

INVERSE KINEMATICS WITH FUZZY REDUNDANCY RESOLUTION FOR A FIRE FIGHTING ROBOT

Bruno Siciliano, Luigi Villani

Dipartimento di Informatica e Sistemistica

Università di Napoli Federico II

{siciliano,lvillani}@unina.it

Abstract The inverse kinematics problem for a fire fighting robot is considered in this paper. The robot is designed for a prompt intervention in road and railway tunnels and is composed by a self-cooling monorail vehicle carrying a fire fighting monitor. The redundant degrees of freedom of the system are used to perform additional control objectives besides the assigned task. A fuzzy technique is adopted to distribute the motion between the monitor and the vehicle while keeping the robot in a security zone. Simulation case studies are developed to demonstrate the effectiveness of the proposed approach.

Keywords: Inverse kinematics, redundancy resolution, fuzzy inference system

1. Introduction

Fire fighting is a very dangerous task, especially in close environments as road or railways tunnels. In fact, the presence of traffic, rails and other obstructions, the high temperature, the intense smoke and heat make a prompt and effective intervention very difficult. Sometimes the fire becomes uncontrollable, like for the 1999 Monte Bianco disaster. On the other hand, the risk of fires in tunnels is very high, due to the possible occurrence of road or rail accidents and to the unavoidable presence of inflammable material.

In such situations robot technology may play a crucial role to guarantee a rapid intervention and continuous water restocking. A possible approach is the ROBOGAT patent (www.robogat.com), consisting in a self-cooling monorail vehicle, mounted on either the tunnel wall or ceiling, carrying a fire fighting monitor. The rail guarantees continuous water supply and the robot may be either teleoperated or autonomous, by using enhanced sensors as thermal and infrared cameras, pyrometers, gas chromatographs.

A typical task for this type of robot may be that of reaching the fire and direct an 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. To this purpose, the task-priority redundancy resolution technique (Nakamura *et al.*, 1897) 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 for a fire fighting robot, based on the weighted pseudo-inverse jacobian and a task-priority strategy integrated with a fuzzy logic algorithm. The approach is inspired to (Antonelli and Chiaverini, 2000) where a fuzzy technique is used to distribute the motion between a manipulator and an underwater vehicle and to handle multiple secondary tasks.

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. Moreover, the FIS may activate the secondary task to keep the monitor in a dexterous configuration during some phases of the primary task.

Numerical simulations have been developed to show the effectiveness of the proposed approach and to prove its superiority with respect to a fixed weights inverse kinematic scheme.

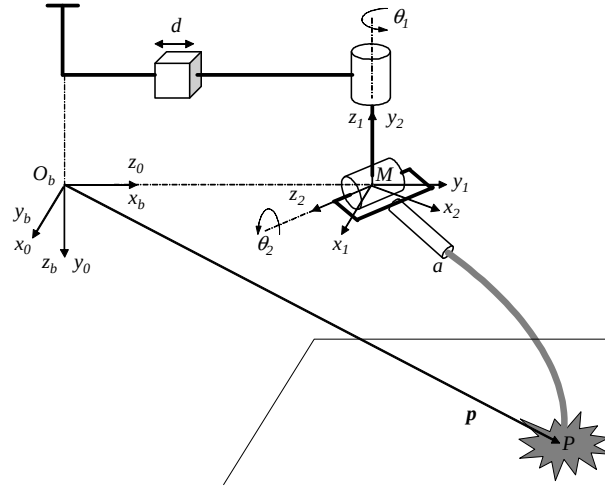


Figure 1. Kinematic model of the fire fighting robot.

2. Direct Kinematics

The kinematic structure of the fire fighting robot (see Fig. 1) 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 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 shape of the water jet can be modified also by varying the area a of the nozzle of the monitor.

The relevant frames required to compute the robot kinematics, chosen according to the Denavit-Hartenberg convention (Sciavicco and Siciliano, 2000), are reported in Fig. 1. It can be seen that the axes z_0 , z_1 and z_2 intersect at the point M .

The direct kinematics equation characterizes the position vector $\mathbf{p} = [x \ y \ z]^T$ of a point P of the water jet, referred to the base frame, in terms of the joint position vector $\mathbf{q} = [d \ \theta_1 \ \theta_2 \ a]^T$.

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

$$\tilde{\mathbf{p}} = \tilde{\mathbf{p}}_M + \mathbf{T}_2 {}^2\tilde{\mathbf{p}} \quad (1)$$

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

$$\mathbf{T}_2 = \begin{bmatrix} c_1 & 0 & s_1 & d_1 \\ -s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2)$$

where s_1 and c_1 denotes $\sin(\theta_1)$ and $\cos(\theta_1)$ respectively.

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:

$${}^2x = lc_2 + v_0c_2t \quad (3)$$

$${}^2y = ls_2 + v_0s_2t - \frac{1}{2}gt^2 \quad (4)$$

$${}^2z = 0, \quad (5)$$

where t is the time variable, l is the length of the second link of the monitor, g is the magnitude of the gravity acceleration, and v_0 is the magnitude of the initial velocity of the water flow. This quantity can be expressed in terms of the nozzle's area a as

$$v_0 = \frac{f}{s}, \quad (6)$$

where f is the flow rate, which is assumed to be constant.

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, for the purpose of this work, it is convenient computing 2x as a function of 2y in the form

$${}^2x = lc_2 + \frac{1}{g} \frac{f^2}{a^2} c_2 \left(s_2 + \sqrt{s_2^2 + 2g \frac{a^2}{f^2} (z + ls_2)} \right) = r(\theta_2, a, z), \quad (7)$$

where the equality ${}^2y = -z$ and Eq. (6) have been taken into account.

Hence, from Eq. (1), the position vector $\mathbf{p}$ can be computed as

$$\mathbf{p}(\mathbf{q}, z) = \begin{bmatrix} d + r(\theta_2, a, z)c_1 \\ -r(\theta_2, a, z)s_1 \\ z \end{bmatrix}. \quad (8)$$

Notice that the quantity $r(\theta_2, a, z)$ 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.

3. Inverse Kinematics

To solve the inverse kinematics problem, the four joint variables d , θ_1 , θ_2 and a must be computed from the three components x , y and z of the vector $\mathbf{p}$ by inverting the first two equations in (8). The complexity of these equations and the existence of multiple solutions due to the redundancy of the system, suggest to solve the problem starting from the differential mapping computed from the first two components of (8)

$$\dot{\mathbf{p}}_{xy} = \mathbf{J}_{xy}(\mathbf{q}, z)\dot{\mathbf{q}} + \mathbf{J}_z(\mathbf{q}, z)\dot{z} \quad (9)$$

where $\mathbf{p}_{xy} = [x \ y]^T$. Notice that the Jacobian matrix $\mathbf{J}_{xy}(\mathbf{q}, z)$ is (2×4) , hence the system has two redundant degrees of mobility.

The simplest way to invert the mapping (9) is to use the pseudo-inverse of the Jacobian matrix:

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger(\mathbf{q}, z)\dot{\mathbf{p}}_{xy} - \mathbf{J}^\dagger(\mathbf{q}, z)\mathbf{J}_z(\mathbf{q}, z)\dot{z} \quad (10)$$

where

$$\mathbf{J}^\dagger = \mathbf{J}_{xy}^T (\mathbf{J}_{xy} \mathbf{J}_{xy}^T)^{-1}. \quad (11)$$

This solution corresponds to the minimization of the joint velocities in a least-square sense (Sciavicco and Siciliano, 2000).

Because of the different characteristic 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 $\mathbf{J}_W^\dagger$

$$\mathbf{J}_W^\dagger = \mathbf{W}^{-1} \mathbf{J}_{xy}^T (\mathbf{J}_{xy} \mathbf{W}^{-1} \mathbf{J}_{xy}^T)^{-1} \quad (12)$$

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 $\beta = 1$ 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 (9) of the form

$$\dot{\mathbf{q}} = \mathbf{J}_W^\dagger(\mathbf{q}, z) \dot{\mathbf{p}}_{xy} - \mathbf{J}_W^\dagger(\mathbf{q}, z) \mathbf{J}_z(\mathbf{q}, z) \dot{z} + \left(\mathbf{I}_4 - \mathbf{J}_W^\dagger(\mathbf{q}, z) \mathbf{J}_{xy}(\mathbf{q}, z) \right) \dot{\mathbf{q}}_a \quad (13)$$

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.:

$$\dot{\mathbf{q}}_a = -\alpha k_w \frac{\partial W(\mathbf{q})}{\partial \mathbf{q}} \quad (14)$$

with $k_w > 0$ and $\alpha \in [0, 1]$, in order to achieve a local minimum for $W(\mathbf{q})$ (Liégeois, 1977).

To avoid numerical drift due to discrete-time integration, a Closed Loop Inverse Kinematics (CLIK) algorithm can be adopted (Sciavicco and Siciliano, 2000), which computes $\mathbf{q}$ from the integration of the vector:

$$\dot{\mathbf{q}} = \mathbf{J}_W^\dagger(\mathbf{q}, z_d) \mathbf{v} - \mathbf{J}_W^\dagger(\mathbf{q}, z_d) \mathbf{J}_z(\mathbf{q}, z_d) \dot{z}_d + \left(\mathbf{I}_4 - \mathbf{J}_W^\dagger(\mathbf{q}, z_d) \mathbf{J}_{xy}(\mathbf{q}, z_d) \right) \dot{\mathbf{q}}_a \quad (15)$$

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 (15) 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 (8).

4. Fuzzy Inverse Kinematics

The CLIK algorithm (15) allows to solve the redundancy of the system by suitably choosing the coefficients β_i of (12) and the coefficient α in (14) 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 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. For example, if the task requires to reach a fire that is initially out of the range of the monitor, a satisfactory inverse kinematics solution should generate a joint trajectory that first moves the vehicle close to the fire zone, up to a security distance, and then adjust the water jet to reach the fire while possibly keeping a dexterous posture for the monitor. Hence, β_1 should be set close to one when the robot is far from the fire and close to zero when the robot reaches the security distance; on the other hand, the choice $\alpha = 1$ should be made only when the robot is close to the fire.

A flexible handling of the variables of interest can be achieved by using a fuzzy approach, 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 (15) according to the Mamdani fuzzy inference system (Driankov *et al.*, 1995).

An example of application of the inverse kinematics algorithm with fuzzy redundancy resolution for the fire fighting robot is described in the case studies.

5. 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 a 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 are $d_m = -100$ m, $d_M = -100$ m, $\theta_{1m} = -\pi/3$ rad, $\theta_{1M} = \pi/3$ rad, $\theta_{2m} = -\pi/6$ rad, $\theta_{2M} = 0$ rad, $a_m = 3.14$ m², $a_M = 32.0$ m². The flow rate is $f = 8 \cdot 10^{-3}$ m³/s. The algorithms have been implemented at a sampling frequency of 1000 Hz, and the gains in (15) have been set to: $\mathbf{K} = \text{diag}\{5, 5\}$, $k_w = 200$.

An articulated task has been considered, which includes three phases:

- **Phase 1:** The water jet is required to reach the core of the fire in the position $\mathbf{p}_f = [30 \ 1.5 \ 6.3]^T$ m from the initial configuration $\mathbf{q}_i = [0 \ 0 \ -\pi/12 \ 5.48 \cdot 10^{-4}]^T$, corresponding to the initial position $\mathbf{p}_i = [11.97 \ 0 \ 6.3]^T$ m. A trajectory is assigned along a segment from $\mathbf{p}_i$ to $\mathbf{p}_f$, according to a quintic polynomial time law of 5 s duration.
- **Phase 2:** The water jet is required to move alternatively along a segment between the positions $\mathbf{p}_f - \boldsymbol{\delta}$ and $\mathbf{p}_f + \boldsymbol{\delta}$, with $\boldsymbol{\delta} = [1 \ 1 \ 0]^T$ m, according to a quintic polynomial time law of 0.5 s duration. This phase starts at $t = 5$ s and ends at $t = 9.5$ s.
- **Phase 3:** The water jet is required to move backwards from $\mathbf{p}_f$ to $\mathbf{p}'_f = [23.5 \ 1.5 \ 0]^T$ m according to a quintic polynomial time law of 5.5 s duration.

Notice that in the **Phase 1** the water jet is initially far from the fire and is required to reach the fire; the **Phase 2** corresponds to the operations required to extinguish the fire; the **Phase 3** represents a situation where the fire moves towards the robot.

The secondary task consists in 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 $a = 5.48 \cdot 10^{-3}$ m².

Two different case studies are considered, with and without using the fuzzy logic algorithm respectively, based on the weighted pseudo-inverse algorithm with secondary task (15).

In the first case study, the weights in (12) and (14) have been fixed to $\beta_1 = \beta_2 = \beta_3 = \beta_4 = 1$, $\alpha = 1$.

The results are reported in Fig. 2 in terms of the time history of the joint variables and of the norm of the position error $\mathbf{p}_d - \mathbf{p}$. It can be observed that during the **Phase 1** the pitch variable θ_2 exceeds the upper joint limit $\theta_{2M} = 0$, in spite of the presence of the secondary task. This is due to the fact that the robot, in order to track the trajectory imposed to the end-point of the water jet, tends to raise the second link of the monitor while the vehicle slowly moves toward the fire. During

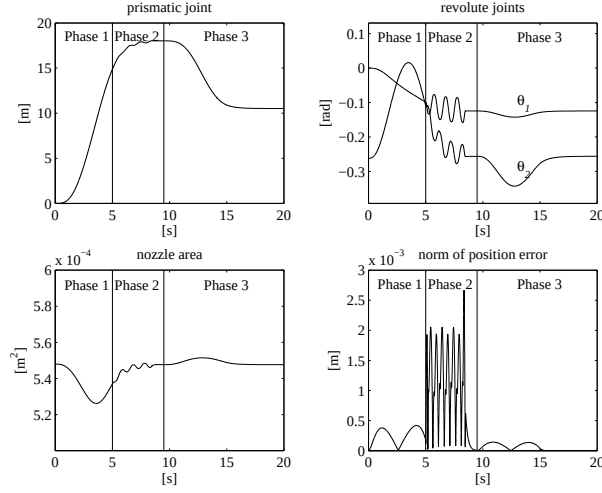


Figure 2. Time histories of the joint positions and of the norm of the position error in the first case study.

the **Phase 2** the vehicle continues to move towards the fire, up to a distance of about 12 m, and keep this distance from the fire also during the **Phase 3**. The position errors remain low along all the motion.

In the second case study, the weights in (12) and (14) are computed via a fuzzy logic approach. To reduce the number of the parameters, they have been chosen as: $\beta_1 = 1 - \beta$, $\beta_2 = \beta_3 = \beta_4 = \beta$.

The Fuzzy Inference System (FIS) receives as input the actual distance of the vehicle from the fire. Hence, a linguistic variable *distance* is considered that can take the values: $distance = \{\text{low, medium, high}\}$. The corresponding membership function is reported in Fig. 3. The outputs are given by the linguistic variables $\beta = \{\text{vehicle, monitor}\}$ and $\alpha = \{\text{low, high}\}$.

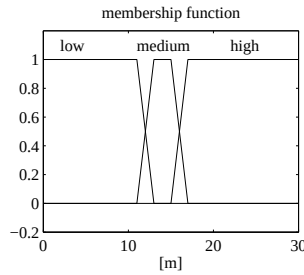


Figure 3. Membership function for the linguistic variable *distance*.

The following fuzzy rules have been used:

- 1) if (*distance* is high) then (β is vehicle) and (α is low);
- 2) if (*distance* is medium) then (β is monitor) and (α is high);
- 3) if (*distance* is low) then (β is vehicle) and (α is high).

On the basis of the first rule, when the robot is far from the fire, the vehicle is asked to approach the fire and the monitor is almost motionless. When the robot reaches a medium distance from the fire (second rule), the monitor is asked to move while keeping a dexterous posture, and the vehicle is left almost motionless. If the fire moves too close to the robot (third rule), the vehicle is asked to move in order to maintain a security distance, possibly keeping the monitor in a dexterous posture.

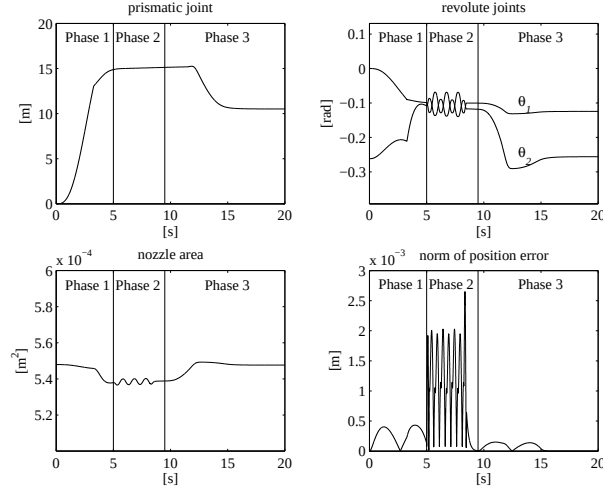


Figure 4. Time histories of the joint positions and of the norm of the position error in the second case study.

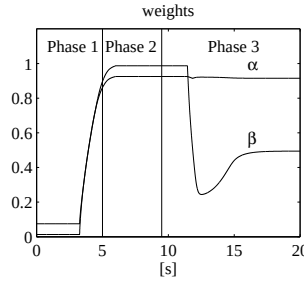


Figure 5. Time histories of the weights α and β in the second case study.

The and-or operations have been calculated by resorting to the *min-max* operations respectively. The defuzzification operation, required to achieve the numerical values of the variables $\beta \in [0, 1]$ and $\alpha \in [0, 1]$, have been performed using the centroid technique and a normalization.

The results of the simulation are reported in Fig. 4 in terms of the time history of the joint variables and of the norm of the position error $\mathbf{p}_d - \mathbf{p}$. The corresponding time histories of the variables β and α computed by the fuzzy logic algorithm are reported in Fig. 5. It can be observed that during the **Phase 1** the vehicle moves quickly toward the fire by reaching a distance of about 15 m while all the joint variables remain far from the joint limits. During the **Phase 2** the vehicle remains motionless while the monitor is in charge to track the prescribed trajectory for the water jet. During the **Phase 3** the robot initially moves the monitor, while the distance from the fire is greater than the security margin; then the vehicle moves back to keep a minimum distance from the fire of about 12 m. Notice that the time history of the position errors is similar to that of the previous case study.

6. Conclusions

An algorithm for redundancy resolution for autonomous tasks of a fire fighting robot has been developed in this paper. The scheme is based on the weighted pseudo-inverse jacobian and a task-priority strategy, combined to a fuzzy technique. Numerical simulations have been developed to demonstrate the effectiveness of the proposed approach.

7. Acknowledgements

The research work reported here was supported by MIUR and ASI.

References

- Antonelli G., and Chiaverini S. (2003), Fuzzy redundancy resolution and motion coordination for underwater vehicle-manipulator systems, *IEEE Trans. on Fuzzy Systems*, vol. 11, pp. 109–120.
- Driankov D., Hellendoorn H., and Reinfrank M. (1995), *An introduction to fuzzy control*, Springer-Verlag, Berlin, D.
- Liégeois, A. (1977), Automatic supervisory control of the configuration and behavior of multibody mechanisms, *IEEE Trans. on Syst., Man, Cybern.*, vol. SMC-7, pp. 868–871.
- Nakamura Y., Hanafusa H., and Yoshikawa T. (1987), Task-priority based redundancy control of robot manipulators, *International Journal of Robotics Research*, vol. 6, no. 2, pp. 3–15.
- Sciavicco, L., and Siciliano, B. (2000), *Modelling and Control of Robot Manipulators*, (2nd Ed.), Springer-Verlag, London, UK.