CONSTANT KINETIC ENERGY ROBOT TRAJECTORY PLANNING

Zoltán ZOLLER and Peter ZENTAY

Department of Manufacturing Engineering
Technical University of Budapest
H–1521 Budapest, Hungary
e-mail: surname@manuf.bme.hu

Abstract

In Continuous Path Control (CPC) problems of robot motion planning three tasks have to be solved. These are the path planning, the trajectory planning and the trajectory tracking. The present paper focuses on the problem of trajectory planning. When solving this problem several criteria can be chosen. These can be requirement of minimum time, minimum energy, minimum force, etc. In this article, we deal with constant kinetic energy motion planning. Comparing with time optimal cruising motion, a much smoother trajectory with more power for solving manufacturing tasks is obtained. This is clearly demonstrated by the results of the sample program. The problem of kinematics and dynamics are solved for 2D polar and 2D linear robots.

Keywords: robot, trajectory planning, constant kinetic energy, LabView.

Introduction

At process planning for robots the most important task is the robot motion planning. When designing robot motion the first task is to define the geometry of the path. In motion planning two kinds of motion can be considered that are widely used in robotics: the PTP (Point To Point) and the CPC (Continuous Path Control) motions. The PTP motion is a kind of motion where only the control point of the motion is programmed and the rest of the path is extrapolated by the robot. In the CPC motion the speed and the position of the robot's TCP (Tool Centre Point) has to be estimated in every point of the path. In this article only the CPC motion is investigated.

When designing robot motion three major tasks have to be solved:

1. The path planning: The robot and its environment are known, the task is to design a path where the motion of the robot's TCP is defined through a given set of positions.
2. Trajectory planning: The path is given which has to be followed by the working point (TCP or end-effector centre point) of the robot, and the corresponding orientation of the tool or gripper attached to it. The task is to find a motion that satisfies a given goal (minimum time, minimum energy, etc.).

3. Trajectory tracking: The task in this part is to plan the control action, which guarantees the realisation of the desired trajectories with the necessary accuracy.

1. Optimal Trajectory Planning

The motion can be designed by several optimality criteria. In industrial application the most widely used criterion is the minimum time criterion where the robot travels through the given path in the shortest possible time. It is needed to reduce the production time and to increase the profit of the production. In some fields the time is not the most important, but the consumed energy is considered as the primal criteria. This can be the case where the amount of available energy is scarce. These cases can be considered in space research, long term space flights or in submarine activities both in military and deep-sea exploration.

The other case when the minimum time criteria are not advisable when a nice smooth path for the motion is needed. When using minimum time criteria we get a spiky trajectory (see literature [1,7]) which can cause unwanted shocks and vibrations. For example when handling non-rigid materials with intrusive (needle) grippers [8] a quick change in the trajectory can move or shake off the handled object from the gripper.

1.1. Time Optimal Trajectory Planning [1]

In the program we use the optimum time motion to calculate the minimum of $\dot{S}$ (see later). So it will be dealt with very briefly.

When using the parametric method for a CPC (Continuous Path Control) task the solution of the inverse geometry problem is obtained in the form

$$q_i = f_i(\lambda), \qquad i = 1, 2, \ldots n, \tag{1}$$

where: $q_i(t)$ are the joint co-ordinates;
$\quad n \quad$ is the number of the degree of freedom of the robot;
$\quad \lambda \quad$ is the length of the path.

The velocity along the path is

$$|v| = \frac{\partial \lambda}{\partial t}. \tag{2}$$

From (1)

$$\dot{q}_i = \frac{\partial f_i(\lambda)}{\partial \lambda} \frac{\partial \lambda}{\partial t} = \frac{\partial f_i(\lambda)}{\partial \lambda} |v|_i, \tag{3}$$

$$|v|_i = \frac{\dot{q}_i x}{\frac{\partial f_i(\lambda)}{\partial(\lambda)}}, \qquad i = 1, 2, \ldots n. \tag{4}$$

At the time optimal cruising motion some of the $\dot{q}_i$ values have their limit value $\dot{q}_{i\,\text{max}}$ so,

$$|v_{i\,\text{max}}| = \frac{\dot{q}_i\,\text{max}}{\frac{\partial f_i(\lambda)}{\partial(\lambda)}}, \qquad i = 1, 2, \ldots n. \tag{5}$$

The optimum velocity may only be the minimum of $|v|_{i\,\text{max}}$ values. That is

$$|v|_{\text{opt}} = \min\{|v|_{i\,\text{max}}\}, \qquad i = 1, 2, \ldots n. \tag{6}$$

For finite differences $\Delta\lambda$

$$|v_{\text{max}}|_i = \frac{\dot{q}_i\,\text{max}}{\frac{f_i(\lambda+\Delta\lambda) - f_i(\lambda)}{\Delta(\lambda)}}. \tag{7}$$

It is easy to prove (see: [2]) that using the equation of the differential motion of the robot one gets:

$$|v_{\text{max}}|_i = \frac{\dot{q}_{i\,\text{max}}}{\sum_{j=1}^{n} J_{ij}^{-1} \frac{\partial x_j}{\partial\lambda}}. \tag{8}$$

x_j are the Cartesian co-ordinates values,

$\dfrac{\partial x_j}{\partial\lambda}$ are the co-ordinates of the unit tangent vector to the path,

$\sum_{j=1}^{n} J_{ij}^{-1}$ are the components of the ith row of the inverse Jacobian matrix.

1.2. Constant Kinetic Energy Trajectory Planning

For the determination of robot motion in such a way that it would provide constant kinetic energy the equation of the dynamics of robot motion in Riemann space may be used. This parameterisation will later be used in our program. The Riemann space is a metric space where the measurement of distance is the base. This distance can be measured between two infinitesimal distant points in space [5]. The length of non-infinitesimal curves can be obtained by integration. The Euclidean space can be considered as a specialised version of Riemann space where the curvature of the space is zero. The straight lines in Riemann space are called geodetic lines [6]. These lines in Euclidean space do not appear straight. For the parameterisation of constant kinetic energy motion we will use the R scalar that is obtained from the curvature of the Riemann space to get the parameterisation of the robot manipulator [3].

1.2.1. The Parametric Model in Riemann Space [2]

The parametric model of robot motion can be obtained by the method of classical mechanics formulated by Synge (see: [3]) where the derivative of the parameter is

$$\dot{S} = \sqrt{2T},\tag{9}$$

where T is the overall kinetic energy of the manipulator system. By using the equation of dynamics of the manipulator in energetic form we get the following equation:

$$\frac{dT}{dt} = N_{dr} - N_{ex},\tag{10}$$

where N_{dr} is the momentary power developed by the drives; N_{ex} is the power given by the external forces effecting the end effector.

   If the frictional forces are neglected at the level of N_{ex} interactive power N_{dr} will be at its maximum ($N_{dr} = N_{ex}$) if the motion occurs at constant kinetic energy $T = T_0 = $ const. When this kind of motion is applied no unnecessary acceleration or deceleration will occur, so the trajectory will be much smoother and the energy for the excessive motions will be zero.

   In the parametric model this can be formulated by the $\dot{S} = \dot{S}_0 = $ constant requirement.

1.2.2. Constant Kinetic Energy Motion on a Given Path

Let the path between points A and B be given in parametric form, where λ is the arc length ($x = x(\lambda)$).

   Formulating the S parameter by using the λ parameter one gets, $\dot{S} = \dot{\lambda}\frac{1}{R}$ where $\dot{\lambda} = \frac{d\lambda}{dt}$ and R is the metric coefficient of Riemann space, an invariant scalar value. The metric coefficient R can be calculated using the formula

$$R = \frac{1}{\sqrt{\tau^T \overline{\overline{H}} \tau}},\tag{11}$$

where

$$\overline{\overline{H}} = \left(\overline{\overline{J}}^{-1}\right)^T \overline{\overline{I}} \left(\overline{\overline{J}}^{-1}\right),\tag{12}$$

$$\overline{\tau} = \left(\frac{\partial x_1}{\partial \lambda}, \frac{\partial x_2}{\partial \lambda}, \frac{\partial x_3}{\partial \lambda}\right)\tag{13}$$

is the tangent vector of the path, J the Jacobian matrix of the robot and I is the inertia matrix of the robot.

$\bar{\tau}, \bar{\bar{J}}, \bar{\bar{I}}$ are the functions of λ that is $R = R(\lambda)$ has a given value in every part of the path. Let us determine the optimal cruising speed for the first point (A) of the path. Let it be called $\dot{\lambda}_A$. With these symbols

$$\dot{S}_A = \frac{\dot{\lambda}_A}{R_A}. \tag{14}$$

To obtain

$$\dot{S} = \dot{S}_A = \frac{\dot{\lambda}}{R(\lambda)}. \tag{15}$$

In any other point the $\dot{\lambda}$ velocity may be determined properly by $\dot{\lambda} = \dot{S}_A R(\lambda)$. If $\dot{\lambda}$ is less or equal to the optimal cruising speed we obtain a motion energetically better than the normal optimal cruising motion [2].

If $\frac{\dot{\lambda}}{R} =$ constant $< \dot{S}_A$, then the constant kinetic energy motion is realised at a lower level.

When the $\dot{S}$ parameter is determined for the whole trajectory and the minimum of it is taken

$$\dot{S} = \text{Min} \, \frac{\dot{\lambda}}{R(\lambda)} = \dot{S}_E. \tag{16}$$

Then determining the velocity as:

$$\dot{\lambda} = \dot{S}_E R(\lambda), \tag{17}$$

the constant kinetic energy motion is realised for the whole trajectory.

2. Computer Program for the Investigation of the Motion

For the investigation of the constant kinetic energy motion the LabView graphical programming environment was used. The use of LabView was chosen because it is easy to make a real robot control by connecting the computer via a board to the robot's primary control. The process can be followed clearly on the XY graph of the control panel and all the changes in the parameters can be seen immediately. The inputs are given in the control panel's input slots and the output is connected to an XY graph. The curves that were obtained during the test runs of the program can be seen later in the paper.

The programming is done in LabView's diagram panel, with the usual programming blocks (icons). The special functions and formulas are programmed by the formula node of LabView.

The functions in the formula node are described in detail in the paper and in the Appendix.

The program was developed for 2D polar and 2D linear robots.

The path of robot motion in the program is a circular path (the parameterisation can be seen in Appendix 1) in the case of both robots.

The initialisation of the program must contain the geometry of the path and the parameters of the robot (inertia, mass, power of drives, etc.). In both programs the calculations along the path are done twice. In the first cycle the minimum value of $\dot{S}$ is determined. In this way the motion of the robot at constant kinetic energy will be at maximum speed set by the speed boundary of the drives. This is illustrated on a diagram in the first test result (*Fig. 4*, *Fig. 8*). The curve of the constant kinetic energy motion touches the curve of the time optimal motion but does not intersect it. It always stays under the time optimal curve.

2.1. Linear Portal Robot

First the parameters of the robot have to be set (see: *Fig. 1*): the masses of the manipulator segments (m_0, m_1), the mass of the actuator (M), the maximal speeds of the robot arms ($x1.\max$, $x2.\max$), the power of the drives ($P1$, $P2$) and a force boundary (constraint). The parameters of the path have to be set as well, in this case the co-ordinates of the centre of the circle ($x1C$, $x2C$), the radius of the circle (R), the angle of the start and end points of the path (alfa A, alfa B), and the step number.

The calculations are done in the FORMULA NODE of the program. The results of the motion can be seen on the XY GRAPH (see: *Fig. 2*).
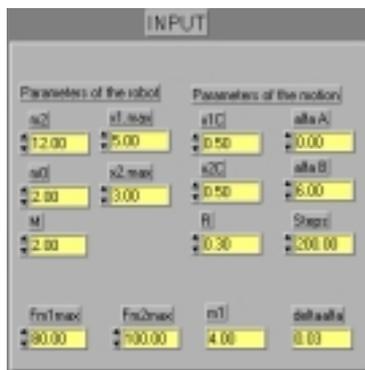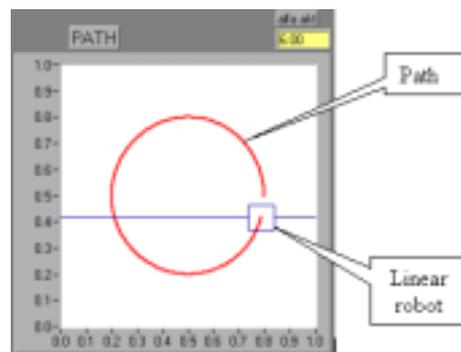

Fig. 1. Input values

Fig. 2. The motion of the actuator

The results of the first calculation cycle can be seen in *Fig. 3*.

The minimum value of $\dot{S}$ can be seen left to the diagram (S.min). The value of R along the path is seen at (A), and T is the max. kinetic energy determined for the time optimal motion.

The whole time of the constant kinetic energy motion is longer than the time of the time optimal motion but the difference is not too much.

In the second calculation cycle the angular speed values are illustrated in one diagram for both time optimal and constant kinetic energy motions (see: *Fig. 4*). In
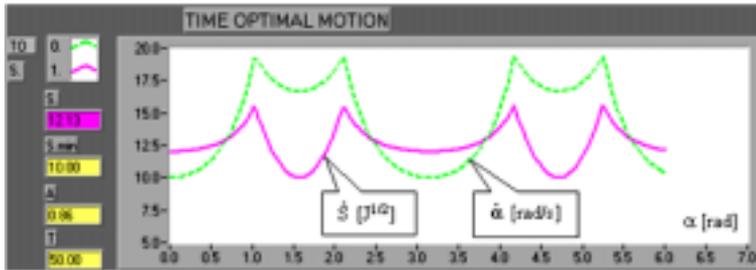
Fig. 3. The determination of the minimum of $\dot{S}$

the diagram it can be seen that the trajectory of the constant kinetic energy motion is much smoother, the speed along the path does not fluctuate as much as in the time optimal motion. In the same diagram the forces of the first joint needed for the robot's motion are also illustrated. It can be clearly seen from the diagram that by using the constant kinetic energy motion the joint forces will be less than by moving at time optimal motion. This is a great advantage of using constant kinetic energy motion.
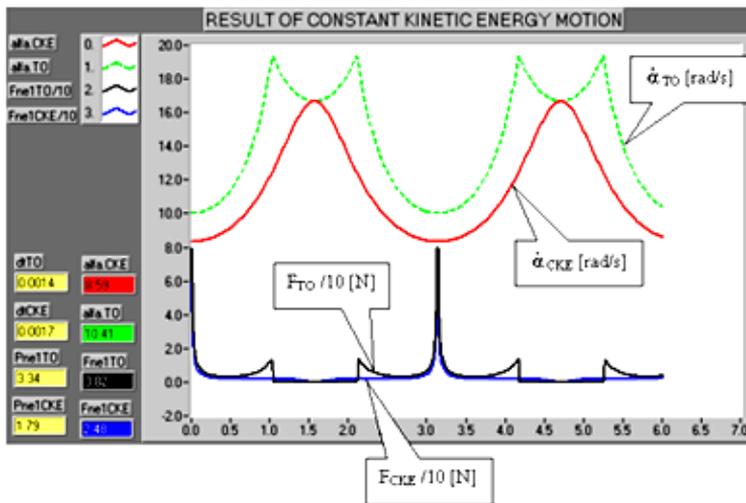

Fig. 4. Result of constant kinetic energy motion (Linear Portal Robot)

2.2. *The Polar Robot*

The parameter of the motion is the same as in case of the linear portal robot only the parameters of the robot are different (see: *Fig. 5*). The parameters of the robot

are: the mass of the robot arm (ma), the mass of the actuator (mp), the length of the robot arm (arm), the maximal speed of joints, the maximal forces that the drives can deliver (Th1, Th2) and an upper boundary (constraint) for the forces.

The calculations are also done in a FORMULA NODE. The robot motion is illustrated on a XY GRAPH (see: *Fig. 6*).
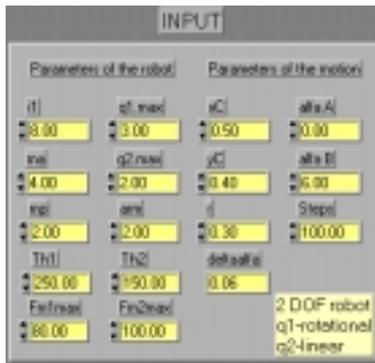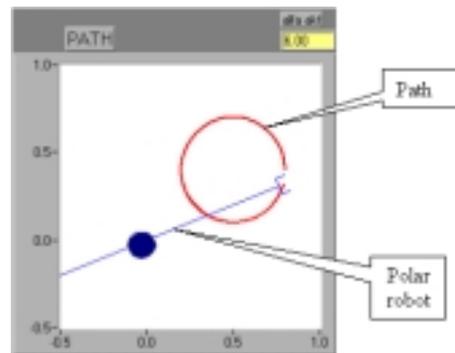

Fig. 5. Input values

Fig. 6. The motion of the actuator

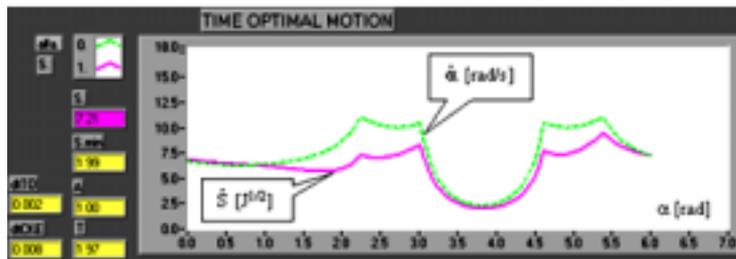The results of the first calculation cycle can be seen in *Fig. 7*.

Fig. 7. The determination of the minimum of $\dot{S}$

The minimum value of $\dot{S}$ can be seen left to the diagram (S.min). The value of R along the path is seen under (A), and T is the max. kinetic energy determined for the time optimal motion. The whole time of the constant kinetic energy motion is longer than the time of the time optimal motion but the difference is not so large.

In the second calculation cycle the angular speed values are illustrated in one diagram for both time optimal and constant kinetic energy motions (see: *Fig. 8*). In the diagram it can be seen that the trajectory of the constant kinetic energy motion is much smoother, the speed along the path does not fluctuate as much as in the time optimal motion. In the same diagram the forces of the first joint needed for the robot's motion are also illustrated. It can be seen from the diagram, as in the case of the polar robot, that by using the constant kinetic energy motion the joint

forces will be less than by moving at time optimal motion. This again illustrates the advantages of constant kinetic energy motion.
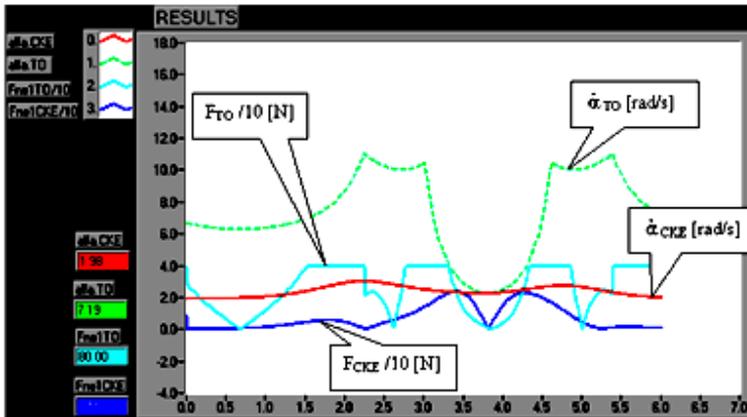

Fig. 8. Result of constant kinetic energy motion (Polar Robot)

3. Conclusions

From the test results of the program, for the two robots, it is clear that when the time is not the most important criterion of robot motion the constant kinetic energy motion provides a much better trajectory characteristic. The motion in the sample programs indicated that the constant kinetic energy motion had taken longer time to accomplish than time optimal motion, however, the difference was not too big. When the difference is as small the constant kinetic energy motion is better because it uses less energy and the trajectory is much smoother (better for handling delicate objects). Sometimes, however, the difference can be considerably larger. In this case it is better to make the calculations for both criteria and to choose the one more suitable for the task.

Acknowledgements

References

[1] SOMLÓ, J. – LANTOS, B. – CAT, P.T.: Advanced Robot Control, Akadémiai Kiadó, Budapest, 1997.
[2] SOMLÓ, J. – LOGINOV, A.: Energetically Optimal Cruising Motion of Robots, *IEEE Conference on Intelligent Engineering Systems 1997*, Budapest.
[3] SOMLÓ, J. – PODURAJEV, J.: A New Approach to the Contour Following Problem in Robot Control (Dynamic Model in Riemann Space), *Mechatronics*, Vol. 3. No. 2, pp. 241–263, 1993.
[4] TÓTH, S.: Optimal Time Cruising Trajectory Planning for SCARA, *CAMP'94 International Conference*, Budapest 1994 Sept. 13–15, pp. 54–61.
[5] HRASKÓ, P.: Introduction to the General Theory of Relativity, Tankönyvkiadó, 1997. Budapest (In Hungarian). pp. 56–103.
[6] LÁNCZOS, K.: Space Through the Ages, Academic Press Inc. London Ltd. 1979, pp. 130-240.
[7] ZENTAY, P. – ZOLLER, Z.: Time Optimal Trajectory Planning for Robots in LabView Programming System, *MicroCAD '99* Miskolc, 1999. February 24–25.
[8] ZOLLER, Z. – ZENTAY, P. – MEGGYES, A. – ARZ, G.: Robotical Handling of Polyurethane Foams with Needle Grippers, *Periodica Polytechnica*, 1999. (to be published).
[9] SZATMÁRI, SZ.: Labview Based Control of Parallel Type Robots, *FMTU'99*, 1999, Cluj-Napoca, Romania, pp. 65–68.
[10] SZATMÁRI, SZ.: Velocity and Acceleration Constrained Motion of Parallel Type Robots, *9. DAAAM Symposium (DAAAM'98)*, 1998, Cluj-Napoca, Romania, pp. 455–456.

Appendix

Appendix 1: Parameterisation of the Path

Let us parameterise a circular path. This path and its parameterisation is used in the program for the motion of both robots (see: *Fig. 1*). The conventional parameterisation is used when the centre of the circle is given by its X, Y co-ordinates and the path is parameterised by its radius (r) and the angle from the starting point (α).
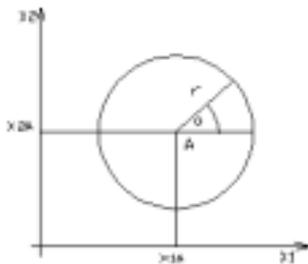

Fig. 1. Arc

To obtain the arc length parameters we use the $r^*\alpha$ (α in radians). The two co-ordinates by the arc length parameter are like the following:

$$x_1 = x_{1A} + r\cos\alpha, \tag{1}$$

$$x_2 = x_{2A} + r\sin\alpha, \tag{2}$$

where: x_{1A} is the first co-ordinate and x_{2A} is the second co-ordinate of the centre of the circular path.

By deriving x_1, x_2 by time the co-ordinate speeds are obtained:

$$\dot{x}_1 \;=\; -r \sin \alpha \dot{\alpha}, \tag{3}$$
$$\dot{x}_2 \;=\; r \cos \alpha \dot{\alpha}. \tag{4}$$

Further derivation results in the value of acceleration:

$$\ddot{x}_1 \;=\; -r(\cos \alpha \dot{\alpha} + \sin \alpha \ddot{\alpha}), \tag{5}$$
$$\ddot{x}_2 \;=\; r(-\sin \alpha \dot{\alpha} + \cos \alpha \ddot{\alpha}). \tag{6}$$

The derivative of x_1, x_2 by α sets the tangent vectors of the path, the τ vector:

$$\tau = \begin{cases} \dfrac{\partial x_1}{\partial \alpha} \;=\; -r \sin \alpha, & (7) \\[2ex] \dfrac{\partial x_2}{\partial \alpha} \;=\; r \cos \alpha. & (8) \end{cases}$$

Appendix 2: The Linear Portal Robot

Kinematics of the Linear Portal Robot

The robot we use for the model is a 2D linear robot. The actuator is moving along the x_1, x_2 axis. The mass of the actuator moving in x_1 direction is m_0, M is the mass of the handled piece so the whole moving mass in x_1 direction is $m_1 = m_0 + M$. The mass moving in x_2 direction is m_2.

From the structure of the robot we get an easy solution for the co-ordinates because the world and the joint co-ordinates are equal to each other ($\mathbf{x} = \mathbf{q}$).

Time Optimal Trajectory Planning

To obtain the time optimal joints' speeds, all the trajectory speeds for every maximal joint speed have to be calculated.

$$\dot{\alpha}_{\text{opt1}} \;=\; \frac{-\dot{x}_{1\,\text{max}}}{r^* \sin \alpha}, \tag{9}$$

$$\dot{\alpha}_{\text{opt2}} \;=\; \frac{\dot{x}_{2\,\text{max}}}{r^* \cos \alpha}. \tag{10}$$

From the two joint speeds the smaller can be realised [1].

$$\dot{\alpha}_{\text{opt}} = \text{Min}\{\dot{\alpha}_{\text{opt1}}, \dot{\alpha}_{\text{opt2}}\}. \tag{11}$$

From this every joint speed can be calculated:

$$\dot{x}_1 = -r \sin \alpha \dot{\alpha}_{\text{opt}}, \tag{12}$$

$$\dot{x}_2 = r \cos \alpha \dot{\alpha}_{\text{opt}}. \tag{13}$$

Constant Kinetic Energy Trajectory Planning

Let us observe the constant kinetic energy motion for a 2D robot when it is moving along a circular trajectory.

The inertia matrix of the robot is $\begin{bmatrix} m_1 & 0 \\ 0 & (m_1 + m_2) \end{bmatrix}$, this matrix is a symmetrical matrix which contains non-zero elements only in its diagonal.

The $\overline{H}$ matrix in this case is equal to the inertia matrix because the Jacobian matrix of a linear portal robot is a unit matrix. From this the R scalar is easily obtained. The $\tau^T \overline{\overline{H}} \tau$ in this case looks like:

$$[-r \sin \alpha \; r \cos \alpha] \begin{bmatrix} m_1 & 0 \\ 0 & (m_1 + m_2) \end{bmatrix} \begin{bmatrix} -r \sin \alpha \\ r \cos \alpha \end{bmatrix}$$

$$= [-m_1 r^2 \sin^2 \alpha + (m_1 + m_2) r^2 \cos^2 \alpha]. \tag{14}$$

From this the value of R is:

$$R = \frac{1}{\sqrt{-m_1 r^2 \sin^2 \alpha + (m_1 + m_2) r^2 \cos^2 \alpha}}. \tag{15}$$

If we take r and use the $\sin^2 \alpha = 1 - \cos^2$, we obtain the easy form for R:

$$R = \frac{1}{r} \frac{1}{\sqrt{m_1 + m_2 \cos^2 \alpha}}. \tag{16}$$

From this $\dot{\lambda} = \sqrt{2T} R$ is obtained.

Accessible External Forces

The less energy for accelerating or decelerating the robot arm (P_{dr1}, P_{dr2}) is used the more energy (or force) is left for manufacturing. If the robot does not use any energy for acceleration (deceleration) all the force in the robot arm can be used for manufacturing. The equations are calculated by their component equations:

$$\frac{\frac{1}{2} m_1 (\Delta v_1)^2}{\Delta t} = P_{\text{dr1}} - P_{\text{ex1}} \to P_{\text{ex1}}, \tag{17}$$

$$\frac{\frac{1}{2} m_2 (\Delta v_2)^2}{\Delta t} = P_{\text{dr2}} - P_{\text{ex2}} \to P_{\text{ex2}}. \tag{18}$$

From the external powers (P_{ex1}, P_{ex2}) the accessible force can be obtained:

$$F_{ex1} = \frac{P_{ex1}}{v_1}, \tag{19}$$

$$F_{ex2} = \frac{P_{ex2}}{v_2}. \tag{20}$$

Appendix 3: The Polar Robot

Kinematics of the Polar Robot

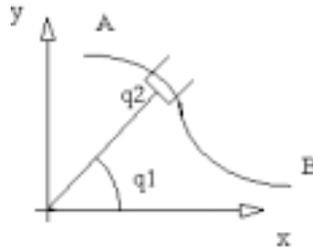The figure of a 2D polar robot can be seen in *Fig. 2.*

Fig. 2. The polar robot

Two tasks can be formulated by observing the polar robot:

1. What kind of motion (speeds and positions) is obtained by given joint co-ordinates and joint speeds. This is the subject of direct kinematics. In this case the next equations can be formulated:

$$x = q_2^* \cos q_1, \tag{21}$$
$$y = q_2^* \sin q_1, \tag{22}$$
$$\dot{x} = \dot{q}_2^* \cos q_1 - q_2^*(\sin q_1)\dot{q}_1, \tag{23}$$
$$\dot{y} = \dot{q}_2^* \sin q_1 - q_2^*(\cos q_1)\dot{q}. \tag{24}$$

2. By giving the positions and speeds of the TCP of the robot the joint speeds and positions have to be calculated. This is called the inverse kinematics task.

To solve this task the next equation can be formulated:
The joint co-ordinates:

$$q_1 = \operatorname{arctg} \frac{y}{x}, \tag{25}$$

$$q_2 = \sqrt{x^2 + y^2}. \tag{26}$$

The joint speeds:

$$\dot{q}_1 = \frac{\dot{y}x - \dot{x}y}{x^2 + y^2}, \tag{27}$$

$$\dot{q}_2 = \frac{x\dot{x} - y\dot{y}}{\sqrt{x^2 + y^2}}. \tag{28}$$

The accelerations of joints:

$$\ddot{q}_1 = \frac{x\ddot{y} - \ddot{x}y}{x^2 + y^2} - \frac{2(x\dot{y} - \dot{x}y)(x\dot{x} + y\dot{y})}{(x^2 + y^2)^2}, \tag{29}$$

$$\ddot{q}_2 = \frac{\dot{x}^2 + x\ddot{x} + \dot{y}^2 + y\ddot{y}}{\sqrt{x^2 + y^2}} - \frac{(x\dot{x} + y\dot{y})^2}{(x^2 + y^2)^{3/2}}. \tag{30}$$

Time Optimal Trajectory Planning

For obtaining the optimal trajectory we need to get the relations between joint and the co-ordinate speeds by using the α parameter. By substituting the parametrical equations into the kinematics equations earlier determined for the polar robot we obtain these relations

$$\dot{q}_1 = \frac{r\cos\alpha\,\dot{\alpha}(x_c + r\cos\alpha) + r\sin\alpha\,\dot{\alpha}(y_c + r\sin\alpha)}{(x_c + r\cos\alpha)^2 + (y_c + r\sin\alpha)^2}, \tag{31}$$

$$\dot{q}_2 = \frac{-(x_c + r\cos\alpha)r\sin\alpha\,\dot{\alpha} + (y_c + r\sin\alpha)r\cos\alpha\,\dot{\alpha}}{\sqrt{(x_c + r\cos\alpha)^2 + (y_c + r\sin\alpha)^2}}. \tag{32}$$

From these equations using $\dot{\alpha}$ we get:

$$\dot{q}_1 = f_1(\alpha)\dot{\alpha}, \tag{33}$$
$$\dot{q}_2 = f_2(\alpha)\dot{\alpha}. \tag{34}$$

The joint speeds are at maximal, so

$$\dot{q}_{1\,\text{max}} = f_1(\alpha)\dot{\alpha}_{01}, \tag{35}$$
$$\dot{q}_{2\,\text{max}} = f_2(\alpha)\dot{\alpha}_{02}. \tag{36}$$

The values of angular velocities that correlate to the maximal joint velocities are obtained by taking the angular velocities from these equations that are proportional to the trajectory speeds:

$$\dot{\alpha}_{01} = \frac{\dot{q}_{1\,\text{max}}}{f_1(\alpha)}, \tag{37}$$

$$\dot{\alpha}_{02} = \frac{\dot{q}_{2\,\text{max}}}{f_2(\alpha)}. \tag{38}$$

From the two values the smaller can be realised:

$$\dot{\alpha}_0 = \text{Min}\,\{\dot{\alpha}_{01}, \dot{\alpha}_{02}\}. \tag{39}$$

Constant Kinetic Energy Trajectory Planning

The Jacobian matrix of the polar robot is:

$$\overline{\overline{J}} = \begin{bmatrix} -q_2 \sin q_1 & \cos q_1 \\ q_2 \cos q_1 & \sin q_1 \end{bmatrix}, \tag{40}$$

The inverse Jacobian matrix is (this is needed for the calculations):

$$\overline{\overline{J}}^{-1} = \begin{bmatrix} -\dfrac{\sin q_1}{q_2} & \dfrac{\cos q_1}{q_2} \\ \cos q_1 & \sin q_1 \end{bmatrix}. \tag{41}$$

The inertia matrix of the polar robot is:

$$\overline{\overline{I}} = \begin{bmatrix} I_0 + m_a(q_2^2 - q_2 kar) + \frac{1}{3}m_a kar^2 + q_2^2 m_p & 0 \\ 0 & m_a + m_p \end{bmatrix}. \tag{42}$$

The value of the R scalar is:

$$R = \frac{1}{\sqrt{\tau^T \overline{\overline{H}} \tau}}, \quad \text{where:} \quad \overline{\overline{H}} = \left(\overline{\overline{J}}^{-1}\right) \overline{\overline{I}} \left(\overline{\overline{J}}^{-1}\right). \tag{43}$$

By substituting the previous equations into R we get the value for it:

$$R = 1 / \Bigg(\Bigg(r^2(m_a + m_p)$$

$$-\frac{r^2(3q_2^2(m_a + m_p) + 3q_2 kar\, m_a - (3i + kar^2 m_a))\cos(\alpha)^2}{3q_2^2} \Bigg) \cos(q_1)^2$$

$$-\frac{2r^2(3q_2 kar\, m_a - (3i + kar^2 m_a))\sin(\alpha)\cos(\alpha)\sin(q_1)\cos(q_1)}{3q_2^2} \tag{44}$$

$$+\frac{r^2(3q_2^2(m_a + m_p) - 3q_2 kar\, m_a + 3i + kar^2 m_a)\sin(\alpha)^2\sin(q_1)^2}{3q_2^2}$$

$$+r^2(m_a + m_p)\cos(\alpha)^2 \Bigg)^{1/2}.$$

Determining the External Forces

The parameters of both time optimal and constant kinetic energy motions were determined. With these parameters we get the forces/torques equations in a parametric form:

$$
\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} =
\begin{bmatrix} I_0 + m_a(q_2^2 - q_2 R) + \frac{1}{3} m_a R^2 q_2^2 m_p & 0 \\ 0 & m_a + m_p \end{bmatrix}
\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix},
$$

$$
+ \begin{bmatrix} [m_a(2q_2 - R) + 2q_2 m_p]\dot{q}_2\dot{q}_1 \\ -\dfrac{1}{2}[m_a(2q_2 - R) + 2q_2 m_p]\dot{q}_1^2 \end{bmatrix}. \tag{45}
$$

By subtracting these values from the forces/torques given by the drives we get the maximal forces/torques for manufacturing:

$$
\begin{aligned}
F_{\text{ne1}} &= F_{\text{dr1}} - \tau_1, \\
F_{\text{ne2}} &= F_{\text{dr2}} - \tau_2.
\end{aligned} \tag{46}
$$