Intel Serv Robotics
DOI 10.1007/s11370-007-0003-2

ORIGINAL RESEARCH

A unified fuzzy logic approach to trajectory planning and inverse kinematics for a fire fighting robot operating in tunnels

A. De Santis · B. Siciliano · L. Villani

Received: 7 February 2007 / Accepted: 12 February 2007

Abstract In this paper a fuzzy logic approach to automatic trajectory planning and closed-loop inverse kinematics for a robotic system purposely designed to extinguish fires in road and railway tunnels is presented. The robot is composed of a self-cooling monorail vehicle carrying a fire fighting monitor. A fuzzy inference system is adopted for the automatic generation of the task-space trajectory for the robot and to distribute the motion among the available joints in the presence of redundant degrees of mobility. Redundancy also allows assigning additional tasks besides the primary task. Simulation case studies are presented to test the performance of the whole system in a typical intervention scenario.

Keywords Rescue robotics · Fire fighting · Inverse kinematics · Fuzzy inference system · Redundancy

1 Introduction

Tunnels are important infrastructures for the European Union and play a crucial role for the development of the regional economies [1]. Hence, the problem of safety in road and railway tunnels and, in particular, the risk connected to fires is of particular concern [2].

A. De Santis (✉) · B. Siciliano · L. Villani
PRISMA Lab, Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli Federico II, Via Claudio 21, 80125 Napoli, Italy
e-mail: agodesa@unina.it

B. Siciliano
e-mail: siciliano@unina.it

L. Villani
e-mail: lvillani@unina.it

The recent disasters in the Monte Bianco tunnel (connecting Italy to France) and in the Tauern tunnel (Austria), in 1999, have evidenced that the capability of intervention of the fire brigades is very limited because of the extreme environmental conditions (high temperature, intense smoke, gas emissions, traffic, rails and other obstructions) which delay or preclude the action of men and machines. On the other hand, a prompt and effective intervention is crucial to keep the fire under control and contain the damage.

In this respect, technology may play an important role to augment safety. Tunnels can be made more and more "intelligent" by installing distributed sensors to measure significant variables (temperature, humidity, wind velocity, presence of smoke or other gas, etc.) that can be collected and suitably elaborated to facilitate human intervention or to guide the operation of automatic devices [3]. In particular, robots can replace or support men in monitoring and intervention in case of fire.

In this work, a robotic system purposely designed for fire extinguishing in tunnels is considered. The robot derives from the ROBOGAT patent [4] and is the subject of a National Operational Program (P.O.N.) financed by the Italian Ministry of Education, University and Scientific Research.

The basic idea of ROBOGAT (see Fig. 1) is that of a robotic system able to perform operations similar to those of the firemen. The system is composed of:

- a self-cooling monorail including a pipe which guarantees continuous supply of water or extinguishing liquid;
- a mobile base moving on the monorail;

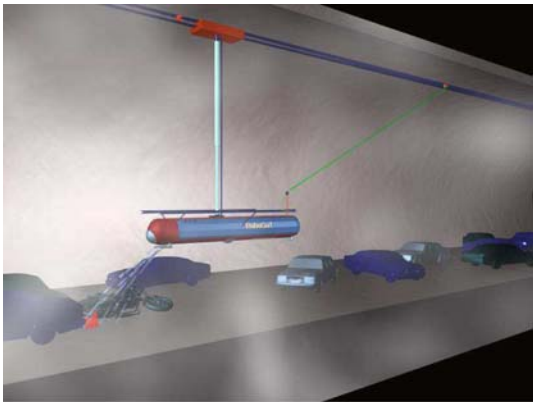

Fig. 1 A sketch of the fire fighting robotic system

- a robotized fire fighting monitor;
- enhanced sensors as infrared video cameras, pyrometers, gas cromatographs.

The monorail may be mounted on the side or on the ceiling of the tunnel, depending on the room available. Along the pipe included in the monorail, every 30 m, suitable inox steel victaulic couplings are installed, where the robot can be connected for water supply when it reaches a place close to the fire. Hence, the robot may reach any point of the tunnel in short time, guaranteeing a prompt intervention. The system is built to resist to high temperatures (up to 1,000°C) and is tele-operated from a remote control room; moreover, it is able to perform some autonomous operation. The mobile base is guided to the place of intervention, where it is automatically connected to the pipe through a telescopic hose which allows the robot to move along the monorail also after the connection to the pipe. The robotized fire fighting monitor is suitably controlled to drive the water jet on the hot spots located by the infrared cameras.

The issue considered in this paper concerns the automatic trajectory planning and inverse kinematics for this robot, for the execution of a typical task, i.e., reaching the fire in the tunnel and direct a high pressure water flow at the base of the flames. During the execution of the task, the robot must protect itself from the fire by keeping a suitable safety distance.

The automatic trajectory planning is a key feature in all the tasks where the speed is important, such operation in hostile environments. Moreover, it has to take into account the kinematic structure and the dynamic capability of the robot. For redundant robotic systems, planned trajectories may result in different motion distributions among the joints, so that a "preferred" distribution can be suitably selected on the basis of task-space considerations (e.g., obstacle avoidance), joint-space optimality criteria (e.g., dexterity issues), or sensory information.

From a kinematical point of view, the robotic system considered in this paper is redundant; hence the redundant degrees of mobility may be exploited to achieve a suitable coordination between the motion of the vehicle and that of the monitor. For example, a task-priority redundancy resolution technique [5] can be adopted to specify a primary task (e.g., a prescribed trajectory for the water jet) which is fulfilled with higher priority with respect to a secondary task (e.g., maintaining a safety distance or a dexterous posture).

In this paper, a redundancy resolution approach is used to solve the inverse kinematics problem, using a closed-loop algorithm. This is based on a weighted pseudo-inverse Jacobian and a task-priority strategy integrated with a fuzzy logic algorithm. The approach is inspired by [6], where it is adopted for underwater systems, and was applied in [7] to the robot considered here. One novelty of this work is that a fuzzy trajectory planning strategy is integrated with the inverse kinematic problem, in order to achieve a water jet trajectory which automatically takes into account the motion distribution among the joints selected by the redundancy resolution algorithm.

In detail, a Fuzzy Inference System (FIS) is in charge of distributing the required water-jet motion between the vehicle and the monitor, by setting the weights of the weighted pseudo-inverse Jacobian. The FIS may also activate a secondary task, e.g., to keep the monitor in a dexterous configuration during some phases of the primary task, or to keep the vehicle at a safety distance from the fire. At the same time, the trajectory planning takes advantage of the information from the FIS to adjust the task-space planned velocity to the actual motion capability of the joints of the robotic system.

The novel contribution of the paper is the adoption of the fuzzy-based trajectory planning and inverse kinematics algorithm to an automatic pointing problem for a water jet, which takes into account the jet trajectory.

Numerical simulations have been developed to show the effectiveness of the proposed algorithm.

2 Direct kinematics

The kinematic structure of the fire fighting robot (see Fig. 2) can be described by considering a prismatic joint corresponding to the vehicle moving on the monorail (joint variable d), two revolute joints corresponding to the monitor mounted under the vehicle (joint variables θ_1 and θ_2), and the water jet shot by the monitor. Neglecting the influence of the motion of the vehicle and of

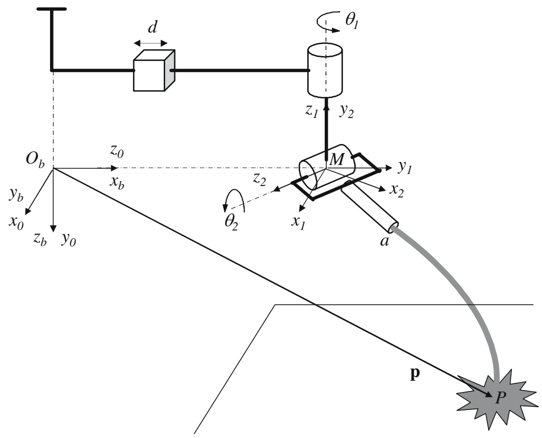

Fig. 2 Kinematic model of the fire fighting robot

the monitor on the shape of the water jet and considering a quasi-static regime, the water trajectory can be described by a parabolic curve lying on the vertical plane of the monitor, corresponding to the parabolic motion of a particle under gravity (i.e., the so-called projectile motion). More complex models of the water jet can be adopted; however, this approximation is reasonable in a wide range of operations [8].

It is assumed that the velocity v_0 at the base of the water jet can be modified by varying the area a of the nozzle of the monitor, according to the simple relation

$$v_0 = \frac{f}{a}$$

where f is the flow rate. The quantity f is usually variable in practical applications; in such a case it is assumed that a flow meter is available to measure f in real time.

The relevant frames required to compute the robot kinematics, chosen according to the Denavit–Hartenberg convention [9], are reported in Fig. 2. It can be seen that the axes z_0, z_1 and z_2 intersect at point M.

The position vector $\boldsymbol{p} = \begin{bmatrix} x & y & z \end{bmatrix}^{\mathrm{T}}$ of a point P of the water jet, referred to the base frame, can be computed in terms of the joint variables d, θ_1 and θ_2 of the robot, of the initial velocity v_0 of the water jet and of the coordinate z.

The vector $\boldsymbol{p}$ can be expressed as

$$\widetilde{\boldsymbol{p}} = \widetilde{\boldsymbol{p}}_M + \boldsymbol{T}_2{}^2\widetilde{\boldsymbol{p}} \tag{1}$$

where $\widetilde{\boldsymbol{v}}$ denotes the representation of a generic vector $\boldsymbol{v}$ in homogeneous coordinates, i.e. $\widetilde{\boldsymbol{v}} = \begin{bmatrix} \boldsymbol{v}^{\mathrm{T}} & 1 \end{bmatrix}^{\mathrm{T}}$, $\boldsymbol{p}_M$ is the position vector of the point M with respect to the base frame, ${}^2\boldsymbol{p} = \begin{bmatrix} {}^2x & {}^2y & {}^2z \end{bmatrix}^{\mathrm{T}}$ is the position vector of the point P with respect to the frame 2, and $\boldsymbol{T}_2$ is the homogeneous transformation matrix of frame 2 with respect to the base frame. The matrix $\boldsymbol{T}_2$ can be easily computed as

$$\mathbf{T_2} = \begin{bmatrix} c_1 & 0 & s_1 & d_1 \\ -s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2}$$

In the above equations, s_i and c_i $(i = 1, 2)$ denote $\sin(\theta_i)$ and $\cos(\theta_i)$, respectively.

The homogeneous representation of vector $\boldsymbol{p}_M$ coincides with the last column of the matrix $\boldsymbol{T}_2$, while the components of the vector ${}^2\boldsymbol{p}$ can be computed on the basis of the projectile equations of motion written in frame 2 as

$${}^2x = lc_2 + v_0 c_2 t \tag{3}$$

$${}^2y = ls_2 + v_0 s_2 t - \frac{1}{2} g t^2 \tag{4}$$

$${}^2z = 0, \tag{5}$$

where t is the time variable, l is the length of the second link of the monitor and g is the magnitude of the gravity acceleration.

The parameter t can be eliminated from Eqs. (3) and (4), and thus ${}^2x$ can be expressed as a function of ${}^2y$ or viceversa. Both the options are feasible, however, it is computationally more convenient computing ${}^2x$ as a function of ${}^2y$ in the form

$${}^2x = lc_2 + \frac{1}{g} v_0^2 c_2 \left(s_2 + \sqrt{s_2^2 + 2g \frac{z + ls_2}{v_0^2}} \right) = r(\theta_2, v_0, z),$$

where the equality ${}^2y = -z$ has been taken into account. Hence, in view of (1) and (2), the vector $\boldsymbol{p}$ can be computed in the form:

$$\boldsymbol{p}(d, \theta_1, \theta_2, v_0, z) = \begin{bmatrix} d + r(\theta_2, v_0, z) c_1 \\ -r(\theta_2, v_0, z) s_1 \\ z \end{bmatrix}, \tag{6}$$

where $r(\theta_2, v_0, z)$ is the range of the water jet evaluated on a horizontal plane placed at a distance z along the z_b-axis of the base frame, which can be computed as

$$r(\theta_2, v_0, z) = lc_2 + \frac{1}{g} v_0^2 c_2 \left(s_2 + \sqrt{s_2^2 + 2g \frac{z + ls_2}{v_0^2}} \right).$$

Notice that the variable z appears in both sides of the direct kinematic equation (6), hence it is both a joint-space and a task-space variable. Therefore, there are five degrees of mobility and three task coordinates, resulting in two redundant degrees of mobility.

3 Inverse kinematics

To solve the inverse kinematics problem, the variables $d, \theta_1, \theta_2, v_0$ and z must be computed from the three components x, y and z of the vector $\boldsymbol{p}$ by inverting the three equations in (6). Since the last equation is an identity, only the first two equations have to be solved. Assuming that the flow rate f is constant, the solution to the problem can be found starting from the differential mapping computed from the first two components of (6)

$$\dot{\boldsymbol{p}}_{xy} = \boldsymbol{J}_{xy}(\boldsymbol{q}, z)\dot{\boldsymbol{q}} + \boldsymbol{J}_z(\boldsymbol{q}, z)\dot{z} \tag{7}$$

where

$$\boldsymbol{p}_{xy} = \begin{bmatrix} x & y \end{bmatrix}^{\mathrm{T}}$$
$$\boldsymbol{q} = \begin{bmatrix} d & \theta_1 & \theta_2 & v_0 \end{bmatrix}^{\mathrm{T}}.$$

Notice that the Jacobian matrix $\boldsymbol{J}_{xy}(\boldsymbol{q}, z)$ is (2×4), and this is consistent with the number (two) of redundant degrees of mobility of the system.

The simplest way to invert the mapping (7) is to use the pseudo-inverse of the Jacobian matrix:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\dagger}(\boldsymbol{q}, z)\dot{\boldsymbol{p}}_{xy} - \boldsymbol{J}^{\dagger}(\boldsymbol{q}, z)\boldsymbol{J}_z(\boldsymbol{q}, z)\dot{z} \tag{8}$$

where

$$\boldsymbol{J}^{\dagger} = \boldsymbol{J}_{xy}^{\mathrm{T}} \left(\boldsymbol{J}_{xy}\boldsymbol{J}_{xy}^{\mathrm{T}}\right)^{-1}. \tag{9}$$

This solution corresponds to the minimization of the joint velocities in a least-square sense [9].

Because of the different characteristics of the available degrees of mobility, it could be required to modify the velocity distribution with respect to the least-square minimal solution. For example, it would be preferable to perform slow gross motion using the vehicle and fast motion of small amplitude using the monitor. This might be achieved by adopting a weighted pseudo-inverse $\boldsymbol{J}_W^{\dagger}$

$$\boldsymbol{J}_W^{\dagger} = \boldsymbol{W}^{-1}\boldsymbol{J}_{xy}^{\mathrm{T}} \left(\boldsymbol{J}_{xy}\boldsymbol{W}^{-1}\boldsymbol{J}_{xy}^{\mathrm{T}}\right)^{-1} \tag{10}$$

with the (4×4) matrix $\boldsymbol{W}^{-1} = \mathrm{diag}\{\beta_1, \beta_2, \beta_3, \beta_4\}$ where β_i is a weight factor belonging to the interval $[0, 1]$ such that $\beta_i = 1$ corresponds to full motion for the ith degree of mobility and $\beta_i = 0$ corresponds to freeze the corresponding joint.

The redundancy of the system can be further exploited by using a task-priority strategy corresponding to a solution to (7) of the form

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_W^{\dagger}(\boldsymbol{q}, z)\dot{\boldsymbol{p}}_{xy} - \boldsymbol{J}_W^{\dagger}(\boldsymbol{q}, z)\boldsymbol{J}_z(\boldsymbol{q}, z)\dot{z} + \left(\boldsymbol{I}_4 - \boldsymbol{J}_W^{\dagger}(\boldsymbol{q}, z)\boldsymbol{J}_{xy}(\boldsymbol{q}, z)\right)\dot{\boldsymbol{q}}_a$$

where $\boldsymbol{I}_4$ is the (4×4) identity matrix, $\dot{\boldsymbol{q}}_a$ is an arbitrary joint velocity vector and the operator $\left(\boldsymbol{I}_4 - \boldsymbol{J}_W^{\dagger}\boldsymbol{J}_{xy}\right)$ projects the joint velocity vector in the null space of the Jacobian matrix. This solution generates an internal motion of the robotic system (secondary task) which does not affect the motion of the water jet end-point P (primary task).

The joint velocity vector $\dot{\boldsymbol{q}}_a$ can be chosen aligned to the gradient of a scalar objective function $W(\boldsymbol{q})$, i.e.:

$$\dot{\boldsymbol{q}}_a = -\alpha k_w \frac{\partial W(\boldsymbol{q})}{\partial \boldsymbol{q}} \tag{11}$$

with $k_w > 0$ and $\alpha \in [0, 1]$, in order to achieve a local minimum for $W(\boldsymbol{q})$ [10].

To avoid numerical drift due to discrete-time integration, a Closed Loop Inverse Kinematics (CLIK) algorithm can be adopted [9], which computes $\boldsymbol{q}$ from the integration of the vector:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_W^{\dagger}(\boldsymbol{q}, z_d)\boldsymbol{v} - \boldsymbol{J}_W^{\dagger}(\boldsymbol{q}, z_d)\boldsymbol{J}_z(\boldsymbol{q}, z_d)\dot{z}_d + \left(\boldsymbol{I}_4 - \boldsymbol{J}_W^{\dagger}(\boldsymbol{q}, z_d)\boldsymbol{J}_{xy}(\boldsymbol{q}, z_d)\right)\dot{\boldsymbol{q}}_a \tag{12}$$

with

$$\boldsymbol{v} = \dot{\boldsymbol{p}}_{xy,d} + \boldsymbol{K}(\boldsymbol{p}_{xy,d} - \boldsymbol{p}_{xy}),$$

where $\boldsymbol{K}$ is a (2×2) positive definite matrix gain to be chosen so as to ensure the convergence to zero of the error $\boldsymbol{p}_{xy,d} - \boldsymbol{p}_{xy}$. Notice that in (12) the subscript d denotes the components of the position and velocity vectors that are input to the CLIK algorithm; the position components without the subscript d are those computed from the joint position vector $\boldsymbol{q}$ (the output of the algorithm) via the direct kinematics equation (6).

In the case that f is not constant, an additional term depending on $\dot{f}$ should be added to (7) to take into account the variation of f with time. However, this term can be neglected in the CLIK algorithm, provided that the correct value of f is used for the computation of the direct kinematics equation (6). In fact, thanks to the robustness of the closed loop scheme, the integration error will remain as lower as higher the gain $\boldsymbol{K}_P$ is.

4 Trajectory planning

The CLIK algorithm (12) allows solving the redundancy of the system by suitably choosing the coefficients β_i

of the weighted pseudo-inverse of the Jacobian (10), as well as the coefficient α in (11) for a given function $W(\boldsymbol{q})$ defining the secondary task.

The choice of the above parameters could be made a priori, according to the task requirements. However, setting constant values would mean to fix the motion distribution among the degrees of mobility of the system and activate the secondary task during the whole development of the primary task.

In fact, it could be useful to change the values of the parameters β_i and α during the operation, in order to better distribute the motion among the joints depending on the particular operating conditions.

Moreover, the planned trajectory itself should be consistent with the motion distribution between the joints, also taking into account the maximum velocity achievable for each joint.

For example, if the task requires to reach a fire that is initially out of the range of the monitor, the trajectory planner should generate a motion for the end point of the water jet that first reaches the fire zone as fast as possible and then moves about the fire to extinguish it.

At the same time, the redundancy of the system should be exploited by the inverse kinematics algorithm so that during the approach phase it is the vehicle that moves close to the fire (i.e., $\beta_1 \approx 1$, $\beta_2 = \beta_3 = \beta_4 \approx 0$, $\alpha \approx 0$) while the monitor and the valve are mainly used during the fire extinguishing phase (i.e., $\beta_1 \approx 0$, $\beta_2 = \beta_3 = \beta_4 \approx 1$), possibly keeping a dexterous posture (i.e., $\alpha \approx 1$).

A simple and effective way to automatically compute a task-space trajectory compatibile with the joint motion distribution can be that of selecting the instantaneous maximum velocity for the end point of the water jet as the weighed mean of the maximum velocity contribution of each joint, where the weights are the same β_i used by the inverse kinematics algorithm, i.e.,

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} \beta_1 v_{1M_x} + \beta_2 v_{2M_x} + \beta_3 v_{3M_x} + \beta_4 v_{4M_x} \\ \beta_1 v_{1M_y} + \beta_2 v_{2M_y} + \beta_3 v_{3M_y} + \beta_4 v_{4M_y} \end{bmatrix}, \tag{13}$$

being v_{iM_x} (v_{iM_y}) the maximum velocity contribution of the joint i along the x-direction (y-direction).

The task-space trajectory planning algorithm developed for the fire fighting task receives as inputs the current position of the fire in the tunnel and the current values of the weights β_i and computes the desired trajectory for the water-jet end point. This is represented by a rectilinear motion parallel to the rail, until the vehicle reaches a safety distance from the fire; then, an alternate motion in the horizontal plane centered on the fire is commanded.

5 Fuzzy inference system

A flexible handling of the variables of interest for trajectory planning and inverse kinematics can be achieved by using a fuzzy algorithm, which is in charge of managing the distribution of the motion between the joints as well as activating the secondary task. This is done by adjusting online the weighting factors β_i and α in (12) according to a Mamdani Fuzzy Inference System [11].

Notice that other kinds of inference systems can be used in principle, however fuzzy systems have been adopted here because they can be simply implemented and easily modified by changing the membership functions or adding new rules.

The crisp inputs of the FIS are the measurements coming form three exteroceptive sensors that are supposed to be available on the robotic system. They are as follows: a fire detector, which provides a binary output ψ which takes the value 1 in the presence of fire in the tunnel and the value 0 otherwise; a temperature sensor, which measures the temperature Θ of the environment close to the robot; a distance sensor (e.g., based on infrared cameras), which measures the distance δ between the robot and the fire along the rail. The crisp outputs are three values ($\beta_d = \beta_1, \beta_m = \beta_2 = \beta_3, \beta_v = \beta_4$) which weight the motion capability of the joints (the prismatic joint d, the monitor joints θ_1 and θ_2, and the water jet initial velocity v_0) in the operations. Two linguistic variables are considered: *temperature* = {low, high}, *distance* = {low, average, high}, while the output linguistic variables are *vehicle* = {standing, moving}, *monitor* = {standing, average, moving}, *valve* = {closed, half open, open}. The membership functions used for the linguistic functions are not reported here for brevity. The fuzzy rules are

1. if (*distance* is high) and (*temperature* is low), then (*vehicle* is moving, *monitor* is standing, *valve* is closed);
2. if (*distance* is medium) and (*temperature* is low), then (*monitor* is average, *valve* is half open);
3. if (*distance* is low) and (*temperature* is low), then (*vehicle* is standing);
4. if (*temperature* is high), then (*vehicle* is standing, *monitor* is moving, *valve* is open).

After the defuzzyfication, the crisp output β_v is multiplied by the output ψ of the fire detector, i.e., the valve is freezed in the absence of fire. Notice that β_m and β_v follow the same rules, however two different crisp outputs are considered, because it could be necessary to freeze

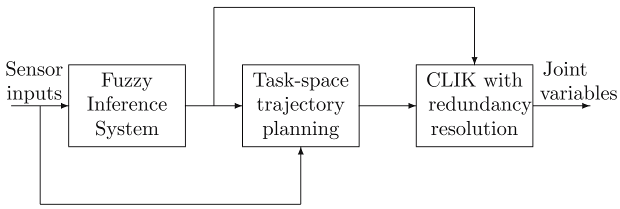

Fig. 3 Schematic of the algorithm with fuzzy trajectory planning and redundancy resolution

the valve (i.e., forcing $\beta_v = 0$ when $\psi = 0$) by letting the monitor free to move (i.e., $\beta_m \approx 0$).

The weight α should be chosen as a function of the secondary task, by introducing suitable fuzzy rules or as a function of the crisp outputs (if the secondary task is specified in the joint space).

For example, if the secondary task is that of keeping the monitor in a dexterous configuration where all the variables are far from the joint limits (i.e., $\theta_1 = 0$ rad, $\theta_2 = -\pi/12$ rad and $v_0 = 14.6$ m/s), then α can be computed as

$$\alpha = \frac{1}{2}(\beta_m + \beta_v).$$

A schematic of the proposed algorithm with fuzzy trajectory planning and redundancy resolution is represented in Fig. 3. An example of application is described in the simulation case study.

6 Simulation

The simulation case studies have been performed using MATLAB with the Fuzzy Logic Toolbox. It is assumed that the robot is mounted under the ceiling of the tunnel which is about 6.30 m high and 10 m large at the base. The length of the second link of the monitor is $l = 0.8$ m. The joint limits for the monitor and valve are

$$\theta_{1m} = -\pi/3\,\text{rad}, \quad \theta_{1M} = \pi/3\,\text{rad}$$
$$\theta_{2m} = -\pi/2\,\text{rad}, \quad \theta_{2M} = 0\,\text{rad}$$
$$v_{0m} = 2.6\,\text{m/s}, \quad v_{0M} = 26.6\,\text{m/s}.$$

The flow rate is $f = 8 \times 10^{-3}$ m^3/s. The algorithm has been implemented at a sampling frequency of 1,000 Hz, and the gains in (12) have been set to

$$\boldsymbol{K} = \text{diag}\{600, 600\} k_w = 200.$$

A fire is sensed in the position

$$\boldsymbol{p}_f = \begin{bmatrix} 20 & 0 & 6.3 \end{bmatrix}^{\text{T}}$$

while the robot is assumed to be at the entrance of the tunnel, where the base frame is located.

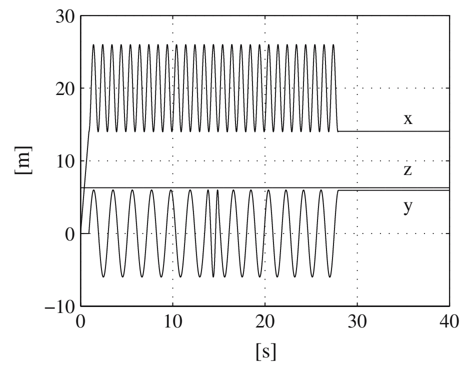

Fig. 4 Time history of the planned task-space coordinates in the first case study

Three case studies are presented. In the first case study the robot has to reach the place of the fire and extinguish it; in the second case study, during the extinguishing phase, the fire suddenly moves toward the robot (e.g., as a consequence of an explosion) and the robot must protect itself by keeping a suitable safety distance. Finally, in the third case study, the flow rate of the water is assumed to be variable along the tunnel and the robustness of the CLIK algorithm is tested.

6.1 First case study

The first case study is aimed at testing the fuzzy trajectory planning and inverse kinematic algorithm in the execution of a fire extinguishing task. For the purpose of the simulation, it is assumed that the temperature sensed by the robot increases proportionally to the inverse of the distance to the the fire, from 30 to 480°C; during the fire extinguishing phase, it assumed that the temperature remains constant for 15 s and then decreases proportionally to the time, until 50°C, in a time interval of 5 s duration.

In Fig. 4, the task-space trajectory for the water jet, generated by the motion planning algorithm, is reported. It can be recognized that the x component (i.e,. the component along the rail) reaches the position of the fire in about 1 s time, when the fire extinguishing phase begins. During this phase, a sinusoidal motion is planned both for the x and y components. Notice that the frequencies of the sinusoids are variable according to the velocity law (13), due to the variation of the weights β_i computed by the FIS. At the end of the fire extinguishing phase, the position of the water jet is stopped in a fixed point.

The time history of the joint positions computed by the inverse kinematics algorithm is shown in Fig. 5. It can be seen that the joint variable d is quickly variable

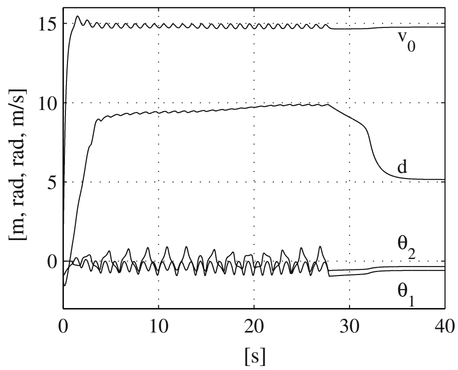

Fig. 5 Time history of the joint positions in the first case study

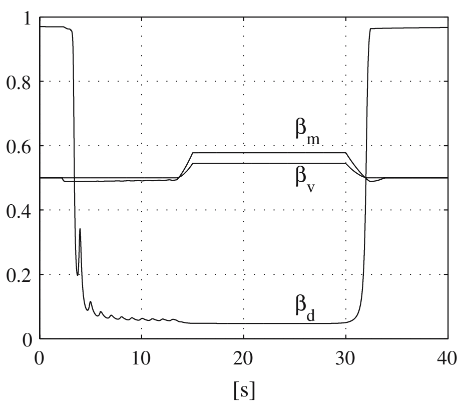

Fig. 6 Time history of the crisp outputs of the Fuzzy Inference System (FIS) in the first case study

during the approach phase and remains practically constant during the fire extinguishing phase; at the end of the fire it assumes a value corresponding to the fulfillment of the secondary task for the monitor and the valve. During the fire extinguishing phase, the two revolute joint variables θ_1 and θ_2 vary according to the primary task, about the dexterous values imposed by the secondary task; these values are exactly reached after the end of the fire. Finally, the variable v_0 starts to increase when the fire is sensed, in order to increase the range of the water jet, then remains almost constant, next to the value set by the secondary task.

In Fig. 6, the time history of the crisp values of the output of the fuzzy inference system is reported. It can be recognized that the weight β_d is high during the approach phase and after the fire extinguishing phase, and this is in accordance to the time history of the joint variable d in Fig. 5. Also, the weights β_m and β_v have both average values during all the task, and are higher when the temperature reaches its maximum value.

To better understand the role played by the FIS in the redundancy resolution, the same task-space trajectory of

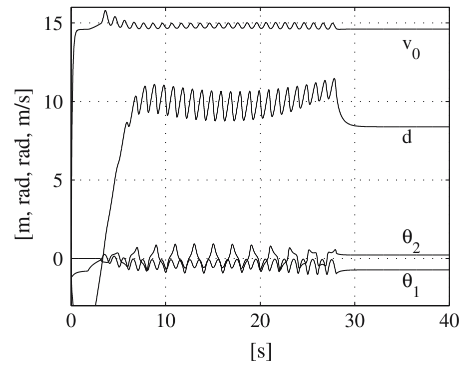

Fig. 7 Time history of the joint positions using fixed weights inverse kinematics in the first case study

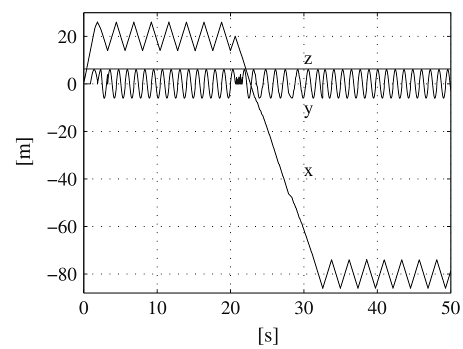

Fig. 8 Time history of the planned task-space coordinates in the second case study

Fig. 4 has been considered, while fixed weights ($\beta_d = \beta_m = \beta_v = 1$) have been used in the pseudo-inverse Jacobian matrix. From the time history of the joint positions computed by the inverse kinematics algorithm, shown in Fig. 7, it can be seen that the vehicle moves with high velocity also during the extinguishing phase, resulting in high power consumption.

6.2 Second case study

The second case study is aimed at testing the capability of the fuzzy trajectory planning and inverse kinematics algorithm to protect the robot from the fire. It is assumed that the robot is extinguishing the fire at a suitable safety distance but, at the time $t = 20$ s, the fire moves back of 100 m in 15 s in the direction of the robot.

The task-space trajectory for the water jet, reported in Fig. 8, clearly shows that the robot moves back with the fire, since the x component from $t = 20$ s and $t = 35$ s decreases of about 100 m. From Fig. 9, where the time history of the crisp values of the output fuzzy inference system is reported, it can be recognized that the weights β_v and β_m reach a value close to 1 that is maintained

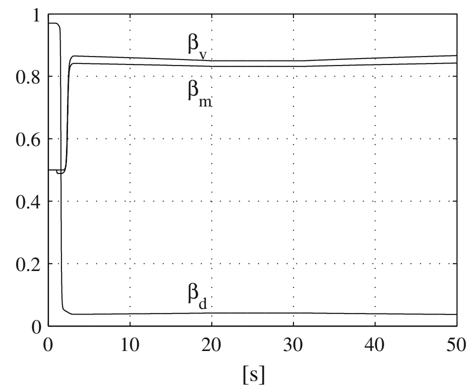

Fig. 9 Time history of the crisp outputs of the FIS in the second case study

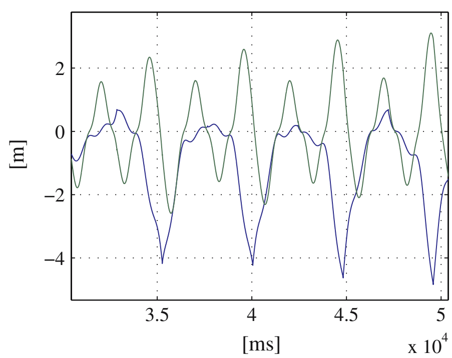

Fig. 10 Time history of the x and y components of the position errors with constant f in the third case study

during all the task, while the weight β_d is close to zero. This means that the robot follows the fire closely in order to extinguish it and moves back to keep the joint values imposed by the secondary task.

6.3 Third case study

In the third case study the same task considered in the second case study is assigned. However, it is assumed that the flow rate f decreases linearly with the distance from the tunnel entrance with a 10% slope. In this case, if a constant flow equal to the minimum value is used in the CLIK algorithm, an error between the desired and actual trajectory occurs. This error is represented in Fig. 10 in terms of the difference between the desired and actual coordinates of the impact point of the water jet on the tunnel floor. On the other hand, considering the actual value of f in the computation of the direct kinematics used in the CLIK algorithm (but neglecting the contribution of $\dot{f}$ to the Jacobian), the error is more than one order of magnitude smaller than before,

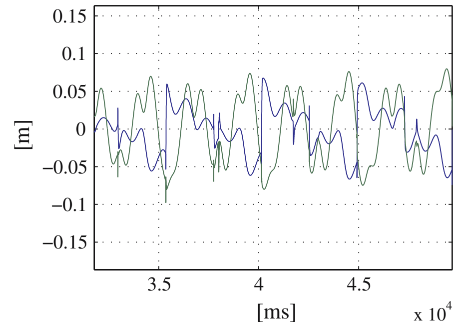

Fig. 11 Time history of the x and y components of the position errors with variable f in the third case study

as Fig. 11 shows. The achieved accuracy is more than satisfactory for the execution of the task at issue.

7 Conclusion

An automatic trajectory planner and a redundancy resolution algorithm for a fire-fighting robot have been developed in this paper. A fuzzy inference system has been adopted to support the planning of the trajectory and the inverse kinematic scheme, which is based on the weighted pseudo-inverse Jacobian with a task-priority strategy. Numerical simulations have been presented to demonstrate the effectiveness of this approach. Future work will be devoted to generalize the unified trajectory planning and inverse kinematics approach to more complex tasks, possibly using self-tuning inference systems.

References

1. Building energy and transport infrastructures for tomorrow's Europe. In: Proceedings of the Second Annual European Energy and Transport Conference, Barcelona, Spain (2002)
2. United Nation Economic Commission for Europe, Tansport Division, Safety in Tunnels. http://www.unece.org/trans/main/tunnels.html
3. Pacilio N, Sacripanti A (2001) Tunnel Intelligenti: Gallerie Dimaniche e Analisi di Rischio Variabile nel Tempo, Progetto FIT, ENEA (in Italian)
4. http://www.robogat.com
5. Nakamura Y, Hanafusa H, Yoshikawa T (1987) Task-priority based redundancy control of robot manipulators. Int J Robot Res 6(2):3–15
6. Antonelli G, Chiaverini S (2003) Fuzzy redundancy resolution and motion coordination for underwater vehicle-manipulator systems. IEEE Trans. Fuzzy Syst 11:109–120
7. Siciliano B, Villani L (2004) Inverse kinematics with fuzzy redundancy resolution for a fire fighting robot. The 9th International Symposium on Advances in Robot Kinematics, Sestri Levante, I, June/July 2004. In: Galletti C, Lenarcic J (eds) On Advances in Robot Kinematics, pp. 283–292. Kluwer Academic, Dordrecht

8. Hatton AP, Leech CM, Osborne MJ (1985) Computer simulation of the trajectories of large water jets. Int J Heat Fluid Flow 6(2):137–141
9. Sciavicco L, Siciliano B (2000) Modelling and Control of Robot Manipulators, 2nd edn. Springer, London
10. Liégeois A (1977) Automatic supervisory control of the configuration and behavior of multibody mechanisms. IEEE Trans Syst Man Cybern 7:868–871
11. Driankov D, Hellendoorn H, Reinfrank M (1995) An Introduction to Fuzzy Control. Springer, Berlin