International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:9, No:5, 2015

Motion Planning of SCARA Robots for Trajectory Tracking

Giovanni Incerti

Abstract—The paper presents a method for a simple and immediate motion planning of a SCARA robot, whose end-effector has to move along a given trajectory; the calculation procedure requires the user to define in analytical form or by points the trajectory to be followed and to assign the curvilinear abscissa as function of the time. On the basis of the geometrical characteristics of the robot, a specifically developed program determines the motion laws of the actuators that enable the robot to generate the required movement; this software can be used in all industrial applications for which a SCARA robot has to be frequently reprogrammed, in order to generate various types of trajectories with different motion times.

Keywords—Motion planning, SCARA robot, trajectory tracking.

I. Introduction

THE SCARA type robots are widely used in industrial operations for "pick and place" operations, which require only a simple transfer of the end-effector between two points of the workspace. Sometimes, however, they are also used to carry out operations of cutting, welding. gluing, etc., for which an accurate trajectory tracking is necessary; in these cases an efficient motion planning is essential to enable the robot to move correctly along the trajectory. In recent years the research activities in this area have focused mainly on the development of mathematical methods for calculation and optimization of the actuators motion commands, in order to obtain the correct execution of complex motion tasks. In this paper a method is described for an efficient motion planning of SCARA robots, in order to obtain the tracking of user defined trajectories; the methodology here presented has been implemented into a general purpose calculation software, that can be employed in all industrial applications that require a frequent re-programming of the robot, to allow movements on different trajectories. The developed software has been structured according to the following points:

- definition of the robot link lengths and assignment of their rotation limits;
- computation and graphical representation of the robot workspace on the computer screen;
- mathematical definition of the end-effector trajectory;
- graphical visualization of the trajectory, in order to allow the user to verify if the curve is entirely contained in the workspace;
- definition of the function $s = s(t)$ that assigns the curvilinear abscissa of the end-effector along the trajectory as function of the time;
- computation of the end-effector motion (Cartesian components of the position, velocity and acceleration vectors);
- computation of the motion commands for the robot rotary axes;
- graphical visualization of the results, to allow a simple and rapid verification of the maximum values of velocity and acceleration.

The following sections describe in detail the calculation procedure here summarized; the matter was divided into four basic parts, consisting of sections from II to V and their respective subsections. In particular, section II briefly summarizes the SCARA robot kinematics and the procedure for the calculation of the robot workspace; section III describes the methods to calculate the actuator motion commands when the end-effector trajectory and the function $s = s(t)$ are known; the fourth section gives some details about the mathematical definition of the end-effector motion law; finally, section V illustrates the results of some simulations obtained by means of a computer program, purposely developed inside this research activity.

G. Incerti is with the Department of Mechanical and Industrial Engineering, University of Brescia, 25123 Brescia, Italy, e-mail: giovanni.incerti@unibs.it.

II. The SCARA Robot

The schematic structure of a SCARA robot is represented in Fig. 1. The system has two degrees of freedom and moves in an horizontal plane. The joint coordinates used for the kinematic analysis of the system are the angles ϑ_1 and ϑ_2, which represent respectively the absolute rotation of the first link and the relative rotation of the second link referred to the first one. Using the sign conventions in Fig. 1 the direct kinematics problem on the position variables is defined by the

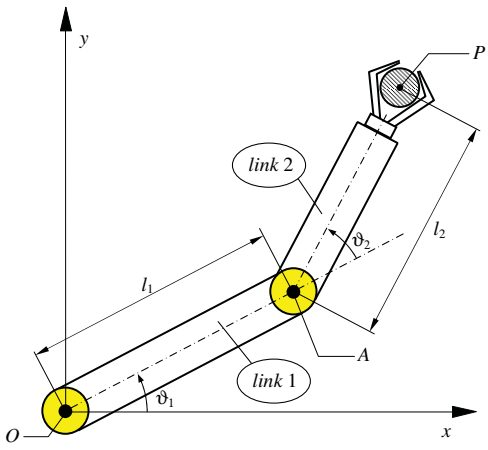

Fig. 1. Schematic structure of a SCARA robot.

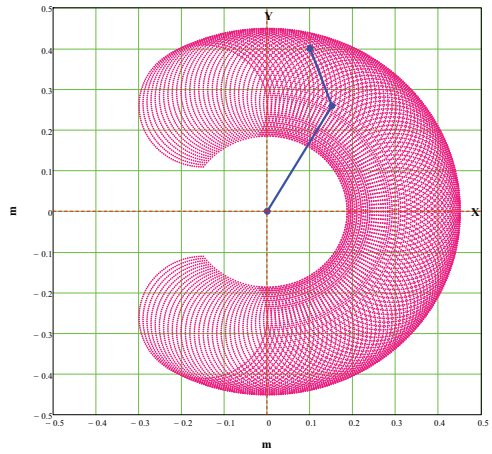

Fig. 2. Workspace of a SCARA robot, calculated with the following data: $l_1 = 300\,\text{mm}$, $l_2 = 150\,\text{mm}$, $\Delta\vartheta_1 = \pm 120°$, $\Delta\vartheta_2 = \pm 150°$.

following relationships[1]:

$$\begin{cases} x_P = f_1(\vartheta_1, \vartheta_2) = l_1 C_1 + l_2 C_{12} \\ y_P = f_2(\vartheta_1, \vartheta_2) = l_1 S_1 + l_2 S_{12} \end{cases} \tag{1}$$

where x_P and y_P indicate the end-effector coordinates (point P in Fig. 1). The robot workspace depends on the length of the links and by their rotation limits; it can be determined through the relations (1) by varying the angles ϑ_1 and ϑ_2 within the allowable ranges and mapping the coordinates of the point P in the xy plane. As an example, Fig. 2 shows the workspace of a SCARA robot, whose links can perform rotations within assigned intervals $\Delta\vartheta_1$ and $\Delta\vartheta_2$.

The velocity of point P along the x and y directions can be calculated by differentiation of (1) with respect to time; thus we have:

$$\begin{cases} \dot{x}_P = \dfrac{\partial f_1}{\partial \vartheta_1}\dot{\vartheta}_1 + \dfrac{\partial f_1}{\partial \vartheta_2}\dot{\vartheta}_2 \\ \dot{y}_P = \dfrac{\partial f_2}{\partial \vartheta_1}\dot{\vartheta}_1 + \dfrac{\partial f_2}{\partial \vartheta_2}\dot{\vartheta}_2 \end{cases} \tag{2}$$

Using the matrix notation we obtain:

$$\dot{\mathbf{p}} = \mathbf{J}\dot{\mathbf{q}} \tag{3}$$

where

$$\mathbf{p} = [x_P \; y_P]^T \qquad \mathbf{q} = [\vartheta_1 \; \vartheta_2]^T \tag{4}$$

are the vectors containing the Cartesian coordinates of the end-effector and the rotations of the links respectively, while

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial f_1}{\partial \vartheta_1} & \dfrac{\partial f_1}{\partial \vartheta_2} \\ \dfrac{\partial f_2}{\partial \vartheta_1} & \dfrac{\partial f_2}{\partial \vartheta_2} \end{bmatrix} \tag{5}$$

indicates the jacobian matrix of the robot [1].

[1]For simplicity, we have adopted the following abbreviations:

$$S_i = \sin\vartheta_i \qquad C_i = \cos\vartheta_i \qquad i = 1, 2$$
$$S_{12} = \sin(\vartheta_1 + \vartheta_2) \qquad C_{12} = \cos(\vartheta_1 + \vartheta_2)$$

The acceleration of point P along the x and y directions is obtained by time differentiation of (3):

$$\ddot{\mathbf{p}} = \mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}} \tag{6}$$

The solution of the inverse kinematic problem on the position variables is not particularly difficult; indeed, from simple geometrical considerations and from the application of Carnot's theorem to the triangle OPA (Fig. 3) we have:

$$\begin{cases} \vartheta_2 = \pm \arccos\left(\dfrac{x_P^2 + y_P^2 - l_1^2 - l_2^2}{2 l_1 l_2}\right) \\ \vartheta_1 = \operatorname{atan2}(y_P, x_P) - \operatorname{atan2}(l_2 S_2, l_1 + l_2 C_2) \end{cases} \tag{7}$$

As regards the computation of the joints velocities and accelerations, we make explicit (3) and (6) with respect to the joint variables $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$:

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}\dot{\mathbf{p}} \tag{8}$$

$$\ddot{\mathbf{q}} = \mathbf{J}^{-1}(\ddot{\mathbf{p}} - \dot{\mathbf{J}}\dot{\mathbf{q}}) \tag{9}$$

The complete expressions of the Jacobian matrix $\mathbf{J}$, its time derivative $\dot{\mathbf{J}}$ and its inverse $\mathbf{J}^{-1}$ are given in Appendix A.

III. Motion on a Predefined Trajectory

This section describes in detail the procedure to calculate the actuators motion commands of a robot when the trajectory in the xy plane is assigned and the curvilinear abscissa of the end-effector is known as function of the time. The mathematical treatment here presented is independent from the kinematic architecture of the manipulator under consideration and therefore it is valid also for not SCARA type robots.

Depending on the mode used for the trajectory definition, the calculation procedure presents some variants: the cases that may occur are described in the following subsections.

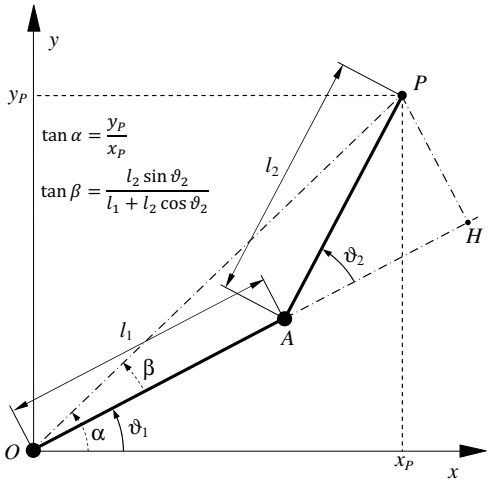

Fig. 3. Kinematic scheme of a SCARA robot.

A. Trajectory Defined in Cartesian Form

Let us suppose that the trajectory is assigned in the xy plane by means of the function $y = f(x)$; the end-effector initial position $P_i(x_i, y_i)$ on the trajectory and the final position $P_f(x_f, y_f)$ are also known (Fig. 4). We denote with the symbol s the curvilinear abscissa of the end-effector and suppose that its time dependence is given through the function $s = s(t)$; finally we suppose that the total motion time T is assigned. Then we can write:

$$s(0) = 0 \qquad s(T) = L \tag{10}$$

where L indicates the length of the trajectory; recalling that $ds = \sqrt{1 + f'(x)^2}dx$, the trajectory length can be calculated as:

$$L = \int_0^L ds = \int_{x_i}^{x_f} \sqrt{1 + f'(x)^2}dx \tag{11}$$

The velocity v of the end-effector along the trajectory is the time derivative of the curvilinear abscissa:

$$v = \frac{ds}{dt} = \frac{ds}{dx}\frac{dx}{dt} = \lambda(x)\dot{x} \tag{12}$$

where $\lambda(x) = \sqrt{1 + f'(x)^2}$. In a similar way we can calculate the acceleration a of the end-effector as the second time derivative of the curvilinear abscissa:

$$a = \frac{d^2s}{dt^2} = \frac{dv}{dt} = \frac{d}{dt}(\lambda\dot{x}) = \lambda'(x)\dot{x}^2 + \lambda(x)\ddot{x} \tag{13}$$

where:

$$\lambda'(x) = \frac{f'(x)f''(x)}{\lambda(x)} \tag{14}$$

Making explicit (13) with respect to $\ddot{x}$ we have:

$$\ddot{x} = \frac{a(t) - \lambda'(x)\dot{x}^2}{\lambda(x)} = \mathcal{A}(x, \dot{x}, t) \tag{15}$$

The differential equation (15) can be numerically integrated, if the acceleration $a(t)$ is known; in this way it is possible to determine the functions x, $\dot{x}$ e $\ddot{x}$ versus time, which represent the horizontal components of the end-effector position, velocity and acceleration. The corresponding vertical components can be easily calculated using the following relationships:

$$\begin{aligned} y &= f(x) \\ \dot{y} &= \frac{dy}{dt} = \frac{dy}{dx}\frac{dx}{dt} = f'(x)\dot{x} \\ \ddot{y} &= \frac{d^2y}{dt^2} = f''(x)\dot{x}^2 + f'(x)\ddot{x} \end{aligned} \tag{16}$$

B. Trajectory Defined in Polar Coordinates

When using polar coordinates the trajectory is defined by the function $\rho = g(\alpha)$, shown in Fig. 5; assuming the initial position $P_i(\alpha_i, \rho_i)$ and the final position $P_f(\alpha_f, \rho_f)$ are known, the length L of the trajectory is calculated by the following integral:

$$L = \int_0^L ds = \int_{\alpha_i}^{\alpha_f} g(\alpha)d\alpha \tag{17}$$

Proceeding in a similar manner to the previous case, it is possible to calculate the velocity and acceleration of the end-effector:

$$v = \frac{ds}{dt} = \frac{ds}{d\alpha}\frac{d\alpha}{dt} = g(\alpha)\dot{\alpha} \tag{18}$$

$$a = \frac{d^2s}{dt^2} = \frac{dv}{dt} = \frac{d}{dt}(g\dot{\alpha}) = g'(\alpha)\dot{\alpha}^2 + g(\alpha)\ddot{\alpha} \tag{19}$$

The differential equation that is obtained is then:

$$\ddot{\alpha} = \frac{a(t) - g'(\alpha)\dot{\alpha}^2}{g(\alpha)} = \mathcal{B}(\alpha, \dot{\alpha}, t) \tag{20}$$

The solution of (20) gives the angular coordinate α and its derivatives versus time; the radial coordinate ρ and its derivatives can be calculated by means of the formulas given below:

$$\begin{aligned} \rho &= g(\alpha) \\ \dot{\rho} &= \frac{d\rho}{dt} = \frac{d\rho}{d\alpha}\frac{d\alpha}{dt} = g'(\alpha)\dot{\alpha} \\ \ddot{\rho} &= \frac{d^2\rho}{dt^2} = g''(\alpha)\dot{\alpha}^2 + g'(\alpha)\ddot{\alpha} \end{aligned} \tag{21}$$

Transforming in Cartesian coordinates, it is immediate to obtain the position, velocity and acceleration components along the x and y axes.

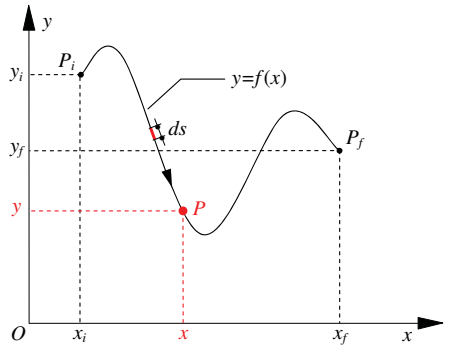

Fig. 4. Trajectory defined in Cartesian form.

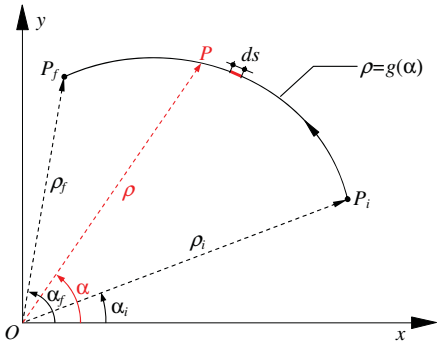

Fig. 5. Trajectory defined in polar coordinates.

C. Trajectory Defined in Parametric Form

In many cases of practical interest the trajectory can be defined in parametric form by means of two equations of the type:

$$\begin{cases} x = x(\gamma) \\ y = y(\gamma) \end{cases} \tag{22}$$

where γ is the parameter. As usual, it is necessary to specify the initial position $P_i(x_i, y_i)$ and the final position $P_f(x_f, y_f)$ of the end-effector, using the corresponding Cartesian coordinates (Fig. 6).

Recalling that $ds = \sqrt{x'(\gamma)^2 + y'(\gamma)^2}d\gamma$, the length L of the trajectory can be calculated through the relation:

$$L = \int_0^L ds = \int_{\gamma_i}^{\gamma_f} \sqrt{x'(\gamma)^2 + y'(\gamma)^2}d\gamma \tag{23}$$

where γ_i and γ_f indicate the parameter values corresponding to the initial and final positions, respectively.

Defining now the function $F(\gamma) = \sqrt{x'(\gamma)^2 + y'(\gamma)^2}$ the velocity v can be calculated as in the previously analyzed cases:

$$v = \frac{ds}{dt} = \frac{ds}{d\gamma}\frac{d\gamma}{dt} = F(\gamma)\dot{\gamma} \tag{24}$$

As regards the acceleration a, we use the following relationship:

$$a = \frac{d^2s}{dt^2} = \frac{dv}{dt} = \frac{d}{dt}(F\dot{\gamma}) = F'(\gamma)\dot{\gamma}^2 + F(\gamma)\ddot{\gamma} \tag{25}$$

where:

$$F'(\gamma) = \frac{x'(\gamma)x''(\gamma) + y'(\gamma)y''(\gamma)}{F(\gamma)} \tag{26}$$

Proceeding as usual, the differential equation that is obtained is the following:

$$\ddot{\gamma} = \frac{a(t) - F'(\gamma)\dot{\gamma}^2}{F(\gamma)} = \mathcal{C}(\gamma, \dot{\gamma}, t) \tag{27}$$

The integration of the differential equation (27) gives the parameter γ and its time derivatives versus time. The Cartesian components of the end-effector position are determined easily by (22); by differentiation, the velocity and acceleration components can be obtained.

TABLE I
END-EFFECTOR COORDINATES AND CORRESPONDING VALUES OF THE PARAMETER γ

Parameter	Point	x	y
$\gamma_0 = 0$	P_0	x_0	y_0
$\gamma_1 = 1/n$	P_1	x_1	y_1
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\gamma_k = k/n$	P_k	x_k	y_k
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\gamma_n = 1$	P_n	x_n	y_n

D. Trajectory Defined by Points

In many practical cases is difficult to provide an analytical expression for the trajectory and, therefore, it is preferred assign the curve to be followed through a series of points. With reference to Fig. 7, let us suppose to use $n + 1$ points P_k $(k = 0, 1, \ldots, n)$, defined by the corresponding Cartesian coordinates. We can associate to each point an arbitrary parameter γ_k and assume, for convenience, that this parameter is between 0 and 1, so that the values $\gamma_0 = 0$ e $\gamma_n = 1$ correspond to the initial point P_0 and the final point P_n of the trajectory respectively. Again for convenience, we adopt an uniform spacing $(\Delta\gamma = 1/n)$ for the parameter, in order to obtain the situation shown in Table I.

By associating the values of the parameter γ to the abscissas x_k it is possible to build by points the function $x = x(\gamma)$; in a similar way the function $y = y(\gamma)$ can be generated by associating the values of the parameter γ to the ordinates y_k.

To to ensure that the two functions are continuous together with their derivatives, we can perform a cubic splines interpolation [2]; using this approach, we obtain two functions similar to those defined in (22) and therefore the trajectory is again defined in parametric form.

The end-effector position, velocity and acceleration components along the x and y axes can be calculated according to the already described procedures. It should be noted that the interpolation by cubic splines is convenient, since it allows to define complex trajectories using a limited number of points.

Moreover, by implementing the procedure on the computer, the trajectory can be easily changed, by varying the

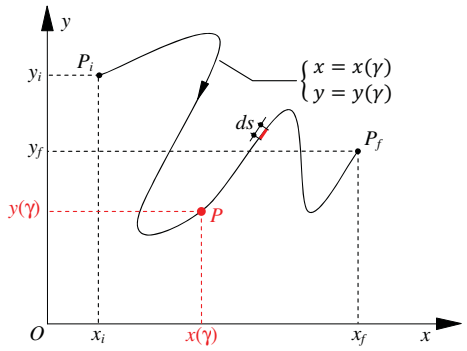

Fig. 6. Trajectory defined in parametric form.

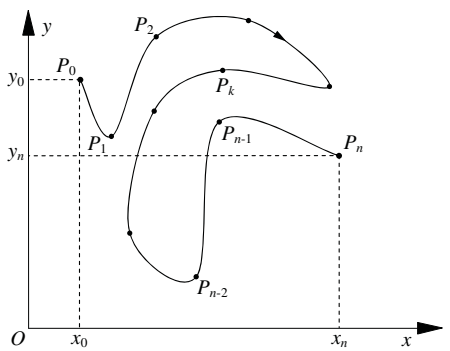

Fig. 7. Trajectory defined by points, using a cubic spline intepolation.

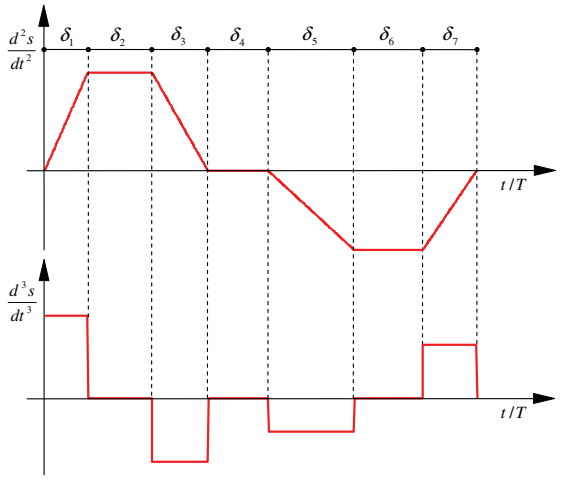

Fig. 8. Trapezoidal motion law (acceleration and jerk diagrams).

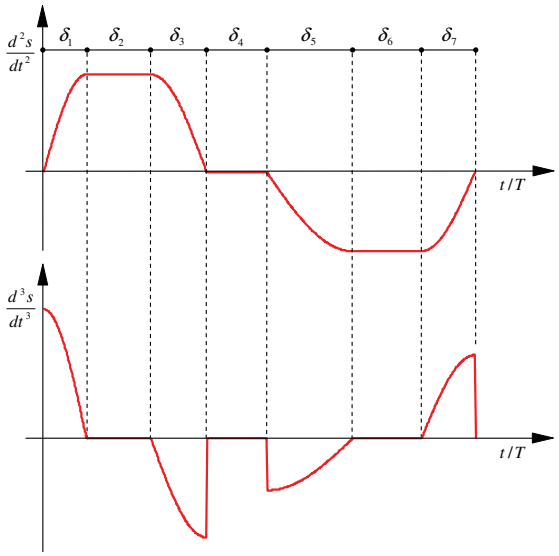

Fig. 9. Modified trapezoidal motion law (acceleration and jerk diagrams).

coordinates of some points, so as to meet the demands of the technological process in which the robot is inserted.

IV. End Effector Motion Law on the Trajectory

In Section III it was stated that, in order to calculate the motion of the robot, it is necessary to define the function $s = s(t)$ that defines the curvilinear abscissa of the end-effector along the trajectory; this function can be defined in analytical form by means of various mathematical expressions.

In the case where the motion law of the end-effector along the trajectory is constituted by three stages (acceleration, constant speed and deceleration), trapezoidal (or modified trapezoidal) acceleration profiles could be used; as is known, these profiles are well established in the technical literature and they are commonly employed for the design of cam mechanisms. For motion laws with a greater number of stages is necessary to use more complex calculation procedures, based on spline functions [3] [4] [5].

As regards the trapezoidal laws, we recall that they have an acceleration diagram consisting of seven intervals (Fig. 8); the duration of each time interval t_i is defined by the user and it is usually expressed as fraction $\delta_i = t_i/T$ of the total motion time T, which correspond to the traveling time of the end-effector along the trajectory.

The 2^{nd} and the 6^{th} time interval have constant acceleration, respectively with values a_{max} and a_{min}; the 4^{th} interval correspond to the constant velocity stage and therefore the acceleration is null; in the odd intervals (1^{st}, 3^{rd}, 5^{th} and 7^{th}) the acceleration varies linearly versus time, so that the discontinuities in the diagram are eliminated.

The total acceleration time t_a and the total deceleration time t_d are respectively:

$$t_a = \sum_{i=1}^{3} t_i \qquad t_d = \sum_{i=5}^{7} t_i \tag{28}$$

Differentiating the acceleration with respect to time, we obtain the jerk function $j(t) = da/dt = d^3s/dt^3$ whose diagram, shown in Fig. 8, has discontinuities at the points of transition from one interval to the next.

To eliminate some of these discontinuities the rectilinear functions are replaced with arcs of sinusoid, thus obtaining the modified trapezoidal acceleration and jerk curves represented in Fig. 9.

The maximum and minimum values of the acceleration are determined by imposing the following conditions on the displacement s and on the velocity v:

$$s(T) = L \qquad v(T) = 0 \tag{29}$$

These relationships generate a linear system of two equations, whose solution gives the maximum acceleration a_{max} and the minimum a_{min} of the end-effector along the trajectory (for a full discussion of this topic see [6]). As final remark, we can observe that the trapezoidal functions allow the designer to generate various types of motion profiles, simply by varying the durations of the time intervals t_i.

V. Examples of Numerical Simulations

In order to illustrate the calculation procedure described in the previous sections, this paragraph shows some examples of motion simulations on different trajectories. The calculations have been developed for robots with different geometric properties and using different motion laws (see Table II).

All numerical simulations are carried out by means of a computer program, purposely developed by the author to allow an easy planning of the robot motion.

For each of the four simulations we have reported:

- the graphical representation of the workspace and the end-effector trajectory;
- the acceleration and jerk diagrams of the end effector motion law;
- the angular position, velocity and acceleration diagrams for both robot links.

SIMULATION EXAMPLE NO.1: TRAJECTORY IN CARTESIAN FORM

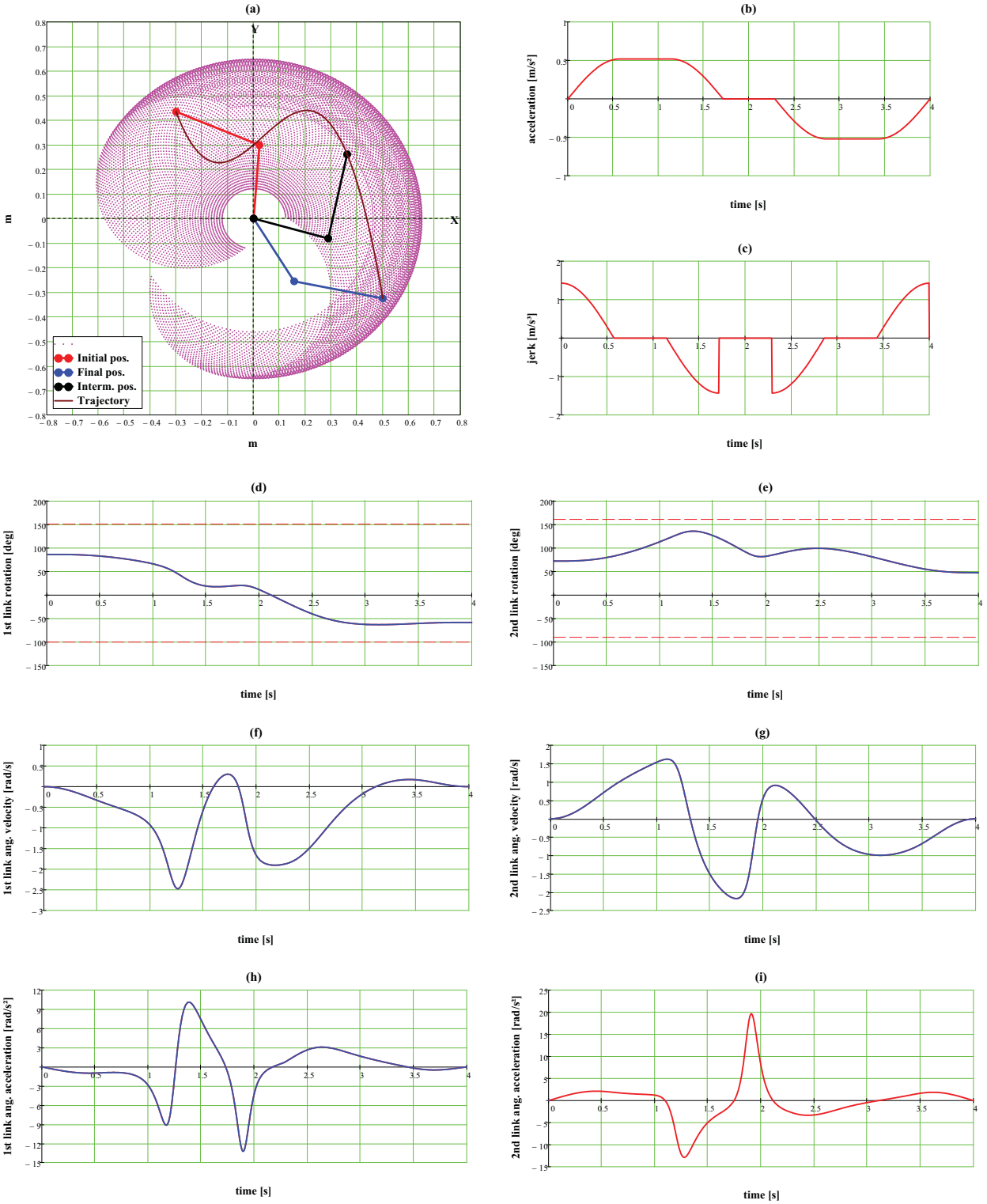

Fig. 10. Simulation results: a) Workspace of the robot and end-effector trajectory; b) Acceleration diagram d^2s/dt^2; c) Jerk diagram d^3s/dt^3; d, e) First and second link angular positions $\vartheta_1(t)$ and $\vartheta_2(t)$; f, g) First and second link angular velocities $\dot{\vartheta}_1(t)$ and $\dot{\vartheta}_2(t)$; h, i) First and second link angular accelerations $\ddot{\vartheta}_1(t)$ and $\ddot{\vartheta}_2(t)$.

Simulation Example No.2: Trajectory in Polar Form

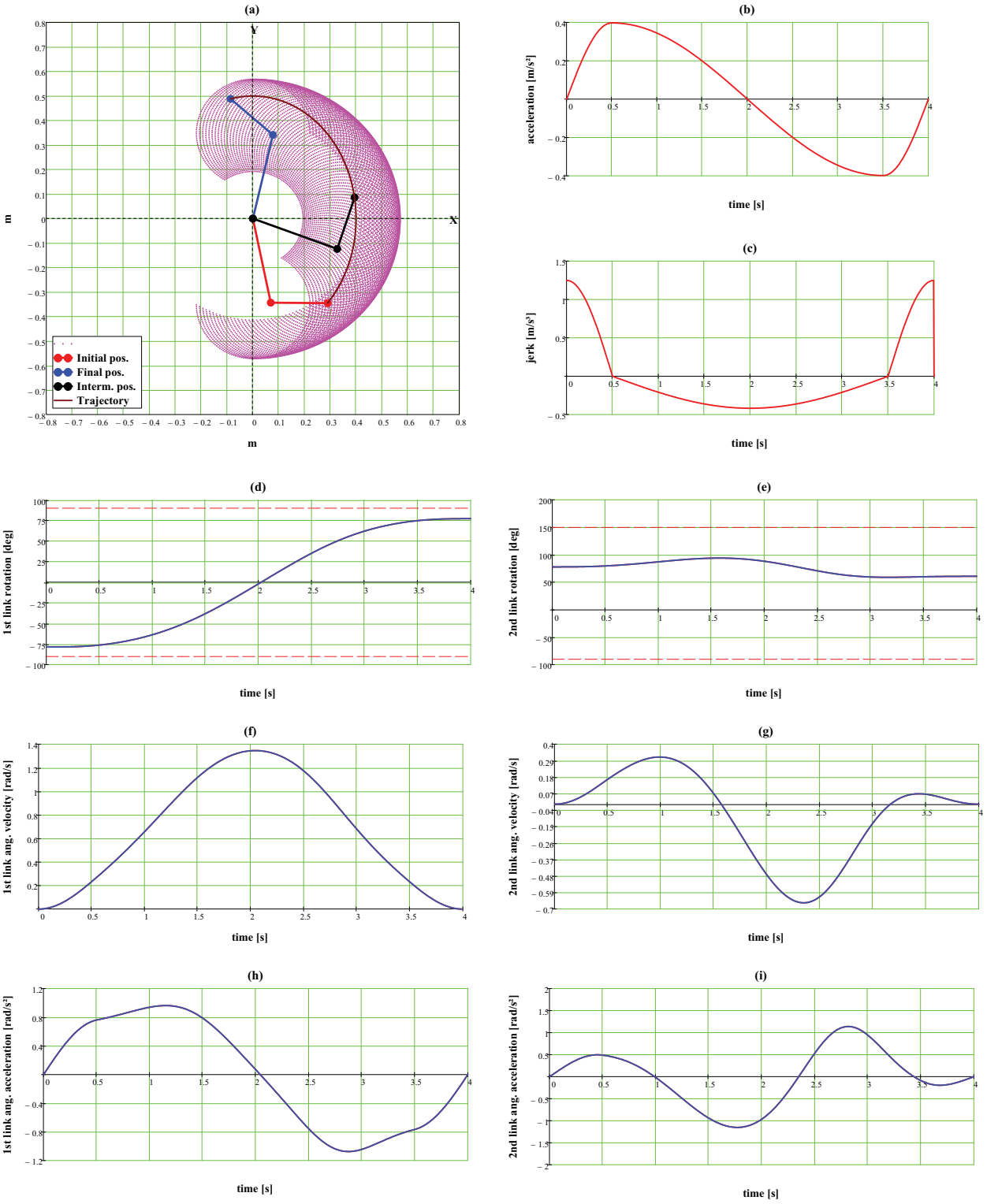

Fig. 11. Simulation results: a) Workspace of the robot and end-effector trajectory; b) Acceleration diagram d^2s/dt^2; c) Jerk diagram d^3s/dt^3; d, e) First and second link angular positions $\vartheta_1(t)$ and $\vartheta_2(t)$; f, g) First and second link angular velocities $\dot{\vartheta}_1(t)$ and $\dot{\vartheta}_2(t)$; h, i) First and second link angular accelerations $\ddot{\vartheta}_1(t)$ and $\ddot{\vartheta}_2(t)$.

Simulation Example No.3: Trajectory in Parametric Form

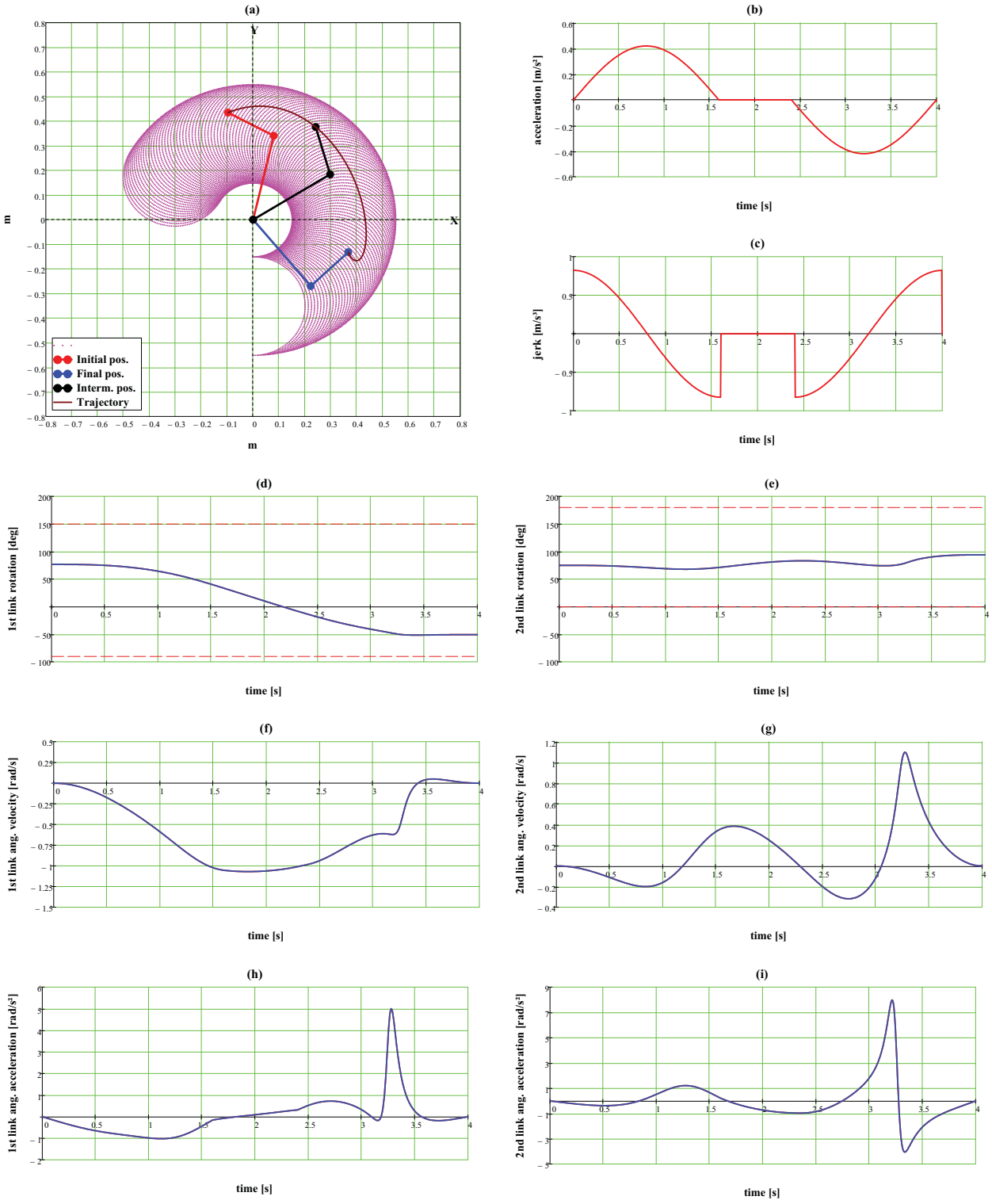

Fig. 12. Simulation results: a) Workspace of the robot and end-effector trajectory; b) Acceleration diagram d^2s/dt^2; c) Jerk diagram d^3s/dt^3; d, e) First and second link angular positions $\vartheta_1(t)$ and $\vartheta_2(t)$; f, g) First and second link angular velocities $\dot{\vartheta}_1(t)$ and $\dot{\vartheta}_2(t)$; h, i) First and second link angular accelerations $\ddot{\vartheta}_1(t)$ and $\ddot{\vartheta}_2(t)$.

SIMULATION EXAMPLE NO.4: TRAJECTORY DEFINED BY POINTS

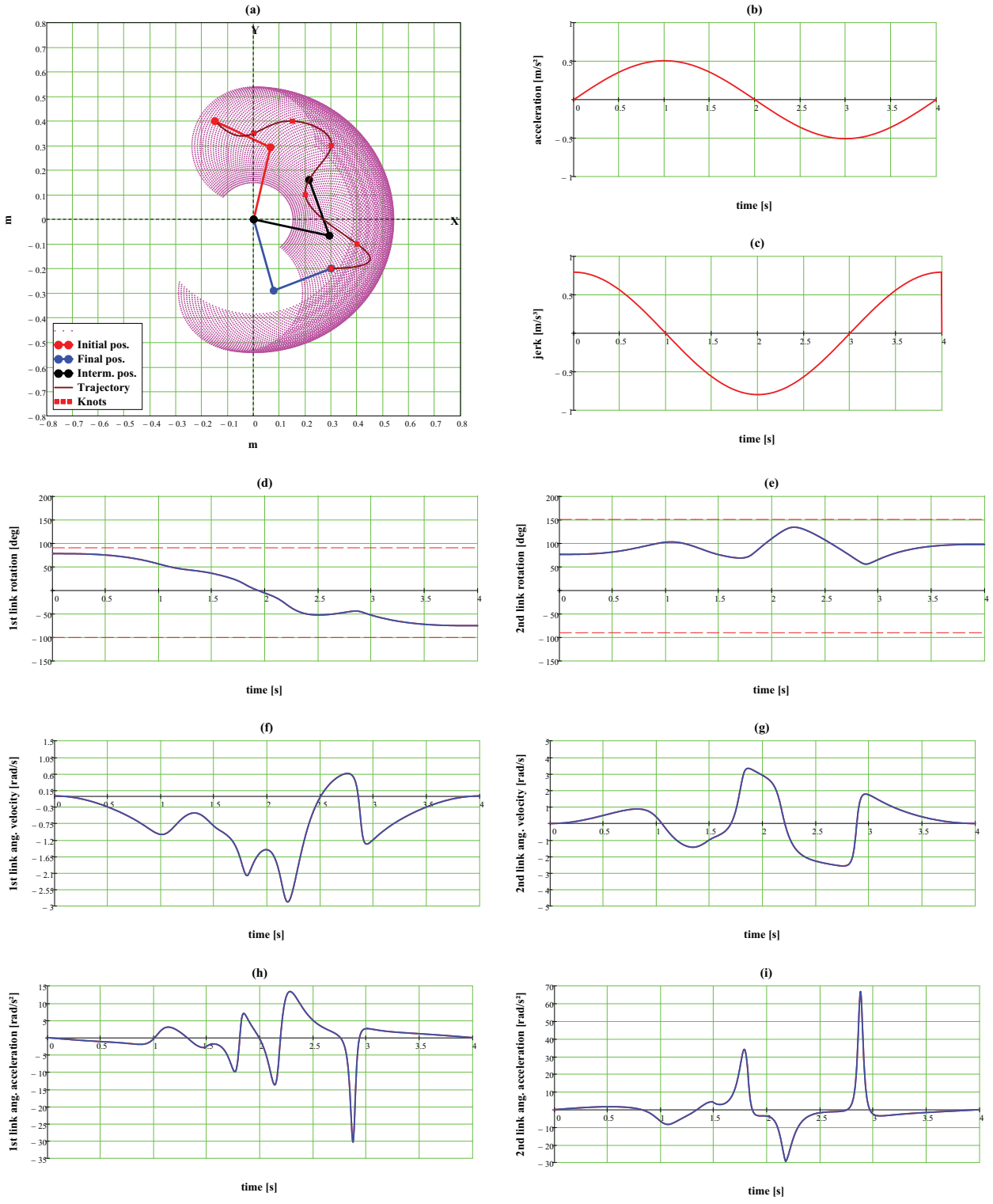

Fig. 13. Simulation results: a) Workspace of the robot and end-effector trajectory; b) Acceleration diagram d^2s/dt^2; c) Jerk diagram d^3s/dt^3; d, e) First and second link angular positions $\vartheta_1(t)$ and $\vartheta_2(t)$; f, g) First and second link angular velocities $\dot{\vartheta}_1(t)$ and $\dot{\vartheta}_2(t)$; h, i) First and second link angular accelerations $\ddot{\vartheta}_1(t)$ and $\ddot{\vartheta}_2(t)$.

TABLE II
PARAMETERS USED FOR SIMULATION EXAMPLES

Description	Example No. 1	Example No. 2	Example No. 3	Example No. 4
1st link length	$l_1 =$ 300 mm	$l_1 =$ 350 mm	$l_1 =$ 350 mm	$l_1 =$ 300 mm
2nd link length	$l_2 =$ 350 mm	$l_2 =$ 220 mm	$l_2 =$ 200 mm	$l_2 =$ 240 mm
Rotation limits (1st link)	$\vartheta_{1min} = -100°$; $\vartheta_{1max} = +150°$	$\vartheta_{1min} = -90°$; $\vartheta_{1max} = +90°$	$\vartheta_{1min} = -90°$; $\vartheta_{1max} = +150°$	$\vartheta_{1min} = -100°$; $\vartheta_{1max} = +90°$
Rotation limits (2nd link)	$\vartheta_{2min} = -90°$; $\vartheta_{2max} = +160°$	$\vartheta_{2min} = -90°$; $\vartheta_{2max} = +150°$	$\vartheta_{2min} = 0°$; $\vartheta_{2max} = +180°$	$\vartheta_{2min} = -90°$; $\vartheta_{2max} = +150°$
Starting point	$P_1 =$ (–0.3, 0.435) [m]	$P_1 =$ (–0.29, –0.345) [m]	$P_1 =$ (–0.097, 0.435) [m]	$P_1 =$ (–0.15, 0.4) [m]
End point	$P_2 =$ (0.5,–0.325) [m]	$P_2 =$ (–0.086,0.488) [m]	$P_2 =$ (0.367,–0.132) [m]	$P_2 =$ (0.3,–0.2) [m]
Trajectory length	$L =$ 1.544 m	$L =$ 1.152 m	$L =$ 1.025 m	$L =$ 1.290 m
Motion time	$T =$ 4 s	$T =$ 4 s	$T =$ 4 s	$T =$ 4 s
Acceleration profile	Modified trapezoid	Modified trapezoid	Modified trapezoid	Modified trapezoid
Parameters δ_i	1/7 - 1/7 - 1/7 - 1/7 - 1/7- 1/7 - 1/7	1/8 - 0 - 3/8 - 0 - 3/8 - 0 - 1/8	1/5 - 0 - 1/5 - 1/5 - 1/5 - 0 - 1/5	1/4 - 0 - 1/4 - 0 - 1/4 - 0 - 1/4

The first simulation has been performed using a trajectory in Cartesian form generated by the following 3rd degree polynomial function:

$$y(x) = a_3x^3 + a_2x^2 + a_1x + a_0 \tag{30}$$

where the coefficients assume the following values: $a_3 = -11\,\mathrm{m}^{-2}$, $a_2 = 1.2\,\mathrm{m}^{-1}$, $a_1 = 0.9$ and $a_0 = 0.3\,\mathrm{m}$.

For the second simulation the trajectory has been defined by an arc of ellipse in polar form whose equation is:

$$\varrho(\alpha) = \frac{ab}{\sqrt{(a\sin\alpha)^2 + (b\cos\alpha)^2}} \tag{31}$$

where $a = 0.4\,\mathrm{m}$, $b = 0.5\,\mathrm{m}$ and $-50° \le \alpha \le 100°$.

For the third simulation the trajectory has been defined by the following parametric equations:

$$\begin{aligned} x(\gamma) &= 0.2\sin\gamma + 0.125\gamma \\ y(\gamma) &= 0.36\cos\gamma + 0.03\gamma + 0.1 \end{aligned} \tag{32}$$

where $-0.3 \le \gamma \le 3.5$. Finally, for the fourth simulation, we have used trajectory defined by the knots visible in Fig. 13a; a cubic splines interpolation has been employed, as described in Section III D.

VI. CONCLUSIONS AND FUTURE DEVELOPMENTS

The paper presented a calculation method that allows an easy motion planning of a SCARA robot on a given trajectory. The work led to the development of an easy to use software application, that can be used in all industrial applications where a frequent re-programming of a SCARA robot is required. The motion laws of the actuators are processed by the calculation program in graphical form, so that the user can verify the maximum values of velocity and acceleration. The generated files are stored on the PC hard disk in tabular form and they can be easily exported in ASCII format in order to be subsequently loaded into the electronic motion controller of the robot. Currently the software is developed for robot SCARA type; however, since the calculation method is general, it is possible to extend the code also for robots with different kinematics architectures.

APPENDIX A

The jacobian matrix **J** of a SCARA robot is given by the following expression:

$$\mathbf{J} = \begin{bmatrix} -l_1S_1 - l_2S_{12} & -l_2S_{12} \\ l_1C_1 + l_2C_{12} & l_2C_{12} \end{bmatrix}$$

Its inverse is:

$$\mathbf{J^{-1}} = \frac{1}{l_1l_2S_2}\begin{bmatrix} l_2C_{12} & l_2S_{12} \\ -l_1C_1 - l_2C_{12} & -l_1S_1 - l_2S_{12} \end{bmatrix}$$

Its time derivative can be calculated as:

$$\dot{\mathbf{J}} = \begin{bmatrix} -l_1\dot{\vartheta}_1C_1 - (\dot{\vartheta}_1 + \dot{\vartheta}_2)l_2C_{12} & -l_2(\dot{\vartheta}_1 + \dot{\vartheta}_2)C_{12} \\ -l_1\dot{\vartheta}_1S_1 - (\dot{\vartheta}_1 + \dot{\vartheta}_2)l_2S_{12} & -l_2(\dot{\vartheta}_1 + \dot{\vartheta}_2)S_{12} \end{bmatrix}$$

REFERENCES

[1] Legnani G., *Robotica industriale*, CEA - Casa Editrice Ambrosiana, Milano, 2003 - (in Italian).
[2] Press W.H., Teukolsky S.A., Vetterling W.T., Flannery B.P., *Numerical Recipes in C*, Cambridge University Press, 1992.
[3] Rothbart H.A. et al., *Cam Design Handbook*, Mc Graw Hill, New York, 2004.
[4] Norton R.L., *Cam Design and Manufacturing Handbook*, Industrial Press, New York, 2001.
[5] Erdman A.G., Sandor G.N., Mechanism Design - Analysis and Synthesis Prentice Hall, Englewood Cliffs, New Jersey, 1991.
[6] Magnani P.L., Ruggieri G., *Meccanismi per macchine automatiche*, UTET, Torino, 1986 - (in Italian).

Giovanni Incerti graduated in Mechanical Engineering in 1990 at the University of Brescia, Italy. At the same university he received the Ph.D in Applied Mechanics in 1995. Actually he is associate professor at the University of Brescia and his teaching activity is related to the courses of Mechanical Vibrations, Applied Mechanics and Vibration Control. He is author of papers dealing with the dynamic analysis of cam systems, the mathematical modeling of devices for industrial automation, the mechanism design by optimization techniques and the study of robots and servomechanisms. He took part in researches funded by the Italian Council of Researches (CNR) and by the Italian Ministry of University and Research. Recently he has been scientific coordinator for the University of Brescia inside the National research project *Design and experimental validation of cam transmissions*.