

Performance Evaluation of Potential Field based Distributed Motion Planning Methods for Robot Collectives

Leng-Feng Lee and Venkat N. Krovi
University at Buffalo
USA

1. Introduction

In the past decade, ongoing revolutions in computing effectiveness and miniaturization of processors/sensors/actuators have facilitated the transition of research focus 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 ranging 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 (Wang, 1991; Desai et al., 2001), virtual structures (Lewis and Tan, 1997; Beard et al., 2001) and virtual leaders (Leonard and Fiorelli, 2001; Ogren et al., 2002). 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 and group feedback (Egerstedt and Hu, 2001; Young et al., 2001; Ogren et al., 2002). 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” (Lawton et al., 2000; Egerstedt and Hu, 2001; Leonard and Fiorelli, 2001; Young et al., 2001; Ogren et al., 2002; Olfati-Saber and Murray, 2006). “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 (Bhatt et al., 2008) or in distributed sensor deployment applications where the robots are to form some geometric pattern and maintain it while moving about in the world (Young et al., 2001).

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 (Haug, 1989; Shabana, 1989; Schiehlen, 1990; García de Jalón and Bayo, 1994; Ascher and Petzold, 1998) to the problem of motion planning of mobile robot collectives.

In this chapter, *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 multiplier 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 development stage have the potential to hinder the subsequent control and requires a careful quantitative examination (Ascher et al., 1997). Hence, we compare and contrast the various approaches on the basis of modular formulation, distributed computation and relative computational efficiency and accuracy. These aspects are studied in the context of the motion-planning of a group of point-mass mobile robots which are constrained together by means of rheonomous holonomic constraints.

The rest of the paper is organized as follows: Section 2 presents a brief discussion of various candidate formulations of forward dynamics approaches for constrained multibody systems. In Section 3, the dynamic model of the system of point-mass robots moving in plane is introduced and the candidate methods are evaluated from viewpoint of distribution of computation. Section 4 discusses the standardized test arena and the performance evaluation metric which is then used in Section 5 to compare and contrast the methods. Section 6 presents a brief discussion and concluding remarks.

2. 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 (Arnold, 1989):

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

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

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

where

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

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

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

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

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

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

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

$\underline{\lambda}$ 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 Eq.(3), with respect to time, yields the velocity-level constraints:

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

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

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

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

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

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.

2.1 Direct Lagrange Multiplier Elimination

In this approach, a simultaneous solution of the augmented linear system of Eq. (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:

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

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

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

which may then be integrated using standard numerical solvers. The main advantage is its 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.

2.2 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 virtual spring type law (based on the extent of the constraint violation and 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). This individual multiplier values can be explicitly calculated as $\lambda_i = K_{P_i} C_i + K_{D_i} \dot{C}_i$ where K_{P_i} is the spring constant, K_{D_i} is the damping constant and $C_i(q)$ is the constraint violation in the direction of the respective λ_i . By substituting the value of $\tilde{\lambda}$ in Eq. (2), the final ODE system can be written as:

$$\dot{\tilde{x}}_{2n \times 1} = \begin{bmatrix} \dot{\tilde{q}}_{n \times 1} \\ \dot{\tilde{q}}_{n \times 1} \end{bmatrix} = \begin{bmatrix} \tilde{v} \\ M^{-1} \left(\tilde{f} - A^T (K_P \tilde{C} + K_D \dot{\tilde{C}}) \right) \end{bmatrix} \quad (9)$$

where $K_P = \text{diag}[K_{P_i}]$ and $K_D = \text{diag}[K_{D_i}]$.

2.3 Constraint Manifold Projection

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

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

Let $S(\underline{q})$ be an $n \times (n-m)$ dimensional full rank matrix whose column space is in the null space of A i.e. $AS = \underline{0}$. The orthogonal subspace is spanned by the so-called constraint vectors (forming the rows of the matrix A) while the tangent subspace *complements* this orthogonal subspace in the overall generalized velocity vector space. All *feasible* dependent velocities, $\dot{\underline{q}}$, 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 S and is parameterized by an $(n-m)$ -dimensional vector of independent velocities, $\underline{\nu}(t)$, yielding the expression for the feasible dependent velocities as:

$$\dot{\underline{q}} = \underline{v} = S \underline{\nu}(t) + \underline{\eta}(\underline{q}) \tag{11}$$

where $\underline{\eta}(\underline{q})$ is the particular solution of (10). Differentiating this further we get:

$$\dot{\underline{v}} = S \dot{\underline{\nu}} + \dot{S} \underline{\nu} + \dot{\underline{\eta}} = S \dot{\underline{\nu}} + \underline{\gamma}(q, \underline{\nu}) \tag{12}$$

where $\underline{\gamma}(q, \underline{\nu}) = \dot{S} \underline{\nu} + \dot{\underline{\eta}}$ needs to be calculated numerically which has potential of introducing errors. In order to avoid this situation, we adopted the method in (Yun and Sarkar, 1998).

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 Eq. (2) by S^T and noting that $S^T A^T = \underline{0}$ we get:

$$S^T M \dot{\underline{v}} = S^T \underline{f} \quad (13)$$

By substituting $\dot{\underline{v}}$ from Eq. (12) into Eq. (13) and solving for $\dot{\underline{v}}$ we get:

$$\dot{\underline{v}} = -(S^T MS)^{-1} (S^T M \underline{\gamma} - S^T \underline{f}) \quad (14)$$

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

$$\dot{\underline{x}}_{(2n-m) \times 1} = \begin{bmatrix} \dot{\underline{q}}_{n \times 1} \\ \dot{\underline{v}}_{(n-m) \times 1} \end{bmatrix} = \begin{bmatrix} S \underline{v} + \underline{\eta} \\ -(S^T MS)^{-1} (S^T M \underline{\gamma} - S^T \underline{f}) \end{bmatrix} \quad (15)$$

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.

2.4 Baumgarte Stabilization

The drawbacks of the Constraint Manifold Projection 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 (Baumgarte, 1983) 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 Eq. (3) are approximated by a first order system of the form, we obtain:

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

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

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

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 (Ascher et al., 1995).

3. Distributed Modeling of the N-Mobile Robot Collective

We consider a collective formed by N -robots, each with point-mass m_i operating in the horizontal plane with a configuration vector $q_i = [x_i, y_i]^T \in \mathbb{R}^2$ w.r.t an inertial frame $\{F\}$, as shown in Fig 1.

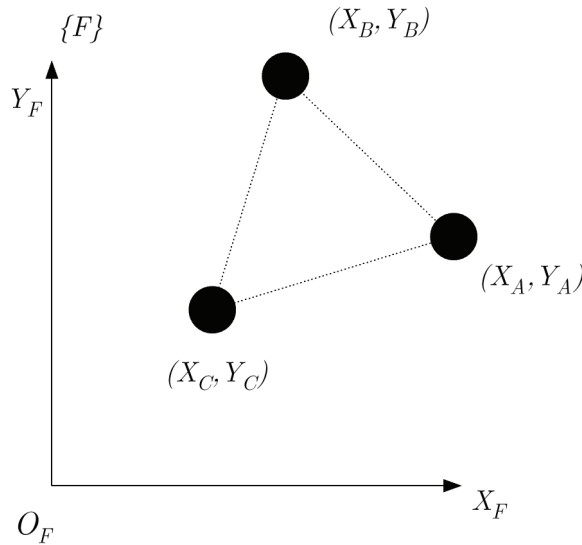


Figure 1. A robot collective form by $N = 3$ point mass robots operating in the horizontal plane w.r.t an inertial frame $\{F\}$

The governing EOM for each robot take the simple form $M_i \ddot{q}_i = \underline{u}$, where $M_i = m_i I_{2 \times 2} \quad \forall i = 1, \dots, N$. The equations of the overall collective moving in formation can be written in an index-3 DAE form as:

$$\begin{aligned}
 \dot{\underline{q}} &= \underline{v} \\
 M(\underline{q}) \ddot{\underline{q}} + \underline{V}(\underline{q}, \dot{\underline{q}}) + \underline{G}(\underline{q}) &= E(\underline{q}) \underline{u} - A^T \underline{\lambda} \\
 \underline{C}(\underline{q}) &= \underline{0}
 \end{aligned} \tag{18}$$

where

$$\underline{q} = \begin{bmatrix} q_1 \\ \vdots \\ q_N \end{bmatrix}; M(\underline{q}) = \begin{bmatrix} [M_1]_{2 \times 2} & \cdots & \underline{0} \\ \vdots & \ddots & \vdots \\ \underline{0} & \cdots & [M_N]_{2 \times 2} \end{bmatrix}; \underline{u} = -k(\nabla_{\underline{q}} V_p)^T; E(\underline{q}) = I_{2N \times 2N}; G(\underline{q}) = \underline{0};$$

$$V(\underline{q}, \dot{\underline{q}}) = \underline{0}$$

We will consider the case where a *rigid formation is desired*. The $2N - 3$ constraint equations (Olfati-Saber and Murray, 2002) that maintain the rigidity are obtained from the requirement that each robot tries to maintain a desired distance with the others:

$$C(\underline{q}) = \left[C_{ij}(\underline{q}) \right]_{(2N-3) \times 1} = \underline{0} \quad (19)$$

Equation (18) represents the centralized form of the governing equations using artificial potentials. We now consider the possibility of distributing the motion-planning computations between the multiple agents. Further details are available in Lee (2004).

3.1 The Penalty Formulation

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

$$[\dot{\underline{x}}_i]_{4 \times 1} = \begin{bmatrix} \dot{q}_i \\ \ddot{q}_i \end{bmatrix} = \begin{bmatrix} v_i \\ M_i^{-1} (E_i u_i - A_i^T (K_{P_i} C_i + K_{D_i} \dot{C}_i)) \end{bmatrix}; \quad \forall i = A, B, C \quad (20)$$

where K_{P_i} , K_{D_i} are the compliance and damping matrices and C_i represents the extent of the constraint violation as pertinent to robot i .

The three dynamic sub-systems, shown in Eq. (20), can be simulated in a distributed manner if at every time step: (i) either the information pertaining to $C_i(\underline{q})$, 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.

3.2 Constraint Manifold Projection

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 $\underline{q} = [q_A^T \quad q_B^T \quad q_C^T]^T$, the projected dynamics equations may be partitioned as:

$$\begin{aligned}
 & \begin{bmatrix} S_A^T & S_B^T & S_C^T \end{bmatrix} \begin{bmatrix} M_A & 0 & 0 \\ 0 & M_B & 0 \\ 0 & 0 & M_C \end{bmatrix} \begin{bmatrix} S_A^T \\ S_B^T \\ S_C^T \end{bmatrix} \dot{q} + \\
 & \begin{bmatrix} S_A^T & S_B^T & S_C^T \end{bmatrix} \begin{bmatrix} M_A & 0 & 0 \\ 0 & M_B & 0 \\ 0 & 0 & M_C \end{bmatrix} \begin{bmatrix} \gamma_A^T \\ \gamma_B^T \\ \gamma_C^T \end{bmatrix} = \begin{bmatrix} S_A^T & S_B^T & S_C^T \end{bmatrix} [I_{6 \times 6}] \begin{bmatrix} u_A^T \\ u_B^T \\ u_C^T \end{bmatrix}
 \end{aligned} \tag{21}$$

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

$$\dot{x}_i = \begin{bmatrix} \dot{q}_i \\ \dot{v}_i \end{bmatrix} = \begin{bmatrix} S_i v_i + \eta_i \\ -(S^T M S)^{-1} (S_i^T (M_i \gamma_i - E_i u_i)) \end{bmatrix}; \quad \forall i = A, B, C \tag{22}$$

By examining Eq. (21), we note that the overall system can be evaluated in a distributed manner if states q_i and v_i 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 $(S^T M S)^{-1}$.

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 $(S^T M S)^{-1}$ computed and the result subsequently broadcasted to all robots.

4. 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 Figure 2(a). Then an APF is developed in the form of a navigation function (Rimon and Koditschek, 1992) to ensure a unique minimum. This is shown as a 3D plot in Figure 2(b) and as a contour plot in Figure 2(c).

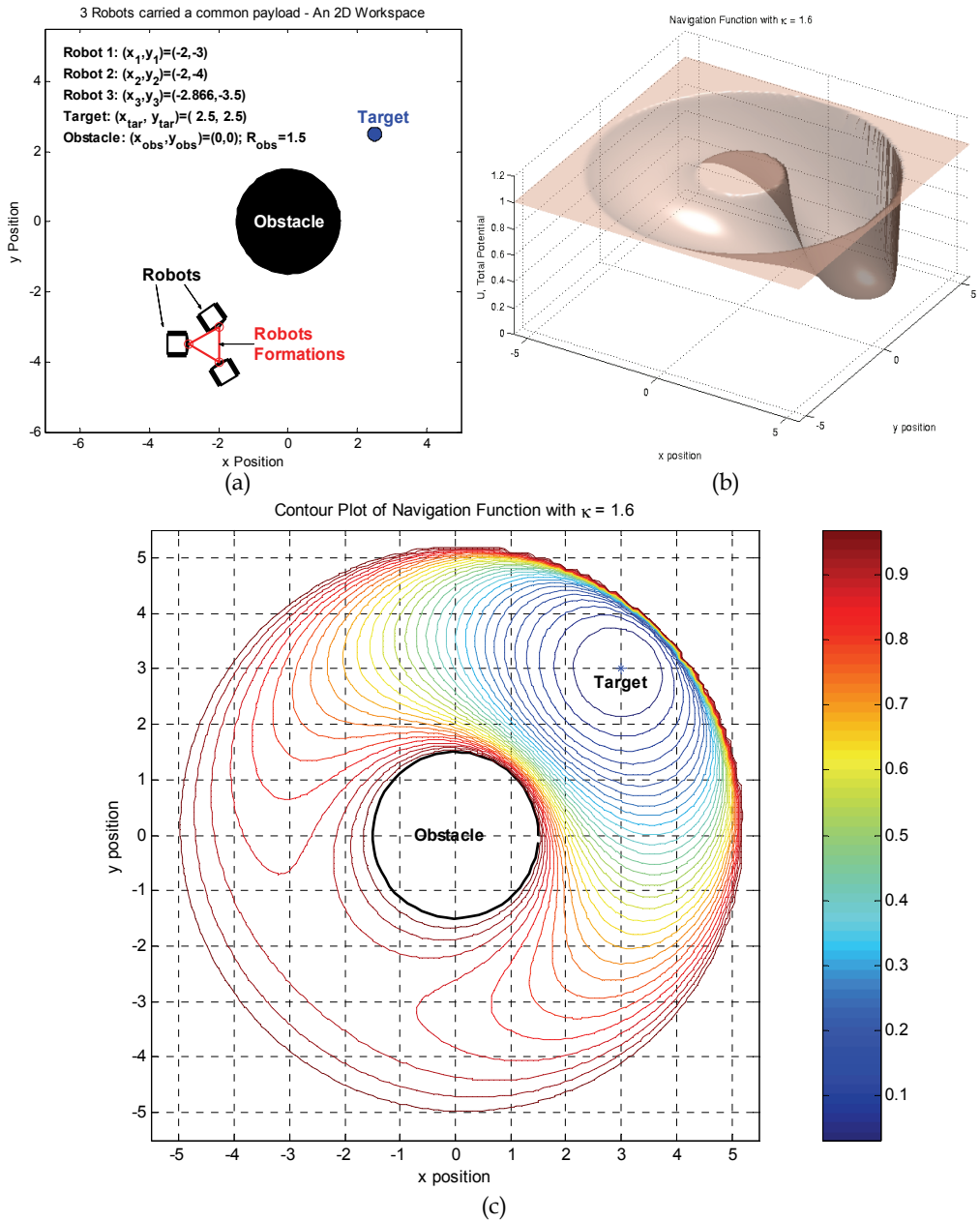


Figure 2. Standard test course for performance measurement (a) Formation, with environment and target; (b) 3-d plot; and (c) contour plot of the generated navigation function

5. Performance Evaluation

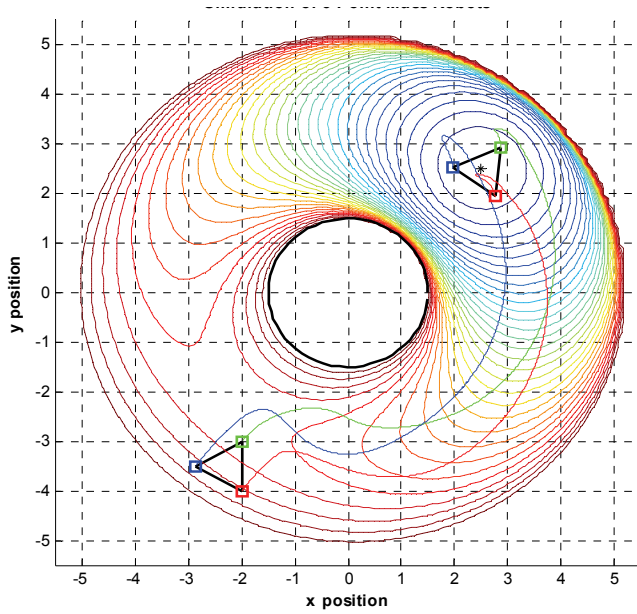
In what follows, we treat the results from the forward dynamic simulations with a fixed time step as generating the motion plans for the robot collective. In Section 3, we examined how both the penalty- and projection-based formulations for motion planning of N -robot collective (for $N = 3, 10$), that could be distributed to run on separate processors (requiring only the exchange of state information at every time instant). This is implemented using MATLAB. Figure 3(a) shows the corresponding simulation result of the 3-robot collective while Figure 3(b) shows the results from a larger 10-robot collective.

We 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

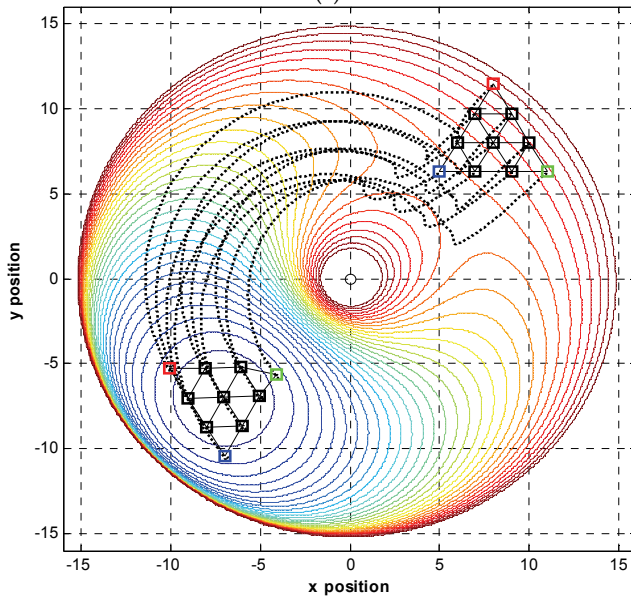
$\|\zeta(q)\|$ and corresponds to the structural error used by Egerstedt and Hu (2001) and Olfati-Saber and Murray (2006). A number of simulations with different values for fixed time-steps (ranging from $1e-2$ to $1e-4$) were performed by Lee (2004). However, only the resulting formation errors from running the two methods for a fixed step size of $1e-3$ seconds are shown in Figure 4. Each method has independent parameters that could potentially affect the performance of the method - the virtual spring/damper parameters (K_{P_i}, K_{D_i}) 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 Lee (2004).

Figure 4(a) shows the results from the benchmark formulation using direct Lagrange elimination method. In Figure 4(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 Figure 4(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.

We also performed a similar simulation studies (with fixed time-step of $1e-3$ seconds) with 10 point mass mobile robots in order to test the scalability of the adopted approaches, the corresponding simulation result is shown in Figure 3(b). The results shown in Figure 5 follow the same general trend observed in Figure 4. However, the distinction between the three methods (as manifested in the total formation error) is far more pronounced.



(a)



(b)

Figure 3. (a) The simulation result showing the three robots in a triangular formation move from their initial position to the target position while maintaining formation; and (b) The simulation result of 10 robots forming an interconnected triangular formation in a workspace with one obstacle

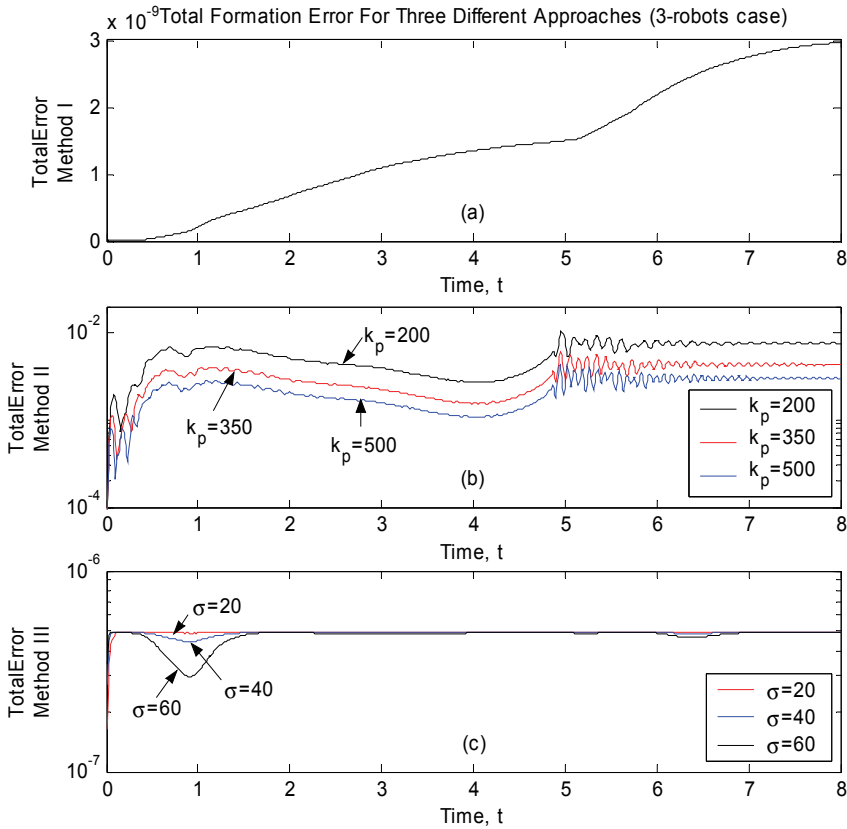


Figure 4. Constraint error for numerical integration with fixed time-step (1e-3 sec) for 3 point mass robots case, using (a) Method I: Direct Lagrange Elimination (Benchmark); (b) Method II: Penalty formulation; and (c) Method III: Constraint manifold projection approaches

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, notwithstanding 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 al.* (2000) 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.

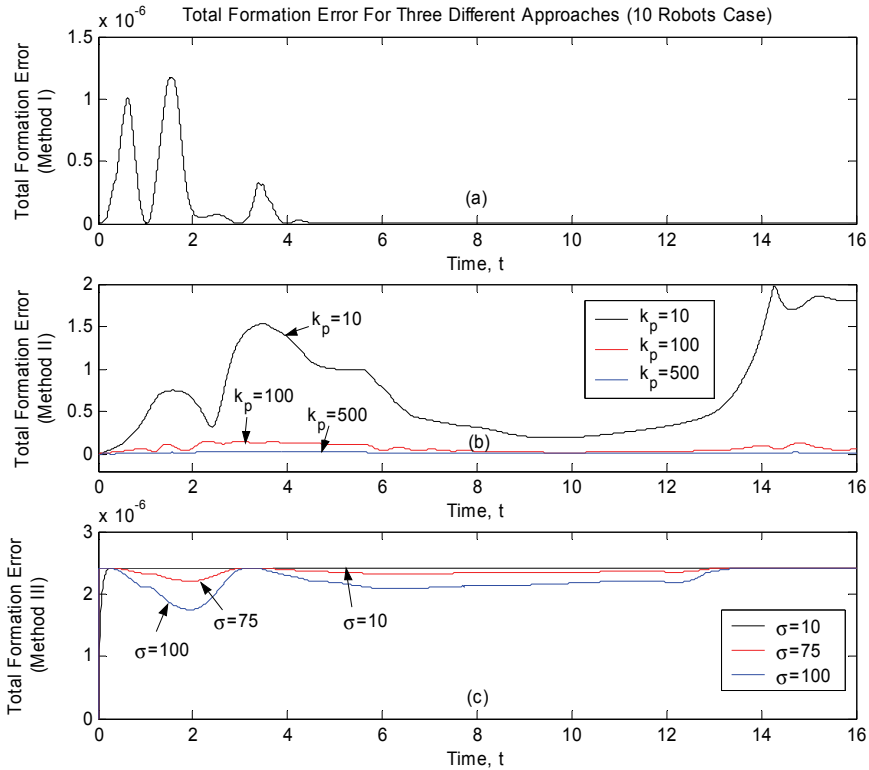


Figure 5. Constraint error for numerical integration with fixed time-step (1e-3 sec) for 10 point mass robots case, using (a) Method I: Direct Lagrange Elimination (Benchmark); (b) Method II: Penalty formulation; and (c) Method III: Constraint manifold projection approaches

6. Discussion & Summary

In this research, 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 and 10-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.

7. 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.

8. References

- Arnold, V. (1989). *Mathematical Methods of Classical Mechanics*, Springer-Verlag, New York.
- Ascher, U., Chin, H., Petzold, L. & Reich, S. (1995). Stabilization of Constrained Mechanical Systems with DAEs and Invariant Manifolds. *Mechanical Structures & Machines*, Vol.23, 135-158.
- Ascher, U., Pai, D. K. & Cloutier, B. (1997). Forward Dynamics, Elimination Methods, and Formulation Stiffness in Robot Simulation. *The International Journal of Robotics Research*, Vol.16, No.6, 749-758.
- Ascher, U. & Petzold, L. (1998). *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, SIAM: Society for Industrial and Applied Mathematics, Philadelphia, PA.
- Baumgarte, J. (1983). A New Method of Stabilization for Holonomic Constraints. *ASME Journal of Applied Mechanics*, Vol.50, 869-870.
- Beard, R. W., Lawton, J. & Hadaegh, F. Y. (2001). A Feedback Architecture for Formation Control. *IEEE Transactions on Control Systems Technology*, Vol.9, No.6, 777-790.
- Bhatt, R. M., Tang, C. P. & Krovci, V. N. (2008). Formation Optimization for A Fleet of Wheeled Mobile Robots - A Geometric Approach. *Robotics and Autonomous Systems*. doi:10.1016/j.robot.2006.12.012
- Desai, J. P., Ostrowski, J. P. & Kumar, V. (2001). Modeling and Control of Formations of Nonholonomic Mobile Robots. *IEEE Transactions on Robotics and Automation*, Vol.17, No.6, 905-908.
- Egerstedt, M. & Hu, X. (2001). Formation Constrained Multi-Agent Control. *IEEE Transactions on Robotics and Automation*, Vol.17, No.6, 947-951.
- García de Jalón, J. & Bayo, E. (1994). *Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time Challenge*, Springer-Verlag, New York.
- Haug, E. J. (1989). *Computer Aided Kinematics and Dynamics of Mechanical Systems: Basic Methods*, Allyn and Bacon Series in Engineering, Prentice Hall.
- Lawton, J., Beard, R. & Young, B. (2003). A Decentralized Approach To Formation Maneuvers. *IEEE Transactions on Robotics and Automation*, Vol.19, No.6, 933-941.

- Lee, L.-F. (2004). Decentralized Motion Planning within an Artificial Potential Framework (APF) For Cooperative Payload Transport by Multi-Robots Collectives. *M.S. Thesis. Mechanical and Aerospace Engineering, University at Buffalo, Buffalo, NY.*
- Leonard, N. E. & Fiorelli, E. (2001). Virtual Leaders, Artificial Potentials and Coordinated Control of Groups, *Proceedings of IEEE Conference on Decision and Control*, pp. 2968-2973, Orlando, FL.
- Lewis, M. A. & Tan, K.-H. (1997). High Precision Formation Control of Mobile Robots Using Virtual Structures. *Autonomous Robots*, Vol.4, 387-403.
- Ogren, P., Egerstedt, M. & Hu., X. (2002). A Control Lyapunov Function Approach to Multi-Agent Coordination. *IEEE Transactions on Robotics and Automation*, Vol.18, No.5, 847-851.
- Olfati-Saber, R. & Murray, R. M. (2002). Graph Rigidity and Distributed Formation Stabilization of Multi-Vehicle Systems, *Proceedings of 41st IEEE Conference on Decision and Control*, pp. 2965-2971, Las Vegas, Nevada.
- Olfati-Saber, R. (2006). Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory. *IEEE Transactions on Automatic Control*, Vol.51, No.3, 401-420.
- Rimon, E. & Koditschek, D. E. (1992). Exact Robot Navigation Using Artificial Potential Functions. *Robotics and Automation, IEEE Transactions on*, Vol.8, No.5, 501-518.
- Schiehlen, W. (1990). *Multibody Systems Handbook*, Springer-Verlag, Berlin.
- Schiehlen, W., A., R. & Schirle, T. (2000). Force Coupling Versus Differential Algebraic Description of Constrained Multibody Systems. *Multibody System Dynamics*, Vol.4, No.4, 317-340.
- Shabana, A. A. (1989). *Dynamics of Multibody Systems*, Wiley, New York.
- Wang, P. K. C. (1991). Navigation Strategies for Multiple Autonomous Mobile Robots Moving in Formation. *Journal of Robotic Systems*, Vol.8, No.2, 177-195.
- Young, B., Beard, R. W. & Kelsey, J. M. (2001). A Control Scheme for Improving Multi-Vehicle Formation Maneuvers, *Proceedings of American Control Conference*, pp. 704-709, Arlington, VA.
- Yun, X. & Sarkar, N. (1998). A Unified Formulation of Robotic Systems with Holonomic and Non Holonomic Constraints. *IEEE Transactions on Robotics and Automation*, Vol.14, No.4, 640-650.