ -\dot{\mathbf{A}}(\mathbf{q})\mathbf{v} \end{bmatrix}_{(n+m)\times 1} \quad (6)$$

In a typical forward dynamics simulation setting, the index-1 DAE systems resulting from the converted ODE approach are then converted into final system of first-order ODEs by: (a) direct Lagrange multiplier elimination; (b) penalty-formulation; or (c) constraint manifold projection.

A. Direct Lagrange Multiplier Elimination

In this approach, a simultaneous solution of the augmented linear system of equations in (6) is obtained at each time step. While an explicit inversion of the augmented system may be avoided by adopting a Gaussian elimination method, the overall approach may still be denoted as:

$$\begin{bmatrix} \dot{\mathbf{v}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{M} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{f} \\ \varphi \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1(\mathbf{q},\mathbf{v}) \\ \mathbf{f}_2(\mathbf{q},\mathbf{v}) \end{bmatrix} \quad (7)$$

Thus the overall system may now be written as a system of first order ODEs as:

$$\dot{\mathbf{x}}_{2n\times 1} = \begin{bmatrix} \dot{\mathbf{q}}_{n\times 1} \\ \ddot{\mathbf{q}}_{n\times 1} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{f}_1(\mathbf{q},\mathbf{v}) \end{bmatrix} \quad (8)$$

which may then be integrated using standard codes. The main advantage is conceptual simplicity and simultaneous determination of the accelerations and the Lagrange multipliers by solving a *linear* system of equations. However, this is a centralized approach and does not scale up very well.

B. Penalty-Formulation

In penalty-based approaches the holonomic constraints are relaxed and replaced by linear/non-linear virtual springs and dampers, thereby incorporating the constraint equations as a dynamical system penalized by a large factor. The Lagrange multipliers are *approximated* using a force-law (based on the extent of the constraint violation and an assumed spring stiffness) and eliminated from the list of $n+m$ unknowns leaving behind a system of $2n$ first order ODEs. While the sole initial drawback may appear to be restricted to the numerical ill-conditioning due to selection of large penalty factors, it is important to note that penalty approaches only *approximate* the true constraint forces and can create unanticipated problems (as will be discussed later). The Lagrange multiplier $\boldsymbol{\lambda}$ in (2) is explicitly calculated as a restoring force provided by a virtual spring. This restoring force, proportional to the extent of constraint

violation, can be expressed as $\boldsymbol{\lambda}_i = \mathbf{K}_{\mathbf{P}_i}\mathbf{C}_i(\mathbf{q}) + \mathbf{K}_{\mathbf{D}_i}\dot{\mathbf{C}}_i(\mathbf{q})$ where $\mathbf{K}_{\mathbf{P}_i}$ is the spring constant, $\mathbf{K}_{\mathbf{D}_i}$ is the damping constant and $\mathbf{C}_i(\mathbf{q})$ is the constraint violation in the direction of the respective $\boldsymbol{\lambda}_i$. By substituting the value of $\boldsymbol{\lambda}$ in (2), the final ODE system can be written as:

$$\dot{\mathbf{x}}_{2n\times 1} = \begin{bmatrix} \dot{\mathbf{q}}_{n\times 1} \\ \ddot{\mathbf{q}}_{n\times 1} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{M}^{-1}(\mathbf{f}(\mathbf{q},\mathbf{v},t,\mathbf{u}) - \mathbf{A}(\mathbf{q})^T(\mathbf{K}_\mathbf{P}\mathbf{C}(\mathbf{q}) + \mathbf{K}_\mathbf{D}\dot{\mathbf{C}}(\mathbf{q})) \end{bmatrix} \quad (9)$$

where $\mathbf{K}_\mathbf{P} = diag[K_{\mathbf{P}_i}]$, $\mathbf{K}_\mathbf{D} = diag[K_{\mathbf{D}_i}]$ and $\mathbf{C}(\mathbf{q})$ is the vector of constraint violations.

C. Constraint Manifold Projection

This approach seeks to take the dynamical equations with constraint-reactions into the *tangent and cotangent* subspace. The rheonomic constraints, $\mathbf{C}(\mathbf{q},t) = \mathbf{0}$, can be written in differential form as:

$$\left[\frac{\partial \mathbf{C}(\mathbf{q},t)}{\partial \mathbf{q}}\right]\dot{\mathbf{q}} + \left[\frac{\partial \mathbf{C}(\mathbf{q},t)}{\partial t}\right] = 0 \Leftrightarrow \mathbf{A}(\mathbf{q})\dot{\mathbf{q}} = \mathbf{a}(\mathbf{q}) \quad (10)$$

where $\mathbf{A}(\mathbf{q})$ is the Jacobian of $\mathbf{C}(\mathbf{q})$. Let $\mathbf{S(q)}$ be a $n\times(n-m)$ dimensional full rank matrix whose column space is in the null space of $\mathbf{A(q)}$ i.e. $\mathbf{A(q)S(q) = 0}$. The orthogonal subspace is spanned by the so-called constraint vectors (forming the rows of the matrix $\mathbf{A(q)}$) while the tangent subspace *complements* this orthogonal subspace in the overall generalized velocity vector space. All *feasible* dependent velocities, $\dot{\mathbf{q}}$, of a constrained multibody system necessarily belong to this tangent space, appropriately called the *space of feasible motions.* This space is spanned by the columns of $\mathbf{S(q)}$) and is parameterized by an $n-m$ dimensional vector of independent velocities, $\boldsymbol{\upsilon}(t)$, yielding the expression for the feasible dependent velocities as:

$$\dot{\mathbf{q}} = \mathbf{v} = \mathbf{S}(\mathbf{q})\boldsymbol{\upsilon}(t) + \boldsymbol{\eta}(\mathbf{q}) \quad (11)$$

where $\boldsymbol{\eta}(\mathbf{q})$ is the particular solution of (12). Differentiating this further we get:

$$\dot{\mathbf{v}} = \mathbf{S}(\mathbf{q})\dot{\boldsymbol{\upsilon}}(t) + \dot{\mathbf{S}}(\mathbf{q})\boldsymbol{\upsilon}(t) + \dot{\boldsymbol{\eta}}(\mathbf{q}) = \mathbf{S}(\mathbf{q})\dot{\boldsymbol{\upsilon}}(t) + \boldsymbol{\gamma}(\mathbf{q},\boldsymbol{\upsilon}) \quad (12)$$

where $\boldsymbol{\gamma}(\mathbf{q},\boldsymbol{\upsilon}) = \dot{\mathbf{S}}(\mathbf{q})\boldsymbol{\upsilon}(t) + \dot{\boldsymbol{\eta}}(\mathbf{q})$ needs to be calculated numerically which has potential of introducing errors. In order to avoid this situation, we adopted the method in [20].

Such a projection process works out well in a Riemannian setting (where the notion of orthogonal complement subspaces exists). Special care needs to be exercised when treating configuration spaces such as $SE(2)$ or $SE(3)$. A family of projections exists depending on selection of dependent/independent velocities. However, once a projection is selected, the dynamic equations of motion can now be projected on to the instantaneous feasible motion directions, to obtain the so-called constraint-reaction-free equations of motion. Pre-multiplying both sides of (2) by $\mathbf{S^T}$ and noting that $\mathbf{S^TA^T = 0}$ we get:

$$\mathbf{S}^T\mathbf{M}(\mathbf{q})\dot{\mathbf{v}} = \mathbf{S}^T\mathbf{f}(\mathbf{q},\mathbf{v},t,\mathbf{u}) \quad (13)$$

By substituting $\dot{\mathbf{v}}$ from (12) into (13) and solving for $\dot{\boldsymbol{\upsilon}}$ we get:

$$\dot{\boldsymbol{\upsilon}} = -(\mathbf{S}^T\mathbf{M}\mathbf{S})^{-1}(\mathbf{S}^T\mathbf{M}\boldsymbol{\gamma} + \mathbf{S}^T\mathbf{f}(\mathbf{q},\mathbf{v},t,\mathbf{u})) \quad (14)$$

The resulting overall system of ODEs may be expressed in state-space form as:

$$\dot{\mathbf{x}}_{(2n-m)\times 1} = \begin{bmatrix} \dot{\mathbf{q}}_{n\times 1} \\ \dot{\boldsymbol{\upsilon}}_{(n-m)\times 1} \end{bmatrix} = \begin{bmatrix} \mathbf{S}\boldsymbol{\upsilon} + \boldsymbol{\eta} \\ -(\mathbf{S}^T\mathbf{M}\mathbf{S})^{-1}(\mathbf{S}^T\mathbf{M}\boldsymbol{\gamma} - \mathbf{S}^T\mathbf{f}(\mathbf{q},\mathbf{v},t,\mathbf{u})) \end{bmatrix} \quad (15)$$

The final solution may be obtained either by numerically integrating a system of *2n-m* first-order ODEs in the *n* dependent velocities and *n-m* independent accelerations.

D. Baumgarte Stabilization

The drawbacks of the converted ODE approach include: (i) the need to provide additional consistent initial conditions; and (ii) the mild instability of the differentiated constraints resulting in state-drift from the position-level constraint manifold. While the growth rate can be reduced by lowering the error tolerance and by using smaller step-sizes or greater numerical precision, this comes at the cost of longer and more expensive computations.

Baumgarte stabilization [21] involves the creation of an artificial first or second-order dynamical system which has the algebraic position-level constraint as its attractive equilibrium configuration. For example, when the holonomic constraints in $\mathbf{C}(\mathbf{q},t) = \mathbf{0}$ are approximated by a first order system of the form, we obtain:

$$\dot{\mathbf{C}}(\mathbf{q},t) + \sigma\mathbf{C}(\mathbf{q},t) = \mathbf{0}, \quad \sigma > 0 \quad (16)$$

where σ is the rate of convergence. The equilibrium condition for this first order system is the constraint manifold $\mathbf{C}(\mathbf{q},t) = \mathbf{0}$ and for any initial condition $\mathbf{q}(0)$, which may not satisfy the holonomic constraint equation $\mathbf{C}(\mathbf{q}(0)) = \mathbf{0}$, the above first order equation guarantees exponential convergence of $\mathbf{C}(\mathbf{q}(t))$ to zero as the time t progresses. The rate of convergence will be determined by σ, which can be chosen based on specific application. Equation (16) can be suitably modified as:

$$\left[\frac{\partial \mathbf{C}(\mathbf{q},t)}{\partial \mathbf{q}}\right]\dot{\mathbf{q}} = -\sigma\mathbf{C}(\mathbf{q},t) - \left[\frac{\partial \mathbf{C}(\mathbf{q},t)}{\partial t}\right] \Leftrightarrow \mathbf{A}(\mathbf{q})\dot{\mathbf{q}} = \bar{\mathbf{a}}(\mathbf{q}) \quad (17)$$

and the rest of solution process remains unchanged. While Baumgarte's technique is very popular in the engineering application community, principally due to the resulting augmented ODE formulation, the practical selection of the parameters of the stabilization system depends both on the discretization methods and step-size and is widely regarded as an open research problem [22].

III. Distributed Modeling of the 3 Mobile Robot Collective

We consider a collective formed by 3 robots, each with point-mass m_i operating in the horizontal plane with a configuration vector $\mathbf{q}_i = \begin{bmatrix} x_i & y_i \end{bmatrix}^T \in \Re^2$ w.r.t an inertial frame $\{F\}$. The governing EOM for each robot take the simple form $\mathbf{M}_i\ddot{\mathbf{q}}_i = \mathbf{u}$, where $\mathbf{M}_i = m_i I_{2\times 2} \quad \forall i = A,B,C$. The equations of the overall collective moving in formation can be written in an index-3 DAE form as:

$$\dot{\mathbf{q}} = \mathbf{v}$$
$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{V}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) = \mathbf{E}(\mathbf{q})\mathbf{u} - \mathbf{A}^T\boldsymbol{\lambda} \qquad (18)$$
$$\mathbf{C}(\mathbf{q}) = \mathbf{0}$$

where

$$\mathbf{M}(\mathbf{q}) = \begin{bmatrix} [\mathbf{M}_A]_{2\times 2} & 0 & 0 \\ 0 & [\mathbf{M}_B]_{2\times 2} & 0 \\ 0 & 0 & [\mathbf{M}_C]_{2\times 2} \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} \mathbf{q}_A \\ \mathbf{q}_B \\ \mathbf{q}_C \end{bmatrix},$$

$$\mathbf{u} = -k\left(\nabla_{\mathbf{q}} V\right)^T, \ \mathbf{E}(\mathbf{q}) = I_{6\times 6}, \mathbf{G}(\mathbf{q}) = \mathbf{0}, \mathbf{V}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{0}$$

We will consider the case where a *rigid formation is desired*. The $2n-3$ [23] constraint equations that maintain the rigidity are obtained from the requirement that each robot tries to maintain a desired distance with the others:

$$\mathbf{C}(\mathbf{q}) = \begin{bmatrix} (x_A - x_B)^2 + (y_A - y_B)^2 - c_{AB}^2 \\ (x_B - x_C)^2 + (y_B - y_C)^2 - c_{BC}^2 \\ (x_C - x_A)^2 + (y_C - y_A)^2 - c_{CA}^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad (19)$$

The centralized governing equations for the robot collective maintaining formation using artificial potentials were discussed in Section II.B. We now consider the possibility of distributing the motion-planning computations between the multiple agents. Further details are available in [24].

A. The Penalty Formulation (Method A)

Noting that the state vector $\mathbf{q} = [\mathbf{q}_A^T \quad \mathbf{q}_B^T \quad \mathbf{q}_C^T]^T$ has state variables belonging to each of the robots A, B and C, the distributed model may be obtained in state-space form as:

$$[\dot{\mathbf{x}}_A]_{4\times 1} = \begin{bmatrix} \dot{\mathbf{q}}_A \\ \ddot{\mathbf{q}}_A \end{bmatrix} = \begin{bmatrix} \mathbf{v}_A \\ \mathbf{M}_A^{-1}(\mathbf{E}_A\mathbf{u}_A - \mathbf{A}_A^T(\mathbf{K}_{\mathbf{P}_A}\mathbf{C}_A + \mathbf{K}_{\mathbf{D}_A}\dot{\mathbf{C}}_A)) \end{bmatrix}$$
$$[\dot{\mathbf{x}}_B]_{4\times 1} = \begin{bmatrix} \dot{\mathbf{q}}_B \\ \ddot{\mathbf{q}}_B \end{bmatrix} = \begin{bmatrix} \mathbf{v}_B \\ \mathbf{M}_B^{-1}(\mathbf{E}_B\mathbf{u}_B - \mathbf{A}_B^T(\mathbf{K}_{\mathbf{P}_B}\mathbf{C}_B + \mathbf{K}_{\mathbf{D}_B}\dot{\mathbf{C}}_B)) \end{bmatrix} \qquad (20)$$
$$[\dot{\mathbf{x}}_C]_{4\times 1} = \begin{bmatrix} \dot{\mathbf{q}}_C \\ \ddot{\mathbf{q}}_C \end{bmatrix} = \begin{bmatrix} \mathbf{v}_C \\ \mathbf{M}_C^{-1}(\mathbf{E}_C\mathbf{u}_C - \mathbf{A}_C^T(\mathbf{K}_{\mathbf{P}_C}\mathbf{C}_C + \mathbf{K}_{\mathbf{D}_C}\dot{\mathbf{C}}_C)) \end{bmatrix}$$

where $\mathbf{K}_{\mathbf{P}_i}$, $\mathbf{K}_{\mathbf{D}_i}$ are the compliance matrices and $\mathbf{C}_i$ represents the extent of the constraint violation as pertinent to robot i, $(i = A, B, C)$.

The three dynamic sub-systems, shown in (20), can be simulated in a distributed manner if at every time step: (i) either the information pertaining to $\mathbf{C}_i(\mathbf{q})$, the extent of the constraint violation, is made available explicitly or (ii) computed by exchanging state information between the robots. The sole coupling between the two sub-parts is due to the Lagrange multipliers, which are now explicitly calculated using the virtual spring. While this is shown for a "three robot system", the process generalizes easily for "n-robot" system.

B. Constraint Manifold Projection (Method B)

We examine this approach as an appropriate alternative to the penalty formulation where again our emphasis is on distribution of the motion planning computations to be performed by the individual robots. Noting that the state vector may be written as $\mathbf{q} = [\mathbf{q}_A^T \quad \mathbf{q}_B^T \quad \mathbf{q}_C^T]^T$, the projected dynamics equations may be partitioned as:

$$[\mathbf{S}_A^T \ \mathbf{S}_B^T \ \mathbf{S}_C^T]\begin{bmatrix} \mathbf{M}_A & 0 & 0 \\ 0 & \mathbf{M}_B & 0 \\ 0 & 0 & \mathbf{M}_C \end{bmatrix}\begin{bmatrix} \mathbf{S}_A \\ \mathbf{S}_B \\ \mathbf{S}_C \end{bmatrix}\dot{\boldsymbol{\upsilon}} + [\mathbf{S}_A^T \ \mathbf{S}_B^T \ \mathbf{S}_C^T]\begin{bmatrix} \mathbf{M}_A & 0 & 0 \\ 0 & \mathbf{M}_B & 0 \\ 0 & 0 & \mathbf{M}_C \end{bmatrix}\begin{bmatrix} \boldsymbol{\gamma}_A^T \\ \boldsymbol{\gamma}_B^T \\ \boldsymbol{\gamma}_C^T \end{bmatrix}$$
$$= [\mathbf{S}_A^T \ \mathbf{S}_B^T \ \mathbf{S}_C^T][\mathbf{I}_{6\times 6}]\begin{bmatrix} \mathbf{u}_A \\ \mathbf{u}_B \\ \mathbf{u}_C \end{bmatrix} \qquad (21)$$

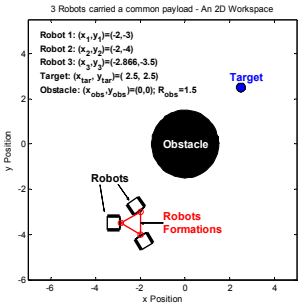
Thus, it is now possible to calculate the state vectors forming separately as:

$$\dot{\mathbf{x}}_A = \begin{bmatrix} \dot{\mathbf{q}}_A \\ \dot{\boldsymbol{\upsilon}}_A \end{bmatrix}_{(2+n-m)\times 1} = \begin{bmatrix} \mathbf{S}_A\boldsymbol{\upsilon} + \boldsymbol{\eta}_A \\ -(\mathbf{S}^T\mathbf{M}\mathbf{S})^{-1}(\mathbf{S}_A^T(\mathbf{M}_A\boldsymbol{\gamma}_A - \mathbf{E}_A\mathbf{u}_A)) \end{bmatrix}$$
$$\dot{\mathbf{x}}_B = \begin{bmatrix} \dot{\mathbf{q}}_B \\ \dot{\boldsymbol{\upsilon}}_B \end{bmatrix}_{(2+n-m)\times 1} = \begin{bmatrix} \mathbf{S}_B\boldsymbol{\upsilon} + \boldsymbol{\eta}_B \\ -(\mathbf{S}^T\mathbf{M}\mathbf{S})^{-1}(\mathbf{S}_B^T(\mathbf{M}_B\boldsymbol{\gamma}_B - \mathbf{E}_B\mathbf{u}_B)) \end{bmatrix} \qquad (22)$$
$$\dot{\mathbf{x}}_C = \begin{bmatrix} \dot{\mathbf{q}}_C \\ \dot{\boldsymbol{\upsilon}}_C \end{bmatrix}_{(2+n-m)\times 1} = \begin{bmatrix} \mathbf{S}_C\boldsymbol{\upsilon} + \boldsymbol{\eta}_C \\ -(\mathbf{S}^T\mathbf{M}\mathbf{S})^{-1}(\mathbf{S}_C^T(\mathbf{M}_C\boldsymbol{\gamma}_C - \mathbf{E}_C\mathbf{u}_C)) \end{bmatrix}$$

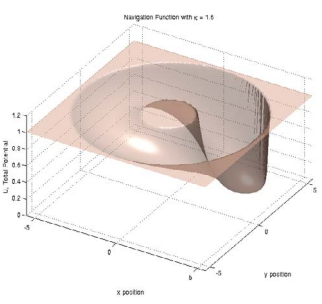
By examining (22), we note that the overall system can be evaluated in a distributed manner if states $\mathbf{q}_i$ and $\boldsymbol{\upsilon}_i$ $(i = A, B, C)$ are made available. Each independent sub-part can now be numerically integrated on a mobile robot thereby permitting independent operation. At each time-instant, the complete state of the system needs to be exchanged between the robots. The coupling between the various sub-parts is due to the existence of the $(\mathbf{S}^T\mathbf{M}\mathbf{S})^{-1}$. This matrix inverse needs to be computed on each and every processor (although we note that the explicit calculation of the inverse is typically avoided by using an optimal equation solver). Alternatively, state information from the slave processors could be collected by a central processor at each time instant, the $(\mathbf{S}^T\mathbf{M}\mathbf{S})^{-1}$ computed and the result subsequently broadcasted to all robots.

IV. The Standardized Test Arena

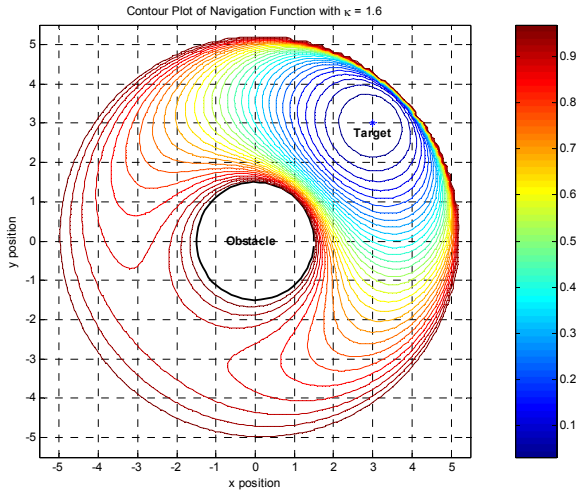
In order to compare the performance of various methods for motion planning of robot collectives within a potential-field framework, we developed a standardized test course. A graphical user interface (GUI) is used to locate the positions of the initial robot configurations, the obstacles and the target. As shown in Fig. 1(a). Then an APF is developed in the form of a navigation function [25] to ensure a unique minimum. This is shown as a 3D plot in Fig. 1(b) and as a contour plot in Fig. 1(c).



(a)



(b)



(c)

Fig. 1: Standard test course for performance measurement (a) Formation, with environment and target; (b) 3D plot; and (c) contour plot of the generated navigation function.

V. Performance Evaluation

In what follows, we treat the results from the forward dynamic simulations with a fixed time step as generating the motion plans for the robot collective. In Section III, we examined how both the penalty- and projection-based formulations for motion planning of a 3-robot collective could be distributed to run on separate processors (requiring only the exchange of state information at every time instant). This is implemented using MATLAB. The numerical parameters of the system are chosen as shown in Table 1.

Mass of robots	$M_1 = 1$	$M_2 = 1$	$M_3 = 1$
Initial x position	$X_1 = -2$	$X_2 = -2$	$X_3 = -2.866$
Initial y position	$Y_1 = -3$	$Y_2 = -4$	$Y_3 = -3.5$
Initial x direction velocity	0	0	0
Initial y direction velocity	0	0	0

Table 1: Relevant numerical parameters for the three robots system

We can then study the performance of the various formulations in the context of accumulated individual constraint errors as well as the overall formation error for a fixed time step and additionally study the effects of varying the time-step. The formation error is computed as $\|C(\mathbf{q})\|$ and corresponds to the structural error used by [8] and [11]. A number of simulations with different values for fixed time-steps (ranging from 1e-2 to 1e-4) were performed in [24]. However, only the resulting formation errors from running the two methods for a fixed step size of 1e-3 seconds are shown in Fig. 2.

Each method has independent parameters that could potentially affect the performance of the method – the virtual spring/damper parameters (K_{Pi}, K_{Di}) in the penalty formulation and the stabilization factor (σ) in the constraint manifold projection method. The effects of these parameters are studied in greater detail in [24].

Fig. 2(a) shows the results from the benchmark formulation using direct Lagrange elimination method. In Fig. 2(c), we note that the selection of the value of the independent parameter σ only plays a minor role since regardless of the selected value the constraint error remains near about 1e-6 which is in agreement with benchmark problem. In contrast, in Fig. 2(b) we see that for small values of the spring stiffness, considerable constraint error results which decreases as K_P is increased. While this constraint error reduces to the order of 1e-3 as the spring stiffness is increased to 500, formation maintenance never reaches the levels observed for the projection-based method.

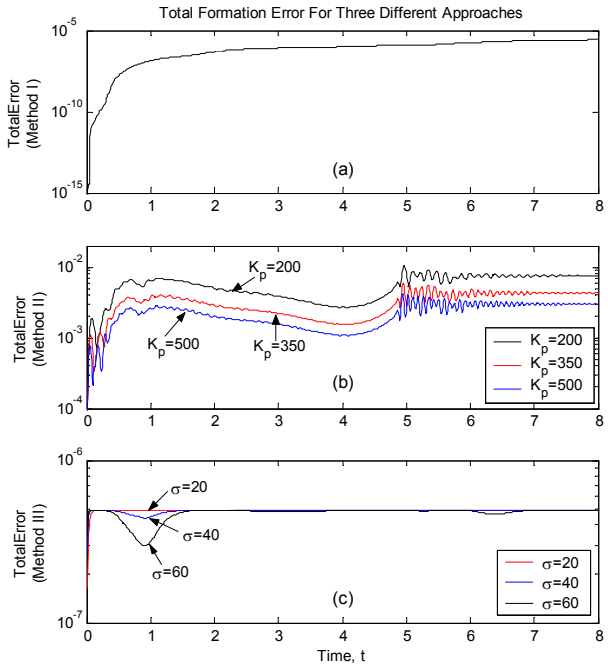


Fig. 2: Constraint error for numerical integration with fixed time-step (1e-3 secs) for (a) Direct Lagrange Elimination (Benchmark), (b) Penalty formulation and (c) Constraint manifold projection approaches.

Many have noted the various advantages of penalty formulation including: automated treatment of appearing/ disappearing constraints, robustness near singularities, in addition to the natural decoupling offered by the formulation. However, the Lagrange multipliers only form a part of the complete picture regarding the constraint forces. They represent the magnitude-type contribution while the other (and perhaps most important part) is the directional information that is embedded in the constraint Jacobian. The imperfect approximation of the Lagrange multipliers, coupled with the (artificial) relaxation of the constraints can over time lead to alternate configurations thereby indirectly affecting the directions of constraint vectors. Hence, not withstanding the small magnitudes of the constraint violations, the incorrect projection of the Lagrange multipliers would: (i) yield seemingly correct but non-physical results; (ii) and additionally act as a continuous source of disturbance. Schiehlen *et at.* [26] noted very similar results when a similar comparison was performed in the context of distributed dynamic simulation by coupling two or more minimal local subsystem with explicit (force-coupled) or implicit (DAE approach) enforcement of holonomic constraints.

VI. Discussion & Summary

In this paper, we examined aspects of the development and performance-evaluation of two alternate methods for distributed motion-planning for robot collectives within an artificial potential framework. These approaches arise by drawing the analogy to formulation methods in use for modular and distributed forward dynamics simulations of constrained mechanical systems. (Similar situations may also be encountered in other arenas where the governing equations take the form of sets of ODEs coupled together by algebraic constraints and solution of the combined system of DAEs needs to be found).

Our preliminary results (examined in the context of distributed motion planning of 3-robot collective discussed in the previous section) indicate that a global unified view of the evaluation of the computational complexity of the simulation is advisable. Specifically, at an algorithmic development level, the penalty-formulation within an APF framework provides a seemingly natural method for decoupling and distributing the computation, reduced computational complexity and an elegant Lyapunov-based setting to prove stability results. However, this is typically at the cost of formation maintenance – the projection-based approach does not distribute as well and is computationally more expensive per time-step. However, in the overall picture, this approach generates motion plans with smaller formation errors for a specified time-step and would have overall computational advantages over using the penalty formulation with a much smaller time-step.

Acknowledgements

We gratefully acknowledge the support from The Research Foundation of State University of New York and National Science Foundation CAREER Award (IIS-0347653) for this research effort.

References

[1] J. P. Desai, J. P. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 905-908, 2001.

[2] P. K. C. Wang, "Navigation strategies for multiple autonomous mobile robots moving in formation," *Journal of Robotic Systems*, vol. 8, no. 2, pp. 177-195, 1991.

[3] R. W. Beard, J. Lawton, and F. Y. Hadaegh, "A feedback architecture for formation control," *IEEE Transactions on Control Systems Technology*, vol. 9, no. 6, pp. 777-790, 2001.

[4] M. A. Lewis and K.-H. Tan, "High precision formation control of mobile robots using virtual structures," *Autonomous Robots*, vol. 4, pp. 387-403, 1997.

[5] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proc. IEEE Conference on Decision and Control*, Orlando, FL, pp. 2968-2973, 2001.

[6] P. Ogren, E. Fiorelli, and N. E. Leonard, "Formations with a mission: Stable coordination of vehicle group maneuvers," in *Proc. 15th International Symposium on Mathematical Theory of Networks and Systems*, 2002.

[7] B. Young, R. W. Beard, and J. M. Kelsey, "A control scheme for improving multi-vehicle formation maneuvers," in *Proc. American Control Conference*, Arlington, VA, pp. 704-709, 2001.

[8] M. Egerstedt and X. Hu., "Formation constrained multi-agent control," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 947-951, 2001.

[9] P. Ogren, M. Egerstedt, and X. Hu., "A control lyapunov function approach to multi-agent coordination," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 847-851, 2002.

[10] J. Lawton, B. Young, and R. W. Beard, "A decentralized approach to elementary formation maneuvers," in *Proc. 2000 IEEE International Conference on Robotics and Automation*, San Francisco, CA, 2000.

[11] R. Olfati-Saber and R. M. Murray, "Flocking with obstacle avoidance: Cooperation with limited communication in mobile networks," in *Proc. 42nd IEEE Conference on Decision and Control*, Maui, Hawaii, 2003.

[12] R. M. Bhatt, "Formation motion planning for payload transport by modular wheeled mobile manipulators," M.S. Thesis, Department of Mechanical & Aerospace Engineering, State University of New York at Buffalo, Buffalo, 2003

[13] U. Ascher and L. Petzold, *Computer methods for ordinary differential equations and differential-algebraic equations*: SIAM, 1998.

[14] J. Garcia de Jalon and E. Bayo, K*inematic and dynamic simulation of multibody systems: The real-time challenge.* New York: Springer-Verlag, 1994.

[15] E. J. Haug, C*omputer aided kinematics and dynamics of mechanical systems: Basic methods:* Allyn and Bacon Series in Engineering, Prentice Hall, 1989.

[16] W. Schiehlen, Mu*ltibody systems handbook.* Berlin: Springer-Verlag, 1990.

[17] A. A. Shabana, Dy*namics of multibody systems.* New York: Wiley, 1989.

[18] U. M. Ascher, D. K. Pai, and B. P. Cloutier, "Forward dynamics, elimination methods, and formulation stiffness in robot simulation," The *International Journal of Robotics Research,* vol. 16, no. 6, pp. 749-758, 1996.

[19] V. Arnold, Ma*thematical methods of classical mechanics,* 2 ed. New York: Springer-Verlag, 1989.

[20] X. Yun and N. Sarkar, "A unified formulation of robotic systems with holonomic and non holonomic constraints," IE*EE Transactions on Robotics and Automation,* vol. 14, no. 4, pp. 640-650, 1998.

[21] J. Baumgarte, "A new method of stabilization for holonomic constraints," AS*ME Journal of Applied Mechanics,* vol. 50, pp. 869-870, 1983.

[22] U. Ascher, H. Chin, L. Petzold, and S. Reich, "Stabilization of constrained mechanical systems with daes and invariant manifolds," M*echanical Structures & Machines,* vol. 23, pp. 135-158, 1995.

[23] R. Olfati-Saber and R. M. Murray, "Graph rigidity and distributed formation stabilization of multi-vehicle systems," in Pr*oc. Decision and Control, 2002, Proceedings of the 41st IEEE Conference on,* vol. 3, pp. 2965-2971 vol.3, 2002.

[24] L.-F. Lee, "Decentralized motion planning within an artificial potential framework (APF) for cooperative payload transport by multi-robot collectives.," Mechanical & Aerospace Engineering, State University of New York at Buffalo, NY, Buffalo, 2004

[25] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," Ro*botics and Automation, IEEE Transactions on,* vol. 8, no. 5, pp. 501-518, 1992.

[26] W. Schiehlen, R. A., and T. Schirle, "Force coupling versus differential algebraic description of constrained multibody systems," Mu*ltibody System Dynamics,* vol. 4, no. 4, pp. 317-340, 2000.