Workshop on Robot Motion Planning:
Online, Reactive, and in Real-time
2012 IEEE/RSJ International Conference on
Intelligent Robots and Systems, IROS 2012
Vilamoura, Algarve, Portugal, October 7-12, 2012

Sensor-based trajectory generation for safe human-robot cooperation

Andrea Maria Zanchettin and Bakir Lacevic

***Abstract*— This paper presents a strategy for sensor-based trajectory generation in unstructured environments which guarantees the achievement of the goal position without incurring in local minima. The passivity of the closed-loop system renders this control scheme well-suited for human-robot cooperation, especially when the robot is supposed physically interact with humans. The given control law has been implemented and experimentally tested in a realistic scenario, demonstrating the effectiveness in driving the robot to a given configuration in a cluttered environment without any offline planning phase.**

I. INTRODUCTION AND MOTIVATIONS

Future paradigms in industrial robotics no longer require a physical separation between robotic manipulators and humans. Moreover, to optimize production, humans and robots will be expected to cooperate to some extent. In this scenario, involving a shared environment between humans and robots, common industrial robot controller might turn to be inadequate for this purpose. In order to obtain a natural and safe collaboration, robots will be equipped with sophisticated sensing devices and with human-aware control/planning capabilities.

In the literature many attempts in developing suitable robot reactions to unforeseen events have been presented by means of trajectory adaptation [4], [7] or modifications based on sensor readings, [2], [5]. However, in case of very unstructured environments, a better solution might be achieved with an advanced sensor-based motion and trajectory generation, rather than an online modification of an offline planned path. The aim of this research is to provide tools to overcome the current limitations in off-the-shelf robot controller and propose an online trajectory generator capable of understanding the environment and of computing a sensor-based trajectory to let the robot perform a prescribed task with a suitable level of safety.

A preliminary version of this work is discussed in [8], while paper complements the previous one by adding more details on the actual implementation of the sensor-based trajectory generation and discussing the outcome of more realistic experiments.

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 - Challenge 2 - Cognitive Systems, Interaction, Robotics - under grant agreement No 230902 - ROSETTA.

A.M. Zanchettin is with Politecnico di Milano, Dipartimento di Elettronica e Informazione, Piazza L. Da Vinci 32, 20133, Milano, Italy (email: zanchettin@elet.polimi.it).

B. Lacevic is with the Faculty of Electrical Engineering, University of Sarajevo, Sarajevo, Bosnia and Herzegovina (email: bakir.lacevic@etf.unsa.ba).

II. BACKGROUND MATERIAL

In this Section, the concept of danger field [6] is briefly outlined. Basically, the danger field is a scalar quantity that captures how much is a specific state of the robot (position and velocity) dangerous with respect to a generic point in the workspace. The intuition behind is that the danger field decreases with the distance from the robot whereas it increases with the robot's velocity, particularly if the robot moves towards the location where the field is computed at. For a simple case of a point robot located at $\boldsymbol{r_s} \in \mathbb{R}^3$, moving with the velocity $\boldsymbol{v_s} \in \mathbb{R}^3$, the elementary danger field at the position $\boldsymbol{r_j} \in \mathbb{R}^3$ could be defined as $DF_e = SDF_e + DDF_e$, where

$$SDF_e = \frac{k_1}{\|\boldsymbol{r_j} - \boldsymbol{r_s}\|^{\lambda_1}}, \tag{1}$$

$$DDF_e = \frac{k_2 \|\boldsymbol{v_s}\|^{\lambda_2} (1 + \cos\angle(\boldsymbol{r_j} - \boldsymbol{r_s}, \boldsymbol{v_s}))}{\|\boldsymbol{r_j} - \boldsymbol{r_s}\|^{\lambda_3}}, \tag{2}$$

and SDF_e and DDF_e are the elementary static and kinetic danger fields respectively, $k_1, k_2, \lambda_1, \lambda_2, \lambda_3$ being positive parameters[1]. The elementary danger field can be generalized to its cumulative version that captures the position and velocity of the robot's i-th link by performing a path integration along the straight line that represents the wire model of the link:

$$DF = \int_0^1 SDF_e(s)ds + \int_0^1 DDF_e(s)ds. \tag{3}$$

For a robot with n links, the cumulative danger field induced at the locations of interest $r_j,\ j = 1, \ldots, n_{obst}$ (e.g., the relevant positions of obstacles) can be expressed as:

$$\begin{aligned} DF = SDF + DDF = \sum_{j=1}^{n_{obst}} \sum_{i=1}^{n} \int_0^1 \frac{k_1 ds}{\|\boldsymbol{r_j} - \boldsymbol{r_{i,s}}\|^{\lambda_1}} + \\ + \sum_{j=1}^{n_{obst}} \sum_{i=1}^{n} \int_0^1 \frac{k_2 \|\boldsymbol{v_{i,s}}\|^{\lambda_2} \rho_{i,j,s}}{\|\boldsymbol{r_j} - \boldsymbol{r_{i,s}}\|^{\lambda_3}} ds, \end{aligned} \tag{4}$$

where $\rho_{i,j,s} = 1 + \cos\angle(\boldsymbol{r_j} - \boldsymbol{r_{i,s}}, \boldsymbol{v_{i,s}})$. Knowing $\boldsymbol{r_{i,s}}$ from the forward kinematics and letting $\boldsymbol{v_{i,s}} = \boldsymbol{J_{i,s}}\dot{\boldsymbol{q}}$, where $\boldsymbol{J_{i,s}}$ represents the Jacobian at point $\boldsymbol{r_{i,s}}$ on the manipulator, the cumulative danger field becomes a function of the configuration $\boldsymbol{q}$ and its time derivative $\dot{\boldsymbol{q}}$. Figure 1 shows volumetric representation of the danger field induced by a 6 DOF robotic manipulator. Notice the typical onion-like charecteristic of the danger-field very similar to the

[1]Note that this is a slight generalization of the danger field with respect to [6].

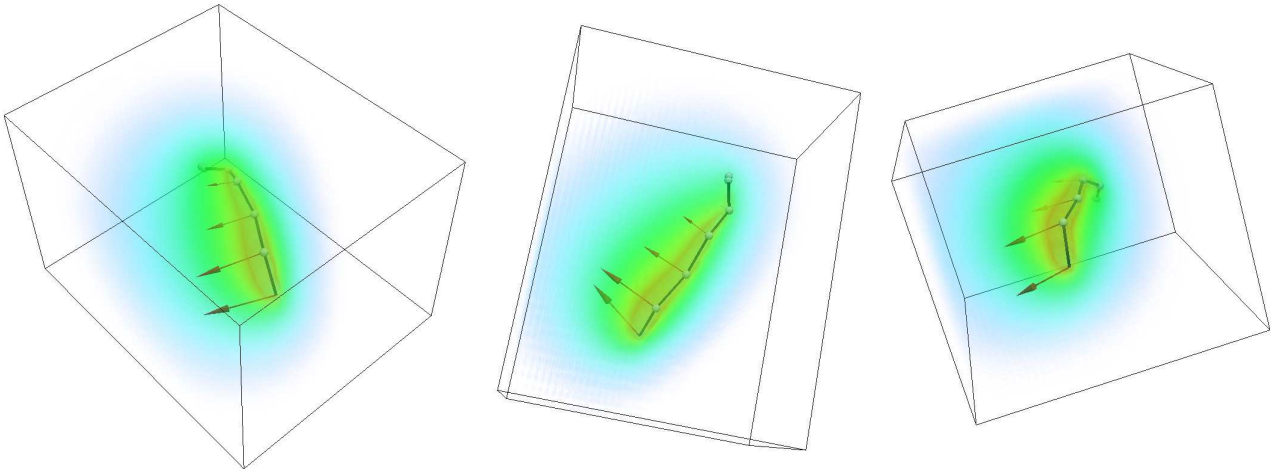

Fig. 1. Danger field around a 6 DOF manipulator: three different perspectives of the same robot state. The velocities of the link endpoints are indicated.

concept of minimum separation distance, developed in [1]. For the computational aspects of the danger field, the reader is referred to [6].

III. IMPLEMENTATION

The sensor-based module for online trajectory generation described in this paper works within a real-time loop of 4 ms, the same as the low-level axis control and consists of three blocks:

- a module for trajectory generation, communicating with the robot controller via a real-time Ethernet connection, see [3], and providing it joint references;
- a module for task description, implementing a state-machine and also responsible of temporary task suspension in case of physical cooperation initiated by the human operator;
- an interface to the workspace surveillance sensor, monitoring and tracking the human or any obstacle and providing a synthetic representation of them to the trajectory generation module.

The three-blocks architecture and their connections are depicted in Fig. 2. In the following, each of the three modules

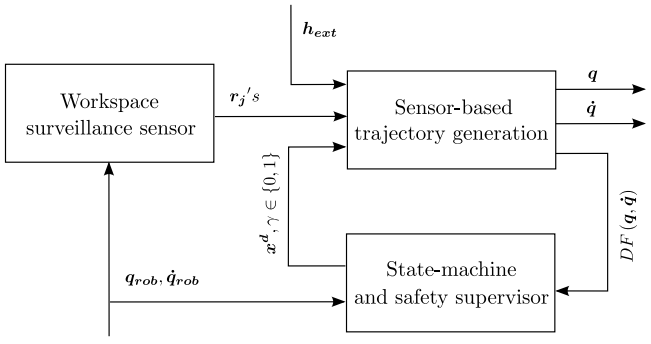

Fig. 2. Components of the trajectory generation algorithm

is detailed.

A. Sensor-based trajectory generation

For a given desired Cartesian position/orientation $\boldsymbol{x_d}$ computed by the state-machine, a trajectory in the configuration, i.e. $\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t)$ space is generated in real-time depending on sensor (proximity and force/torque) readings. In particular, the trajectory is computed by integrating the following dynamic system:

$$\begin{aligned}\ddot{\boldsymbol{q}} =& \boldsymbol{J}^T\left(\gamma \boldsymbol{K_P}(\boldsymbol{q})\,\boldsymbol{e} + \boldsymbol{h_{ext}}\right) - \boldsymbol{K_D}\dot{\boldsymbol{q}} \\ &+\alpha(\boldsymbol{q})\,\boldsymbol{F}(\boldsymbol{\mu},\dot{\boldsymbol{q}})\,\boldsymbol{\mu} - \beta(\boldsymbol{q},\dot{\boldsymbol{q}})\left(\frac{\partial DF}{\partial \dot{\boldsymbol{q}}}\right)^T\end{aligned} \tag{5}$$

where

$$\boldsymbol{\mu} = -\frac{1}{2}\left[\|\boldsymbol{\nabla} U\|\,\|\boldsymbol{\nabla} SDF\| + \boldsymbol{\nabla} U\,(\boldsymbol{\nabla} SDF)^T\right](\boldsymbol{\nabla} SDF)^T \tag{6}$$

is an evasive action preventing the robot hitting obstacles (or humans) in its working space,

$$\begin{aligned}\boldsymbol{F}(\boldsymbol{\mu},\dot{\boldsymbol{q}}) &= \begin{cases}\boldsymbol{I_n}, & \text{if } \boldsymbol{\mu}^T\dot{\boldsymbol{q}} \le 0\\ \boldsymbol{P}_\perp(\dot{\boldsymbol{q}}), & \text{otherwise}\end{cases}\\ \boldsymbol{P}_\perp(\dot{\boldsymbol{q}}) &= \boldsymbol{I_n} - \frac{\dot{\boldsymbol{q}}\dot{\boldsymbol{q}}^T}{\|\dot{\boldsymbol{q}}\|^2}\end{aligned} \tag{7}$$

is a weighting matrix, while $\alpha(\boldsymbol{q}), \beta(\boldsymbol{q},\dot{\boldsymbol{q}}), \gamma$ are positive weighting parameters and $\boldsymbol{K_P}, \boldsymbol{K_D}$ are definite positive matrices. The overall system is depicted in Fig. 3. The dynamic system (5) has different properties such as guaranteed goal achievement with proved absence of local minima, enforced passivity mapping between external wrenches $\boldsymbol{h_{ext}}$ and joint velocities $\dot{\boldsymbol{q}}$ and others. The reader is referred to [8] for a more rigorous explanation of these concepts.

B. Workspace surveillance and geometric representation of obstacles

For workspace surveillance a range camera (MICROSOFT KINECT) with the OPENNI drivers have been selected. The output of the sensor consists of a segment representation

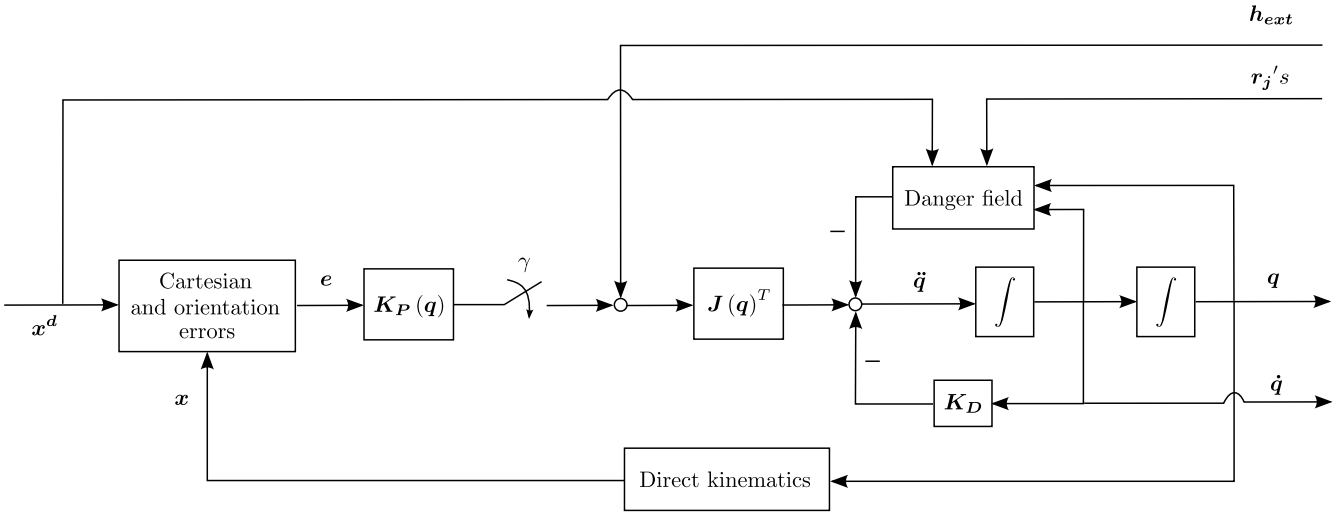

Fig. 3. Block diagram of the sensor-based trajectory generation

of the human silhouette capturing the position of anatomical points along the body (head, shoulder, elbow, wrist, hip, etc.), see Fig. 4.

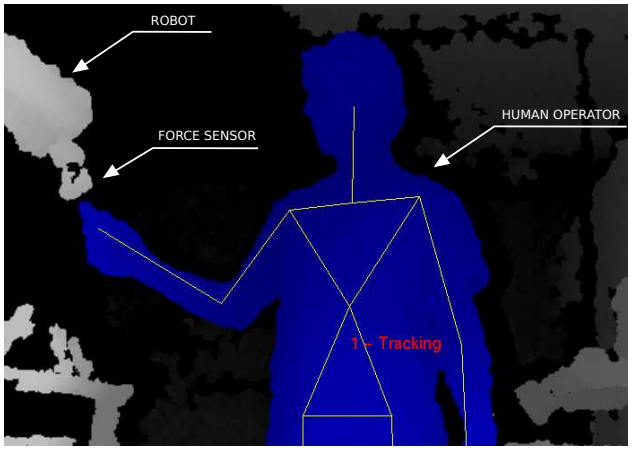

Fig. 4. Robot's workspace as seen from the surveillance range camera

A set of interesting points where to compute the danger field should be selected. Therefore, a simple and fast algorithm compatible with the representation of obstacles provided by the KINECT is described in the following.
While the robot is described as a chain of segments, a generic obstacle (the human operator in this work) can be decomposed in a set of segments, spheres or represented by a more generic triangular meshed surface. Algorithmic primitives based on quadratic optimization have been developed to estimate the closest points of each obstacles to the robot. For example, a segment $\boldsymbol{\mathcal{S}}$ can be parameterized by means of a vertex $\boldsymbol{P}$ and a vector $\boldsymbol{d}$ such that all the points belonging to the segment can be written as follows:

$$\boldsymbol{\mathcal{S}}(t) = \boldsymbol{P} + t\boldsymbol{d},\ 0 \leq t \leq 1 \tag{8}$$

The minimum distance between two segments, hence the closest point to each link of the robot, can be obtained by solving the following constrained optimization problem

$$\begin{aligned} &\min_{t_1,t_2} \|\boldsymbol{\mathcal{S}_1}(t_1) - \boldsymbol{\mathcal{S}_2}(t_2)\|^2 \\ &\text{subject to } 0 \leq t_1, t_2 \leq 1 \end{aligned} \tag{9}$$

On the other hand, a triangle $\boldsymbol{\mathcal{T}}$ is parameterized by means of a vertex $\boldsymbol{V}$ and two vectors $\boldsymbol{e_0}, \boldsymbol{e_1}$. This way all the points belonging to the triangle can be written as follows:

$$\boldsymbol{\mathcal{T}}(u,v) = \boldsymbol{V} + u\boldsymbol{e_0} + v\boldsymbol{e_1} \tag{10}$$

subject to the following constraints

$$\begin{aligned} 0 \leq u \leq 1, \\ 0 \leq v \leq 1, \\ u + v \leq 1 \end{aligned} \tag{11}$$

Therefore the closest point on the robot link can be computed by simply solving the following optimization problem:

$$\begin{aligned} &\min_{t,u,v} \|\boldsymbol{\mathcal{S}}(t) - \boldsymbol{\mathcal{T}}(u,v)\|^2 \\ &\text{subject to } 0 \leq u, v, t \leq 1, u + v \leq 1 \end{aligned} \tag{12}$$

In this case study, where only the human operator represents an obstacle, its body is decomposed into four segments (right and left arm and forearm), one triangle (torso) and one sphere (head). The algorithm described so far to compute the closest point to the robot is then applied providing a set of points where the trajectory generation algorithm will compute the danger field. The depicted scenario is sketched in Fig. 5.

C. State-machine and safety supervisor

A state-machine has been implemented to both monitor the execution and eventually suspend the task when the user gets too close to the robot. During the normal execution, the state-machine communicates with the trajectory generation module sending the sequence of Cartesian position/orientation references. When the value of the danger field exceeds a prescribed threshold DF_{thr}^{sup}, the task is suspended by setting $\gamma = 0$. In this situation the trajectory generation is still

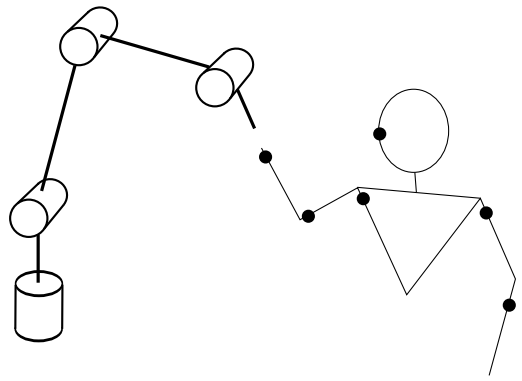

Fig. 5. Points on the human body where danger field will be computed

active but without any reference position/orientation. This is meant to allow the user to manually guide the robot motion, e.g. to teach the robot new positions. When the

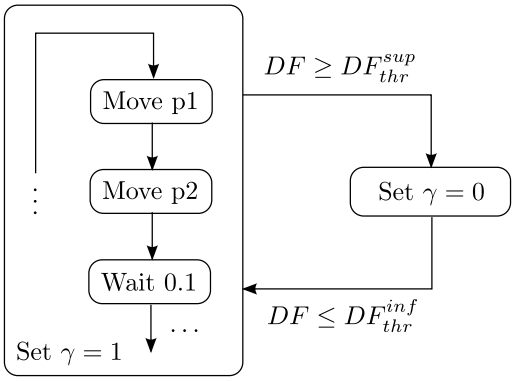

Fig. 6. State-machine governing task execution and suspension

danger field assumes values lower than another threshold $DF_{thr}^{inf} < DF_{thr}^{sup}$, the main task is resumed from the point where it was suspended by setting $\gamma = 1$. Figure 6 sketches the task suspension/resumption mechanism.

IV. EXPERIMENTS

As a validation of the proposed control strategy, some experimental tests have been carried out on an industrial manipulator. The 6 axes ABB IRB-140 robot with 6 kg payload was used for this purpose. The manipulator is equipped with an ATI force/torque sensor mounted on the robot end-effector and interfaced with the controller. All the sensors are acquired and processed within an external real-time LINUX PC interfaced with the ABB IRC5 industrial robot controller using a communication link developed, see [3].

A realistic industrial scenario has been arranged in order to resemble a typical machine tending task: the robot handles a workpiece from a storage station and transports it to a position located on the other side of its workspace.

The operator, regarded as a moving obstacle, is able at any time to enter the working area of the robot for inspection and for this reason a proper safety action should be guaranteed, possibly without interrupting the production. During the

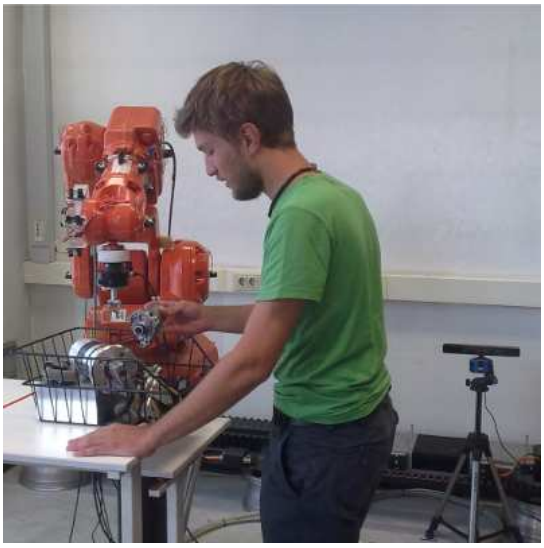

Fig. 7. Human-robot coexistence ($\gamma = 1$)

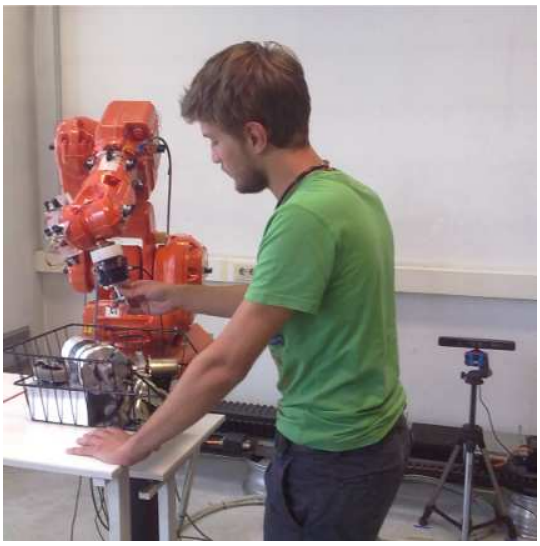

Fig. 8. Human-robot physical cooperation ($\gamma = 0$)

experiment, the same production cycle has been repeated 3 times. During the last two repetitions the human operator enters the scene and walks towards the pick-up station approximately at time instants $t = 25s$ and $t = 70s$. This is confirmed by the profile of the danger field, see Fig. 9, which captures a more dangerous situation due to the vicinity of the human. Correspondingly the robot first tries to reduce the speed and then, since the danger field exceeds the desired threshold, suspends the task to allow the physical cooperation with the operator (this situation is highlighted with gray bands in the Figures). The accompanying video shows the execution of the experiment.

V. CONCLUSIONS

This paper complements [8] detailing the implementation of a newly conceived passivity-based control scheme for robotic manipulators in cluttered environments. The control

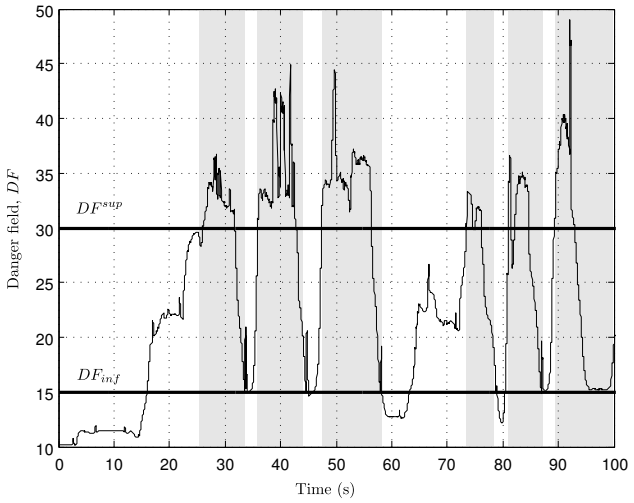

Fig. 9. Profile of the danger field and safety thresholds

law has been experimentally verified in a scenario involving an industrial manipulator physically cooperating with a human operator, demonstrating the possibility to safely move the robot in given configurations without any offline planning phase.

REFERENCES

[1] *ISO/TS 15066 "Robots and Robotic Devices - Collaborative industrial robots"*.

[2] L. Bascetta, G. Magnani, P. Rocco, R. Migliorini, and M. Pelagatti. Anti-collision systems for robotic applications based on laser time-of-flight sensors. In *Advanced Intelligent Mechatronics (AIM), 2010 IEEE/ASME International Conference on*, pages 278 –284, july 2010.

[3] A. Blomdell, I. Dressler, K. Nilsson, and A. Robertsson. Flexible application development and high-performance motion control based on external sensing and reconfiguration of abb industrial robot controllers. In *Proc. ICRA 2010 Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications*, Anchorage, AK, jun 2010.

[4] O. Brock and O. Khatib. Elastic strips: A framework for motion generation in human environments. *International Journal of Robotics Research*, 21(12):1031–1052, 2002.

[5] S. Haddadin, R. Belder, and A. Albu-Schaffer. Dynamic motion planning for robots in partially unknown environments. In *IFAC World Congress 2011, IFAC 2011*, pages 6842–6850, 2011.

[6] B. Lacevic and P. Rocco. Kinetostatic danger field - a novel safety assessment for human-robot interaction. In *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010*, pages 2169–2174, 2010.

[7] S. Quinlan and O. Khatib. Elastic bands: connecting path planning and control. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 802–807, may 1993.

[8] A.M. Zanchettin, B. Lacevic, and P. Rocco. A novel passivity-based control law for safe human-robot coexistence. In *IEEE/RSJ 2012 International Conference on Intelligent Robots and Systems, IROS 2012*, 2012.