Integrated Constrained Motion Planning and Control for Compliant Robots

Abstract—**In this paper we present a new approach to integrate motion planning and control for robots under task constraints. The method is comprised of two stages. In a first stage, the constraint is relaxed, replacing the lower–dimensional constraint manifold with a narrow but full-dimensional boundary layer. Motion is planned using RRT* on points within the boundary layer, which can be efficiently sampled. The plan resulting from this first stage will not satisfy the task constraint, and generates displacements with respect to the constraint manifold. In the second stage of the algorithm, we introduce an elastic model in the system, associating such displacements to interaction forces acting outside the tangent space of the constraint manifold. The elastic model can be fictitious or represent physical compliance in case of soft robotic systems. In the second stage the plan is refined by regulating positions along the constraint manifold, and forces in the complement. This is achieved by a novel geometric control scheme that is proved to be noninteracting (i.e. makes position errors irrelevant to force control, and viceversa) for a linearized model of the system. The final outcome of the two-stage planner is therefore a trajectory that satisfies constraint with arbitrarily good approximation, asymptotically rejecting perturbations coming from sample displacements.**

Fig. 1. The motion planning method presented in this paper is developed for systems with closed kinematic loops.

I. Introduction

In robotics, motion planning and control are two problems typically addressed separately, the first usually dealing with geometry and kinematics, while the second copes with dynamics. These two problems become harder when we consider constraints, such as closed kinematic chains, task constraints, accelerations and torque limits, among others.

In this paper we present an integrated approach for motion planning and control of robots subject to kinematic constraints, i.e. closed kinematic chains such as those arising when robots are in contact with the environment or with themselves (see e.g. fig. 1).

It should be noticed that an ever increasing number of robotic systems designed for interaction use compliant actuation (e.g. series elastic or variable stiffness actuators). In these systems, compliance can be regarded as a convenient buffer between planning (always imperfect) and execution, allowing to relax the constraints that the interaction task imposes to the mechanism.

The method for constrained motion planning we propose in this paper for arbitrary robots, is inspired to the control of soft robots, but is also applicable to traditional rigid structures.

Planning motions for robots while dealing with constraints imposed by the task into their structure is known as constrained motion planning problem. This problem has been widely studied in recent years. The main issues in this problem are: 1) working in high dimensional spaces and 2) dealing with constraints. Randomized algorithms such as RRT have been shown to be a viable solution for unconstrained problems [7, 6]. An important extension of RRT to introduce optimality criteria in the results has been proposed in the RRT* algorithm [5].

The planning problem with constraints is challenging because an explicit description of the admissible manifold in the configuration space ($\mathcal{CS}$) is typically not available. This implies that not all randomly generated samples in the $\mathcal{CS}$ can be considered as a possible configuration to explore because the constraint describes a zero–measure lower–dimensional submanifold in the original $\mathcal{CS}$. There exist some approaches to generate motions for robots with constraints, which are based either on the decomposition of the chain in a passive and an active chain [3], or in the projection of any random sample to the valid submanifold [10, 1, 11]. All this approaches guarantee that the samples are in the constraint manifold, but suffer from the introduction of computational complexity in either explicitly solving the constraint or in iteratively finding the projection of samples onto the manifold.

In this paper we present a two–stage planning method for robots under task constraints. In the first stage we consider (or pretend) that the robot is compliant, so that the constraint manifold is relaxed to a boundary layer. Random points are chosen in this boundary layer with an effective biasing mechanism that has high probability of sampling points within the boundary layer. These samples are then used as a basis for the application of RRT*, thus obtaining an asymptotically optimal plan in the boundary layer.

Such plan however will not necessarily satisfy the task constraint, as it generates displacements with respect to the constraint manifold. If used on a real compliant robot, this plan would generate a motion that is approximately correct, but also internal forces proportional to the displacement from the manifold and depending on the robot and environment combined stiffness.

As such interaction forces can be unacceptable, we introduce a second stage to filter the plan and regulate interaction forces to zero (or a small value). We show that it is indeed possible to achieve this regulation of internal forces without jeopardizing the plan.

To build the filter we adopted the geometric algorithm for the analysis of the output functional controllability of general manipulation systems presented in [8, 9]. Then, for a linearized model of the system, a parameterization for a noninteracting control to regulate positions of the mechanism and internal forces is presented.

The final outcome of the two-stage planner is therefore a trajectory that satisfies constraint with arbitrarily good approximation, asymptotically rejecting perturbations coming from sample displacements.

The rest of this paper is organized as follows. First and second stages of the planning approach presented in this papers are detailed in sections II and III. Section IV we apply our approach in a simple but explanatory example. Finally section V gives the conclusions.

II. Sample–Based Planning Algorithm on Manifolds

Random sampling-based methods for path planning have an excellent performance when they are able to explore the whole $\mathcal{CS}$ of robot manipulators. The good performance comes from the fact that manipulators are able to explore the whole environment performing any kind of motions in any directions in the $\mathcal{CS}$, so any point in it is a valid configuration and can be connected to any other one. On the other hand when the system is subject to constraints, such as closed kinematic chains, this fact is not true any more.

In this section we propose a sampling technique for constrained systems and a sample based planning algorithm.

A. Motion Planning for Constrained Robots

Consider a configuration space $\mathcal{M} \in \mathbb{R}^d$ that is a compact set of configurations q_i. Let $\mathcal{O} \subset \mathcal{M}$ be the obstacle region and $\mathcal{M}_{free} := \mathcal{M} \setminus \mathcal{O}$ the configuration set free of obstacles. Introducing a kinematic constraint $C(q) = 0$ that limits the robot configurations and hence motion, we define a nonlinear submanifold of $\mathcal{M}$ as $\mathcal{M}_v := \{q : q \in \mathcal{M}_{free}, C(q) = 0\}$ to describe all configurations where none of the links of the mechanism collide neither with objects in the environment nor with other links, and satisfy the constraint. The motion planning problem is to find a continuous path $\sigma : [0, 1] \to \mathcal{M}_v$; with $\{\sigma(0) = q_{init},\ \sigma(1) = q_{final}\}$.

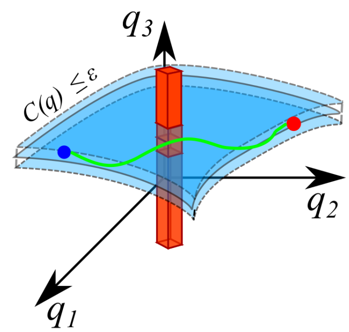


Fig. 2. Motion planning problem under relaxed constraints. Initial position q_{init} in blue. Final position q_{final} in red. Planned path in green. Constraint $C(q)$ in baby blue.

B. Introducing Compliance to Relax Constraints

As mentioned in the introduction we consider (or pretend) that the system is compliant. The idea is to consider the compliance, in the planning phase, as a parameter to relax the constraint, i.e. $C(q) \leq \epsilon$. With this assumption, the submanifold describing the relaxed constraint can be considered as a narrow but fully dimensional boundary layer in the $\mathcal{CS}$. Thanks to this we can use rejection techniques to randomly sample in $\mathcal{M}_r := \{q : q \in \mathcal{M}_{free} \wedge C(q) \leq \epsilon\}$, and thus speed up the planning process. Now the planning problem is to find a continuous path $\sigma : [0, 1] \to \mathcal{M}_r$, with, $\{\sigma(0) = q_{init},\ \sigma(1) = q_{final}\}$. This relaxed problem is graphically described in Fig. 2.

C. Sample-Based Planning Algorithm

The random based–sampling algorithm used in this paper is the $Soft\!-\!RRT^*$ reported in Algorithm 1. The difference with the original RRT* algorithm is that instead of just checking for collision we also check if the new configuration is inside the relaxed constraint.

Algorithm 1 $\mathcal{T} = (V, E) \leftarrow$ soft-RRT*(x_{init})

```
T ← InitTree();
for i = 1 to N do
    x_rand ← Sample(i);
    x_nearest ← Nearest(V, x_rand);
    x_new ← Steer(x_nearest, x_rand);
    if Constraints(X_nearesr, x_new) then
        X_near ← Near(T, x_new)
        x_min ← BestParent(X_near, x_new)
        T ← T ∪ (x_new, x_min)
        T ← Rewire(T, X_near, x_new)
    end if
end for
return G = (V, E).
```

The main functions use in the Algorithm 1 are

- In function **Sample** a configuration x_{rand} is generated using the algorithm presented in section II-D, which converges to a uniform distribution of random points within the boundary layer.

- Function **Nearest** returns the (previously sampled) configuration $x_{nearest}$ closest to x_{rand}.
- Function **Steer** connects two configurations if possible otherwise a configuration x_{new} is obtained as in RRT*. In this work we are using simple interpolation in joint space as a local steering procedure.
- **Near** function returns a set X_{near} containing points which are inside a ball centered on x_{near}. For details on the parameterization of the ball the reader can refer to [5].
- The main difference with the original RRT* algorithm is in the **Constraints** function. In our case this function includes not only collision checking but also the validation of the configuration x_{new} to be in the boundary layer.
- **BestParent** function select the best configuration $x_{min} \in X_{near}$ to connect with x_{new}.
- Function **Rewire** rearrange the tree if one of the configurations on X_{near} could be better connected to the tree passing through x_{new}.

D. Biased Random Sampling

The first step in randomized path planners is perfomed in the function **Sample** and it consists on generating a new sample in $\mathcal{M}_r$. Typically, random configurations are taken using a uniform distribution to explore equally all regions in $\mathcal{CS}$. In our problem, the probability of sampling a new point in $\mathcal{M}_r$ can be computed as

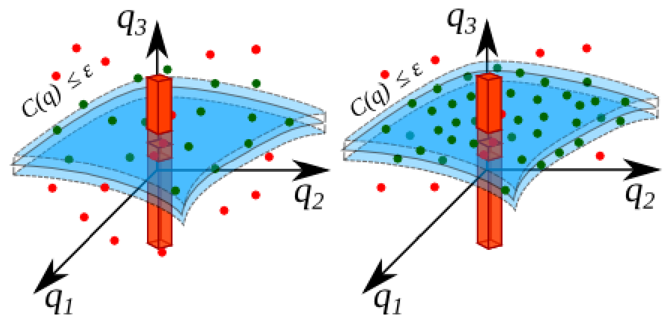
$$\rho = volume(\mathcal{M}_r)/volume(\mathcal{M}), \tag{1}$$

where the volume of $\mathcal{M}$ depends on the mechanism, more precisely by the range of motion of all joints. On the other hand, the volume $\mathcal{M}_r$ depends of the relaxing parameter ϵ. As a consequence, the probability of sampling a point in $\mathcal{M}_r$ goes to 0 with ϵ. In other words, bigger is ϵ higher is the probability of sampling in $\mathcal{M}_r$.

It is evident that, for small values of ϵ, most of the samples will be rejected by the **Constraints** function because they do not lay in the relaxed constraint. To minimize the impact of this fact, in the proposed Soft-RRT* we used the algorithm presented in [2] to bias the random sampling procedure to converge to a uniform distribution not in $\mathcal{M}$ but in $\mathcal{M}_r$, The algorithm consists in building a k-d tree to approximate $\mathcal{M}_r$. The idea is graphically presented in the Fig. 3.

E. Equilibrium configurations

In order to apply the planning algorithm proposed in this paper to closed kinematic mechanism it is necessary to define the points to evaluate the interaction forces. A viable solution is to consider the central link as an object being manipulated and their previous and following joints as contact points. In the case of robots contacting a surface the end effector can be considered as the manipulated object, the contact point in the environment together the joint linking the end effector to the robot are considered as contact points. This details are important to evaluate the equilibrium position of the mechanism and the object (in case of multiple robots



(a) (b)

Fig. 3. Graphical explanation of the difference of using uniform distribution (a) and applying the algorithms presesnted in [2] (b) to get random sampling configurations in $\mathcal{M}_r$.

manipulating an object), which is then used in the collision detector included in function **Constraints**.

Equilibrium configurations can be computed performing a kineostatic analysis of the system as in [4]. In that work authors presented the equilibrium manifold of the multi-robot system subject to synergistic underactuation and variable stiffness in the joint actuation. In this work we are considering a fully actuated system. For the particular case of multi-robot system manipulating and abject, the equilibrium configurations are described by the following equations

$$w + G(u) f_h = 0, \tag{2}$$
$$\tau - J^T(q) f_h = 0, \tag{3}$$
$$f_h - K_c p_h = 0, \tag{4}$$
$$\tau - K_q(q_r - q) = 0, \tag{5}$$

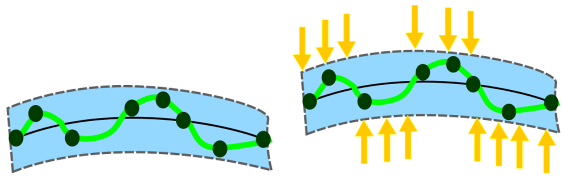
where $q \in \mathbb{R}^d$ is the vector of joint positions, $q_r \in \mathbb{R}^d$ is the reference joint position, joint torques are represented by $\tau \in \mathbb{R}^d$, contact forces by $f_h \in \mathbb{R}^6$ and object positions by $u \in \mathbb{R}^6$. $G(u)$ is the so called grasp matrix of the system, $w \in \mathbb{R}^6$ is the external wrench in the object, $J^T \in \mathbb{R}^{c \times d}$ is the multi-robot Jacobian matrix, $K_c \in \mathbb{R}^{c \times c}$ is the contact stiffness matrix and $K_q \in \mathbb{R}^{d \times d}$ is the joint stiffness matrix. Dimension c is the number of contact constraints.

Equations (2) to (5) describe the equilibrium manifold in the system that by adding quotes into the contact forces f_h becomes the same as the one used in the planning phase. In order to find equilibrium configurations we firstly get the random sampling configurations q_r and then we solve the following optimization problem to the get the rest of the variables

$$\begin{aligned} \min_{q,u,f_h,\tau} \quad & x^T \mathbf{W} x \\ \text{subject to} \quad & \Phi(x, q, u, f_h, \tau) = 0, \end{aligned} \tag{6}$$

where $x \in \mathbb{R}^{2d+12}$ is an equilibrium residual vector and $\Phi(x) \in \mathbb{R}^{2d+12}$ groups the equations describing the equilibrium manifold.

Once the planned path is obtained, a proper controller able to let the robot follow the planned path must be sensitized. The



(a) Undesired forces arising from planning on the relaxed contraint

(b) The task of the controller is to project the undesired forces back to the manifod

Fig. 4. Lateral view of the relaxed constraint. In green are the pushing and pulling forces against the constraint. The black dots are the nodes extracted from the tree generated by the soft-RRT*.

main challenge comes from the fact that the closed kinematic constraint has been relaxed producing undesired contact forces arise during interactions, see a graphical example in Fig. 4(a). A force control law must thus ensure that the nominal constraint is satisfied during the whole execution, see Fig. 4(b). For example, in case of bimanual manipulation the planning stage provide a path along which the object handled by the robot does not fall but can be damaged since high squeezing/stretching forces can appear.

III. Geometric Noninteracting Control

For force planning stage we are adopting a hybrid control scheme to regulate points along the constraint manifold and forces in the complement.

Consider a mechanism with n actuated joint variables q, and c constraints $C(q) = 0$. Let m denote the residual mobility of the system (i.e., the no. of dof's compatible with the constraints). Now consider the constraints as elastic elements, i.e. remove the constraint and allow for violations $C(q) \leq \epsilon$ and associate internal forces $f = K\epsilon$ where K can be interpreted as a suitable stiffness matrix resulted from contacts and joints stiffness. Looking at the space of internal forces that can be actively controlled by joint torques, let p be the dimension of self-balanced (internal) forces, i.e. those which do not affect the overall position of the mechanism.

A. The linearized model

The linearized model, with no disturbances, of the lumped parameter compliant model for the multiple robot-object dynamics presented in [9] is,

$$\dot{x} = Ax + B_\tau \tau' + B_\omega \omega, \tag{7}$$

defined at the equilibrium configuration

$$\begin{aligned} x &= \left[q_{eq}^T \; u_{eq}^T \; 0^T \; 0^T\right]^T \\ \tau' &= \tau - J^T t_{eq} \\ \omega &= G t_{eq}. \end{aligned} \tag{8}$$

Where q_{eq} and u_{eq} stand for the equilibrium joint positions and object position respectively, J is the Jacobian of the contact points and G is known as grasp matrix. Under the assumptions reported in the previously mentioned work, the dynamics matrix A, joint torque input matrix B_τ, and external disturbance matrix B_ω have the form

$$A = \left[\begin{array}{c|c} 0 & I \\ \hline -L_k & -L_b \end{array}\right], \; B_\tau = \left[\begin{array}{c} 0 \\ 0 \\ \hline M_h^{-1} \\ 0 \end{array}\right]; B_\omega = \left[\begin{array}{c} 0 \\ 0 \\ \hline 0 \\ M_o^{-1} \end{array}\right], \tag{9}$$

where

$$L_k = M^{-1} P_k; \; L_b = M^{-1} P_b, \tag{10}$$

$$\begin{aligned} M &= \begin{bmatrix} M_h & 0 \\ 0 & M_o \end{bmatrix} \\ P_k &= \begin{bmatrix} J^T \\ -G \end{bmatrix} K \begin{bmatrix} J & -G^T \end{bmatrix} \\ P_b &= \begin{bmatrix} J^T \\ -G \end{bmatrix} B_q \begin{bmatrix} J & -G^T \end{bmatrix}, \end{aligned} \tag{11}$$

M_h and M_o are the multiple-arm and object dynamic matrices. K and B_q are the stiffness and damping matrices at the contact points.

We are interested in the combinations of states giving object positions and internal forces as outputs, which can be respectively selected by matrices $C_u = [0 \; I \; 0 \; 0]$ and $C_t = \left[KJ \; -KG^T \; B_q J \; -BG^T\right]$. The output matrix is hence

$$C = \begin{bmatrix} \Gamma_u^+ C_u \\ E^+ C_t \end{bmatrix}. \tag{12}$$

Image space of matrices Γ_u consist of rigid body motions of the object being manipulated while E is a base matrix for the asymptotic internal forces.

For non-redundant mechanism, Theorem 1 in [8] states that $m + p = n$. This means that the input-output representation of the minimal A-invariant functionally controllable subspace of states is square. This implies that it is possible to devise a linear controller that uses the n inputs to decouple and control independently rigid motions in the constraint manifold and internal forces in the complementary direction.

B. Noninteracting Control

The control objective is to project back to the constraint the motions generated during planning phase, and at the same time regulate the internal forces in the mechanism. In practice it is useful to control each of the outputs independently, meaning that the controller should be able to regulate each of output without affecting the others. Finding a control law where the input i affects just the corresponding output i is known as noninteracting control. The procedure is to differentiate the output vector $y = Cx$ until the control appears. In case of the output corresponding to $E^+ C_t = E^+[KJ \; -KG^T \; BJ \; -BG^T]$, the control appears in the first derivate

$$\begin{aligned} \dot{y} &= C_t A x + C_t B \tau_t^* \\ &= C_t A x + E^+ B_q J M_h^{-1} \tau_t^*. \end{aligned} \tag{13}$$

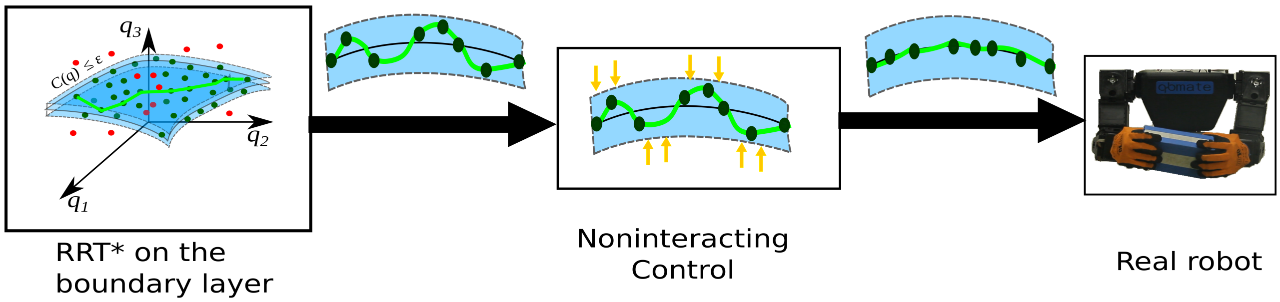


Fig. 5. Pipeline of the constrained motion planning approach proposed in this paper. First the kinematic planning is performed in fully dimensional boundary layer obtained from the ϵ relaxed constraint. Then using the noninteracting control the trajectory is projected to the constraint and the interaction forces are regulated to zero.

For the output corresponding to the object motions $\Gamma_u^+ C_t = \Gamma_u^+ [0 \ I \ 0 \ 0]$ it is necessary to compute up to the third derivate. Indeed,

$$\begin{aligned} \dot{y} &= C_u A x + C_u B_\tau \tau_u^* \\ &= C_u A x + 0 \tau_u^* \end{aligned} \tag{14}$$

$$\begin{aligned} \ddot{y} &= C_u A^2 x + C_u A B_\tau \tau_u^* \\ &= C_u A^2 x + 0 \tau_u^* \end{aligned} \tag{15}$$

$$\begin{aligned} \dddot{y} &= C_u A^3 x + C_u A^2 B_\tau \tau_u^* \\ &= C_u A^2 x + \Gamma_u^+ M_o^{-1} G B_q J M_h^{-1} \tau_u^* . \end{aligned} \tag{16}$$

Thus the corresponding output vector can be rewrite as

$$\hat{y} = P x + Q B_\tau \tau^* \tag{17}$$

where $\hat{y} = [\dddot{y}_u \ \dot{y}_t]^T$,

$$P = \begin{bmatrix} C_u A^3 \\ C_t A \end{bmatrix} \text{ and } Q = \begin{bmatrix} C_u A^2 \\ C_t \end{bmatrix} . \tag{18}$$

Concluding, for the system (7) , the control law

$$\tau = -Q^{-1}(P x + \tau^*), \tag{19}$$

provides the output vector

$$\hat{y} = \tau^* = [\tau_u \ \tau_t]^T . \tag{20}$$

In other words the proposed control law is noninteracting since the rejection of position errors does not influence the force control and viceversa.

The whole motion planning approach presented in this paper is explained in figure 5

IV. Example

In this section we present a simple but explanatory example which is described in Fig. 6. It is a robotic hand composed by two fingers (each one with one degree of freedom) holding an object. All valid configurations are described by the following constraint manifold

$$C(q) = F_1(q_1) - F_2(q_2) = 0 \tag{21}$$

where $F_1(q_1)$ and $F_2(q_2)$ stand for the forward kinematics of the fingers.

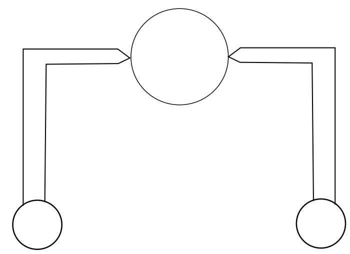


Fig. 6. Robotic hand used as an explanatory example

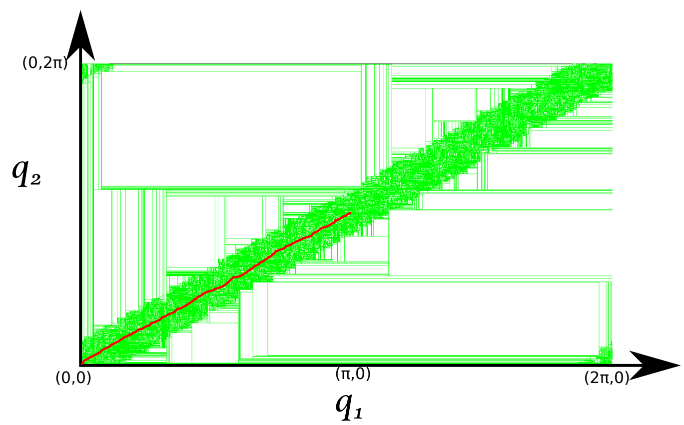


Fig. 7. This figure shows reconstructed boundary layer obtained using the adaptive k-d tree, the sampling procedure converges to a uniform distribution in the relaxed constraint.

The dimension of the $\mathcal{CS}$ for this example is 2 while the dimension of the state space for the linear system used in the second stage is 10. The parameter defining the relaxation limits was $\epsilon = 0.5$. For this example just stiffness at the contact points was considered. The outcome of the fists stage is depicted in Fig. 7 where the k-d tree approximation of the boundary layer is shown together with the path generated by the algorithm 1.

For the second stage of the algorithm, we linearized the dynamic model of the hand on the configuration $q_{init} = [0 \ 0]$ which is the same starting position for the first stage. The output trajectory resulting from the first and second stage are

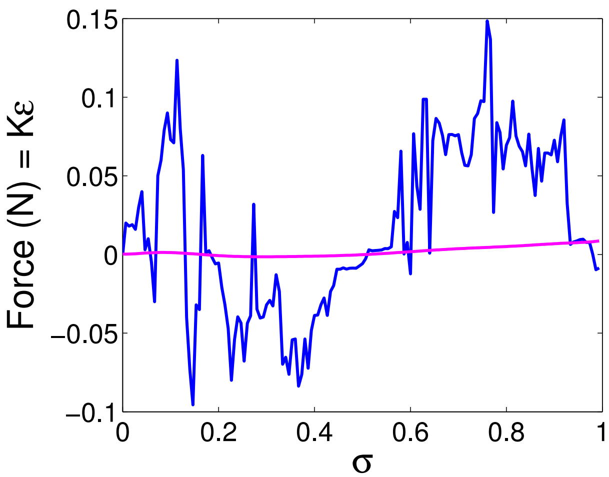


Fig. 8. Comparison of forces resulting from the planned path in the first stage and forces filtered in the control phase

compared in Fig. 8. The maximum constraint violation in the first phase was 0.1486 while in the second part was reduced to 0.00864.

The example presented in this section shows the validity of the presented approach.

V. Conclusions

In this paper we presented an integrated approach for motion planning and control of closed kinematic chains (given by the robot interaction with the environment or by the mechanism itself). The method consists in two parts, the first consist in planning a path replacing the lower–dimensional constrained manifold with a narrow boundary layer with the same dimension of the configuration space of the mechanism. RRT* algorithm together with an adaptive k-d tree is performed in the mentioned boundary layer. The output of the first stage is a path which violates the constraint less than an ϵ parameter. This parameter is associated to internal forces thanks to a model of compliance considered for the interactions of the robot and the environment. The introduction of the compliance in the planning enable the algorithm to avoid projections during planning phase. Internal forces are then filtered in the second stage which consist on a geometric control algorithm, specifically noninteracting controller, applied to a linearized model of the system to regulate forces. The final output of the presented approach is a motion trajectory for the mechanisms as well as the control inputs which can be applied in the real robot.

Future work is directed to 1) considere the redundant motion subspace in the linearized model and in the controller, 2) compare the proposed algorithm with other algorithms with the same purpose using more complex examples such as bimanual manipulation using redundant arms.

References

[1] Dimitry Berenson. *Constrained Manipulation Planning*. PhD thesis, 2011. URL http://repository.cmu.edu/dissertations/172/.

[2] Joshua Bialkowski, Michael Otte, and Emilio Frazzoli. Free-configuration biased sampling for motion planning. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1272–1279, 2013. doi: 10.1109/IROS.2013.6696513.

[3] J. Cortes, T. Simeon, and J-P Laumond. A random loop generator for planning the motions of closed kinematic chains using PRM methods. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 2, pages 2141–2146 vol.2, 2002. doi: 10.1109/ROBOT.2002.1014856.

[4] E. Farnioli, M. Gabiccini, M. Bonilla, and A. Bicchi. Grasp compliance regulation in synergistically controlled robotic hands with vsa. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2013*, pages 3015 –3022, Tokyo, Japan, November 3-7 2013. doi: 10.1109/IROS.2013.6696783.

[5] Sertac Karaman and Emilio Frazzoli. Incremental sampling-based algorithms for optimal motion planning. *CoRR*, abs/1005.0416, 2010.

[6] L.E. Kavraki, P. Svestka, J.-C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *Robotics and Automation, IEEE Transactions on*, 12(4):566–580, 1996. ISSN 1042-296X. doi: 10.1109/70.508439.

[7] Steven M. Lavalle. Rapidly-exploring random trees: A new tool for path planning. Technical report, 1998.

[8] Domenico Prattichizzo and Antonio Bicchi. Consistent task specification for manipulation systems with general kinematics. *in ASME Journal of Dynnamic System Measurements and Control. Accepted*, pages 760–767, 1997.

[9] Domenico Prattichizzo and Antonio Bicchi. Dynamic analysis of mobility and graspability of general manipulation systems. *IEEE Transactions on Robotics and Automation*, 14:241–258, 1998.

[10] M. Stilman. Global manipulation planning in robot joint space with task constraints. *Robotics, IEEE Transactions on*, 26(3):576–584, 2010. ISSN 1552-3098. doi: 10.1109/TRO.2010.2044949.

[11] Dawen Xie and N.M. Amato. A kinematics-based probabilistic roadmap method for high dof closed chain systems. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 1, pages 473–478 Vol.1, April 2004. doi: 10.1109/ROBOT.2004.1307194.