The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Simultaneous Local Motion Planning and Control for Cooperative Redundant Arms

Gustavo Arechavaleta, Arturo Barrios, Gerardo Jarquín and Vicente Parra-Vega

Abstract— **We propose a scheme to deal simultaneously with local motion planning and dynamic control of redundant cooperative robots subject to holonomic, posture and loop-closure constraints. In contrast to previous contributions, an iterative method, that glue together the problem of kinematic motion planning with dynamic control, generates the sequence of feasible collision-free motions by combining in a single strategy the task-priority redundancy formalism with a joint model-free smooth force-motion control scheme based on a well-posed projection, which allows motions of the system only in the tangent subspace of the contact constrained manifold. Overall the new algorithm is able to compute fast and accurately local solutions involving severely constrained robotic motions. We have successfully verified in simulations the effectiveness of the proposed iterative method in the context of cooperative manipulation tasks.**

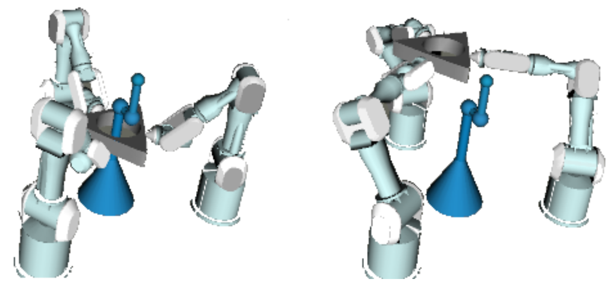

Fig. 1. A set of redundant robots have to transport an object from one place to another by applying the corresponding contact forces on the object surface. The residual redundancy is used to accomplish the goal while avoiding obstacles and joint limits. Left frame shows the initial configuration. A possible solution is depicted on the right.

I. Introduction

Planning and control manipulation tasks by a set of cooperative robots constitutes a classical and complex problem in robotics.

In motion planning, there exist a wide variety of particular instances of the problem that have been solved. Manipulation planning algorithms are mainly focused on the computation of *global* solutions. They need a precise geometric description of the problem and assume a discrete or continuous set of stable grasps and placements [1]. Although these methods are *global*, they do not consider the task-constrained dynamics arising when at least two bodies are in contact.

A. Problem statement

In this work we will be concerned with the problem of generating the sequence of dynamically feasible motions of a set of constrained redundant robots for object-carrying tasks. The inputs of the problem are the initial configuration of the whole system, the final configuration of the object and the desired magnitude of the contact force exerted by each robot. We assume that the end-effectors have already established a stable grasp and the arms are holding the object. The solution of the manipulation task should consider several contact force and kinematic closed-loop constraints while transporting the object from one place to another. In the case of redundant robots, additional bilateral and unilateral posture constraints can be satisfied (see Fig. 1).

B. Related work

There exist partial solutions obtained by some motion planners reported in the literature (e.g. [2]). Most of them are based on a two-stage strategy. At the first stage, a *global path* linking the initial and final configurations is constructed using a simplified kinematic model of the system. Then, the motion is generated for the whole system. The redundancy is exploited by applying either a task functional decomposition approach (e.g. [3]) or differential kinematics with priorities (e.g. [4]). These planners do not consider in a single strategy the interaction between motion and force at contact points while satisfying postural constraints through residual redundancy.

Most of the control approaches for constrained manipulation tasks use hybrid force-motion control schemes. This framework was originally introduced in [5]. The operational space approach incorporates explicitly the end-effector dynamics in the controller [6]. Based on a well-defined decoupling of the dynamics of constrained mechanical systems, several force-motion control schemes have been proposed [7], [8], [9]. Liu and Li [9] provided a detailed geometric interpretation of the projection method introduced in [10].

Among the previously mentioned motion planners and force-motion controllers, only the task-priority redundancy formalism based on the operational space has been used to handle force and motion constraints simultaneously while residual redundancy is used to satisfy additional constraints. This strategy has been integrated as a local method embedded in a *global* motion planner [11].

C. Contribution

The main contribution of this work is an iterative local method based on a joint space formulation to generate dynamical cooperative manipulation tasks that efficiently exploits the residual redundancy of the whole system while

Authors are with Robotics and Advance Manufacturing Group, Centro de Investigación y de Estudios Avanzados del IPN, Saltillo, Coah. México. {garechav,vicente.parra}@cinvestav.edu.mx

preserving the compliant behavior. This is possible by projecting the feasible motions of the system into the tangent space of the contact constrained manifold. The controller compensates explicitly joint constrained dynamics to enforce exponential convergence of position, velocity and contact force errors for tracking regime, without relying on inertial nor Coriolis matrices nor gravitational torques, contrary to the popular but computationally expensive hybrid computed torque controller [5], let alone operational space control strategies [6].

The remaining of this paper is structured as follows. In Section II we recall the classical prioritized inverse-kinematics and the general form of a constrained mechanical system. In Section III we describe the force-motion controller. Then, Section IV introduces the proposed iterative method. Section V illustrates the effectiveness of the method by a series of simulation experiments. Finally, we provide in Section VII some concluding remarks.

II. Modeling constrained redundant robots

There exist several constraints to be satisfied for transporting a bulky object by a set of redundant robotic arms. First, we introduce the set of bilateral and unilateral posture constraints. Then, we define the holonomic constraints that appear when the robots are in contact with the environment. Because we assume frictionless point contacts, the forces exerted by the arms and the object only occur in the normal direction. Note that closed kinematic loops are formed when at least two robotic arms are holding the same object.

A. Posture constraints

Let $q \in \mathbb{R}^n$ be the configuration of the robot in the n-dimensional joint space where n represents the number of degrees of freedom (d.o.f). The location of any body of the robot can be represented by $x(q)$ where $x \in \mathbb{R}^m$ is a coordinate vector and $m \leq n$. Thus, a typical task consists in finding q for a desired location x_d such that

$$e(q) = 0 \tag{1}$$

where $e(q) = x(q) - x_d$. These nonlinear equations are in terms of q and can be seen as equality posture constraints. Differentiating $e(q)$ w.r.t. time we obtain the following linear differential system

$$\dot{e}(q) = J_e \dot{q} \tag{2}$$

where $J_e = \frac{\partial e}{\partial q} \in \mathbb{R}^{m \times n}$ is the Jacobian of the task. If $m < n$, it means that the robot is redundant w.r.t. the task and J_e is not invertible. The robotics literature on numerical methods for solving Eq. (2) is wide. Newton-Raphson algorithm and Levenberg-Marquardt method are among the best techniques to solve this kind of problems [12]. The general solution is written as

$$\dot{q} = J_e^+ \dot{e} + P_e z \tag{3}$$

where $\dot{q} \in \mathbb{R}^n$ is the time derivative of q, J_e^+ denotes the pseudoinverse of J_e, $P_e = I - J_e^+ J_e$ is the projection operator spanning the $n - m$ null space of J_e and z is any vector in $\mathbb{R}^n$. The recursive method proposed in [13], and improved in [14], for simultaneously solving a stack of k prioritized tasks is sketched in Algorithm 1.

Algorithm 1. Given k tasks with priorities, find $\dot{q} \in \mathbb{R}^n$ where n is the number of d.o.f. of the robot

```
Set P_{e0} ← I_{n×n}
Set q̇ ← 0
for i = 1, 2, ..., k
    Compute J_{ei} and ė_i
    Set Ĵ_{ei} ← J_{ei} P_{ei−1}
    Compute Ĵ_{ei}^+
    Set q̇_i ← Ĵ_{ei}^+ (ė_i − J_{ei} q̇_{i−1})
    Set q̇ ← q̇ + q̇_i
    Set P_{ei} ← P_{ei} − Ĵ_{ei}^+ Ĵ_{ei}
end
```

Inequalities of the form

$$e(q) \leq 0 \tag{4}$$

are more difficult to handle. They can be treated by an active set method (i.e. the subset of inequalities that become equalities define the active set). The computational experience reported in [15] indicates that quadratic programming methods are strong competitors for mixed systems. The inherent discontinuities arising in Eq. (4) has been solved in [16]. Examples of these constraints are joint limits and collision avoidance.

B. Contact constraints

We consider $r \leq n$ frictionless point contact constraints as nonlinear equalities of the form

$$\Phi(q) = (\phi_1(q), \ldots, \phi_r(q))^T = 0 \tag{5}$$

Differentiation of (5) w.r.t. time leads to

$$J_\Phi \dot{q} = 0 \tag{6}$$

where $J_\Phi = \frac{\partial \Phi}{\partial q} \in \mathbb{R}^{r \times n}$ is the Jacobian of the constraint w.r.t. q. Eq. (6) specifies that any $\dot{q}$ must belong to the null space of J_Φ. It implies that the following operator

$$P_\Phi = I - J_\Phi^+ J_\Phi \tag{7}$$

allows to project the joint velocity vector $\dot{q}$ into the tangent plane at the contact point. If $r < n$ and using Algorithm 1 together with Eq. (7), the joint velocity can be defined as

$$\dot{q} = P_\Phi \dot{q}_e \tag{8}$$

where $\dot{q}_e$ is the output of Algorithm 1 that satisfies all posture constraints.

C. Constrained dynamic model

The Euler Lagrange modeling formalism applied to constrained robots deliver a set of nonlinear ordinary differential equations of motion coupled with a holonomic algebraic constraint, built upon the constrained Lagrangian $L = K - P + \Phi^T(q)\lambda$, for K, P the scalars kinetic and potential energy, respectively, and $\lambda \in \mathbb{R}^r$ the Lagrange multipliers for r independent contacts, that is the constraint $\Phi(q)$ is full row rank. The resulting model is a differential algebraic equation of index 2 (the constraint $\Phi(q)$ is required to be derived twice to obtain the control input) as follows

$$\begin{aligned} M(q)\ddot{q} + h(q,\dot{q}) &= \tau + J_\Phi^T\lambda \\ \Phi(q) &= 0 \end{aligned} \tag{9}$$

where $M(q) \in \mathbb{R}^{n\times n}$ denotes the symmetric positive definite inertial matrix, $h(q,\dot{q}) \in \mathbb{R}^n$ contains the Coriolis, centrifugal and gravitational terms, $\tau \in \mathbb{R}^n$ represents the generalized input torques. For solving Eq. (6), notice that the constraint Jacobian J_Φ is not rank-deficient, due to $rank(\Phi(q)) = r \;\; \forall q \in \mathbb{R}^n$, consequently differentiating Eq. (6) w.r.t. time, we get

$$J_\Phi\ddot{q} + \dot{J}_\Phi\dot{q} = 0 \tag{10}$$

It is preferable to replace (5) with (10) in (9) to obtain a well-posed system representation of the state space system as follows

$$\begin{pmatrix} M(q) & J_\Phi^T \\ J_\Phi & 0 \end{pmatrix}\begin{pmatrix} \ddot{q} \\ -\lambda \end{pmatrix} = \begin{pmatrix} \tau - h(q,\dot{q}) \\ -\dot{J}_\Phi\dot{q} \end{pmatrix} \tag{11}$$

which can be solved implicitly for both $\ddot{q}$ and λ, by using an implicit Runge Kutta solver. Notice that a high-gain numerical constrained stabilizer u can be added to the right hand side of Eq. (10), as it is customary in numerical simulations of DAE-2 systems, say the Baumgarten u stabilizer [17], then (12) becomes

$$\begin{pmatrix} \ddot{q} \\ -\lambda \end{pmatrix} = \begin{pmatrix} M(q) & J_\Phi^T \\ J_\Phi & 0 \end{pmatrix}^{-1}\begin{pmatrix} \tau - h(q,\dot{q}) \\ -\dot{J}_\Phi\dot{q} + u \end{pmatrix} \tag{12}$$

This allows numerical stabilization of the solution by keeping bounded, below physical resolution of encoders, the nonlinear integration error, maintaining the structural integrity of the DAE-2 system.

III. Force-motion control scheme

On one hand, motion planning techniques should produce feasible smooth trajectories to the robot. On the other, control schemes should guarantee that the robot is well behaved, in a sense that real trajectories does not deviate from desired ones. However, in the context of cooperative robots, there is an unavoidable coupling of both, the planner and the controller to maintain the dynamic feasibility of the assigned task. In other words, they must account for the interaction of all robotic arms through the rigid object while satisfying the set of contact and posture constraints. As a consequence, the design of the control system for the coupled problem of simultaneous local motion planning and control of a complex holonomic robotic system deserves further discussions before presenting a convenient control design.

A. Preliminary discussion

There is a variety of cooperative robot control schemes to choose from, nonetheless there are two issues to highlight for redundant multirobot systems, one is the high dimension of the composed regressor which turns to be a time-consuming process. Then it is convenient to derive a smooth regressor-free controller; the second issue is the model of the mechanical robotic system, which yields differential equations tightly constrained by an algebraic kinematic implicit function (see Eq. (9)). A departing point is to design a controller for the DAE-2 system, not for the classical ODE formulation of robot dynamics. Finally, notice that the controller is required to exhibit fast and robust convergence to desired trajectories so as to the physical robots wholly comply too with all contact and posture constraints for every sampling guided by the planner.

B. Control design

Our previous cooperative regressor-free control scheme [18] is extended to joint space with similar stability properties, that is exponential convergence of all position, velocity and contact force tracking errors. This scheme exhbits low computation cost since it does neither depend on knowledge of parameters nor regressor, consequently meets the two issues exposed above to comply adequately with all closed kinematic constraints at each instant. To this end, consider a decentralized model-free scheme which enforces exponential convergence of position/velocity in the tangent subspace at the contact point while guaranteeing contact force tracking to comply with the holonomic constraint. This passivity-based control system is independent of structural knowledge of the dynamic robots, though it explicitly accounts for dynamic compensation, rendering a smooth second order sliding mode in both orthogonalized tangent position/velocity P_Φ and normal force J_Φ^T subspaces, at the contact point. The joint extension of our previous proposed controller stands for a nonlinear decentralized regressor-free position/velocity controller is

$$\begin{aligned} \tau \;\; = \;\; & -K_p(q)\Delta q - K_v(q)\Delta\dot{q} - K_i(q)\int_{t_0}^{t_f}\mathrm{sgn}(A) \\ & -K_{pf}(q)(B) - K_{if}(q)\int_{t_0}^{t_f}\mathrm{sgn}(B) \\ & -J_\Phi^T(\lambda_d - \tanh(\nu B) - \eta(\gamma_p B + \gamma_i\int_{t_0}^{t_f}\mathrm{sgn}(B)) \\ & -J_\Phi^T\dot{\Xi}_f - K_{pr}(q)\Delta\dot{P} - K_{dr}(q)\Delta P \end{aligned} \tag{13}$$

where $\Delta q = q - q_d$ stands for the tracking error, $A = \Delta\dot{q} - \alpha\Delta q - \Xi_q$, $\Xi_q = (\Delta\dot{q}(t_0) - \alpha_q\Delta q(t_0))e^{-\beta_q t}$, $B = \Delta F - \Xi_f$, $\Delta F = \int_{t_0}^{t_f}(\lambda - \lambda_d)dt$, $\Xi_f = \Delta F(t_0)e^{-\beta_f t}$, $K_p(q) = K_d P_\Phi\alpha$, $K_v(q) = K_d P_\Phi$, $K_i(q) = K_d P_\Phi K_i$ stand for the configuration dependent feedback matrices of a nonlinear PID-like position-velocity controller; while $K_{pf}(q) = K_d\gamma_p J_\Phi^+$ and $K_{if}(q) = K_d\gamma_i J_\Phi^+$ build a nonlinear PI-like force controller. The term $K_{pr}(\dot{P} - \dot{P}_d)$, builds a compensating term to enforce the constraint velocities, which stand for the joint velocities deviates onto the force subspace, that is $\dot{P}_i = \sum_j^l J_{\Phi j}\dot{q}_j$, for $i \neq j, i = 1, ..., l$, where l represents the number of robots. $\dot{P}_{di} = \sum_j^l J_{\Phi j}\dot{q}_{dj}$ stands for the desired constraint velocity, built upon desired joint velocity. Feedback gains $K_{pr}(q) = K_d J_\Phi^+$, $K_{dr}(q) = K_d J_\Phi^+ \alpha_r$ and the remaining K_d, K_p, K_v, K_i, K_{pf}, K_{if}, α, α_q, α_r, ν, η, β, β_q, β_f, γ_i, γ_p are positive definite feedback gains of appropriate dimensions. Ξ_p, Ξ_f are exponentially vanishing terms useful to place the system on the sliding surface at t_0, then removing the reaching phase both in the position/velocity and force subspaces, respectively; finally, $tanh$ and sgn are the nonlinear *hyperbolic tangent* and *signum* functions, respectively.

The strict stability proof follows closely to [18], therefore details are omitted. Notice that this proof relies on Lyapunov and second order sliding mode stability arguments to guarantee exponential tracking towards desired position, velocity and contact forces coming, at each sampling period from the planner, without any overshoot. Once the control is applied to each robot, afterwards sensors read the state of the whole joints which are sent back to the planner to close the iterative loop and build a simultaneous motion planning and control system.

Finally, the apparent involved control structure is nothing but an output state feedback controller easy to compute on line (values and range of feedback gains are established in the stability proof), without any dependence of the dynamic model. The stability proof certainly is involved, the control structure is not, with stability properties that meet all performance requirements.

IV. The Iterative Strategy

The iterative algorithm we describe in this section can be seen as a local motion planner that encapsulates an internal force-motion control module. At each iteration, the nested controller computes a feasible torque vector for each robotic arm to ensure local convergence.

A. Prioritizing the tasks

We first define $q_c = (q_{r1}, \ldots, q_{rp})^T$ as the composite vector containing the configuration of all the specified robots as well as the object to be manipulated. The number of entities is p. This representation allows to define a global inertial reference frame where q_c is the configuration of the whole system. Note that the object continues to be a free-flying body and the robots are branches of the same kinematic tree. Thus, all of them are attached to the same inertial reference frame.

The goal of the cooperative task is to transport an object from a given location to a desired one. This task defines an equality constraint in terms of the location of the object as a function of q_c. To achieve the goal, additional equality and inequality constraints must be satisfied to maintain contacts. Also, joint limits and collision avoidance should be considered. The task-redundancy formalism is a natural way to define all these constraints w.r.t q_c. We propose the following stack of tasks organized in decreasing priority:

1) contacts between the tip of each arm and the object,
2) joint limits,
3) collision avoidance and
4) move the object to a desired location.

Note that we handle the compromise between the achievement of the goal and the satisfaction of constraints by assigning the lowest priority to the object-carrying task. Clearly, it is preferable to maintain the contacts even if the result is an approximate solution w.r.t. the desired location of the object.

B. Algorithm outline

The iterative process starts by calling Algorithm 1 to determine the next desired velocity vector $\dot{q}_c^d$ for the whole system. Then, an appropriate generalized torque is computed using Eq. (13) to track the joint velocity reference while preserving the desired contact forces. After that, an efficient constrained dynamics method solves Eq. (12) to get the feasible joint acceleration vector and contact forces. By integrating the joint acceleration we get the next input to Algorithm 1. The general strategy is summarized in Algorithm 2. The process terminates whether the task errors converges to a given threshold or a predefined number of iterates is exceeded.

Algorithm 2. Choose a desired location of the object x_d such that $e_k(q) = 0$; define the initial configuration of the whole system q_c; define the desired contact forces λ^d; choose $tol > 0$; define $k-1$ additional tasks with priorities; construct the stack $E(q) = (e_1(q), ..., e_k(q))$.

Begin
$\dot{q}_c \leftarrow 0$
Evaluate $E(q_c)$ by

$$e_i(q_c) \quad \text{for} \quad i = 1, 2, \ldots, k$$

while $E(q_c) \geq tol$ **do**
Compute $\dot{q}_c^d$ with $(E(q_c), q_c)$ by means of Algorithm 1.
Set $q_c^d \leftarrow q_c + \dot{q}_c^d$
for $j = 1, 2, ..., p$
Set $q_j, \dot{q}_j, q_j^d$ and $\dot{q}_j^d$ with the actual and desired configurations and joint velocities of robot j from $q_c, \dot{q}_c, q_c^d$ and $\dot{q}_c^d$ respectively.
Solve (12) for λ_j
Compute τ with $(\lambda_j, \lambda_j^d, q_j, \dot{q}_j, q_j^d, \dot{q}_j^d)$ by means of (13)
Solve (12) for $\ddot{q}_j$
Update $q_c, \dot{q}_c$ at j
end for
$\dot{q}_c \leftarrow \dot{q}_c + \ddot{q}_c$
$q_c \leftarrow q_c + \dot{q}_c$
Evaluate $E(q_c)$ by

$$e_i(q_c) \quad \text{for} \quad i = 1, 2, \ldots, k$$

otherwise exit.
End

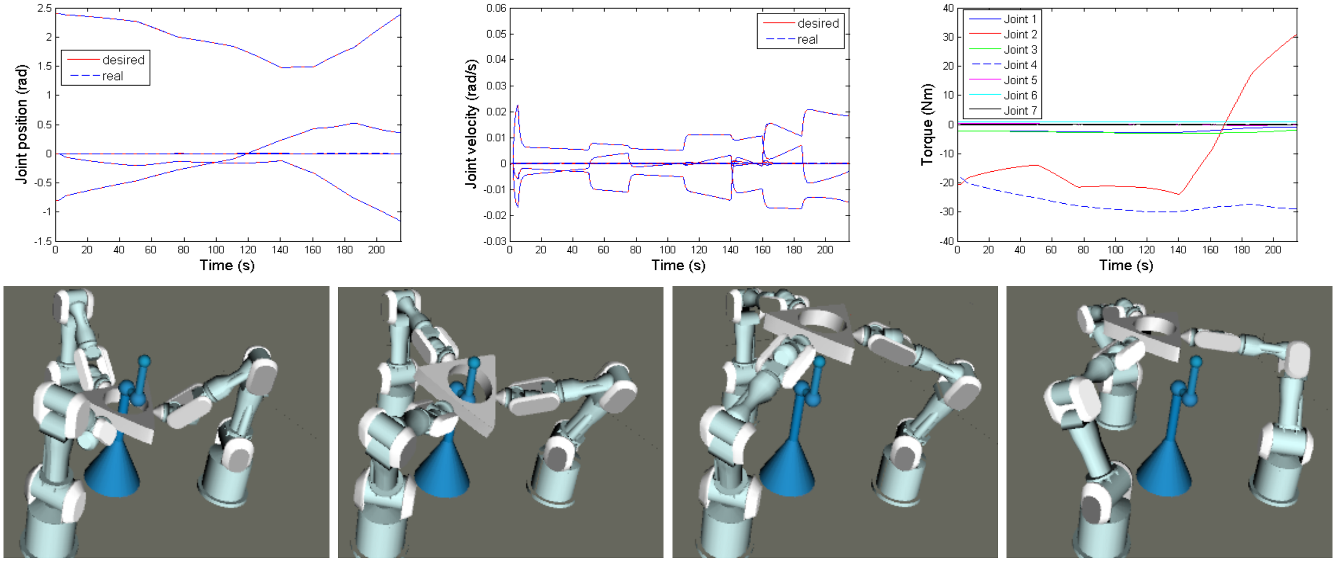

Fig. 2. From top-left to right the joint position, velocity and input torque tracking profiles respectively. Bottom: a representative sequence of the solution.

C. Numerical considerations

Regarding Algorithm 1, we have replaced the Moore-Penrose's pseudoinverse by the well-known mixed minimization formulation [19]:

$$\min_{\dot{q}}\{||\dot{e}_i - J_{ei}||^2 + \dot{q}_i^T W \dot{q}_i\}$$

which corresponds to

$$\dot{q}_i = ({J_{e}}_i^T J_{ei} + W)^{-1} {J_{e}}_i^T \dot{e}_i$$

where W is a diagonal matrix with damping factors. A complete numerical sensitivity study on the selection of W is reported in [12]. Thus, we have defined $W = e_i I + \overline{W}$ where $\overline{W}$ is a bias diagonal matrix and its elements are set to 1.0×10^{-3}.

Algorithm 2 does not need to compute the costly second-order differential kinematics. However, inside the algebraic loop (see lines 5-7) the constrained dynamics in Eq. (12) has to be solved. To overcome this difficulty, we have adopted recursive algorithms with low computational cost described in [20] to efficiently compute inverse and forward dynamics. A straightforward method is

- Compute $h(q, \dot{q})$ by applying the compact version of the recursive Newton-Euler algorithm with spatial operators.
- Compute the joint space inertia matrix $M(q)$ by applying the composite-rigid-body algorithm.

With vector $h(q, \dot{q})$ and matrix $M(q)$, Eq. (12) can be solved for λ and $\ddot{q}$.

It is important to remark that the main loop of Algorithm 2 is based on a numerical optimization strategy and consequently it is sensitive to become trapped in local minima. However, the iterative method can be coupled with sampling-based planners such as RRT-like algorithms to compute *global* solutions [21].

V. Simulation Experiments

The algorithms were implemented in C++ and compiled using GNU C++. The experiments have been performed on a AMD Turion 64 X2 dual-core 3.0 GHz. We used Blas and Lapack libraries for all the linear algebra operations. We have implemented the Featherstone's algorithms [20] to compute inverse and forward dynamics. The final visualization was with Matlab.

We compared our iterative method with a two-stage strategy commonly used in local motion planning. One of the main differences between them is that our algorithm considers the constrained dynamics of the system at each iteration of the task-priority loop. The two-stage strategy consists in computing a kinematic trajectory with Algorithm 1. Then, at the second stage the trajectory is transformed into a dynamically feasible one using forward and inverse dynamics. It turns out that this transformation may break some constraints that were satisfied in the previous stage at kinematic level. Examples of that are the so-called residual collisions. As a consequence, such a strategy should iterate between both stages until a feasible motion satisfies the constraints.

We used the same scenario for both two-stage strategy and Algorithm 2 in order to illustrate the differences. In our experiment we considered 3 PA10 robotic platforms and a 3D bulky object in a constrained environment. The main constraint is represented by one static obstacle. We also defined the initial configuration of a movable object near the static obstacle as it is illustrated on the left frame of Fig. 1. Note that the robots are already in contact with the object surface, generating forces at these contacts. There exist 2 kinematic closed-loops. The whole system has 27 d.o.f. and 15 d.o.f. after considering the closed-chains.

The robots have to cooperate in order to pull the object out without loosing contacts while avoiding joint limits

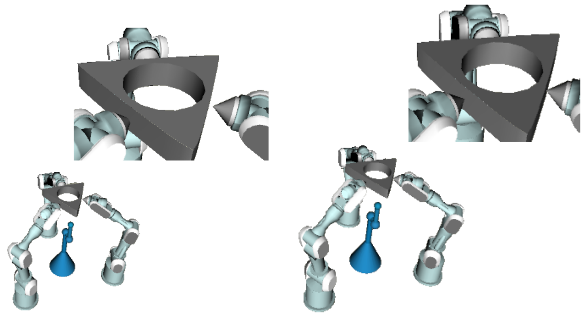

Fig. 3. The comparison between the two-stage strategy (left) and our iterative method (right). The former consists in computing a kinematic trajectory for the whole system. The resulting trajectory satisfies all posture constraints computed with Algorithm 1. Then, the trajectory is transformed into a dynamically executable trajectory through the controller. The right frame shows the same example computed with Algorithm 2.

and obstacles. The desired location was slightly out of the obstacle. Hence, a pure translation could cause collisions (see Fig. 2). To automatically compute the whole motion it was necessary to impose one via point, otherwise the system get trapped in a local minimum. In Fig. 3 the left frame shows a representative example of the two-stage strategy. There exist some collisions between the end-effector and the object. Moreover, at least one robot lost the contact with the object. As a consequence, Algorithm 1 had to recompute again a kinematic trajectory. The right frame shows the same example but computed with Algorithm 2.

Simulation sequences presented in this work can be found at: http://148.247.22.5/saltillo/robotica/garechav/cooperative/

VI. ACKNOWLEDGMENTS

This work has been supported by the National Council of Science and Technology (CONACyT) project No. 84855.

VII. CONCLUSION AND FUTURE WORK

Most of the planners only solve the problem from a geometric and kinematic point of view providing eficient motion trajectories for the robotic system, whilst ignoring the dynamical behavior of the robot when it is constrained by a rigid environment. For complex systems, such as cooperative or humanoid robots, this approach may fail since the real dynamical state may deviate from the ideal trajectories computed unilateraly by the planner, typically off-line. This fact motivates to coupling on-line the local planner with the dynamical control for practical applications in robotics. In this paper, we proposed an iterative algorithm to deal with severely constrained cooperative robots within a task-priority framework. Moreover, the method accounts for frictionless contact problems in manipulation tasks to blend seemingly our previous force-motion scheme [18] in joint space based on constrained robot dynamics. Preliminary results show that our method can be successfully applied in practice. We conjecture that our proposal can be extended, in particular for challenging problems arising in hyper-redundant constrained robotic systems. For instance, the problem of whole-body stable object motion planning and manipulation [4]. Our current work is focusing on validating the use of our method in such robotic problems.

REFERENCES

[1] T. Simeon, J.-P. Laumond, J. Cortes, and A. Sahbani, "Manipulation planning with probabilistic roadmaps," *The International Journal of Robotics Research*, vol. 23, no. 7-8, pp. 729–746, 2004.

[2] E. Yoshida, C. Esteves, I.-R. Belousov, J.-P. Laumond, T. Sakaguchi, and K. Yokoi, "Planning 3-d collision-free dynamic robotic motion through iterative reshaping," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1186–1198, 2008.

[3] G. Arechavaleta, C. Esteves, and J.-P. Laumond, "Planning fine motions for a digital factotum," in *IEEE-RSJ International Conference on Intelligent Robots and Systems*, Japan, December 2004.

[4] E. Yoshida, M. Poirier, J.-P. Laumond, O. Kanoun, F. Lamiraux, R. Alami, and K. Yokoi, "Pivoting based manipulation by a humanoid robot," *Autonomous Robots*, vol. 28, no. 1, pp. 77–88, 2010.

[5] M.-H. Raibert and J. j. Craig, "Hybrid position/force control of manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 102, no. 2, pp. 126–133, 1981.

[6] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal of Robotics and Automation*, vol. 3, no. 1, pp. 43–53, February 1987.

[7] N.-H. McClamroch and D.-W. Wang, "Feedback stabilization and tracking of constrained robots," *IEEE Transactions on Automatic Control*, vol. 33, no. 4, pp. 419–426, May 1988.

[8] S. Arimoto, "Joint-space orthogonalization and passivity for physical interpretations of dextrous robot motions under geometric constraints," *International Journal of Robust and Nonlinear Control*, vol. 5, no. 4, pp. 269–284, 1995.

[9] G. Liu and Z. Li, "A unified geometric approach to modeling and control of constrained mechanical systems," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 574–587, August 2002.

[10] W. Blajer, "A geometric unification of constrained system dynamics," *Multibody System Dynamics*, vol. 1, no. 1, pp. 3–21, March 1997.

[11] Y. Yang and O. Brock, "Elastic roadmaps: Globally task-consistent motion for autonomous mobile manipulation," in *Robotics: Science and Systems*, Philadelphia, August 2006.

[12] T. Sugihara, "Solvability-unconcerned inverse kinematics based on levenberg-marquardt method with robust damping," in *IEEE-RAS International Conference on Humanoids Robots*, Paris, France, December 2009, pp. 555–560.

[13] B. Siciliano and J.-J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *IEEE International Conference on Advanced Robotics*, Pisa, Italy, June 1991, pp. 1211–1216.

[14] P. Baerlocher and R. Boulic, "An inverse kinematic architecture enforcing an arbitrary number of strict priority levels," *The Visual Computer: International Journal of Computer Graphics*, vol. 20, no. 6, pp. 402–417, August 2004.

[15] O. Kanoun, F. Lamiraux, P.-B. Wieber, F. Kanehiro, E. Yoshida, and J.-P. Laumond, "Prioritizing linear equality and inequality systems: application to local motion planning for redundant robots," in *IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 2009, pp. 724–729.

[16] N. Mansard, O. Khatib, and A. Kheddar, "A unified approach to integrate unilateral constraints in the stack of tasks," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 670–685, June 2009.

[17] J. Baumgarte, "Stabilization of constraints and integrals of motion in dynamical systems," *Computer Methods in Applied Mechanics and Engineering*, vol. 1, pp. 1–16, June 1972.

[18] D. Navarro-Alarcón, V. Parra-Vega, and E. Olguín-Díaz, "Minimum set of feedback sensors for high performance decentralized cooperative force control of redundant manipulators," in *IEEE International Workshop on Robotic and Sensors Environments*, Ottawa, Canada, October 2008, pp. 114–119.

[19] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *Journal of Dynamic Systems, Measurement and Control*, vol. 108, pp. 163–171, 1986.

[20] R. Featherstone and D. Orin, "Robot dynamics: Equations and algorithms," in *IEEE International Conference on Robotics and Automation*, San Francisco, CA, April 2000, pp. 826–834.

[21] S. LaValle and J. Kuffner, "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.