

Fuzzy Trajectory Planning and Redundancy Resolution for a Fire Fighting Robot Operating in Tunnels

Agostino De Santis, Bruno Siciliano, and Luigi Villani
PRISMA Lab, Dipartimento di Informatica e Sistemistica
Università degli Studi di Napoli Federico II
via Claudio 21, 80125 Napoli, Italy
{agodesa,siciliano,lvillani}@unina.it

Abstract—In this paper the problem of automatic trajectory planning and inverse kinematics for a robotic system purposely designed to extinguish fires in road and railway tunnels is considered. The robot is composed by 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 solve the inverse kinematics problem in the presence of redundant degrees of freedom. 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.

Index Terms—Rescue robotics, fire fighting, inverse kinematics, fuzzy inference system, redundancy

I. INTRODUCTION

Tunnels are important infrastructures for the European Union and play a crucial role for the development of the regional economies [1]. It should be remarked, however, that most of the existing tunnels in Europe do not have a safety system adequate to the actual traffic volume. For this reason, the risk of serious tunnel disasters is significantly higher than in the past years and their costs become more and more relevant [2].

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 in tunnels 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.

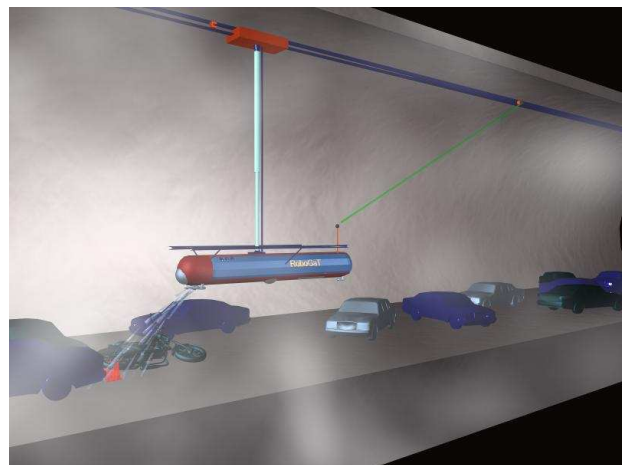


Fig. 1. A sketch of the fire fighting robotic system.

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

- a self-cooling monorail which guarantees continuous water supply;
- a mobile base moving on the monorail;
- a robotized fire fighting monitor;
- enhanced sensors as infrared video cameras, pyrometers, gas chromatographs.

The monorail may be installed on the side or on the ceiling of the tunnel, depending on the room available; 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 1000 degrees) 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 a fire hose (fire hoses are suitably disposed along the monorail at a distance of about 30 m). After the connection to the water supply, the robot may slide in both senses along the monorail thanks to the adoption of telescopic tubes. 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. From a kinematical point of view, the system 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 security distance or a dexterous posture).

In this paper, a redundancy resolution approach is adopted to solve the inverse kinematics problem, based on the weighted pseudo-inverse Jacobian and a task-priority strategy integrated with a fuzzy logic algorithm. This approach is inspired to [6], where it is adopted for underwater applications, and was applied in [7] to the robot considered here. With respect to [7], different fuzzy rules are adopted and a fuzzy trajectory planning strategy is remarkably integrated in the system.

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.

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.

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

II. 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 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). 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, which is assumed to be constant.

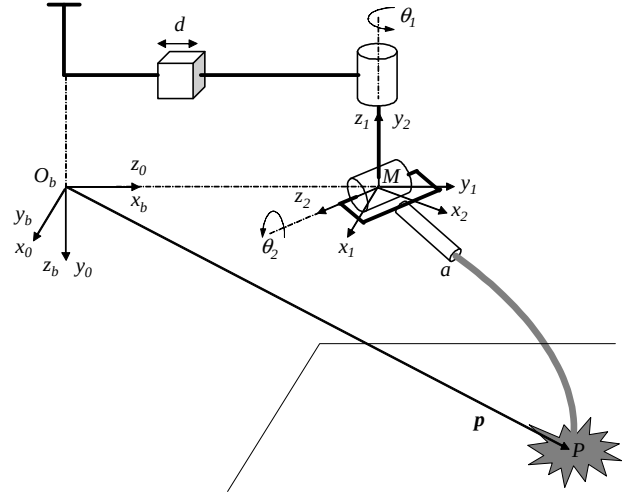


Fig. 2. Kinematic model of the fire fighting robot.

The relevant frames required to compute the robot kinematics, chosen according to the Denavit-Hartenberg convention [8], are reported in Fig. 2. It can be seen that the axes z_0 , z_1 and z_2 intersect at the point M .

The procedure for the computation of the direct kinematic equation is reported in the Appendix. The position vector $\mathbf{p} = [x \ y \ z]^T$ 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 in the form:

$$\mathbf{p}(d, \theta_1, \theta_2, v_0, z) = \begin{bmatrix} d + r(\theta_2, v_0, z)c_1 \\ -r(\theta_2, v_0, z)s_1 \\ z \end{bmatrix}, \quad (1)$$

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

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

In the above equations, s_i and c_i ($i = 1, 2$) denote $\sin(\theta_i)$ and $\cos(\theta_i)$, respectively, l is the length of the second link of the monitor and g is the magnitude of the gravity acceleration.

Notice that the variable z appears in both sides of the direct kinematic equation (1), 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.

To solve the inverse kinematics problem, the variables d , θ_1 , θ_2 , v_0 and z must be computed from the three components x , y and z of the vector $\mathbf{p}$ by inverting the three equations in (1). Since the last equation is an identity, only the first two equations have to be solved.

Hence, the solution of the problem can be found starting from the differential mapping computed from the first two components of (1)

$$\dot{\mathbf{p}}_{xy} = \mathbf{J}_{xy}(\mathbf{q}, z)\dot{\mathbf{q}} + \mathbf{J}_z(\mathbf{q}, z)\dot{z} \quad (2)$$

where

$$\mathbf{p}_{xy} = [x \quad y]^T$$

$$\mathbf{q} = [d \quad \theta_1 \quad \theta_2 \quad v_0]^T.$$

Notice that the Jacobian matrix $\mathbf{J}_{xy}(\mathbf{q}, z)$ 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 (2) is to use the pseudo-inverse of the Jacobian matrix:

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger(\mathbf{q}, z)\dot{\mathbf{p}}_{xy} - \mathbf{J}^\dagger(\mathbf{q}, z)\mathbf{J}_z(\mathbf{q}, z)\dot{z} \quad (3)$$

where

$$\mathbf{J}^\dagger = \mathbf{J}_{xy}^T(\mathbf{J}_{xy}\mathbf{J}_{xy}^T)^{-1}. \quad (4)$$

This solution corresponds to the minimization of the joint velocities in a least-square sense [8].

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 $\mathbf{J}_W^\dagger$

$$\mathbf{J}_W^\dagger = \mathbf{W}^{-1}\mathbf{J}_{xy}^T(\mathbf{J}_{xy}\mathbf{W}^{-1}\mathbf{J}_{xy}^T)^{-1} \quad (5)$$

with the (4×4) matrix $\mathbf{W}^{-1} = \text{diag}\{\beta_1, \beta_2, \beta_3, \beta_4\}$ where β_i is a weight factor belonging to the interval $[0, 1]$ such that $\beta_i = 1$ corresponds to full motion for the i -th 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 (2) of the form

$$\dot{\mathbf{q}} = \mathbf{J}_W^\dagger(\mathbf{q}, z)\dot{\mathbf{p}}_{xy} - \mathbf{J}_W^\dagger(\mathbf{q}, z)\mathbf{J}_z(\mathbf{q}, z)\dot{z} + \left(\mathbf{I}_4 - \mathbf{J}_W^\dagger(\mathbf{q}, z)\mathbf{J}_{xy}(\mathbf{q}, z)\right)\dot{\mathbf{q}}_a \quad (6)$$

where $\mathbf{I}_4$ is the (4×4) identity matrix, $\dot{\mathbf{q}}_a$ is an arbitrary joint velocity vector and the operator $\left(\mathbf{I}_4 - \mathbf{J}_W^\dagger\mathbf{J}_{xy}\right)$ 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 $\dot{\mathbf{q}}_a$ can be chosen aligned to the gradient of a scalar objective function $W(\mathbf{q})$, i.e.:

$$\dot{\mathbf{q}}_a = -\alpha k_w \frac{\partial W(\mathbf{q})}{\partial \mathbf{q}} \quad (7)$$

with $k_w > 0$ and $\alpha \in [0, 1]$, in order to achieve a local minimum for $W(\mathbf{q})$ [9].

To avoid numerical drift due to discrete-time integration, a Closed Loop Inverse Kinematics (CLIK) algorithm can

be adopted [8], which computes $\mathbf{q}$ from the integration of the vector:

$$\dot{\mathbf{q}} = \mathbf{J}_W^\dagger(\mathbf{q}, z_d)\mathbf{v} - \mathbf{J}_W^\dagger(\mathbf{q}, z_d)\mathbf{J}_z(\mathbf{q}, z_d)\dot{z}_d + \left(\mathbf{I}_4 - \mathbf{J}_W^\dagger(\mathbf{q}, z_d)\mathbf{J}_{xy}(\mathbf{q}, z_d)\right)\dot{\mathbf{q}}_a \quad (8)$$

with

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

where $\mathbf{K}$ is a (2×2) positive definite matrix gain to be chosen so as to ensure the convergence to zero of the error $\mathbf{p}_{xy,d} - \mathbf{p}_{xy}$. Notice that in (8) 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 $\mathbf{q}$ (the output of the algorithm) via the direct kinematics equation (1).

III. FUZZY INFERENCE SYSTEM

The CLIK algorithm (8) allows solving the redundancy of the system by suitably choosing the coefficients β_i of the weighted pseudo-inverse of the Jacobian (5), as well as the coefficient α in (7) for a given function $W(\mathbf{q})$ 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 automatically 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 flexible handling of the variables of interest 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 (8) according to a Mamdani Fuzzy Inference System [10].

A schematic of the inference system, for a single input single output case, is reported in Fig. 3.

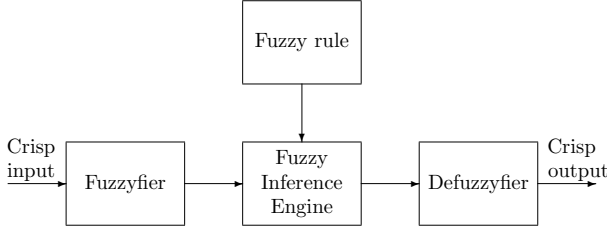


Fig. 3. Mamdani Fuzzy Inference System.

The crisp inputs of the FIS are the measurements coming from three exteroceptive sensors that are supposed to be available on the robotic system. They are: a fire detector, which provides a binary output π 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, which measures the distance δ between the robot and the fire along the rail. The crisp outputs are three values ($\beta_d = \beta_1$, $\beta_m = \beta_1 = \beta_2$, $\beta_v = \beta_4$) 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 defuzzification, the crisp output β_v is multiplied by the output π 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 the valve (i.e., forcing $\beta_v = 0$ when $\pi = 0$) 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:

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

The task-space trajectory planning algorithm receives as input the position of the fire in the tunnel and the crisp output variables of the fuzzy inference system. Then, it computes the instantaneous maximum velocity for the end point of the water jet as a weighted mean of the maximum velocities of the joints and of the water jet, where the weights are β_d , β_m and β_v , i.e.,

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} \beta_d v_{dM} + \beta_m v_{mM} + \beta_v r_{0M} \\ \beta_m v_{mM} + \beta_v r_{0M} \end{bmatrix} \quad (9)$$

where v_{dM} , v_{mM} denote the maximum velocity of the prismatic joint and of the revolute joints of the water jet, respectively, while r_{0M} represents the maximum rate of variation of the initial velocity of the water jet. When the robot is far from the fire (at a distance greater than a safety value), a rectilinear motion parallel to the rail is commanded; when the robot reaches the boundary of the safe zone, an alternate motion in the horizontal plane and centered on the fire is commanded.

A schematic of the proposed algorithm with fuzzy trajectory planning and redundancy resolution is represented in Fig. 4. An example of application is described in the simulation case study.

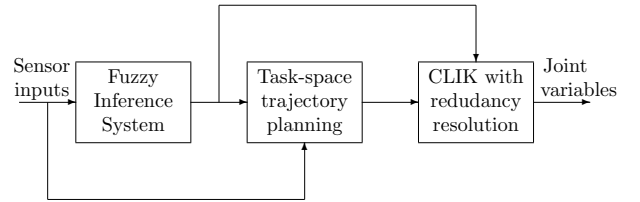


Fig. 4. Schematic of the algorithm with fuzzy trajectory planning and redundancy resolution.

IV. SIMULATION

The simulation case study has 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:

$$\begin{aligned} \theta_{1m} &= -\pi/3 \text{ rad} & \theta_{1M} &= \pi/3 \text{ rad} \\ \theta_{2m} &= -\pi/2 \text{ rad} & \theta_{2M} &= 0 \text{ rad} \\ v_{0m} &= 2.6 \text{ m/s} & v_{0M} &= 26.6 \text{ m/s.} \end{aligned}$$

The flow rate is $f = 8 \cdot 10^{-3}$ m³/s. The algorithm has been implemented at a sampling frequency of 1000 Hz, and the gains in (8) have been set to:

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

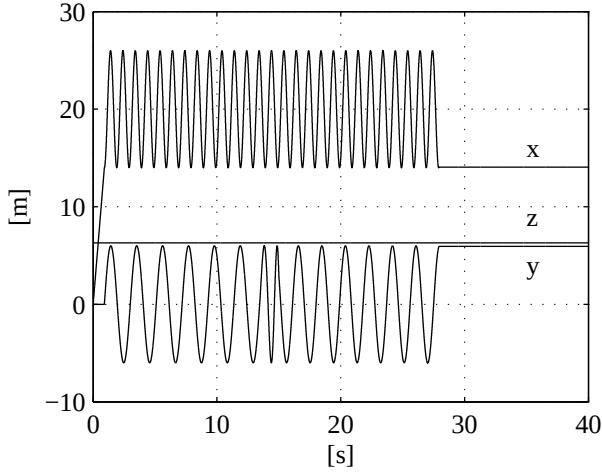


Fig. 5. Time history of the planned task-space coordinates.

A fire is sensed in the position

$$\mathbf{p}_f = [20 \ 0 \ 6.3]^T$$

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

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 fire, from 30°C to 480°C; during the fire extinguishing phase, it is 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. 5, 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 (9) (with $v_{dM} = 10$ m/s, $v_{mM} = 0.3$ rad/s, $r_{oM} = 20$ m/s²), 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. 6. It can be seen that the joint variable d is quickly variable 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,

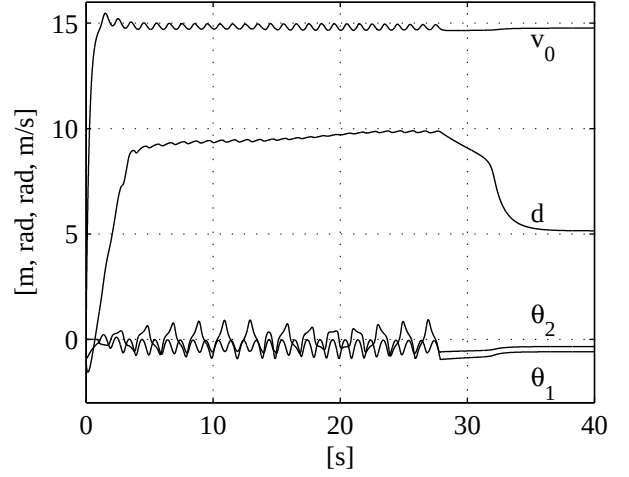


Fig. 6. Time history of the joint positions.

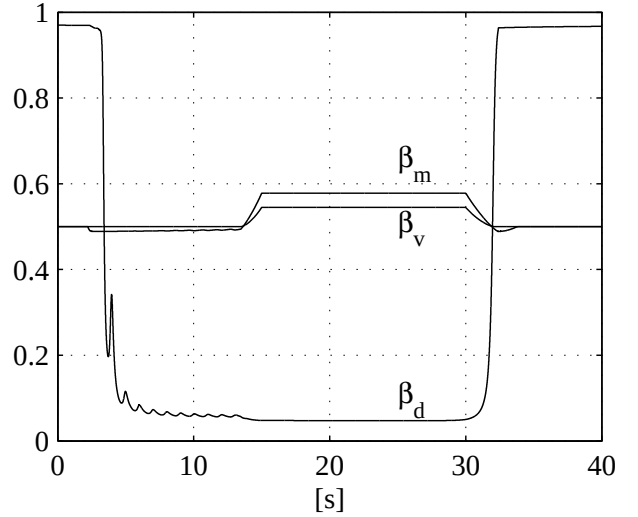


Fig. 7. Time history of the crisp outputs of the FIS.

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. 7, 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. 6. 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. 5 has been considered, while fixed weights ($\beta_d = \beta_m = \beta_v = 1$) have been used in the pseudo-inverse

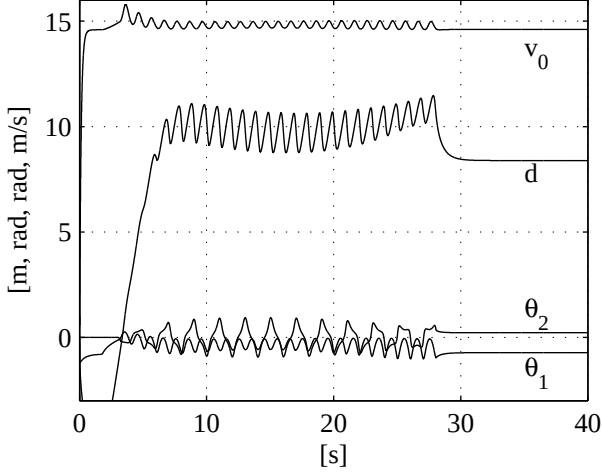


Fig. 8. Time history of the joint positions using fixed weights inverse kinematics.

Jacobian matrix. From the time history of the joint positions computed by the inverse kinematics algorithm, shown in Fig. 8, it can be seen that the vehicle moves with high velocity also during the extinguishing phase, resulting in high power consumption.

V. 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 the proposed approach.

VI. APPENDIX

The vector $\mathbf{p}$ can be expressed as:

$$\tilde{\mathbf{p}} = \tilde{\mathbf{p}}_M + \mathbf{T}_2 {}^2\tilde{\mathbf{p}} \quad (10)$$

where $\tilde{\mathbf{v}}$ denotes the representation of the vector $\mathbf{v}$ in homogeneous coordinates, i.e. $\tilde{\mathbf{v}} = [\mathbf{v}^T \ 1]^T$, $\mathbf{p}_M$ is the position vector of the point M with respect to the base frame, ${}^2\mathbf{p} = [{}^2x \ {}^2y \ {}^2z]^T$ is the position vector of the point P with respect to the frame 2, and $\mathbf{T}_2$ is the homogeneous transformation matrix of frame 2 with respect to the base frame. The matrix $\mathbf{T}_2$ can be easily computed as

$$\mathbf{T}_2 = \begin{bmatrix} c_1 & 0 & s_1 & d_1 \\ -s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (11)$$

The homogeneous representation of vector $\mathbf{p}_M$ coincides with the last column of the matrix $\mathbf{T}_2$, while the components of the vector ${}^2\mathbf{p}$ can be computed on the basis of the

projectile equations of motion written in the frame 2 as:

$${}^2x = lc_2 + v_0 c_2 t \quad (12)$$

$${}^2y = ls_2 + v_0 s_2 t - \frac{1}{2}gt^2 \quad (13)$$

$${}^2z = 0, \quad (14)$$

where t is the time variable.

The parameter t can be eliminated from Eqs. (12) and (13), 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 (10) and (11), the equality (1) follows.

REFERENCES

- [1] Building Energy and Transport Infrastructures for Tomorrow's Europe, *Proceedings of the Second Annual European Energy and Transport Conference*, Barcelona, Spain, 2002.
- [2] United Nation Economic Commission for Europe, Transport Division, Safety in Tunnels. <http://www.unece.org/trans/main/tunnels.html>.
- [3] N. Pacilio and A. Sacripanti, *Tunnel Intelligenti: Gallerie Dimaniche e Analisi di Rischio Variabile nel Tempo*, Progetto FIT, ENEA (in Italian), 2001.
- [4] <http://www.robogat.com>.
- [5] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *Int. J. of Robotics Research*, vol. 6, no. 2, pp. 3–15, 1987.
- [6] G. Antonelli and S. Chiaverini, "Fuzzy redundancy resolution and motion coordination for underwater vehicle-manipulator systems," *IEEE Trans. on Fuzzy Systems*, vol. 11, pp. 109–120, 2003.
- [7] B. Siciliano and L. Villani, "Inverse kinematics with fuzzy redundancy resolution for a fire fighting robot," *9th Int. Symp. on Advances in Robot Kinematics*, Sestri Levante, I, June/July 2004, in *On Advances in Robot Kinematics*, C. Galletti and J. Lenarcic (Eds.), Kluwer Academic Publishers, Dordrecht, NL, pp. 283–292, 2004.
- [8] L. Sciacivco and B. Siciliano, *Modelling and Control of Robot Manipulators*, (2nd Ed.), Springer-Verlag, London, UK, 2000.
- [9] A. Liégeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Trans. on Syst., Man, Cybern.*, vol. 7, pp. 868–871, 1977.
- [10] D. Driankov, H. Hellendoorn, and M. Reinfrank, *An Introduction to Fuzzy Control*, Springer-Verlag, Berlin, D, 1995.