

Reactive Motion Generation for Robots in Dynamic Environments

Sami Haddadin, Rico Belder, and Alin Albu-Schäffer

Institute of Robotics and Mechatronics, DLR - German Aerospace Center, Wessling, Germany, Tel: +49(0)8153 28 1047; e-mail: sami.haddadin@dlr.de

Abstract: In the near future robots are sought to become an integral part of human everyday life. Also in industrial settings robotic Co-Workers are expected to become a commodity. Even though the particular application areas may vastly change, a robot always needs to act in a dynamic and partially unknown environment. It shall reactively generate motions and prevent upcoming collisions. If contact is desired or inevitable, it has to handle it robustly and safely. For preventing collisions in a real-time fashion the Circular Fields method is a powerful scheme, which we developed further and evaluated it extensively. After an initial analysis in rather complex 2D simulations, we extend the evaluation to 3D as well as 6D, where we introduce a hybrid strategy based on Circular and Potential Fields. Finally, the 6D implementation of a hybrid Circular & Potential Fields approach is used to perform the experimental analysis for static multi-object parcours and to avoid a dynamically moving human in a 6D task motion. Based on the algorithms for collision avoidance we also develop and experimentally verify an algorithm for tactile exploration of complex planar 3D wire elements, whose structure is a-priori unknown.



Fig. 1. Physical cooperation between humans and robots.

1. INTRODUCTION

In the near future robots are sought to become an integral part of our daily life as multi-purpose service assistants in our homes. Apart from such domestic applications, flexible and versatile robots may also relieve us from monotonous and physically demanding work in industrial settings. In dangerous or even life-threatening surroundings, as e.g. deep-space or underwater, they may replace humans entirely. Especially when being used in disaster areas or underground scenarios a robot may search and rescue people from hardly accessible locations. However, despite vastly changing particular application areas, a robot always needs to be able to act in a dynamic and partially unknown environment. It shall reactively generate motions and prevent upcoming collisions. If contact is desired or inevitable, it needs to robustly and safely handle it.

Especially physical Human-Robot Interaction (pHRI) is a field in which such behavior is of immanent importance. As human and robot shall collaborate very closely (see Fig. 1) the problem of generating “human-friendly” motions is of large interest. Even though the close interaction of

human and robot in the domestic and industrial sector was always proclaimed to open up entirely new possibilities for service applications and production processes, several problems are still to be tackled before finally achieving this ambitious goal. A particularly important problem indeed is the generation of safe motions in human vicinity, which safely circumvent dynamic obstacles.

However, up to now reactive motion capabilities were usually developed for mobile robots, Yamamoto and Yun [1995], Ögren et al. [2000], where the complexity of the avoidance problem is limited by nature. For multi-degree of freedom (DoF) articulated manipulators, on the other hand, only few algorithms that are tailored to their needs were developed, Siciliano and Khatib [2008], Khatib [1985], Yamamoto and Yun [1995], Ögren et al. [2000], Brock and Khatib [2002] and only few extensively tested experimentally. This is mainly due to the fact that such abilities were unnecessary in real-world applications because of numerous reasons. Articulated robots were exclusively used for industrial applications with only static or very predictable environmental constraints. These applications require only precomputed trajectories that usually remain unchanged and humans are segregated with fences from the robot. Recently, however, first articulated robots have gained the mechanical and control capabilities for coping with local uncertainties in their environment and during (physical) interaction with humans. Powerful and highly sensorized arms as e.g. the DLR Lightweight Robot III (LWR-III), Albu-Schäffer et al. [2007] or the Barrett WAM Arm, Townsend and Guertin [1999] were developed over the last decade. Those systems are particularly well designed for applications that incorporate Human-Robot Interaction. Well suited control strategies were developed for this new type of robots for nominal interaction control and sensitive collision detection and reaction, Albu-Schäffer et al. [2007], Haddadin et al. [2008], De Luca et al. [2006], Wang et al. [2007], Ebert and Henrich [2002]. This significant progress finally necessitates the development of appropriate real-time collision avoidance methods that

* This work has been partially funded by the European Commission's Sixth Framework Programme as part of the project VIAC-TORS under grant no. 231554.

are particularly well suited for the robot's demands and capabilities in pHRI tasks.

In this paper we approach the aforementioned problem by extensively evaluating a particularly promising approach: the Circular Fields method. We analyze the scheme in numerous 2D, 3D, and 6D scenarios with respect to its ability to generate reactive motions in real-time with limited local knowledge of a possibly dynamic and complex environment. Furthermore, simultaneous goal convergence, while providing coordinated movement in translation and orientation, is a further primary requirement. For implementing such a motion behavior we significantly extend and combine Circular Fields with Potential Fields such that they generate smooth, intuitive trajectories and/or virtual disturbance signals that can be fed to low-level controllers. We apply the method to different levels of control and motion generation in order to analyze its respective effectiveness. Furthermore, we provide promising experimental results for rather complex real-world examples. All methods are experimentally verified on the DLR Lightweight Robot III (LWR-III).

In addition to addressing the reactive motion control for pHRI, we also extend the algorithms such that a powerful scheme for physically exploring unknown wire objects with tactile information only is generated. The algorithm is used to explore complex planar 6D wires in both simulation and experiment. It enables the robot to incrementally built a geometric interaction map of the object and updates it according to the respective sensory input.

2. ALGORITHM

In this section we introduce the Circular Field method together with some modifications, which significantly enhance the original scheme. Apart from generating better motion behavior in the sense of intuitive behavior for obstacle avoidance, we also outline how to significantly reduce the calculation load.

2.1 Point mass dynamics

In this paper we associate a virtual attractor dynamics with the robot, which is directly affected by virtual forces \mathbf{F}_v generated by the CF algorithm. This causes the virtual particle m to avoid collisions, while converging to the final goal. The assumed point mass dynamic system equation is

$$m\ddot{\mathbf{x}} = -k_a(\mathbf{x} - \mathbf{x}_d) + \mathbf{F}_v - k_d\dot{\mathbf{x}}, \quad (1)$$

where the quadratic potential field for goal attraction is

$$U_a = \frac{1}{2}k_a(\mathbf{x} - \mathbf{x}_d)^T(\mathbf{x} - \mathbf{x}_d). \quad (2)$$

and the virtual obstacle force $\mathbf{F}_v = \sum_j \mathbf{F}_{ob,j}$ is composed of the subforces generated by the distinct obstacle objects (OOs). $\mathbf{x}, \mathbf{x}_d, k_a, k_d$ are the position, desired goal, (positive) attractive constant, and (positive) damping constant.

2.2 Circular Fields

The Circular Fields approach (CFs), Singh et al. [1996], is based on the generation of artificial electro-magnetic-fields (B-Fields) that are generated by virtual current elements associated to the surface of obstacles. Thus, in contrast to electrostatic charges for Potential Fields (PFs) the analogon of dynamical electric charges are consulted. This adaptation of a B-Field \mathbf{B} generates obstacle forces

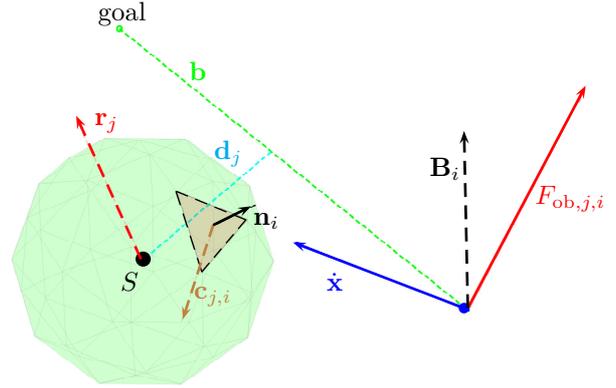


Fig. 2. Principle of the current definition for Circular Fields.

$\mathbf{F}_{ob,j}$ that are vertical to the avoiding object (AO) velocity vector $\dot{\mathbf{x}}$, see Fig. 2.

$$\mathbf{F}_{ob,j} = \dot{\mathbf{x}} \times \sum_i \mathbf{B}_i \quad (3)$$

The generated forces cause a re-orientation of the instantaneous velocity vector and yield a guidance of the avoiding object around the obstacle object j . This process dissipates no kinetic energy from the system. The local circular field \mathbf{B}_i of each surface element i (of an obstacle), acting on the virtual particle is defined as

$$\mathbf{B}_i = I_K \frac{\mathbf{c}_{j,i} \times \frac{\dot{\mathbf{x}}}{\|\dot{\mathbf{x}}\|}}{l_i^2} da_i, \quad (4)$$

where I_K is the virtual current factor, $\mathbf{c}_{j,i}$ is the current direction vector of surface element i , $l_i = \|\mathbf{x} - \mathbf{x}_{\mathbf{n}_i}\|$ is the distance of the current position of the point mass \mathbf{x} , and $\mathbf{x}_{\mathbf{n}_i}$ is the position of the obstacle surface element¹.

2.3 Adapting the CF Surface Current Rotation Vector

In the original CF approach, Singh et al. [1996], two schemes for current definition were given. They generate independent current elements for every surface and therefore, induce especially for the 3D case unwanted oscillatory behavior (see Singh et al. [1996] Fig. 6-7). Due to various problems in applying these algorithms already in simulation, we chose an alternative approach to define the current directions. In particular, we select the current of all CF obstacle surfaces to depend on each other. The chosen definition uses the vector from the actual AO position \mathbf{x} to the desired position \mathbf{x}_d , the goal vector \mathbf{b} and the center of mass of the respective OO $\mathbf{m}_{og,j}$, see Fig. 2. The local circular field $\mathbf{B}_{j,i}$ of each surface element, acting on the virtual particle k is now defined as

$$\mathbf{B}_i := I_K \frac{(\mathbf{n}_i \times \mathbf{r}_j) \times \frac{\dot{\mathbf{x}}}{\|\dot{\mathbf{x}}\|}}{l_i^2} da_i, \quad (5)$$

where \mathbf{n}_i is the normal of surface element i and $\mathbf{x}_{\mathbf{n}_i}$ is the position of the surface element. The surface current of (4) is defined as $\mathbf{c}_{j,i} = \mathbf{n}_i \times \mathbf{r}_j$.

\mathbf{r}_j is the field rotation vector of obstacle j , which is defined as

¹ This artificial electro-magnetic-field equation is a modification of Equ. (2) in Singh et al. [1996] by indices and our variable notation. It has roughly nothing in common with the real world background.

$$\mathbf{r}_j := \frac{\mathbf{d}_j \times \mathbf{b}}{\|\mathbf{d}_j \times \mathbf{b}\|}, \quad (6)$$

with \mathbf{d}_j being the shortest distance between the center of mass of the OO $\mathbf{m}_{og,j}$ and the goal vector \mathbf{b} :

$$\mathbf{d}_j = \mathbf{x} + \mathbf{b} \frac{(\mathbf{m}_{og,j} - \mathbf{x}) \cdot \mathbf{b}}{\|\mathbf{b}\|^2} - \mathbf{m}_{og,j} \quad (7)$$

With the given definition the current is continuous around the object and therefore oscillating robot behavior is avoided as long as the damping and attractor are chosen accordingly.

2.4 Velocity Angle Adaptation

The chosen current definition leads to diverting forces in front of the obstacle and forwarding forces behind it. Even though this behavior will already lead to satisfactory results it may lead to a ‘‘caching’’ effect, which may cause the robot to be follow the obstacle surface for some cycles (however no global minimum present). Changing the direction according to the obstacle rotation vector \mathbf{r}_j about a constant angle ψ

$$\dot{\mathbf{x}}^* = \mathbf{R} \left(\frac{\mathbf{r}_j}{\|\mathbf{r}_j\|} \psi \right) \dot{\mathbf{x}} \quad (8)$$

leads to

- earlier deviation (also if being parallel to an infinite wall)
- and to reducing the CF influence after passing by the obstacle (when not being that deep in the CF anymore).

This reduces the aforementioned effect significantly².

2.5 Calculation Load Reduction

The construction of obstacles and AOs based on surfaces results in an exponentially increasing calculation effort for the CF approach. Thus, it is of large interest to simplify the algorithm without losing performance and to analyze possible parallelizability. For AO n and OO j consisting of surfaces k and i the virtual force equation includes two sum formulations.

$$\mathbf{F}_{obj,j,n} = \sum_k \sum_i (\dot{\mathbf{x}}_{k,i,rel} \times \mathbf{B}_{i,k}) \quad (9)$$

In simulation experiments it turned out to be disadvantageous to include the relative velocity $\dot{\mathbf{x}}_{n,k}$ of surfaces k to the body center point of the object n . Thus, we consider the relative velocity to the body center point approximately to be zero and therefore negligible. The velocity of every avoiding surface element is now the relative velocity $\dot{\mathbf{x}}_{n,relative}$ of AOs n to OOs j . Because of this simplification it is possible to reduce (9). This leads to

$$\mathbf{F}_{obj,j,n} = \dot{\mathbf{x}}_{rel} \times \sum_k \sum_i \mathbf{B}_{i,k}, \quad (10)$$

where j is the actual OO and n is the AO the force acts on. In the following, we define $\dot{\mathbf{x}} = \dot{\mathbf{x}}_{rel}$. \mathbf{B}_i may be written as

$$\mathbf{B}_{i,k} := I_K \frac{(\mathbf{n}_{j,i} \times \mathbf{r}_j) \times \frac{\dot{\mathbf{x}}}{\|\dot{\mathbf{x}}\|}}{l_i^2} da_i da_k, \quad \text{with } \mathbf{c}_{j,i} = \mathbf{n}_i \times \mathbf{r}_j. \quad (11)$$

In addition, we consider that the current $\mathbf{c}_{j,i}$ is defined to be constant or calculated ahead. If we also regard that the

² In fact, we did not observe it anymore in any simulation.

objects are constructed with approximately equally sized surfaces, the calculation effort reduces further and (11) is simplified to

$$\mathbf{B}_{i,k} := I_K \frac{\mathbf{c}_{j,i} \times \frac{\dot{\mathbf{x}}}{\|\dot{\mathbf{x}}\|}}{l_i^2} da^2. \quad (12)$$

Another reduction is achieved by including an individual obstacle surface size factor k_{B_i} in the individual current direction vector of every surface. With this it is possible to change the sum formulation as follows.

$$\sum_k \sum_i \mathbf{B}_{i,k} = I_K da^2 \sum_k \sum_i \frac{\mathbf{c}_{j,i}}{l_{i,k}^2} \times \frac{\dot{\mathbf{x}}}{\|\dot{\mathbf{x}}\|} \quad (13)$$

$$= -k_{B_i} \frac{\dot{\mathbf{x}}}{\|\dot{\mathbf{x}}\|} \times \sum_k \sum_i \frac{\mathbf{c}_{j,i}}{l_{i,k}^2} \quad \text{with } k_{B_i} = I_K da^2 \quad (14)$$

Thus, the object force may be written as

$$\mathbf{F}_{obj,j,n} = -k_{B_i} \left(\dot{\mathbf{x}} \times \left(\frac{\dot{\mathbf{x}}}{\|\dot{\mathbf{x}}\|} \times \sum_k \sum_i \frac{\mathbf{c}_{j,i}}{l_{i,k}^2} \right) \right). \quad (15)$$

Now, the calculation effort is already significantly smaller and may be further reduced with a voxel space approximation for $l_{i,k}^2$. This leads to determining the distance $l_{i,k}$ by the voxel grid. Another result would be that the algorithm could be highly parallelized. Furthermore, a parallelization of the parts of the avoiding robot is advantageous for multiple avoiding objects (e.g. several objects attached to the different robot links).

Additional Calculations for PFs Since later on for the 6D case we combine CFs and PFs (CFs for translation and PFs for orientation), we also calculate the corresponding PFs for each obstacle surface acting on every avoiding surface. Therefore, results from the CF force algorithm can be used for the generation of CF forces. If only the potential field forces need to be calculated for every avoiding surface,

$$\mathbf{F}_{n,k} = k_r \sum_i \frac{\mathbf{x} - \mathbf{x}_{n_i}}{\|\mathbf{x} - \mathbf{x}_{n_i}\|^2} \quad (16)$$

has to be solved³. This means that the minimal calculation effort $\forall i$ is 8 additions, 3 multiplications, and 3 divisions. This leads to 14 floating point operations (FLOs) in total. Using the data $\mathbf{d}\mathbf{x}_i = \mathbf{x} - \mathbf{x}_{n_i}$ and $l_i^2 = \|\mathbf{x} - \mathbf{x}_{n_i}\|^2$ already calculated for the CF forces, the remaining calculations are

$$\mathbf{F}_{n,k} = k_r \sum_i \frac{\mathbf{d}\mathbf{x}_i}{l_i^2}. \quad (17)$$

In other words, 14 FLOs are reduced to 6 FLOs, or 3 additions and 3 divisions, so the reduction of calculation load is 43 %.

Next, we discuss various simulation and experimental results obtained with the described method.

3. COLLISION AVOIDANCE: SIMULATIONS & EXPERIMENTS

In this section the experimental performance of the algorithm is shown for various 2D, 3D, and 6D scenarios. Therefore we assume all objects and calculated or estimated mass point as global known. In the following the only global information used are the mass points of

³ This is of course one possible choice for Potential Fields, however, a very advantageous one by means of calculation load reduction.

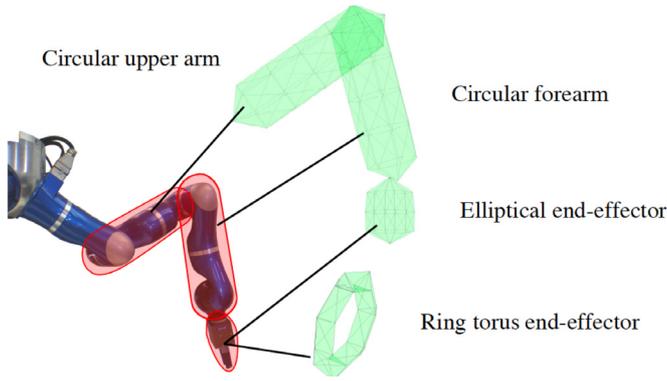


Fig. 3. The implemented, geometrical robot hull, for the LWR-III.

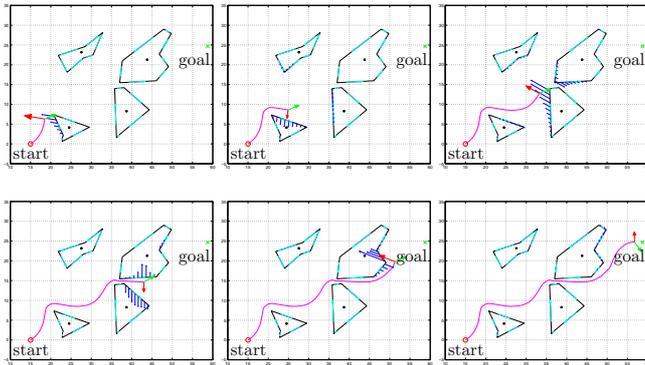


Fig. 4. 2D example for the circular field method surpassing a narrow passage.

every OO in influencing range. Present we working at solutions for local definition of mass points, whereby, it is of advantage to projected the mass point inside of the object. However, in the following for force calculation real limited view is used.

3.1 Robot Hull Design

In this paper a volumetric (only used for the 6D case) robot is formally separated into several avoiding objects (AOs) that create a partitioned geometric representation of the robot. These hulls cover the robot with surface elements and endow it with a volume representation for all actively movable parts, see Fig. 3. Thereby, we can generate virtual forces and moments acting on each separate robot segment⁴.

3.2 Simulation Random 2D

In Figure 4 a 2D example of a narrow passage problem is shown. The polygons are randomly generated⁵ and six sample steps from the full simulation were chosen to indicate the performance of the method. The virtual particle has a limited view range, which is indicated by the virtual forces calculated for the discretized surface elements. The resulting external force (red arrow) acting on the virtual particle and the attractor force (green) are shown. Furthermore, the associated current direction is

⁴ The current implementation is only used for evaluation purposes and will be substituted by automatically generated hulls, Frese and Täubig [2009].

⁵ We tested 50 random parcours, which all were successfully solve. Several examples can be seen online at www.safe-robots.com.

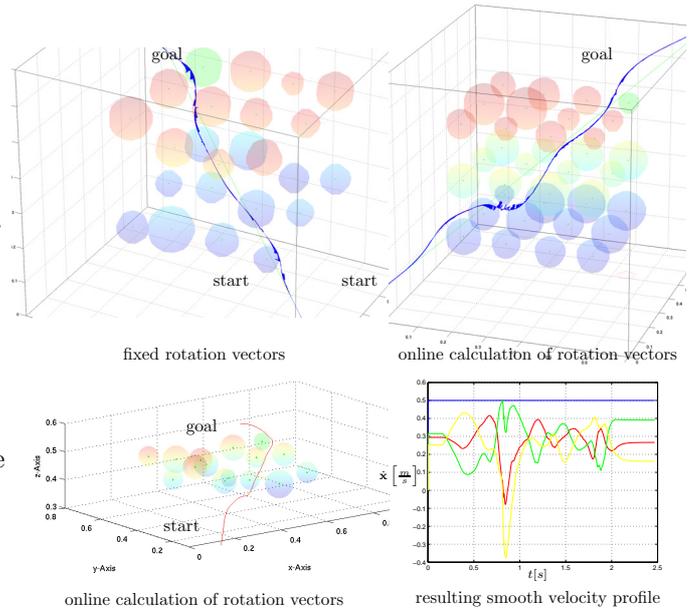


Fig. 5. 3D avoidance of spherical objects with a point mass.

indicated on the object borders. In contrast to potential fields, which always generate repulsive forces in normal direction to the surface element, this approach generates a very intuitive force response for the presented problem, smoothly guiding the robot to the goal.

3.3 Static random 3D spheres

For the 3D case we again chose to evaluate random test-beds, see Fig. 5. The obstacle influence range is between 0.03 and 0.05 units. The *upper left* figure depicts the behavior for fixed rotation vectors \mathbf{r}_j . The virtual point mass takes one of the shortest trajectories towards the goal. The images *down left* and *top right* are simulated for calculating the rotation vectors online. The derivation of \mathbf{r}_j leads to a virtual point mass trajectory with less risk of colliding with obstacles while moving towards the goal. The trajectory is similarly smooth as the previous example and needs approximately the same number of iteration steps for goal convergence, i.e. the generated trajectory is of similar quality as for fixed \mathbf{r}_j . Finally, in the image *down right* an example for a resulting velocity profile is given, showing the smooth behavior during a motion.

3.4 3D Trap Simulation Scenarios

To further analyze the capabilities of this algorithm in 3D, we simulated more complex problems, c.f. Fig. 6 and Fig. 7. Figure 6 shows a 3D trap from different angles. The robot has a limited view and thus enters the trap. However, after it is able to sense the walls of the object, it escapes and converges to the goal. Figure 7 depicts a box with only a single small entry. Even though the robot has only very limited view (indicated by the sparsely visualized grey spheres), it is able to escape and converge very smoothly to the goal. As one can see from the presented simulations, the collision avoidance based on circular fields is also able to cope with more complex obstacles having local minima and non trivial geometries.

3.5 Dynamic Objects

The behavior in a highly dynamic environment was also simulated in order to judge and adjust the behavior for

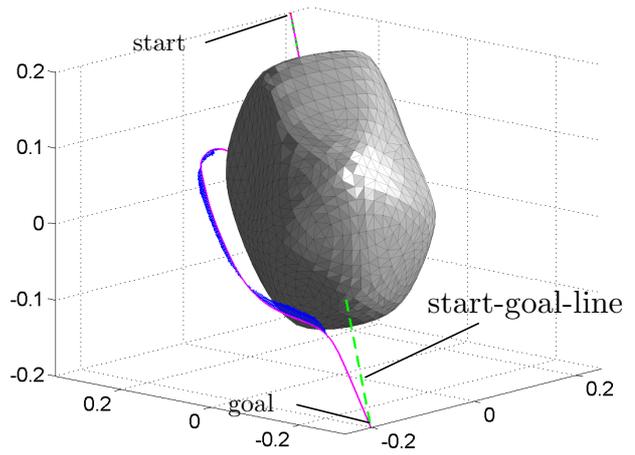
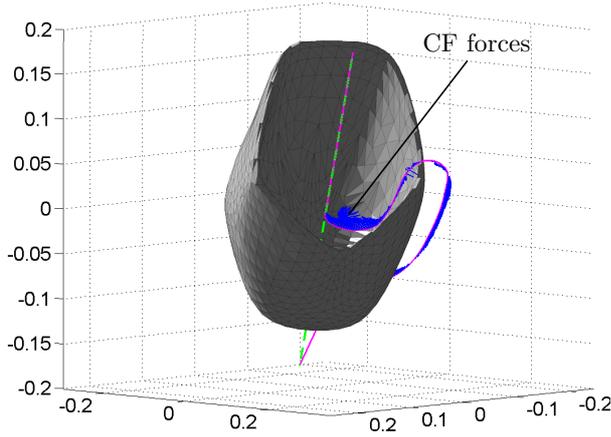


Fig. 6. Circular field based motion generation reactively

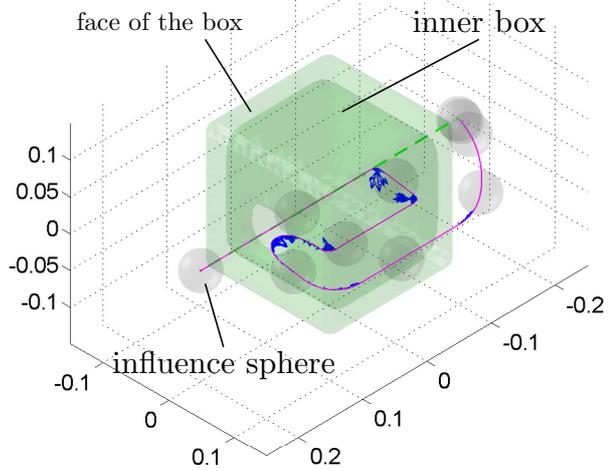
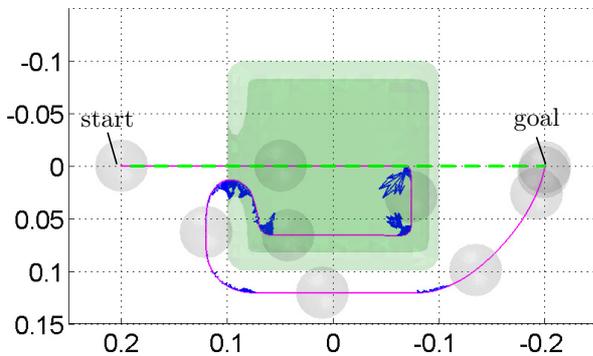


Fig. 7. Using CFs to reactively pass a complex dead-end (top and 3D View).

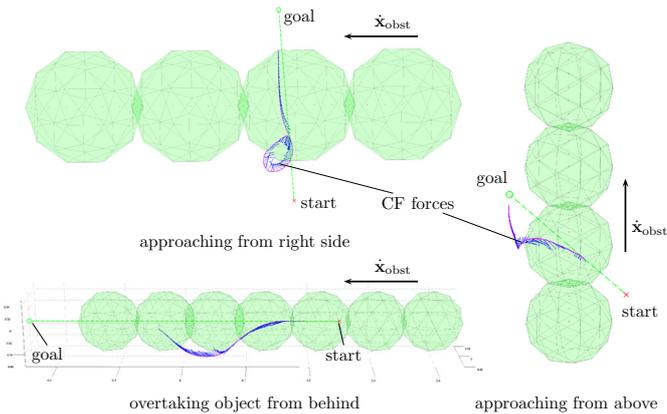


Fig. 8. Point mass avoiding dynamically moving objects.

this especially important situation. For this the algorithm is tested for some special cases of obstacle motion where the velocity of the obstacles is approximately the same or higher than the robot velocity, see Fig. 8. In order to avoid the approaching obstacles we chose different strategies that incorporate the current situation, differentiating whether the object approaches from behind or from the side and above. Avoiding an obstacle approaching from the front at high velocity can be performed with the approach we already used up to now. Therefore, we omit the discussion

of this particular case: For the other cases we chose following strategies.

- Relative velocity based
- Overtaking: switch orientation of obstacle rotation vector by 180° if the relative velocity is a towards motion (AO) or towards the goal (different approaches possible)
- Object from side/below/above: continuous turning of obstacle rotation vector depending on the (position to) OO and the relative velocity.

An open problem we still encounter is how to consistently combine the used strategies or to find **one** continuous strategy. However, this is left for future work.

3.6 6D End-Effector Collision Avoidance Experiment

Figure 9 depicts a classical operational space control loop consisting of

- (1) the motion generator for providing the desired reference motion in terms of generalized operational space coordinates \mathbf{x}_d as a function of time,
- (2) the controller that provides the desired operational space force, which is then transformed via the Jacobian transpose to desired torque commands, and
- (3) the physical robot that transforms the desired torque command via a low-level motor torque loop into

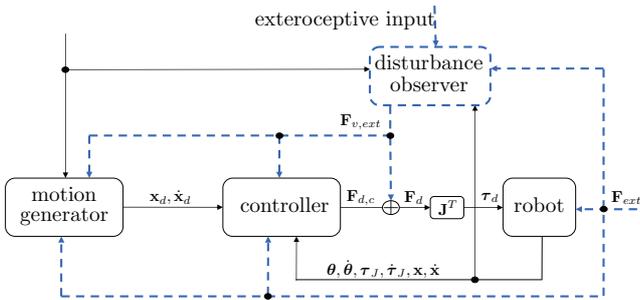


Fig. 9. Using disturbance inputs at different motion control levels for realizing effective disturbance response.

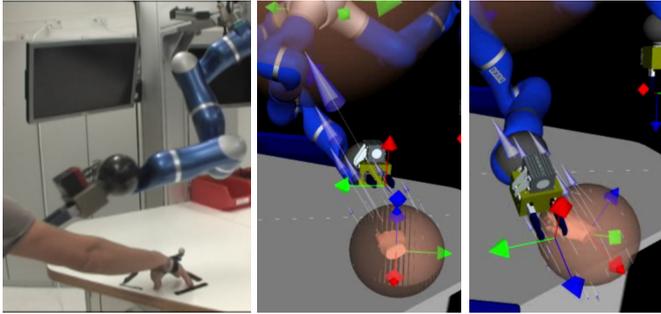


Fig. 10. Avoidance behavior for a Cartesian point to point motion. The depicted arrows denote the occurring forces calculated by the RCA.

motor torques that generate the respective robot motion.

A reactive disturbance response⁶ can be fourfold:

- (1) Physical forces act on the robot and inherently produce a dynamic response of the robot.
- (2) The controller can implement a purely passive disturbance response (the measurement of external forces is not directly incorporated) with respect to external forces, or actively react to them (a classical example is inertia shaping). A combination of both is of course possible as well.
- (3) Furthermore, generalized virtual forces, e.g. generated by repulsive potentials, can directly act as a motor command input and add a respective behavior to the controller.
- (4) The motion generator provides motion commands that directly take into account the presence of physical or generalized virtual forces, leading e.g. to a collision retraction or reactive collision maneuvers.

Usually, disturbance reaction schemes act isolated in the sense that the particular response is exclusively carried out by a single scheme. However, in order to provide more sophisticated and situation dependent behavior, it is important to equip a robot with the capability to react on multiple levels of abstraction simultaneously. For this we developed various control and motion generation schemes over the last years that are able to process various sensorial inputs as described in the following.

In the present experiment we evaluated the 6D case for a robot that avoids a dynamically moving human. The resulting forces from the obstacles are depicted for such a run in Fig. 10. As already mentioned we use CFs for translation and PFs for rotation as this generates

⁶ Please note that we do refer to hard real-time reaction and not adaptation of via points or motion fragments that are e.g. provided by a global motion planner.

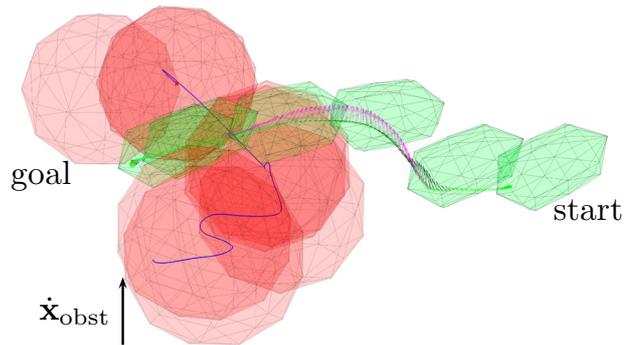


Fig. 11. Visualization of the human behavior (red spheres) and the robot end-effector (green object), which smoothly circumvents the dynamically moving human.



Fig. 12. The hot wire assembly.

a much more intuitive behavior of the robot in reality. On the left side the real posture of the OO (passive marker on hand) and the final configuration of the robot arm are illustrated. In the middle the 3D visualization before passing and on the right the one while passing the dynamic obstacle is shown. Additionally, the environment frames, their respective changes, and the progression of the obstacle forces are illustrated. The reorientation of the end-effector, which also has large impact on the motion of the forearm and secondary on the upper arm of the robot can be observed. A time evolution of the process is depicted in Fig. 11. The green end-effector hull is a partly robot representation, see Fig. 3 but not depict in Fig. 10, which consists of static charge particles for every surface element.

Please note that in general, if the obstacle approaches the end-effector too closely, PF forces generate a translational movement that prevents a possible collision.

Next, we outline how to use our method such that tactile exploration of an unknown object becomes possible.

4. TACTILE EXPLORATION: SIMULATIONS & EXPERIMENTS

In this section we describe how we use the Hybrid CF-PF approach together with a finite reactive planner to successfully solve a complex tactile exploration problem, the ‘‘Hot Wire’’ problem. For this we use the information about interaction forces measured from the LWR-III⁷ to

⁷ We use a nonlinear disturbance observer based on the generalized momentum of the manipulator together with the integrated joint torque sensors to obtain a good estimation of the external torques acting on the robot. These are then transformed into Cartesian wrenches via the Jacobian.

generate a virtual obstacle map. This is interpreted as an obstacle we want to circumvent. This enables us to explore complex 6D objects and, after the exploration phase is over, to perform motions that do not collide with the obstacles anymore. Figure 12 gives a perspective view on the problem and showcases multiple scenarios that were experimentally verified.

The ‘‘Hot Wire’’ exploration simulation is intended as a show-case that it is possible to explore a complex wire by tactile contact information only⁸. Furthermore, we use contact information to generate a tactile map of the object that subsequently can then be used for future motion generation.

4.1 Simulation

Approach The task to explore an object by interoceptive perception capabilities of a robot is still a major challenge in robotics. The combination of tactile exploration and local motion generation algorithms is a novel way to approach the problem. The underlying idea is to associate the robot with a virtual 6D end-effector object, which is guiding the motion in a virtual environment. This environment is incrementally built based on interaction forces/torques sensed during motion. In our implementation we use a torus as the virtual representation of our gripper to explore a wire labyrinth that is not known a-priori, see Fig. 12. We generate translational and angular velocities to control the robot end-effector directly in Cartesian impedance control.

The basic approach is to explore the physical object (wire) by using force and position measurements of the internal sensors of the robot and to construct an ‘‘avoidance map’’ based on a simple assumption on the unknown object geometry. Initially, we assume the wire to be an infinite straight object with certain radius. Furthermore, we assume that the robot initial pose is aligned with the beginning of the wire-type object to be explored. Based on the sensed tactile information as e.g. force, moment, position, and orientation we re-orientate this wire element incrementally.

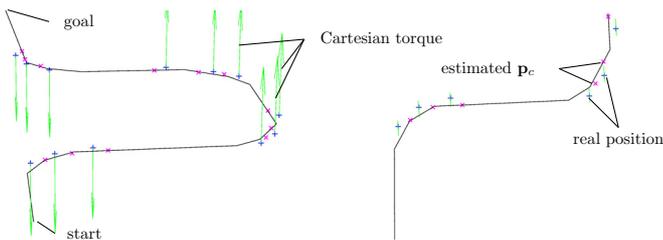


Fig. 13. Hot wire generation sketch.

The a-priori assumption on the orientation $\mathbf{x}_{\text{base}0}$ of the wire element is used to rotate the basic wire element object into the respective start pose.

For then calculating the rotation vector based on sensor input two approaches can be used. First, we may assume that sensed forces are directly aligned with the geometric normal of the object. This leads to aligning new elements orthogonally to this vector. Secondly, the rotation can be performed according to the measured torques (Fig. 13 green arrows). According to some initial testing, the torque approach appeared to be the more robust approach due to

non negligible friction effects during contact. The update rule for the new pose is

$$\mathbf{x}_{\text{baseNew}} = \mathbf{R}(f(M_{\text{ext},z}))\mathbf{x}_{\text{base}0}, \quad (18)$$

where f is a suitable function to limit the calculated rotation for high torques $M_{\text{ext},z}$ (the z -axis is the relevant rotation for the problem. We are currently working on the 6D ‘‘Hot Wire’’ problem). $\mathbf{x}_{\text{baseNew}}$ is the re-oriented wire direction. The re-orientation is then performed at the estimated wire center point \mathbf{p}_c of the contact (Fig. 13 magenta marks). This is obtained by the actual position of the torus $\mathbf{x}_{\text{torus}}$ (blue marks) and the external torque $M_{\text{ext},z}$.

$$\mathbf{p}_c = R_{\text{torus}} \frac{M_{\text{ext},z} \times \mathbf{x}_{\text{base}0}}{\|M_{\text{ext},z} \times \mathbf{x}_{\text{base}0}\|} + \mathbf{x}_{\text{torus}} - r_{\text{torus}} r_{\text{wire}} \frac{M_{\text{ext},z} \times \mathbf{x}_{\text{baseRot}}}{\|M_{\text{ext},z} \times \mathbf{x}_{\text{baseRot}}\|} \quad (19)$$

In the following \mathbf{p}_c and the new orientation $\mathbf{x}_{\text{baseRot}}$ are used to calculate the crossing point with the already existing wire element to redefine the internal model of the wire that is a polygon of line elements (Fig. 13 black line).

The motion generation approach for the problem uses only CFs (no damper and attractor), thus a free floating mass for translation and PFs with damping and no attractor for rotation (rotation energy is always decreasing: $D \propto E_{\text{rot}}$). There is no ‘‘long-term’’ rotational local minimum due to map building.

Reactive Planner To control the overall robot behavior during the task we designed a finite state reactive planner, which main chart is shown in Fig. 14. It consists of three main states:

- (1) the ‘‘avoid’’ state,
- (2) the ‘‘in collision’’ state, and
- (3) the ‘‘reached goal’’ state,

where the end-effector is stopped and waits for new commands. The ‘‘avoid’’ state consists of 5 sub-states that are self-explanatory: the ‘‘wait for start’’, ‘‘stop at this position’’, ‘‘acceleration forward’’, ‘‘acceleration backward’’, and the ‘‘explore’’ state. The latter tracks the current internal wire model and is left if external forces become too large (leads to a change into the sub-state ‘‘in collision’’). ‘‘in collision’’ determines the course of action during a collision: The end-effector is stopped and the contact forces observed for creating a new internal model. Then, by calculating the crossing point of the new line element the new destination for retraction is obtained. The avoidance map is not updated while moving back in order to avoid discontinuous virtual forces. During the ‘‘acceleration backward’’ and ‘‘constant velocity’’ sub-states the robot then moves back. If the task is completed (finished), the motion is stopped. Next the avoidance map is updated and the state change to the ‘‘avoid’’ is executed.

With the described behavior it is possible to fully explore the wire and afterwards navigate with the robot through the wire without causing any collisions anymore.

Results In order to judge the quality of the algorithm in simulation, we tested various randomly generated wires, of which two trials are depicted in Fig. 15. As already described the task is to explore an unknown generated wire (green) with a ring torus (green) by continuously deforming the initial iteration (red). The orientation of the ring torus is also marked with red and green marks. For both wire examples the resulting torques on the torus are calculated and this residual is integrated until a certain threshold is surpassed. If this incident occurs, the instantaneous difference of moments is associated with a contact moment and used to re-orientate the wire (red). As

⁸ With visual feedback this would be a rather trivial problem

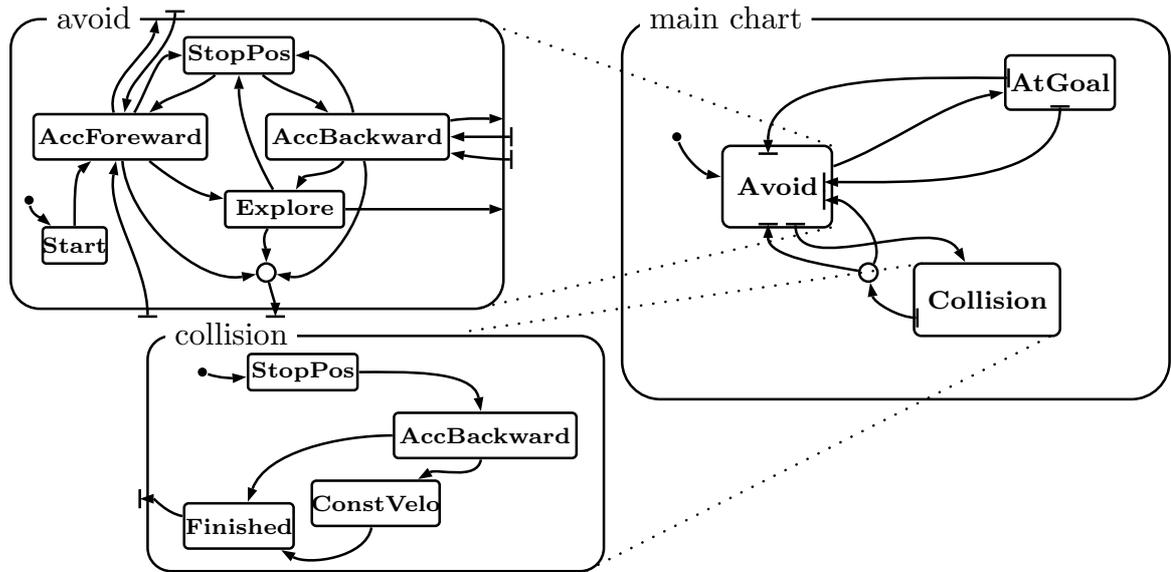


Fig. 14. Implementation of the hot wire State Flow planner.

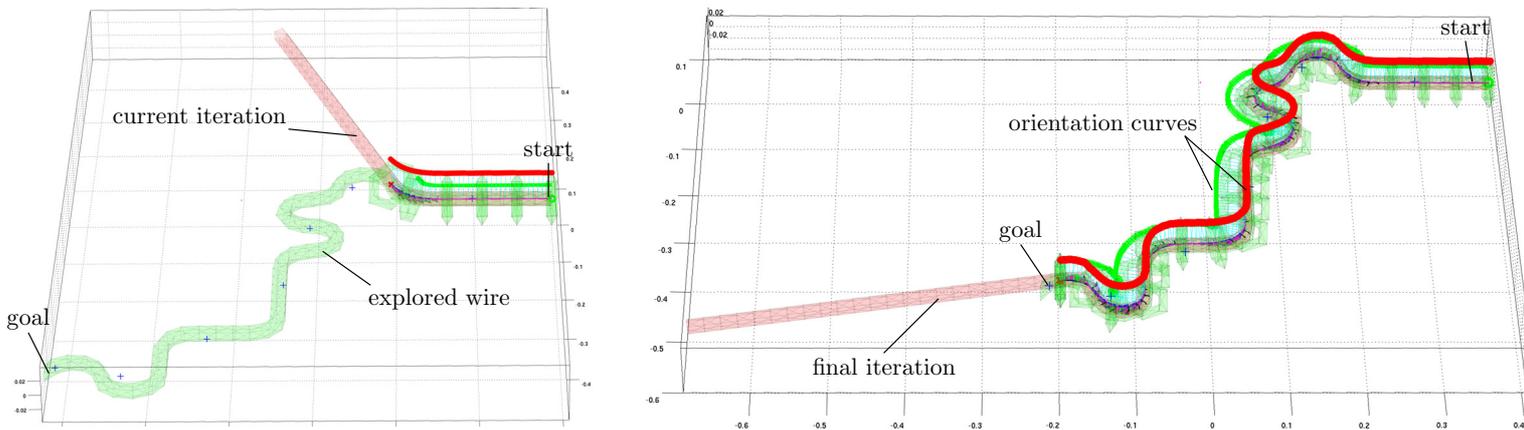


Fig. 15. Simulation of the hot wire exploration: started (left). Simulation hot wire exploration: finished (right).

shown in Fig. 15 (right) the entire object can be explored such that the full geometry of the object is reconstructed.

Next, we discuss the experimental implementation and results.

4.2 Experiment

The “Hot Wire” experiment is basically performed with the same approach as for simulation. The only difference is that the model of the collision avoidance simulation is of course different from the real wire. Therefore, we include measured robot states to observe the real-world. The control-loop is closed via the desired velocity of the robot. This means that the collision avoidance is used to command only the reference translational and angular velocities. However, in case a contact between gripper and the real wire occurs we use the real state measurements of the robot to generate the avoidance map as accurate as possible. The contact treatment is left to the local behavior of the impedance controller and sensing of external forces. In Fig. 16 the results of one of the hot wire experiments are depicted.

The virtual wire element is re-orientated at every contact point according to the true wire. At the time instant con-

tact is detected, the desired state is illustrated by a yellow torus. The real robot position is marked with a blue cross mark. The difference between real and desired state of the robot can be observed by the position difference. Magenta crosses mark the points of the estimated position of the wire center during contact. For the contact incidents also the contact torques (green arrows) are depicted (visible for the two left turns of the wire).

This experiment was performed with multiple wire configurations, see Fig 12, with the wire structure being changed also online. Examples of generated wire exploration maps are depicted in Fig. 17. On the left side the 2D movement (a projection of the true motion) is shown for a quite complex wire scenario. The “dead-end movement” represents the contact behavior with the wire. On the right hand side the force measurements are depicted. They are clearly not normal with respect to the imaginable real wire. This observation justifies the chosen approach without needing to estimate fiction properties.

5. CONCLUSION

In this paper we presented and analyzed a real-time collision avoidance scheme and discussed its applicability to

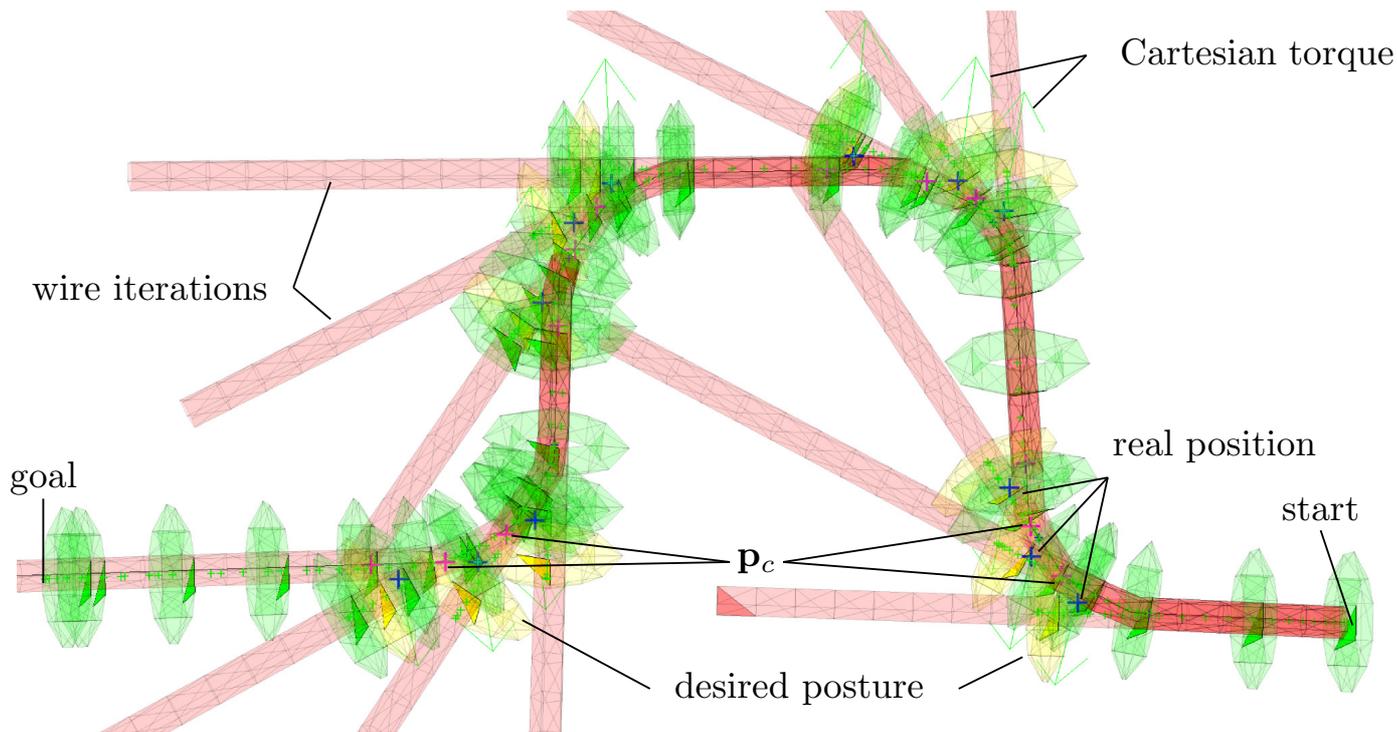


Fig. 16. Perspective view on the explored avoidance map and intermediate steps.

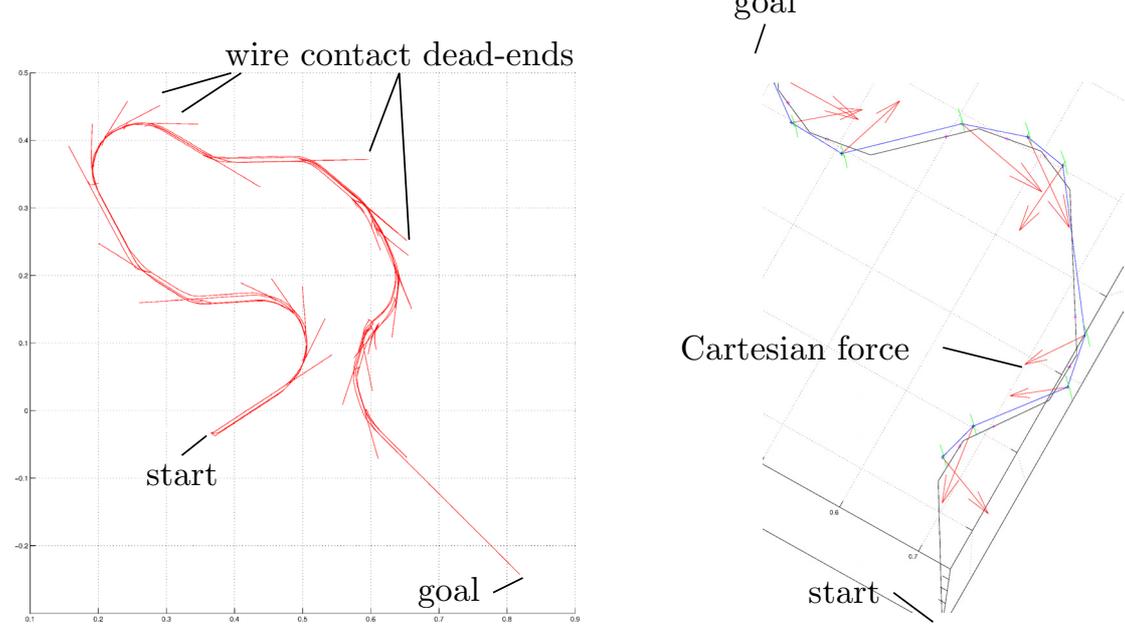


Fig. 17. Hot wire experiment, motion and force-moments.

articulated manipulators in dynamically changing environments for applications in physical Human-Robot Interaction. We extended the original definition of CFs and finally developed a hybrid CF-PF approach for 6D reactive Operational space real-time motion. We analyzed collision avoidance for static multi-object parcours and were able to show the performance for avoiding dynamically moving humans. Furthermore, we developed an algorithm for performing tactile exploration of complex planar 3D wire elements, whose structure is a-priori unknown. All stated problems were successfully solved experimentally and showcase the effectiveness of the designed algorithms.

Please find videos that show numerous results of the paper at www.safe-robots.com

ACKNOWLEDGEMENTS

We would like to thank Sven Parusel for his support during the experimental phase of this thesis. Furthermore, we would like express our gratitude to Tim Rokahr and Andrea Schwier for building the Hot Wire parcours.

REFERENCES

Alin Albu-Schäffer, Sami Haddadin, Christian Ott, Andreas Stemmer, Thomas Wimböck, and G. Hirzinger.

- The DLR lightweight robot lightweight design and soft robotics control concepts for robots in human environments. *Int. J. Rob. Res.*, 34(5):376–385, 2007.
- Alin Albu-Schäffer, Christian Ott, and Gerd Hirzinger. A unified passivity-based control framework for position, torque and impedance control of flexible joint robots. *Int. J. Rob. Res.*, 26(1):23–39, 2007. ISSN 0278-3649.
- Oliver Brock and Oussama Khatib. Elastic Strips: A Framework for Motion Generation in Human Environments. *Int. J. Robotics Research*, 21(12):1031–1052, 2002.
- Alessandro De Luca, Alin Albu-Schäffer, Sami Haddadin, and Gerd Hirzinger. Collision detection and safe reaction with the DLR-III lightweight manipulator arm. pages 1623–1630, 2006.
- Dirk Ebert and Dominik Henrich. Safe human-robot-cooperation: Image-based collision detection for industrial robots. In *In: IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1826–1831, 2002.
- Udo Frese and Holger Täubig. Modelling and calibration technique of laser triangulation sensors for integration in robot arms and articulated arm coordinate measuring machines. *Research Report RR-09-01*, 2009.
- Sami Haddadin, Alin Albu-Schäffer, Gerd Hirzinger, and Alessandro De Luca. Collision detection and reaction: a contribution to safe physical human-robot interaction. In *Proceedings and presented at IROS 2008*, pages 3356 – 3363, 2008. ISBN 978-1-4244-2057-5.
- O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. volume 2, pages 500–505, 1985.
- P. Ögren, N. Egerstedt, and X. Hu. Reactive mobile manipulation using dynamic trajectory tracking. In *ICRA*, pages 3473–3478, 2000.
- Bruno Siciliano and Oussama Khatib, editors. *Springer Handbook of Robotics*. Springer, 2008.
- L. Singh, H. Stephanou, and J. Wen. Real-time robot motion control with circulatory fields. pages 2737–2742, 1996.
- William T. Townsend and Jeffrey A. Guertin. Teleoperator slave - WAM design methodology. *Int. J. Rob. Res.*, 26(3):167–177, 1999.
- Shuguo Wang, Jin Bao, and Yili Fu. Real-time motion planning for robot manipulators in unknown environments using infrared sensors. *Robotica*, 25(2):201–211, 2007. ISSN 0263-5747.
- Yoshio Yamamoto and Xiaoping Yun. Coordinated obstacle avoidance of a mobile manipulator. In *ICRA*, pages 2255–2260, 1995.