Control and Obstacle Collision Avoidance Method applied to Human-robot Interaction

5 authors, including:

Antonio C. Leite
Pontifícia Universidade Católica do Rio de Janeiro
35 PUBLICATIONS   **164** CITATIONS

SEE PROFILE

Pål Johan From
Norwegian University of Life Sciences (NMBU)
94 PUBLICATIONS   **433** CITATIONS

SEE PROFILE

Fernando Lizarralde
Federal University of Rio de Janeiro
118 PUBLICATIONS   **1,391** CITATIONS

SEE PROFILE

Liu Hsu
Federal University of Rio de Janeiro
295 PUBLICATIONS   **4,272** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project   Robotics Laboratory View project

Project   TIBA - Tankette for Intelligent BioEnergy Agriculture View project

Control and Obstacle Collision Avoidance Method applied to Human-Robot Interaction

Antonio C. Leite, Thiago B. Almeida-Antonio, Pål J. From, Fernando Lizarralde and Liu Hsu

Abstract— In this work, we present a control and obstacle collision avoidance method for redundant robot manipulators operating in partially structured environments in the presence of humans. The control algorithm is based on the concept of artificial potential fields and it uses the pseudo-inverse of the Jacobian matrix with a weighting factor for the mechanical joint limits, taking advantage of the robot redundancy for the purpose of obstacle avoidance and control goal achievement. The detection algorithm uses a depth sensor based on the structured light to obtain a 2-1/2-D description of the surroundings from a point cloud. Repulsive fields are created around the detected obstacles, allowing for the robot to perform the task of interest without collisions. A filtering methodology based on geometric elements is presented to filter the RGB-D scene captured by the depth sensor, eliminating the robot body and the obstacles located outside its workspace. Experimental results, obtained with a Motoman DIA10 robot and a Microsoft Kinect^TM, illustrate the feasibility of the proposed scheme.

I. INTRODUCTION

Nowadays, one of the main challenges in the robotics area consists of dealing with the interaction between humans and robots, enabling them to work together in a natural, effective and safe manner. In this sense, human-robot interaction applications have been developed to establish different types of cooperative, collaborative or coordinated behaviors. To perform these interaction tasks, the robot must be aware of its surrounding and move without jeopardizing the safety of operators around it. Thus, the robot has to be able to avoid collisions with any obstacles located within its workspace and achieve the control goals simultaneously [1].

A number of motion planning and obstacle avoidance methods have been proposed over the years with the purpose of generating a collision-free path or trajectory among two or more points of interest [2]. However, as the current proposal is to develop technical resources necessary to foster the interaction between robots and humans in a dynamic and unstructured environment, the use of off-line techniques may not be suitable since most of them consider that the modeling of the robot and the task scenario are fully known and free of uncertainties. An alternative solution consists of using on-line or reactive techniques which are able to deal with

the presence of unknown and dynamic objects as well as uncertain environments [3].

A well-known obstacle avoidance approach, well suited for dynamic environments, is based on the concept of artificial potential fields, where attractive and repulsive virtual fields are generated in the robot workspace and, thus, the robot end effector can move towards a desired point, and at the same time avoids colliding with obstacles in the environment [4], [5]. Another interesting approach is to employ circulatory fields, which act to circumvent the robot path around the obstacles rather than just repel the robot, as is usual in the potential fields [6]. Circulatory fields and potential fields can also be combined in order to take the advantages of both approaches [7]. In these works, the motions of the obstacles are assumed to be known a priori, ignoring the need for external sensing components or detection equipment.

An obstacle detection method based on the information obtained from a Microsoft Kinect^TM is proposed in [8] for collision avoidance in interaction tasks between humans and robots. A calibration method for Kinect and a comparative method to distinguish robots and obstacles, when they share the same working place is presented in [9]. Following this trend, some works devoted to the navigation problem of mobile robots in indoor environments have been developed by using the Kinect for obstacle detection instead of using a detection approach based on 3D laser scanner or stereo cameras [10], [11]. Still in the human-robot interaction framework, some authors have used the Kinect to recognize different body gestures and generate a visual interface or assist in the execution of collaboration tasks, whose phases are conducted by gestures and voice commands [12], [13].

In this work, we present a control and obstacle collision avoidance method for redundant robot manipulators operating in human-populated environments. The control algorithm is based on the concept of artificial potential fields and it uses the pseudo-inverse of the Jacobian matrix with a weighting factor for the mechanical joint limits, taking advantage of the redundancy characteristic of the robot for obstacle avoidance and control purposes. A depth sensor, based on the structure light technology, is used to provide a 2-1/2-D geometrical description of the environment from a RGB-D point cloud. Repulsive fields are generated around the detected obstacles, allowing the robot to perform the task of interest without collisions and ensuring the safety of the operators. A filtering methodology based on geometric elements is devised to filter the point cloud, removing first the points related to the robot joints and links, and after the points located outside the robot workspace. Experimental results, obtained with

Antonio C. Leite, Thiago B. de Almeida-Antonio, Fernando Lizarralde, and Liu Hsu are with the Department of Electrical Engineering, COPPE/Federal University of Rio de Janeiro (UFRJ), P. O. Box 68504, 21941-972 RJ, Rio de Janeiro, Brazil, e-mail: toni@coep.ufrj.br, fernando@coep.ufrj.br, liu@coep.ufrj.br

Pål J. From is with the Department of Mathematical Sciences and Technology, Norwegian University of Life Science (NMBU), P. O. Box 5003, NO-1432 Ås, Norway, e-mail: pal.johan.from@nmbu.no

a Motoman DIA10 robot and a natural interface device Microsoft Kinect^{TM}, are presented to illustrate the feasibility of the proposed scheme.

II. PROBLEM FORMULATION

Objective: In this work, we consider that a redundant robot manipulator has to perform a particular task in a partially structured environment, in the presence of humans working and moving freely within the robot workspace (Fig. 1).

Task: The task of interest consists of moving the robot end effector from an initial point P_0 to a final point P_f, following a reference trajectory $r(t)$ and keeping a desired constant orientation.

Assumption: Here, we assume that the task is previously planned meaning that the reference trajectory $r(t)$ is known, free of singularities and generated by a trajectory planning algorithm. We also assume that the geometrical description of the working place is partially known.

Proposal: The geometrical description of the environment is obtained by using a natural interface device (NID) equipped with a depth sensor, which is mounted in front of the robot. The depth sensor uses the structured light technology registering the whole scene in the form of a 3D point cloud. Thus, static and dynamic objects, including people, can be properly identified and the robot can perform the task safely without colliding with these obstacles.
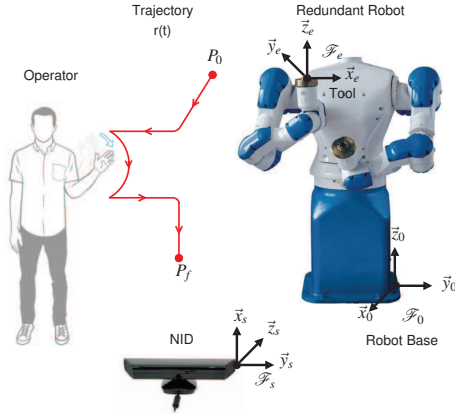

Fig. 1.   Problem formulation for a simple human-robot interaction.

Since most of industrial robots have an internal velocity control loop to directly drive the joints, a purely kinematic control law based on speed commands can be implemented to ensure the satisfactory performance for the robot motion during the task execution [14].

III. KINEMATICS: MODELING AND CONTROL

Consider the kinematic control problem for a robot manipulator with n degrees of mobility. In this context, when the robot has high gear reduction ratios and low speeds or slow accelerations are employed to perform the task, the robot motion can be described simply by:

$$\dot{\theta}_i = u_i, \qquad i = 1, \cdots, n, \qquad (1)$$

where $\theta_i \in \mathbb{R}^n$ and $\dot{\theta}_i \in \mathbb{R}^n$ are the vectors of angular position and angular velocity of the i-th joint respectively, $u_i \in \mathbb{R}^n$ is the velocity control signal applied to the motor driver of the i-th joint and n is the number of joints.

From the kinematic modeling, the joint space variables are related to the Cartesian space variables by means of the following *forward kinematics* map and its differential equation:

$$p = k(\theta), \qquad \dot{p} = J_p(\theta)\dot{\theta}, \qquad (2)$$

where $k(\theta)$ is an m-dimensional vector function, nonlinear in general, $J_p(\theta) = (\partial k/\partial \theta) \in \mathbb{R}^{3 \times n}$ is the *analytical Jacobian* of the position term, and $p, \dot{p} \in \mathbb{R}^3$ denote the position and the linear velocity of the robot end effector, expressed in the base frame $\mathscr{F}_0$ respectively.

Now, consider that the orientation of the robot end effector can be represented by the *unit quaternion* $q \in \mathbb{H}^1$, corresponding to the rotation matrix $R \in SO(3)$, describing the orientation of the end-effector frame $\mathscr{F}_e$ with respect to the base frame $\mathscr{F}_0$. The unit quaternion representation can be given in terms of a four-dimensional vector $q = [\, q_s \ q_v^\mathsf{T} \,]^\mathsf{T}$, where $q_s \in \mathbb{R}$ is the scalar part and $q_v \in \mathbb{R}^3$ is the vector part, subject to the unit norm constraint $q_s^2 + q_v^\mathsf{T} q_v = 1$. The so-called *quaternion propagation* rule relates the time-derivative of the unit quaternion $\dot{q} \in \mathbb{H}$ with the angular velocity of the robot end effector $\omega \in \mathbb{R}^3$ as:

$$\dot{q} = \frac{1}{2} E(q)\omega, \qquad E(q) = \begin{bmatrix} -q_v^\mathsf{T} \\ q_s I - Q(q_v) \end{bmatrix}, \qquad (3)$$

where $Q(\cdot): \mathbb{R}^3 \mapsto so(3)$ denotes the skew-symmetric matrix operator and $J_r(q) = 2E^\mathsf{T}(q) \in \mathbb{R}^{3 \times 4}$ is the well-known *representation Jacobian* [14].

The *differential kinematics* equation provides the relationship between the joint velocities vector and the corresponding linear and angular velocity of the robot end effector as:

$$v = \begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = \begin{bmatrix} J_p(\theta) \\ J_o(\theta) \end{bmatrix} \dot{\theta} = J(\theta)\dot{\theta}, \qquad (4)$$

where $J(\theta) \in \mathbb{R}^{6 \times n}$ is the *geometric Jacobian* of the robot manipulator. Notice that, without loss of generality, the unit quaternion can be obtained in terms of the manipulator joint angles as $q = h(\theta)$, where $h(\cdot)$ is a vector function, nonlinear in general. Thus, taking into account (3) and (4) we have that: $J_o(\theta) = J_r(q)(\partial h(\theta)/\partial \theta)$. Therefore, considering the kinematic control approach (1) and from (4), we obtain the following control system:

$$v = \begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = J(\theta)u. \qquad (5)$$

For the case where the Jacobian matrix is rectangular ($n > 6$) the velocity control signal $u \in \mathbb{R}^n$ is given by:

$$u(t) = J^\dagger(\theta)v_c, \qquad v_c = \begin{bmatrix} v_p \\ v_o \end{bmatrix}, \qquad (6)$$

where $J^\dagger = J^\mathsf{T}(JJ^\mathsf{T})^{-1}$ is the right pseudo-inverse of J and $v_c \in \mathbb{R}^6$ is a Cartesian control signal, composed of the

---

[1]The symbol $\mathbb{H}$ denotes the unit quaternion group satisfying the algebra of quaternion [14].

position and orientation control signals, v_p and v_o, which are independently designed to command the end-effector pose. It is worth mentioning that the control signal (6) locally minimizes the norm of the velocities of the joints, provided that the robot kinematics is *known* and $v_c(t)$ does not lead the robot to *singular configurations*, where the Jacobian matrix has deficient rank. The failure of this last condition is a fairly open-problem in robotics and other correlated areas, and it has been widely discussed in the control literature [14].

A. Avoiding the Joint Limits

Here, we will show that the velocity control signal (6) can be modified to avoid that the robot motions reach the mechanical limits of the joints. Considering the kinematic control problem for a non-redundant robot manipulator, the following joint-space control law based on the weighted pseudo-inverse of the Jacobian matrix can be used [15]:

$$u_w = W^{-1} J^{\mathsf{T}} \left(J W^{-1} J^{\mathsf{T}} \right)^{-1} v_c = J_w^\dagger v_c , \qquad (7)$$

where $W \in \mathbb{R}^{n \times n}$ is a weighting matrix, symmetric and positive definite and $J J_w^\dagger = I$. Notice that, the relation (7) holds provided that $v_c(t)$ does not drive the robot through singular configurations.

The matrix W is designed so that when a joint of the robot reach its limit, the diagonal elements tend to infinity. This fact ensures that the joint velocities tend to zero in the neighborhood of a mechanical limit. The diagonal elements of W^{-1} approach to zero as the joint reaches its limit, and are equal to 1 when the joint angle is in the middle of its range. Thus, each diagonal element of the matrix W is defined as:

$$W_{i,i} = 1 + |\partial H(\theta) / \partial \theta_i| , \qquad (8)$$

with

$$\frac{\partial H(\theta)}{\partial \theta_i} = \frac{(\theta_{i_{max}} - \theta_{i_{min}})^2 (2\theta_i - \theta_{i_{max}} - \theta_{i_{min}})}{4 (\theta_i - \theta_{i_{max}})^2 (\theta_i - \theta_{i_{min}})^2} , \qquad (9)$$

where $\theta_{i_{max}}$ and $\theta_{i_{min}}$ are the upper and lower bounds of the *i*-th joint respectively. In fact, when $\theta_i \to \theta_{i_{max}}$ or $\theta_i \to \theta_{i_{min}}$ we have that $|\partial H / \partial \theta_i| \to \infty$, and when $\theta_i = (\theta_{i_{max}} + \theta_{i_{min}})/2 = \theta_i^*$ implies that $|\partial H / \partial \theta_i| = 0$.

B. Position Control

Here, we consider the position control problem for a robot manipulator. In this context, we assume that the control goal for a given task consists in following a time-varying desired position $p_d(t)$ from the current position p. The control goal can be simply described as:

$$p \to p_d(t), \qquad e_p = p_d(t) - p \to 0, \qquad (10)$$

where $e_p \in \mathbb{R}^3$ is the position error. From (5) and (7), we obtain $\dot{p} = v_p$ and, consequently, the following position error equation $\dot{e}_p = \dot{p}_d - v_p$. Considering a position control law based on a proportional action and a feed-forward term:

$$v_p = K_p e_p + \dot{p}_d , \qquad (11)$$

with $K_p = k_p I$ being the position gain matrix, the position error dynamics is governed by $\dot{e}_p + K_p e_p = 0$. Thus, by choosing k_p as a positive scalar, the closed-loop control system is exponentially stable and, consequently, $\lim_{t \to \infty} e_p(t) = 0$.

C. Orientation Control

Consider the orientation control problem for a robot manipulator. In this context, we assume that the control goal for a given task consists in regulating the current orientation R to a desired constant orientation R_d. The control goal can be simply described as:

$$R \to R_d , \qquad R_q = R R_d^{\mathsf{T}} \to I , \qquad (12)$$

where $R_q \in SO(3)$ is the orientation error matrix expressed in the base frame $\mathscr{F}_0$. In order to design a orientation control system based on the *unit quaternion formulation*, a suitable orientation error should be defined. First, let q and q_d be the unit quaternions associated with the rotation matrices R and R_d respectively. Now, let $e_q = [e_{qs} \ e_{qv}^{\mathsf{T}}]^{\mathsf{T}} \in \mathbb{H}$ be the unit quaternion error, associated with R_q, where e_{qs} and e_{qv} are the scalar and vector parts, subject to the unit norm constraint. The orientation error can be defined in terms of the unit quaternion error as follows $e_q = q * q_d^{-1}$, where "$*$" denotes the quaternion product operator. Notice that, by using the unit quaternion properties, we have $e_q = [1 \ 0^{\mathsf{T}}]^{\mathsf{T}}$ if and only if the orientation matrices R and R_d are aligned.

From (5) and (7), we obtain $\omega = v_o$ and using the quaternion propagation rule (3) it should be observed that the orientation error equation is nonlinear [14]. Considering a orientation control law based on a proportional action and a feed-forward term:

$$v_o = K_o e_{qv} + \omega_d , \qquad (13)$$

where $K_o = K_o^{\mathsf{T}} > 0$ is the orientation gain matrix and $\omega_d \in \mathbb{R}^3$ is the desired angular velocity, we can show that the equilibrium points $e_q = (\pm 1, 0^{\mathsf{T}})$ are almost globally asymptotically stable. The stability and the convergence analysis for the orientation control system is presented in [16] by using the Lyapunov stability theory.

D. Exploiting the Robot Redundancy

The redundancy characteristic of the robot manipulator, given by the extra degrees of mobility, can be exploited to generate internal motions at the joints, allowing the reconfiguration of its mechanical structure without changing the position and orientation of the robot end effector. This functionality can be obtained by adding a second term, named homogeneous solution, to the previously calculated joint control signal (7) as:

$$u_r = J_w^\dagger(\theta) v_c + \alpha J_w^\# \mu , \qquad (14)$$

where α is a positive scalar gain, $J_w^\# = (I - J_w^\dagger J)$ is a projection matrix, $\mu = W^{-1} \sigma$ and $\sigma \in \mathbb{R}^n$ is a vector of arbitrary joint velocities. The matrix $J_w^\#$ projects the vector μ in the null space of the weighted Jacobian matrix J_w without violating the constraint given by $v = J(\theta) \dot{\theta}$. It is worth mentioning that the first term of (14) corresponds to the minimum norm of joint velocities and the second one satisfies an additional constraint with secondary priority, specified by means of σ. A usual choice for the additional constraint is given by:

$$\sigma = (\partial g(\theta) / \partial \theta)^{\mathsf{T}} , \qquad (15)$$

where $g(\theta)$ is a secondary objective function to be maximized, for instance, the distance from an obstacle. Notice that, the solution for (15) moves along the direction of the gradient of the objective function to maximizes it locally, satisfying the primary objective given by the kinematic constraint (4). The reason for using the term μ, instead of only the term σ, is to avoid oscillatory joint motions when a particular joint is close to its limits as the joint velocity tends to zero [15].

IV. OBSTACLE AVOIDANCE METHODS

An efficient approach for the online motion planning is based on the concept of a particular point of the robot manipulator can move in the Cartesian space under the action of artificial potential fields. In this context, attractive fields drive the robot toward to a desired point, whereas repulsive fields are created around the obstacles, static and dynamic, to avoid collisions [4]. In this work, two types of repulsive potential fields are employed: the first is defined in terms of an additional constraint where the links are considered as the control points for the distance calculation [5]; the second acts directly on the robot end effector and can be included in the position control law, ensuring that robot follows a reference trajectory free of collisions [8].

A. Obstacle Avoidance for the Links

Here, we present the repulsive potential field generated to avoid the collision between links and obstacles. Considering a generic link L_i and an obstacle O_k, a repulsive potential field can be defined by [4]:

$$U_{L,O}(\theta) = \begin{cases} \frac{k_{L,O}}{2}\left(\frac{1}{\eta_{L,O}(\theta)} - \frac{1}{\eta_{lim}}\right)^2, & \eta_{L,O}(\theta) \leq \eta_{lim} \\ 0, & \eta_{L,O}(\theta) > \eta_{lim} \end{cases}$$

where $k_{L,O} > 0$ is a gain factor, $\eta_{L,O}(\theta)$ is the distance between the link and the obstacle and η_{lim} is the range of influence of the obstacle, or the boundary for obstacle detection. Notice that, the magnitude of the repulsive potential field $U_{L,O}$ depends on the inverse of the distance $\eta_{L,O}$ and, thus, if $\eta_{L,O} \to 0$, implies that $U_{L,O} \to \infty$. The resulting additional constraint is given by:

$$\sigma = \sum_{k=1}^{n_o} \nabla U_{L,O_k}(\theta), \qquad (16)$$

with n_o being the number of obstacles and $\nabla U_{L,O}$ is the gradient of the repulsive potential obtained from:

$$\nabla U_{L,O}(\theta) = -\frac{k_{L,O}}{\eta_{L,O}^2(\theta)}\left(\frac{1}{\eta_{L,O}(\theta)} - \frac{1}{\eta_{lim}}\right)\nabla\eta_{L,O}(\theta),$$

for $\eta_{L,O} \leq \eta_{lim}$, where $\nabla\eta_{L,O}$ is the gradient of the distance. The resulting repulsive force $f_{L,O}$ acting on the obstacle is given by:

$$f_{L,O}(\theta) = -\nabla U_{L,O}(\theta). \qquad (17)$$

B. Obstacle Avoidance for the Robot End effector

Let $U_{E,O}(\theta)$ be the repulsive potential field generated to avoid the collision between a generic robot end effector E_i and an obstacle O_k. The repulsive force corresponding to $U_{E,O}(\theta)$ is defined by [8]:

$$f_{E,O} = -\nabla U_{E,O}(\theta) = v_{mag}(\eta_{E,O})\frac{\eta_{E,O}(\theta)}{||\eta_{E,O}(\theta)||}, \qquad (18)$$

and it can be expressed in terms of the direction of the unit vector $\vec{\eta}_{E,O}(\theta)$ and its magnitude:

$$v_{mag}(\eta_{E,O}) = \frac{v_{max}}{1 + e^{\left(||\eta_{E,O}||(2/\rho)-1\right)b}}, \qquad (19)$$

where $\eta_{E,O} \in \mathbb{R}$ is the distance between the robot end effector and the obstacle, $v_{max} \in \mathbb{R}$ is the maximum magnitude, $\rho \in \mathbb{R}$ is the range of influence of the obstacle and $b \in \mathbb{R}$ is a constant value which defines the slope of the curve. Notice that, when $\eta_{E,O} \to 0$ implies that $v_{mag} \to v_{max}/(1+e^{-b})$, when $\eta_{E,O} \to \rho$ we have $v_{mag} \to v_{max}/(1+e^{b})$, and when $\eta_{E,O} \to \bar{\rho}$ yields $v_{mag} \to 0$ where $\bar{\rho} \in [\rho, \infty)$.

Considering the contribution of all obstacles for the reaction motion of a particular robot end effector E, the resulting repulsive force f_t is given by:

$$f_t = \sum_{k=1}^{n_o} f_{E,O_k}. \qquad (20)$$

However, in the presence of multiple obstacles the use of (20) can result in a resulting repulsive force with high magnitude. This means that the higher the number of obstacles is the higher the magnitude of f_t will be. A workaround to this issue is to obtain the direction of f_t and consider only the force magnitude due to the closest obstacle to the robot end effector, denoted by $v_{mag}(\eta_{E,O_{min}})$. Thus, it is possible to obtain a repulsive force f_r with smoother variations as:

$$f_r = -\nabla U_r = v_{mag}(\eta_{E,O_{min}})\frac{f_t}{||f_t||}, \qquad (21)$$

where $f_r/||f_r||$ is the unit vector in the direction of resulting repulsive force. From the force-velocity duality, we can consider the term f_r as a repulsive velocity vector and, therefore, add (21) to the position control law (11) to obtain:

$$v_{p,r} = K_p e_p + \dot{p}_d + f_r, \qquad (22)$$

ensuring that the robot end effector successfully performs the task of interest while avoiding obstacles located at its workspace. The boundedness of the control law (22) can be discussed analyzing the magnitudes of the repulsive velocity term ∇U_r and the position error proportional term.

Notice that, according to (21) we observe that the repulsive force is based only on the direction of minimum distance between the robot end effector and the obstacle. When an obstacle approaches to the robot end effector with a higher velocity than the robot motion capability it may be difficult to avoid collisions retracting the robot end effector to the same direction of the obstacle velocity. A more effective reaction strategy, similar to the human reaction, is to escape from

the collision moving the robot end effector in a direction approximately normal to the obstacle velocity. This effect can be obtained using the time variation of the repulsive force f_r denoted by $\dot{f}_r = df_r/dt$ to change the direction of the repulsive vector in the neighborhood of a particular point located at the robot end effector [8].

The main concern related to the use of the potential fields approach is the local minima problem, which occurs when the total force field is zero (force equilibrium). Indeed, the total potential obtained by the superposition of attractive and repulsive potentials, can exhibits local minima except for very particular cases. Alternative solutions to overcome the local minima problem are based on the randomized best-first algorithm or navigation functions [14].

V. OBSTACLE DETECTION ALGORITHM

A suitable obstacle detection approach allows the robot to operate in the presence of humans in safe and efficient manner, detecting static and dynamic obstacles located in its workspace. Here, to prevent that the obstacle detection algorithm considers the robot arm as an obstacle, we present a delimitation method for the links and joints based on the use of geometric elements to remove the robot body from the point cloud obtained by the depth sensor.

A. Description of the Environment

The RGB-D image captured by the depth sensor provides a 2-1/2-D geometrical description of the environment by means of a triplet (x_p, y_p, z_p), where the pair $\{x_p, y_p\}$ denotes the coordinates of the projection of a generic Cartesian point on the image plane and the element z_p denotes the depth coordinate between the point and the plane. The depth sensor can be modeled as a *pin-hole* camera in terms of a projection matrix T_p with the *intrinsic parameters*, and a homogeneous transformation matrix T_{rd} with the *extrinsic parameters* as:

$$T_p = \begin{pmatrix} f_d\alpha_x & 0 & x_{d_0} \\ 0 & f_d\alpha_y & y_{d_0} \\ 0 & 0 & 1 \end{pmatrix}, \quad T_{rd} = \begin{pmatrix} R_{rd} & p_{rd} \\ 0^\top & 1 \end{pmatrix},$$

where f_d is the focal length of the lens, α_x, α_y are the scaling factors in pixel per millimeters and x_{d_0}, y_{d_0} are the offsets related to the position of the center of the image plane with respect to the optical axis. The pair $\{R_{rd}, p_{rd}\}$ denotes the relative rotation matrix and the position vector between the depth sensor frame $\mathscr{F}_d$ and the reference frame $\mathscr{F}_r$. Here, without loss of generality, we can assume that the reference frame and the base frame are coincident and, thus, $\mathscr{F}_r \equiv \mathscr{F}_0$.

The pixel coordinates on the image plane and the depth coordinate of the generic Cartesian point can be expressed in the depth sensor frame $\mathscr{F}_d$ by means of the triplet (x_d, y_d, z_d) as follow:

$$\begin{aligned} x_d &= ((x_p - x_{d_0})z_p)/(f_d\alpha_x), \\ y_d &= ((y_p - y_{d_0})z_p)/(f_d\alpha_y), \\ z_d &= z_p. \end{aligned} \quad (23)$$

Finally, the coordinates (x_d, y_d, z_d) can be expressed in the reference frame $\mathscr{F}_r$ by means of the triplet (x_r, y_r, z_r), which

can be used as the input data for the obstacle detection algorithm through the following coordinate transformation:

$$\begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = R_{rd} \begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} + p_{rd}. \quad (24)$$

B. Filtering the Point Cloud

A point cloud corresponding to generic Cartesian points can be composed of objects located near or far from the robot, as well as, of its links and joints. Then, when the depth sensor is inserted in the working place, the detection algorithm could consider the elements of the robot body as obstacles. In this context, a delimitation method is proposed to remove the robot from the scenario obtained by the sensor, enclosing the links and joints by a sequence of geometric elements, such as, spheres, cylinders and ellipsoids. The delimitation method based on spheres can be implemented according to the following steps:

Step 1: Consider two consecutive joints J_i e J_{i+1} connected by a link L_j with length ℓ and radius r (see Fig. 2);

Step 2: Calculate the position of the robot joints by means of the forward kinematics map $p_{J_i} = k(\theta_i)$ for $i = 1, \cdots n$;

Step 3: Define equidistant points inside the link L_j along the smaller straight line segment connecting the joints J_i and J_{i+1}. The number of points n_p is defined by: $n_p = \left(\frac{\ell}{d}\right) + 1$, where $d = \left(\frac{\ell}{\lambda}\right)$ and $\lambda \in \mathbb{N}^*$;

Step 4: Use joints and intermediate points as center of the spheres and create spheres in all links of the robot arm;

Step 5: From the radius of the link r and the distance between the points d obtain the radius of the spheres r_s by using the Pythagoras theorem: $r_s^2 = \left(\frac{d}{2}\right)^2 + r^2$;

Step 6: Remove all points of the cloud located inside the spheres and subtract their radius from the distance calculation, such that, $\eta'_{L,O} = \eta_{L,O} - r_s$.
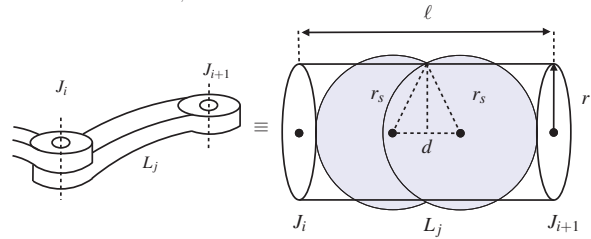

Fig. 2. Delimitation method for the robot body based on spheres.

It is worth mentioning that to completely enclose the links and joints using the sphere-based removal method, we should employ a high number of points, which result in increasing of the computational cost and reduction of the dimension of the collision-free workspace. In addition, objects located very close to the robot can be considered as part of it and, in general, there is an overlapping of the geometric elements. An alternative solution to overcome these drawbacks is to replace the spheres by other figures, such as cylinders or ellipses, allowing for the delimitation of the links using a small number of geometric elements and avoiding the

overlap. For the sake of saving space, the steps of these two delimitation methods will be omitted here.

Notice that, the depth sensor of the Kinect has two depth ranges: the default range and the near mode range. So, when a given object is located out of these detection ranges the measured depth value is zero and that object is not considered as an obstacle. Furthermore, to reduce the computational load of the obstacle detection method we can discard objects that are placed far away from the robot in a given period of time, but which still are within its workspace. First, the points located outside the robot workspace are removed from the point cloud. Next, the dimension of the geometric elements, previously created to remove the robot body from the cloud, are increased until it reaches the range of influence of obstacle. Finally, the points located outside the geometric elements are removed, eliminating the objects out of reach.

The drawback of the obstacle detection method based on a single depth sensor is the possible occurrence of occlusion regions or occluded objects, increasing the risks of collision. A simple solution to reduce the occurrence of the occlusion is to increase the sensors density, using more than one depth sensor with their lines of sight orthogonal to each other. In addition, we can perform a sensorial fusion between the Kinect and other external sensor that provides a geometrical representation of the environment as a point cloud.
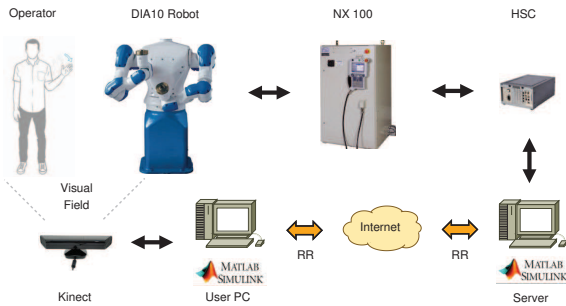

Fig. 3.   Experimental setup: DIA10 robot and Kinect.

VI. EXPERIMENTAL RESULTS

In this section, we present the description of the experimental apparatus used to perform the experiments. Graphical results are also shown to illustrate the feasibility of the proposed scheme. The robotic system is composed of a DIA10 dual-arm robot (Motoman Inc.) with 15-DoF (degrees of freedom) or axes of motion, 7-DoF per arm and plus 1-DoF for the base rotation, a low-level industrial controller NX100 and a HSC system (High-speed Synchronous Controller). In the experiments, only the left arm was used ($n = 8$) and the task of interest is to reach a desired position and orientation in the Cartesian space, with only 2 degrees of redundancy.

For the DIA10 robot, the reference signals are generated by an external computer (User PC), running the user application developed in Matlab/Simulink on Windows OS, that is connected to the NX100 controller through the HSC system. The signals are sent to a position control loop, built-in Simulink blocks (Server) and executed at the frequency

of $500\,Hz$. As it is usual for industrial robot arms, position and/or velocity controllers can be implemented in the joint space at a kinematic level. The access to the HSC system is performed via Matlab server running a communication protocol provided by *Robot Racounter* (RR), an architecture and communication library which is developed for robotic and automation applications [17]. In our experimental setup, RR is used as an interface to the HSC system enabling distributed control over networks, local control, and interface to supported Matlab/Simulink languages.

A natural interface device Microsoft Kinect[TM], endowed with a depth sensor and a video camera, is positioned in front of the robot, facing it, around at a distance of $2.25\,m$ and a height of $0.96\,m$. These values were obtained with respect to an inertial frame fixed on the robot base, and it was measured by hand *in-situ* or remotely calibrated. The depth sensor captures images 320×240 *pixel* of resolution at the frequency of $30\,Hz$ and the intrinsic parameters of the depth sensor were considered previously known or obtained by using a calibration method proposed in [18]. The default range of the depth sensor is $[0.8, 4.0]\,m$. The control parameters empirically tuned are: $K_p = 70I\,s^{-1}$, $K_o = 70I\,rad\,s^{-1}$, and $\alpha = -10^{-2}$. The obstacle avoidance parameters are: $k_{L,O} = 5$ and $\eta_{lim} = 0.6\,m$ for all links. Other parameters are: $v_{max} = 0.35\,m\,s^{-1}$, $\rho = 0.6\,m$, $b = 9$, $\dot{V}_{r_{max}} = 1\,m\,s^{-1}$, and $c = 9$.

Two experiments were performed to evaluate the behavior of the robot during the execution of contact-free tasks, with (or without) collision avoidance, in the presence of human operators moving in its workspace[2]:

Experiment 1 - the task of interest is to regulate the position and orientation of the end effector while the operator moves toward and away from the robot (see Fig. 4). The initial position and orientation are $p_0 = [\,0.816\;0.656\;1.337\,]^\mathsf{T}\,m$ and $R_0 = I$ respectively. The motions performed by the robot end effector avoiding the collision with the operator are depicted in Fig. 5. The time history of the position error and the norm of the orientation error, without and with collision avoidance, is presented in Fig. 6, (a)-(d). It is possible to observe the changes in the end-effector position due to the reactive behavior of the robot to prevent the contact. Fig. 7, (a) and (b), shows the behavior in time of the joint control signals without and with collision avoidance respectively.

Experiment 2 - the task of interest consists of moving the robot end effector through four Cartesian points or way-points, drawing a rectangular trajectory in a plane, while the operator continuously approaches and moves away from the robot. Fig. 8, (a) and (b), depicts the task execution without and with collision avoidance, as well as the way-points and the motion directions. Fig. 9(a) presents the end-effector motions during the task execution without collision avoidance, while Fig. 9(b) shows the same motion with collision avoidance, where it can be observed the reaction of the end effector when the operator suddenly approaches the robot. The behavior in time of the position error and the

---

[2]Please, visit the corresponding weblink to watch the accompanying video clip of the two experiments: `https://youtu.be/OU2Qigky360`.

norm of orientation error respectively is depicted in Fig. 10, (a) and (b), where we can observe the variations due to the reactive action of the robot end effector to avoid collision and reaches the way-points simultaneously. Fig. 11, (a) and (b), presents the time history of the joint control signals, without and with collision avoidance respectively.
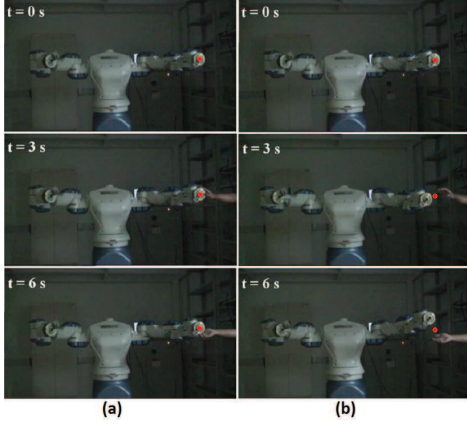

Fig. 4. Experiment 1: sequence of images for the regulation task (a) without collision avoidance and (b) with collision avoidance.
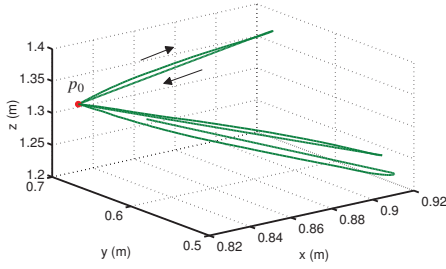

Fig. 5. Experiment 1: initial position of the end effector p_0 and the end-effector reaction motions for the regulation task. Legend: $p_0(\bullet)$, p (−).

VII. CONCLUDING REMARKS

A control and obstacle collision avoidance method is presented to allow a natural, safe and efficient interaction between robots and humans. The position and orientation control algorithm is based on the kinematic control approach and uses the weighted pseudo-inverse of the Jacobian matrix for weighting the mechanical joint limits and takes advantage of the redundancy characteristics of the robot structure for on-line collision avoidance. An obstacle avoidance algorithm is designed in terms of two repulsive potential fields considering the links and the robot end effector as points of interest to avoid collisions with static and dynamic obstacles during the execution of contact-free tasks. It is noteworthy that, the occurrence of local minima has not been observed in the experimental tests since the proposed method does not use attractive potential fields to achieve the control goal. The obstacle detection algorithm uses a depth sensor based on the structured light technology to obtain a 2-1/2-D geometrical description of the environment from a point cloud. Since the
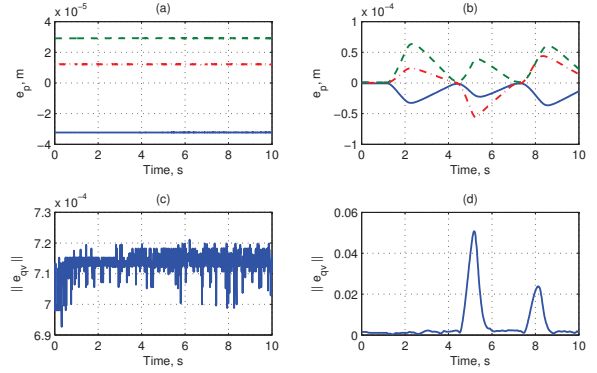

Fig. 6. Experiment 1: position error and norm of orientation error for the regulation task (a)-(c) without collision avoidance and (b)-(d) with collision avoidance. Legend: e_{px} (−), e_{py} (−−), e_{pz} (−.), $||e_{qv}||$ (−).
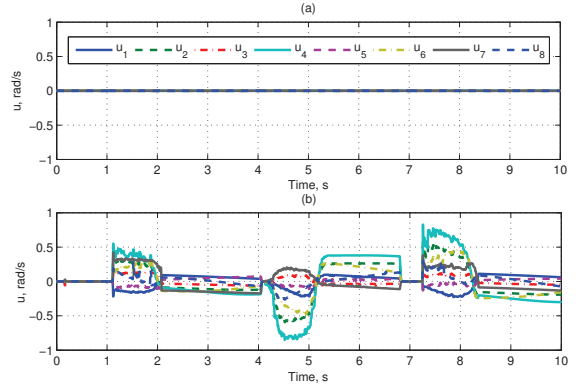

Fig. 7. Experiment 1: joint control signals for the regulation task (a) without collision avoidance and (b) with collision avoidance.
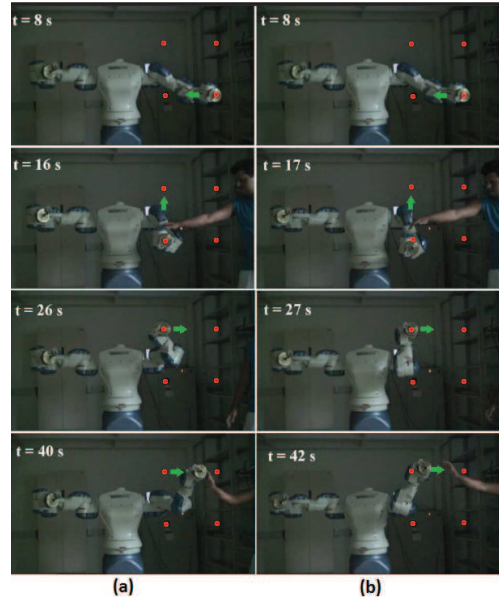

Fig. 8. Experiment 2: sequence of images for the way-points tracking task (a) without collision avoidance and (b) with collision avoidance.
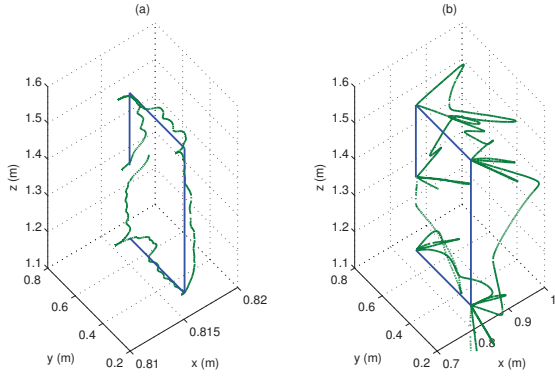
Fig. 9. Experiment 2: end-effector reaction motion for the way-points tracking task (a) without collision avoidance and (b) with collision avoidance. Legend: p_d (—), p (— —).
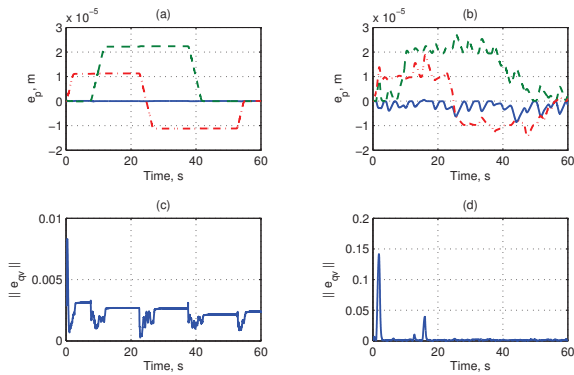

Fig. 10. Experiment 2: position error and norm of orientation error for the way-points tracking task (a)-(c) without collision avoidance and (b)-(d) with collision avoidance. Legend: e_{px} (—), e_{py} (— —), e_{pz} (—·), $\|e_{qv}\|$ (—).

robot elements, joint and links, can also be considered as obstacles, a delimitation method is presented to remove the robot from the RGB-D scene captured by the depth sensor, covering the robot body with a sequence of spheres, cylinders and ellipses. Practical tests are carried out with a Motoman DIA10 robot performing regulation and trajectory tracking tasks in a human-populated environment and monitored by a Microsoft Kinect[TM]. Experimental results are included to illustrate the feasibility of the proposed scheme.

According to our experiments, the sphere-based delimitation method requires less computational effort and it is easier to implement compared to other solutions. The cylinder-based delimitation method uses a smaller number of geometric elements and there is no significant reduction in the dimension of the collision-free workspace. Actually, the performance of all delimitation methods depends on the robot geometric structure and the link-joint configuration.

REFERENCES

[1] O. Brock and O. Khatib, "Elastic strips: A Framework for Motion Generation in Human Environments," *Int. J. Robot. Res.*, vol. 21, no. 12, pp. 1031–1052, 2002.

[2] L. E. Kavraki and S. M. LaValle, "Motion Planning," in *Springer Handbook of Robotics*. Springer-Verlag Ltd., 2008, pp. 109–131.
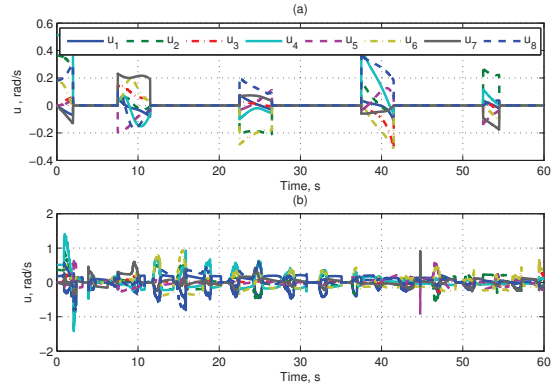
[3] J. Minguez, F. Lamiraux, and J.-P. Laumond, "Motion Planning and Obstacle Avoidance," in *Springer Handbook of Robotics*. Springer-Verlag Ltd., 2008, pp. 827–852.

[4] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, 1986.

[5] C. Y. Chung, B.-H. Lee, and J.-H. Lee, "Obstacle Avoidance for Kinematically Redundant Robots using Distance Algorithm," in *Proc. of the IEEE/RSJ Int. Conf. on Intell. Rob. and Syst.*, vol. 3, Sep 1997, pp. 1787–1793.

[6] L. Singh, H. Stephanou, and J. Wen, "Real-time Robot Motion Control with Circulatory Fields," in *Proc. of IEEE Int. Conf. on Rob. and Autom.*, vol. 3, Apr 1996, pp. 2737–2742.

[7] S. Haddadin, R. Belder, and A. Albu-Schäffer, "Dynamic Motion Planning for Robots in Partially Unknown Environments," in *Proc. of the 18th IFAC World Congress*, vol. 18, 2011, pp. 6842–6850.

[8] F. Flacco, T. Kroger, A. De Luca, and O. Khatib, "A Depth Space Approach to Human-robot Collision Avoidance," in *Proc. of the IEEE Int. Conf. on Rob. and Autom.*, 2012, pp. 338–345.

[9] P. Rakprayoon, M. Ruchanurucks, and A. Coundoul, "Kinect-based Obstacle Detection for Manipulator," in *Proc. of the IEEE/SICE Int. Symp. on Syst. Int.*, Dec 2011, pp. 68–73.

[10] D. Correa, D. Sciotti, M. Prado, D. Sales, D. Wolf, and F. Osorio, "Mobile Robots Navigation in Indoor Environments Using Kinect Sensor," in *Proc. of the 2nd Brazilian Conf. on Crit. Embed. Syst.*, May 2012, pp. 36–41.

[11] M. Greuter, M. Rosenfelder, M. Blaich, and O. Bittel, "Obstacle and Game Element Detection with the 3D-Sensor Kinect," in *Res. and Educ. in Robot.* Springer-Verlag Ltd., 2011, vol. 161, pp. 130–143.

[12] L. Cheng, Q. Sun, H. Su, Y. Cong, and S. Zhao, "Design and Implementation of Human-robot Interactive Demonstration System based on Kinect," in *Proc. of the 24th Chinese Contr. and Dec. Conf.*, May 2012, pp. 971–975.

[13] A. De Luca and F. Flacco, "Integrated Control for pHRI: Collision Avoidance, Detection, Reaction and Collaboration," in *Proc. of the IEEE RAS EMBS Int. Conf. on Biom. Robot. and Biomech.*, Jun 2012, pp. 288–295.

[14] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control.* Springer-Verlag Ltd., 2009.

[15] M. Bjerkeng, A. A. Transeth, K. Y. Pettersen, E. Kyrkjebo, and S. A. Fjerdingen, "Active Camera Control with Obstacle Avoidance for Remote Operations with Industrial Manipulators: Implementation and Experimental Results," in *Proc. of the IEEE/RSJ Int. Conf. on Intell. Rob. and Syst.*, Sep 2011, pp. 247–254.

[16] A. C. Leite, F. Lizarralde, and L. Hsu, "Hybrid Adaptive Vision-Force Control for Robot Manipulators Interacting with Unknown Surfaces," *Int. J. Robot. Res.*, vol. 28, pp. 911–926, Jul 2009.

[17] J. D. Wason and J.-Y. Wen, "Robot Raconteur: A Communication Architecture and Library for Robotic and Automation Systems," in *Proc. of the IEEE Conf. on Autom. Science and Eng.*, Aug 2011, pp. 761–766.

[18] D. Herrera C., J. Kannala, and J. Heikkil, "Joint Depth and Color Camera Calibration with Distortion Correction," *IEEE Trans. on Patt. Anal. and Mach. Intell.*, vol. 34, no. 10, pp. 2058–2064, Oct 2012.

Fig. 11. Experiment 2: joint control signals for the way-points tracking task (a) without collision avoidance and (b) with collision avoidance.