International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878
Volume 2 Issue 2, February 2014

Decision of Robotics Trajectory based on Curve Fitting Techniques

[1]Abhinash, [2]Ashwani Kumar Jha

CAAD Department
[1,2]BIT Mesra, Ranchi-835215

Abstract: ***The proposed study is based upon the estimation of equation of curve path which is used as robot trajectory. It enables the robot to identify the hurdle and avoiding it by opting curvilinear path. The decision of curvilinear path is based upon occurrence of obstacle in path. In this case robots move continuously in forward direction but while encountering the hurdle, it slightly changes the direction and move on the curvilinear path. The equation of required curvilinear path is evaluated using Matlab program and further this equation is feed into the microcontroller for simulating the path in real world.***

Keywords: Mobile robots, Trajectory generation, Obstacles avoidance, Time complexity.

1. Introduction

In order to operate in the human createdenvironment the mobile robot should understand its own dynamics. It is therefore natural to incorporate different models for defined trajectory motion. Trajectory generation is one of the major problems which should be incorporated in order to reach from initial position to the final position. While it is require to study these two boundary conditions to get the classical solution to the given problem. In order to generate a smooth trajectory so that there would not be any collision or avoidance we need to get a nonlinear differential equation which is further solved to get the require trajectory or path.In addition to the given proposition the problem is complicated when we have to solve this nonlinear differential equation.

In this context we introduce our proposed method to solve that require nonlinear differential equation to extract the curvilinear path which is helpful for this purpose. This trajectory generated curvilinear path helps the robot to reach its required destination. For the autonomous operation trajectory generation is used to acquire specific locations in context to the fixed goal point.Trajectory generation can also be cast as the core component of the target reaching planning. It can also be used as the mechanics to encode different location in order to specify the trajectory of path to reach the destination. In this context trajectory generation is the key of encoding various location in space so that it can meet all require constraints to generate a smooth nonlinear path.This not only helps in achieving the goal but also saves time to the great extends. This give rise to an algorithm which is required for selecting the shortest path or trajectory so as to achieve the target in the minimum possible time allotted.In the context of robot path tracing , most researchers in trajectory generation has deal with few obstacles or obstacles free path subjected to constraints like smooth linear environment. Two basis techniques exist. The first is nearness diagram algorithm. This technique produces a solution for limited number of obstacles present in front of the source in left and right trajectory. The second technique is the use of Bezier curve which produces a S-shaped path for this purpose having high degree of accuracy. Some of the first work in trajectory generation involved composing optimal paths where the obstacles are in front of the robot in different trajectory [11]. The desire of meeting the higher order geometric primitives was intended to enable different platforms or environment for robot path tracing.

The present generation of mobile robots is content to move the robots from initial position to the final position and perhaps avoid the obstacles within the path followed by the robot.Further the real time algorithm is required so as to gather the information for the application. Continuous motion of mobile robots is the core task to be performed.

2. Experimental Methodology

Here we are basically working on the movement or locomotion of bipedal robots. It may be divided into two parts one single-support phase (that is one foot on the ground) and double-support phase. Simulation has been used to test and check the algorithms used for locomotion of the robots to reduce the chances of failure in its hardware functioning of the robots parts.

One of the best examples of such kind of robot is 5 DOF robots which has introduced by [1]-[3].There are many mathematical formulation which can be used for motion of bipedal robots are: the Lagrangian formulation and the Newton-Euler formulation. In present study path is searched on the basis of confronting obstacles. The trajectory generation algorithm has a new approach for defining suitable path for robots. It consists of two levels one is trajectory generation with numerical approach having minimum error and second level is path prediction. The initial and final level boundary condition defines the input to the trajectory generation while the output is the trajectory. This approach not only saves time but also provides the best trajectory for the robot to meet the destination. Here in this approach the curvilinear path completely depends upon the path to be traced.Robotics movement deals with the mathematical formulations of the equations of robotics limb motion. The dynamic equations of manipulator motion are a set of equations describing the dynamic behavior of the manipulator. Such equations of motion are useful for computer simulation of robot arm motion, the design of suitable control equations for a robot arm, and the evaluation of the kinematic design and structure of a robot arm. Various approaches are available to formulate robot arm dynamics, such as the Lagrange-Euler, the Newton-Euler, the recursive Lagrange-Euler, and the generalized d'Alembert principle formulations [4]. Deriving the dynamic model of a manipulator using the L-E method [9]is simple and systematic. The resultant equations of motion, excluding the dynamics of the electronic control device and the gear friction, are a set of second order, coupled nonlinear differential equations.

One approach that has the advantage of both speed and accuracy is based on the N-E vector formulation [5] was used in this work. The derivation is simple, although messy, and involves vector cross-product terms. The resultant dynamic equations, excluding the dynamics of the control device and the gear friction, are a set of forward and backward recursive equations. These equations can be applied to the robot links sequentially. There are two problems related to manipulator dynamics that are important to solve:

- Inverse dynamics in which the manipulator's equations of motion are solved for given motion to determine the generalized forces. and
- Direct dynamics in which the equations of motion are integrated to determine the generalized coordinate response to applied generalized forces.

The equations of motion for an *n*-axis manipulator are given by:

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) \qquad (1)$$

Where, if we have 4DOF

$$q = [\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4]^T,\ \dot{q} = \frac{dq}{dt},\ \ddot{q} = \frac{d^2q}{dt^2}$$

Obtaining the dynamic equations of motion using the MatLab program (robotics toolbox) was very powerful process; the toolbox use N-E approach to compute the equation of motion by feeding the program by necessary data about the robot system by using the functions (dyn.m, robot.m) to introduce the robot object in the MatLab program, the robot object then can be used in the MatLab program to define Simulink blocks. Because of the nature of the formulation and the method of systematically computing the torques, computations are much simpler, allowing a short computing time. With this algorithm, about three milliseconds are needed to compute the feedback joint torques per trajectory set point.

Robot Control: Robot control is the spine of robotics. It consists in studying how to make a robot manipulator do what it is desired to do automatically; hence, it includes in designing robot controllers. Typically, these take the form of an equation or an algorithm which is realized via specialized computer programs. Then, controllers form part of the so-called robot control system which is physically constituted of a computer, a data acquisition unit, actuators (typically electrical motors), the robot itself and some extra "electronics". In this work two types of control problems was studied feed forward control and computed torque control.

Computed Torque Control: In order to overcome drawbacks of the PD controller[10], a more sophisticated scheme in which the magnitude of the nonlinear disturbing and loading torques is computed using the dynamic equations and used to compensate these disturbances by means of a feed forward may be employed. It must be noted that the basic control method is still PD controller with both position and velocity feedback [6].

The dynamic model (1) that characterizes the behavior of robot manipulators is in general, composed of nonlinear functions of the state variables (joint positions and velocities). This feature of the dynamic model might lead us to believe that given any controller, the differential equation that models the control system in closed loop should also be composed of nonlinear functions of the corresponding state variables. Nevertheless, there exists a controller which is also nonlinear in the state variables but which leads to a closed-loop control system which is described by a linear differential equation. This controller is capable of fulfilling the motion control objective. The computed-torque control law is given by [7]:

$$\tau = M(q)[\ddot{q}_d + k_v\dot{\tilde{q}} + k_p\tilde{q}] + C(q,\dot{q}) + G(q) \qquad (2)$$

Where the gains k_p, k_v are chosen to meet some specific properties of the system and these gains should be adjusted to reduce the errors[8]. The block diagram that corresponds to computed-torque control of robot manipulators is presented in Figure 1.

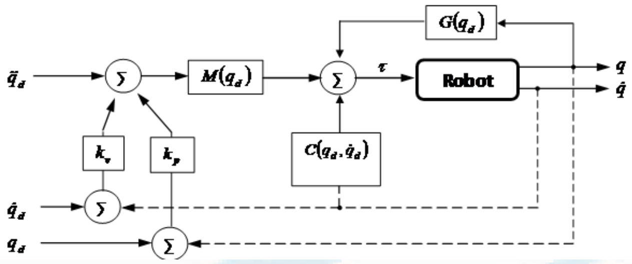

Figure 1: Block diagram of Computed-torque control of robot manipulators

Feed forward Control: Among the conceptually simplest control strategies that may be used to control a dynamic system we find the so-called open-loop control, where the controller is simply the inverse dynamics model of the system evaluated along the desired reference trajectories. For the case of linear dynamic systems, this control technique may be roughly presented as follows [7]: by using (1) applying a torque τ at the input of the robot, the behaviours of its outputs q and $\dot{q}$ are governed by:

$$\frac{d}{dt}\begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \dot{q} \\ M(q)^{-1}[\tau - C(q,\dot{q}) - G(q)] \end{bmatrix} \quad (3)$$

If the behaviour of the outputs q and $\dot{q}$ need to be equal to that specified by q_d and $\dot{q}_d$ respectively, it seems reasonable to replace q, $\dot{q}$ and $\ddot{q}$ by $q_d, \dot{q}_d$, and $\ddot{q}_d$ in the Eq. (2) and to solve for τ. This reasoning leads to the equation of the feed forward controller, it can be expressed as:

$$\tau = M(q_d)\ddot{q}_d + C(q_d, \dot{q}_d)\dot{q}_d + G(q_d) \quad (4)$$

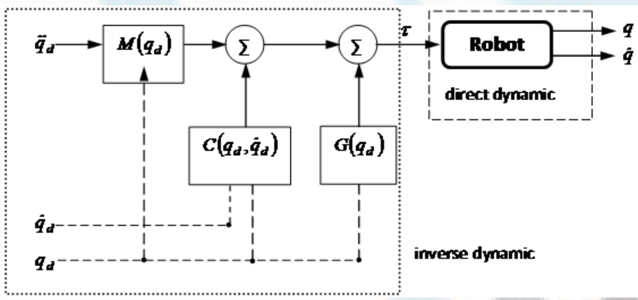

Figure 2: Block Diagram – Feed forward Control

In figure 2 the block-diagram corresponding to a robot under feed forward control is presented; the control action τ does not depend on q nor on $\dot{q}$, that is, it is an open loop control.

The wide practical interest in incorporating the smallest number of computations in real time to implement a robot controller has been the main motivation for the PD plus feed forward control law, Feed forward control (3) may be modified by the addition, of a *feedback* Proportional–Derivative (PD) term shown in figure 3 given by:

$$\tau = M(q_d)\ddot{q}_d + C(q_d, \dot{q}_d)\dot{q}_d + G(q_d) + k_v\dot{\tilde{q}} + k_p\tilde{q} \quad (5)$$

3. Conclusion

The proposed method of designing control based on retrieved equation which is traced upon a curve. Since the curve is traced by obtaining position of an obstacle occurring on the path. The forward moving experiment on the robots verifies the path followed on traced curve. It is already shown above that the transformation of trajectory equation into corresponding movement of robots by opting two different types of control.A separate experiment is carried out in order to further verify the capability of the robot in handling external disturbance which is not reported here.

The results of the experiments vindicate that the system is able to handle and correct the unknown external disturbance up to certain intensity. This limitation is related to the limitations of hardware components, such as the weight of the balancing mass and the actuator response.

In general, the reported experimental results prove that the curve fitting techniques is very useful and bring revolution in the design of minimalist robots, impact free trajectory planning, and obstacleavoidance trajectory with improvement of the control system sensitivity and response to the external disturbances if occurred.This thorough study still gives a valuable insight into the path recognition of robotics movement.

References

[1] Furusho, J., & Masubuchi, M. (1986). Control of a dynamical biped locomotion system for steady walking. *Journal of Dynamic Systems, Measurement, and Control*, *108*(2), 111-118.

[2] Cheng, M. Y., & Lin, C. S. (1995, October). Measurement of robustness for biped locomotion using linearized Poincare'map. In *Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century, IEEE International Conference on* (Vol. 2, pp. 1321-1326). IEEE.

[3] Wahde, M., & Pettersson, J. (2002, June). A brief review of bipedal robotics research. In *Proceedings of the 8th UK Mechatronics Forum International Conference (Mechatronics 2002)* (pp. 480-488).

[4] Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2006). *Robot modelling and control* (pp. 163-182). New York: John Wiley & Sons.
[5] Corke, P. I. (1996). A robotics toolbox for MATLAB. *Robotics & Automation Magazine, IEEE*, *3*(1), 24-32.
[6] Inigo, R. M., & Morton, J. S. (1991). Simulation of the Dynamics of an Industrial Robot. *Education, IEEE Transactions on*, *34*(1), 89-99.
[7] Kelly, R., & Moreno, J. (2005). Manipulator motion control in operational space using joint velocity inner loops. *Automatica*, *41*(8), 1423-1432.
[8] Kelly, R., & Salgado, R. (1994). PD control with computed feedforward of robot manipulators: A design procedure. *Robotics and Automation, IEEE Transactions on*, *10*(4), 566-571.
[9] Huber, T., Torda, A. E., & van Gunsteren, W. F. (1994). Local elevation: a method for improving the searching properties of molecular dynamics simulation. *Journal of computer-aided molecular design*, *8*(6), 695-708.
[10] Tomei, P. (1991). A simple PD controller for robots with elastic joints. *Automatic Control, IEEE Transactions on*, *36*(10), 1208-1213.
[11] Durham, J. W., &Bullo, F. (2008, September). Smooth nearness-diagram navigation. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on* (pp. 690-695). IEEE.

Author Profile

Abhinash received the B.Tech degree in Electronics and Communication Engineering from Biju Patnaik University of Technology, Rourkela. Currently pursuing his ME degree from Birla Institute of Technology, Mesra Ranchi. He is currently working on Robotics automation using artificial intelligence and human machine interaction.

Ashwani Kumar Jha received the B.Tech degree in Computer Science from Punjab Technical University. Now he is pursuing ME in Computer Aided Analysis and Design from Birla Institute of Technology.His current researchinterests are focussed on computational intelligence, hardware and software implementations of biologically plausible artificialneural networks, brain computer interfacing, and intelligent systems inrobotics.