JOURNAL OF ENGINEERING annals OF FACULTY OF ENGINEERING HUNEDOARA
Tome V (year 2007), Fascicole 2, (ISSN 1584 – 2665)

CARTESIAN ROBOT ARM PROBLEM USING A NOVEL DORMAND AND PRINCE TECHNIQUE

R. PONALAGUSAMY, S. SENTHILKUMAR

NATIONAL INSTITUTE OF TECHNOLOGY, DEPARTMENT OF MATHEMATICS, TAMILNADU, INDIA

Abstract: The aim of this article is focused on providing numerical simulation for system of first order Cartesian robot arm problem using the Novel Dormand and Prince Algorithm. The parameters governing the arm model of a robot control problem have been discussed through Novel Dormand and Prince Algorithm. This paper emphasizes an efficient numerical solution for the control logic and system dynamics of a robot arm having three cartesian axes plus two rotational axes of the wrist. Several numerical techniques are used and their results are compared for the integration of the nonlinear differential equations for the individual axes. Many system parameters are discussed for the simulated motion and are adjusted to fit experimental data on the actual arm motion. The précised solution of the system of equations representing the arm model of a Cartesian robot has been compared with the corresponding approximate solutions at different time intervals. Results and comparison show the efficiency of the numerical integration algorithm based on the absolute error between the exact and approximate solutions. It is observed that Novel Dormand and Prince Algorithm is efficient and takes less CPU time compared with RK-Sixth order algorithm.

Key-words: Novel Dormand and Prince Algorithm, Different Numerical Integration Techniques, Ordinary Differential Equations, System of Equations of First Order, Cartesian robot arm problem.

1. INTRODUCTION

Extensive research work is still being carried out on variety of aspects in the field of robot control, especially about the dynamics of a robotic motion and their governing equations. The simulations of the dynamics of Cartesian robot arms are found in the literature Astrom,K.J, (1983), Derby,S, (1983), Orlandea,N and Berenyi,T. (1981), Walker,R.K, Gregory,C and Shah.S. (1984) and Nelson, W.L and Ghang, J.D. (1984) Research in this area is still active and its applications are enormous. This is because of its nature of extending accuracy in the determination of approximate solutions and its flexibility. Because of the-linear and coupled characteristics nature, the design of a robot control system is made complex.

Runge-Kutta (RK) techniques are being applied to compute numerical solutions for the problems, which are modeled as Initial Value Problems (IVPs) differential equations by Alexander and Coylc (1990), Evans (1991), Hung (2000), Shampine and Watts [9-10].Runge-kutta techniques have become very popular, both as computational techniques as well as subject for research, which were discussed by Butcher, J.C. (1964), Butcher. J.C. (1987) and Butcher, J.C. (1990), Shampine(1975) and Shampine(1977). This technique was derived by Runge-kutta and extended by Kutta to solve differential equations efficiently which are equivalent of approximating the exact solutions by matching ‘n’terms of the Taylor series expansion.

Butcher, J.C. (1964), Butcher. J.C. (1987) and Butcher, J.C. (1990), derived the best RK pair by all statistical measures appeared to be the RK algorithm and also an error estimate was given. The RK-Butcher algorithm is nominally considered sixth order since it requires six function evaluations, but in actual practice the 'working order' is five but still exceeds all the other algorithms examined including RK-Fehlberg, RK-Centrodial mean and RK-Arithmetic mean.

Bader (1987) and Bader (1998) introduced the RK-Butcher algorithm for finding the truncation error estimates and intrinsic accuracies and the early detection of' stiffness in coupled differential equations that arises in Theoretical Chemistry Problems.

Mohsen et al., (2004) have proposed an observer-based control strategy and investigated on Cartesian robot arm problem with theory and practice. Also, the feasibility of the controller is validated through both numerical simulations and experimental testing. Significant matching between experimental results and numerical simulation is observed.

Jindong Tan et al., (2004) have discussed about the coordination of multi-robot and human systems in a perceptive reference frame. Dana Petcu et al., (2005) constructed a performance model for parallel iterative numerical techniques under the assumption of a message-passing computing system. Also they have proved the inefficiency of the explicit iterative techniques for ordinary differential equations on distributed memory multiprocessor systems.

Zanariah Abdul Majid (2006) have developed the two-point fully implicit block techniques for solving large systems of ordinary differential equations (ODEs) using variable step size on a parallel shared memory computer. The technique calculates the numerical solution at two equally spaced points simultaneously within a block. The stability of the techniques is also investigated.

A new family of extended Runge–Kutta formulae has been presented by Xinyuan Wu and Jianlin Xia (2006). It is assumed that the user will evaluate both f and f' readily when solving the autonomous system $y'=f(y)$ numerically. Liu et al., (2007) has discussed about the stability of Runge-Kutta techniques in the numerical solution of linear impulsive differential equations.

Foroush Bastani and Mohammad Hosseini (2007) have presented a new adaptive time stepping algorithm for strong approximation of stochastic ordinary differential equations and employed two different error estimation criteria for drift and diffusion terms of the equation, both of them based on forward and backward moves along the same time step.

A construction of continuous extensions to a new representation of two-step Runge–Kutta techniques for ordinary differential equations has been discussed by Bartoszewski and Jackiewicz (2007). This representation makes possible the accurate and reliable estimation of local discretization error, facilitates the efficient implementation of these techniques in variable stepsize environment, and adapts readily to the numerical solution of a class of delay differential equations.

John Butcher (2007) has introduced general linear techniques as the natural generalizations of the classical Runge–Kutta and linear multistep techniques. A survey of general linear techniques is also presented, including recent results on techniques with the inherent RK stability property.

B.Min,W.J..Lee and N.Park (2000) discussed about the fast numerical techniques for solving non-linear differential equations have been proposed and they are based on a fixed constant step size say the average-power analysis (one step technique). Hodgkinson (2004) discussed average power step technique and Liu (2003) discussed the four step technique but the automatics step-adjustment technique has the capacity for automatic step-size control based on the Runge-Kutta-Fehlberg formula. Liu (2003) implemented a adaptive step-size control requires stepping algorithm return information about its performance-most importantly, an estimate of its truncation error because it shares the stage information, the embedded fourth-fifth-order Rungee-Kutta-Fehlberg technique.

Based on the Dormand-Prince (1980) formula, a fast, effective and efficient technique is used which can implement automatic step-size adjustment to solve non-linear differential equation. Simulation results show that (i). The CPU time of the novel technique is smaller than that of the DP technique (ii). The novel Dormand and Prince technique reduced the running time by more than two orders of magnitude in comparison with the other techniques and (iii). In comparison with the DP technique the novel technique can increase the computational speed while solving the non-linear equations. In the present paper, the Cartesian robot arm problem is solved with different approach using the algorithms such as Explicit Euler, RK-Gill, RK-Fifth, RK-Sixth order and Novel Dormand and Prince Algorithm to yield higher accuracy, with less error.

The importance of numerical simulation (virtual) study is that it is possible to try experimental control algorithm without the danger of hurting the robot or personnel. In addition to the above the numerical simulator accurately reflects a robotic dynamics nature.

2. DIFFERENT NUMERICAL INTEGRATION TECHNIQUES

2.1. RK-Fifth Order Algorithm

The Fifth Order Runge-Kutta algorithm is an explicit technique and discussed by Morris Badder (1987,1988). It starts with a simple Euler technique. The increase of the state variable x^{ij} is stored in the constant k_1^{ij} . This result is used in the next iteration for evaluating k_2^{ij}. The same procedure must be repeated to compute the values of k_3^{ij}, k_4^{ij}, k_5^{ij} and k_6^{ij}.

$$k_1^{ij} = \tau f'(x_{ij}(n\tau)), k_2^{ij} = \tau f'(x_{ij}(n\tau) + \frac{1}{4}k_1^{ij}),\ k_3^{ij} = \tau f'(x_{ij}(n\tau) + (\frac{1}{8})k_1^{ij}) + (\frac{1}{8})k_2^{ij}),$$

$$k_4^{ij} = \tau f'(x_{ij}(n\tau) - \frac{1}{2}k_2^{ij} + k_3^{ij}),\ , k_5^{ij} = \tau f'(x_{ij}(n\tau) + \frac{3}{16}k_1^{ij} + \frac{9}{16}k_4^{ij})$$

$$k_6^{ij} = \tau f'(x_{ij}(n\tau) - \frac{3}{27}k_1^{ij} + \frac{2}{7}k_2^{ij} + \frac{12}{7}k_3^{ij} - \frac{12}{7}k_4^{ij} + \frac{8}{7}k_5^{ij}\ ) \tag{10}$$

Therefore, the final integration is a weighted sum of the five calculated derivatives which is given below.

$$x_{ij}((n+1)\tau) = x_{ij}(n\tau) + \frac{1}{90}[7k_1^{ij} + 32k_3^{ij} + 12k_4^{ij} + 32k_5^{ij} + 7k_6^{ij}] \tag{11}$$

where *f(.)* is computed according to (1).

2.2. RK-Sixth Order Algorithm

The RK-Sixth order Algorithm is an explicit technique and discussed by Ponalagusamy and Senthilkumar(2007). It starts with a simple Euler technique. The increase of the state variable x^{ij} is stored in the constant k_1^{ij}. This result is used in the next iteration for evaluating k_2^{ij}. The same procedure must be repeated to compute the values of k_3^{ij}, k_4^{ij}, k_5^{ij} and k_6^{ij}, and k_7^{ij}.

$$k_1^{ij} = \tau f'(x_{ij}(n\tau)),\ k_2^{ij} = \tau f'(x_{ij}(n\tau) + \frac{1}{2}k^{ij}{}_1),\ k_3^{ij} = \tau f'(x_{ij}(n\tau) + (\frac{2}{9})k_1^{ij}) + (\frac{4}{9})k_2^{ij});$$

$$k_4^{ij} = \tau f'(x_{ij}(n\tau) + \frac{7}{36}k_1^{ij} + \frac{2}{9}k_2^{ij} - \frac{1}{12}k_3^{ij}),$$

$$k_5^{ij} = \tau f'(x_{ij}(n\tau) - \frac{35}{144}k_1^{ij} - \frac{55}{36}k_2^{ij} + \frac{35}{48}k_3^{ij} + \frac{15}{8}k_4^{ij}) \tag{12}$$

$$k_6^{ij} = \tau f'(x_{ij}(n\tau) - \frac{1}{360}k_1^{ij} - \frac{11}{36}k_2^{ij} - \frac{1}{8}k_3^{ij} + \frac{1}{2}k_4^{ij} + \frac{1}{10}k_5^{ij})$$

$$k_7^{ij} = \tau f'(x_{ij}(n\tau) - \frac{41}{260}k_1^{ij} + \frac{22}{13}k_2^{ij} \frac{43}{156}k_3^{ij} - \frac{118}{39}k_4^{ij} + \frac{32}{195}k_5^{ij} + \frac{80}{39}k_6^{ij})$$

Therefore, the final integration is a weighted sum of the five calculated derivatives which is given below.

$$x_{ij}((n+1)\tau) = x_{ij}(n\tau) + [\frac{13}{200}k_1^{ij} + \frac{11}{40}k_3^{ij} + \frac{11}{40}k_4^{ij} + \frac{4}{25}k_5^{ij} + \frac{4}{25}k_6^{ij} + \frac{13}{200}k_7^{ij}] \quad (13)$$

where *f(.)* is computed according to (1).

2.3. Novel Dormand and Prince: A Fifth Order Technique

On the basis of the Dormand Prince (1980) formula a novel technique is carried out where the fifth-order technique uses six function evaluation at each time step. The novel Dormand and Prince algorithm is an explicit exponential technique introduced by Xueming Lie (2005). The increase of the state variable x^{ij} is stored in the constant k_1^{ij}. This result is used in the next iteration for evaluating k_2^{ij}. The same procedure must be repeated to compute the values of k_3^{ij}, k_4^{ij}, k_5^{ij} and k_6^{ij}.

$$k_1^{ij} = \tau f'(x_{ij}(n\tau)),\ k_2^{ij} = \tau f'(x_{ij}(n\tau)\exp(\frac{k_1^{ij}}{5})),\ k_3^{ij} = \tau f'(x_{ij}(n\tau)\ \exp(\frac{3}{40})k_1^{ij}\ \exp(\frac{9}{40})k_2^{ij}) \quad (13)$$

$$k_4^{ij} = \tau f'(x_{ij}(n\tau)\exp(\frac{44}{45})k_1^{ij}\ \exp(\frac{-56}{15})k_2^{ij}\ \exp(\frac{32}{9})k_3^{ij})$$

$$k_5^{ij} = \tau f'(x_{ij}(n\tau)\exp(\frac{9372}{6561})k_1^{ij}\ \exp(\frac{-25360}{2187})k_2^{ij}\ \exp(\frac{64448}{6561})k_3^{ij}\ \exp(\frac{-212}{729})k_4^{ij}$$

$$k_6^{ij} = \tau f'(x_{ij}(n\tau)\exp(\frac{9017}{3168})k_1^{ij}\ \exp(\frac{-355}{33})k_2^{ij}\ \exp(\frac{-46732}{4247})k_3^{ij}\ \exp(\frac{49}{176})k_4^{ij}\ \exp(\frac{-5103}{18656})k_5^{ij})$$

Therefore, the final integration is a weighted sum of the five calculated derivatives which is given below.

$$x_{ij}((n+1)\tau) = x_{ij}(n\tau)\exp(\frac{35}{384})k_1^{ij}\ \exp(\frac{500}{1113})k_3^{ij}\ \exp(\frac{125}{192})k_4^{ij}\ \exp(\frac{-2187}{6784})k_5^{ij}\ \exp(\frac{11}{84})k_6^{ij} \quad (14)$$

where *f(.)* is computed according to the given function.

2.4. Novel Dormand and Prince: A Sixth Order Technique

The Novel Dormand and Prince Sixth order algorithm discussed by Xueming Lie (2005) is an explicit exponential technique introduced by Dormand and Prince uses seven function evaluation at each time step. The increase of the state variable x^{ij} is stored in the constant k_1^{ij}. This result is used in the next iteration for evaluating k_2^{ij}. The same procedure must be repeated to compute the values of k_3^{ij}, k_4^{ij}, k_5^{ij}, k_6^{ij} and k_7^{ij}

$$k_1^{ij} = \tau f'(x_{ij}(n\tau)),\ k_2^{ij} = \tau f'(x_{ij}(n\tau)\exp(\frac{k_1^{ij}}{5})),\ k_3^{ij} = \tau f'(x_{ij}(n\tau)\ \exp(\frac{3}{40})k_1^{ij}\ \exp(\frac{9}{40})k_2^{ij}),$$

$$k_4^{ij} = \tau f'(x_{ij}(n\tau)\exp(\frac{44}{45})k_1^{ij}\ \exp(\frac{-56}{15})k_2^{ij}\ \exp(\frac{32}{9})k_3^{ij})$$

$$k_5^{ij} = \tau f'(x_{ij}(n\tau)\exp(\frac{9372}{6561})k_1^{ij}\ \exp(\frac{-25360}{2187})k_2^{ij}\ \exp(\frac{64448}{6561})k_3^{ij}\ \exp(\frac{-212}{729})k_4^{ij}$$

$$k_6^{ij} = \tau f'(x_{ij}(n\tau)\exp(\frac{9017}{3168})k_1^{ij}\ \exp(\frac{-355}{33})k_2^{ij}\ \exp(\frac{-46732}{4247})k_3^{ij}\ \exp(\frac{49}{176})k_4^{ij}\ \exp(\frac{-5103}{18656})k_5^{ij})$$

$$k_7^{ij} = \tau f'(x_{ij}(n\tau)\exp(\frac{35}{384})k_1^{ij}\ \exp(\frac{500}{1113})k_3^{ij}\ \exp(\frac{125}{192})k_4^{ij}\ \exp(\frac{-2187}{6784})k_5^{ij}\ \exp(\frac{11}{84})k_6^{ij}) \quad (15)$$

Therefore, the final integration is a weighted sum of the six calculated derivatives which is given below.

$$x_{ij}((n+1)\tau) = x_{ij}(n\tau)\exp(\frac{5179}{57600})k_1^{ij}\exp(\frac{7571}{16695})k_3^{ij}\exp(\frac{393}{640})k_4^{ij}$$
$$\exp(\frac{-92097}{339200})k_5^{ij}\exp(\frac{187}{2100})k_6^{ij}\exp(\frac{k_7}{40}) \quad (16)$$

where *f(.)* is computed according to the given function.

3. CARTESIAN ROBOT ARM MODEL AND ESSENTIAL OF VARIABLE STRUCTURE

3.1. Working Principle and Description of Cartesian Robot Arm Dynamics

The AID 600 robot arm has Cartesian xyz axes of motion, together with a wrist two-axis revolute pitch and roll motion. The compass reading of these axes with respect to the robot arm is shown in the Figure 1. The gross dynamics of the robot arm can be simplified by assuming the xyz axes uncoupled and the wrist a constant-mass payload with negligible coupling effects, allowing the individual axes to be simulated independently of each other. The existing program considers the x axis; appropriate substitution of constants can yield models for the y and z axes.

The system block diagram for one of the robot axes is given in fig. 2 where the elements containing a "D" denotes delay. In that, J_m represents the moment of inertia of the axis servo-motor, and B_m represents the motor damping. The linear dynamics of the robot axis can be described by the two state variables position, y_0 and velocity y_1. Because, the effects of motor gear backlash are to be investigated, the nonlinear effects of the coupling and uncoupling of the rack and pinion system of the arm and motor gear are described with two additional variables.(i) Gear angle y_2 and (ii). Gear angular velocity y_3 with the drive force transmitted through the stiffness and friction of the gear train when the teeth are in contact.

Within the voltage saturation limits, the driver acts as a controlled current source with transconductance K_i, as maintained by a current feedback loop in the controller circuitry. When the motor reaches the voltage saturation limit, it can be characterized by a single-pole inductance-resistance model with a back EMF proportional to the motor angular velocity. The motor is driven by a controller with an input for a reference (set-velocity) voltage, and an inverting input for a tachometer feedback voltage. The controller has a switching mode power drive unit, preceded by a preamplifier consistng of a lag-lead (pole-zero) compensation network, an overall frequency rolloff at about 1000 HX, and an output voltage limit V_L. The preamplifier's two poles and one zero can be described by the linear combination of two auxiliary state variables, y_4 and y_5. The nonlinear dynamics of the motor-controller are represented by the motor current, which is the state variable y_6.

3.2 Working Principle and Description of Slave Control Unit:

The AID 600 slave control unit calculates the set velocity voltage for each movement axis in the high-level commands software, using information from position and speed-schedule inputs, (marked as 'track' data input in Fig. 2) plus feedback data from optical encoder on the motor shaft. The set-velocity voltage output is modeled as a sample-data device of period 5 ms. The input of new track data from the master control unit occurs at 80 ms intervals. The control logic used in the slave control unit is a combination of position, velocity, and integrated position functions, as shown in the lower portion of Fig. 2. The dynamics of the closed-loop control system for a single axis of the robot arm motion is represented by the seven state variables which are shown in the table 1.

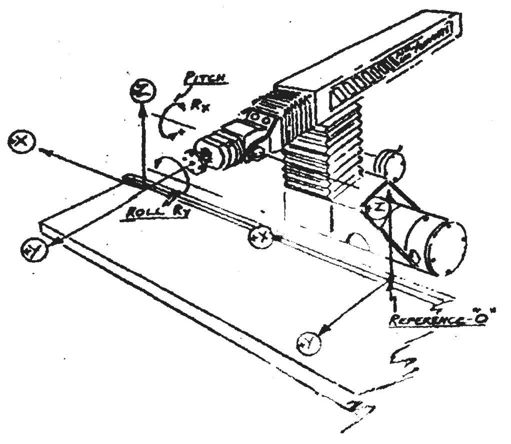

Table 1. Seven State variables

Seven State Variables	Description and Units
y_0	arm position (m.)
y_1	arm velocity (m/s.)
y_2	motor angle (rad.)
y_3	motor velocity (rad/s.)
y_4	controller preamp. variable 1 (volts)
y_5	controller preamp. Variable 2 (Volts/s.)
y_6	motor current (amps.)

Figure 1. Robot arm axes of motion

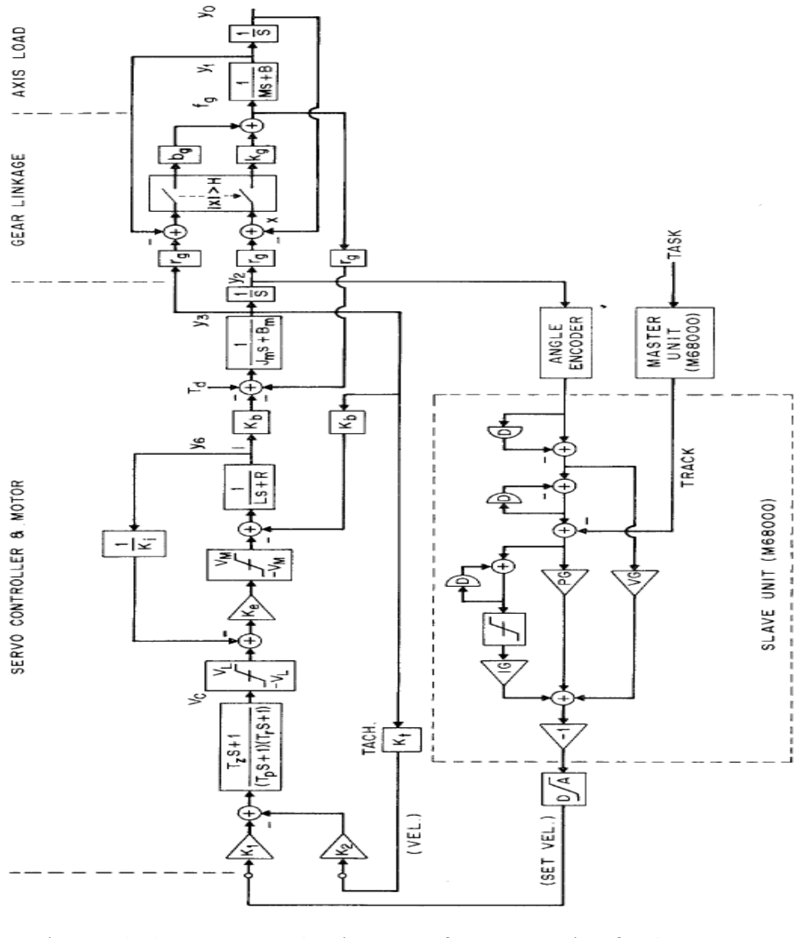

Figure 2. System Block Diagram for one axis of robot arm

The following table 2, shows the four auxiliary variables which are functions of the state variables.

Table 2. Four auxiliary Variables

Auxiliary Variables	Description and specific Units
f_g	force coupled through the gear drive in Newtons (N)
F_s	the static friction force (N)
V_m	the motor voltage
V_c	the controller preamplifier output voltage

In terms of seven state variables, the differential equations of motion are given by,

$$\dot{y}_0 = y_1,\ \dot{y}_1 = (-B_a y_1 - F_s + f_g)/M\ ,\ \dot{y}_2 = y_3,\ \dot{y}_3 = (-B_m y_3 - r_g f_g + k_b y_6)/J_m$$

$$\dot{y}_4 = y_5,\ \dot{y}_5 = (-y_4 - (T_p + T_r)y_5 + k_1U - K_2k_ty_3)T_pT_r,\ \dot{y}_6 = (V_m - R_ay_6 - K_by_3)/L_a \quad (10)$$

where B_a represents the viscous friction coefficient and M is the mass of the arm-rack mechanism; B_m represents the rotational friction coefficient, m is the inertia, K_b represents the torque constant of the motor, and r_g represents the radius of the pinion gear. The controller parameters are the compensation pole time constant, T_p high frequency roll oft time constant, T_r, and the signal 1and 2 input gains, K_1 and K_2. The controller inputs are the set-velocity voltage, U(t) and the tachometer feedback voltage, which is K_t times the motor angular velocity $y_3(t)$.

The controller preamplifier output voltage is given by,

$$v_c = y_4 + T_zy_5,\ |v_c| \le v_L \quad (11)$$

As indicated in Fig. 2, the motor voltage is proportional to the preamplifier output minus the current feedback.

$$v_m = K_e\,(v_c - y_6 / K_i,\ |v_m| \le v_M \quad (12)$$

where K_c represents the forward gain of the power unit and K_i represents the conductance of the current feedback path. If K_e is sufficiently large, and $|V_m| \le V_M$ then the controller acts as a transconductance amplifier with the current, $y_6 \approx K_iV_c$. However, the equation for $\dot{y}_6$ in (10) and the voltage equations (11) and (12) with their limit conditions are necessary to adequately simulate the nonlinear conditions which generally prevail in the controller during arm movements.

The gear coupling force is determined by the coupling stiffness, k_g, the coupling friction coefficient, b_g, and the difference between the position of the pinion gear and the rack relative to the backlash distance, H. Let the pinion-rack position difference be denoted by x.

$$x = r_gy_2 - y_0 \quad (13)$$

Then the coupling force is given by,

$$f_g = \begin{cases} (x-H)K_g + (r_gy_3 - y_1)b_g, x > H \\ (x+H)K_g + (r_gy_3 - y_1)b_g, x < -H \\ 0, -H \le x \le H \\ \end{cases} \quad (14)$$

The axis-load friction has two components. First, a viscous friction proportional to velocity, which is reasonably well modeled by the linear force, B_ay_1 and second, a static friction, F_s which is modeled as a force doublet in the region of zero velocity.

$$F_s = \begin{cases} F_0\,\mathrm{sgn}[y_1], & |y_1| < v \\ 0 & otherwise \end{cases} \quad (15)$$

where v is the threshold velocity level at which the static friction component disappears.

The set of Equations (9)-(14) from the basis for the nonlinear dynamics of each axis of the robot arm motion simulated in the computer program for this study. If the backlash gap, H, and the static friction, F_s were negligibly small, and the gear coupling stiffness, k_g, voltage limits, V_L and V_M, and gain, K_e, were arbitrarily large, the nonlinear, seventh-order system represented by Equations (9)–14) reduces to a linear, fourth-order system. In this case, the controller power stage is replaced by a current source, $y_6 = K_iV_c$ and the gear linkage and axis load (see fig. 2) are replaced by an equivalent loaded motor inertia, J, and rotational friction coefficient, B, given by

$$J = J_m + r_g^2M, B = B_m + r_g^2B_a; \quad (16)$$

and the output position and velocity are simply gear radius, r_g times the motor angular position and velocity. This approximation model was used in part of the testing phase of the study, described in the next section. However, nonlinear effects of the gear backlash, static friction and especially the saturation limits, have a significant effect on the dynamics of the arm movement; hence the inclusion of these nonlinearities in the simulation is important for reasonably accurate representation of the actual arm movement.

4. NUMERICAL SOLUTION TO CARTESIAN ROBOT ARM MODEL PROBLEM: A COMPARISON

The state variable differential equations (10) are integrated using Novel Dormand and Prince Sixth order algorithm and RK-Sixth order algorithm. The following x-axis parameters used in the program were provided by manufacturers pecifications (1978) and data provided by engineers at Automatirx, Inc.:

Table 3. X-axis Parameters and Specifications

Parameters/Symbol	Description	Units
R_a	Armature resistance	0.56 ohms
L	Armature inductance	0.00235 henries
K_b	Motor torque constant	0.289 volt sec/rad
K_b	Motor torque constant	0.289 N m/amp
J_m	Armature inertia (no-load)	0.000847 N m sec^2
B_m	Armature damping	0.000733 N m sec
K_t	Tachometer gain	0.0668 volt sec/rad
r_g	Motor gear radius	0.00573 m
H	Gear backlash	0.00001 m
M	Arm mass	146 kg

Table 4. System Parameters and Specifications

Parameters/ Symbol	Description	Units
T_z	Compensation zero time constant	0.00705 sec
T_p	Compensation pole time constant	0.0071 to 4.74 sec
T_r	Preamp roll off time constant	0.000159 sec
K_1	Set velocity gain	0 to 3333
K_2	Tachometer feedback gain	80 to 870
K_i	Transconductance gain	3 amps/volt
V_M	Motor voltage limit	95 volts

The remaining system parameters needed for the simulation are the arm and gear tooth friction coefficients, the gear tooth spring constant and the actual values of T_p, K_1, K_2 and V_L in the controller. The values of these parameters are estimated by comparing the simulation results and actual measurement on the individual axes of the robot arm, using an accelerometer mounted on the arm by Jensen and Nelson and Ghang (1984) along with tachometer and controller test-point measurements. The validity of some of the other system parameters can also be examined using the same procedure.

The controller parameters are used to establish the desired set-velocity responses. They are the time constant T_p, of the pole in the lead-lag compensation network; the set-velocity gain k_1, and the feed back velocity (tachometer) gain K_2. The procedure recommended in the servo controller manual (1978) is to set the gain K_2 at its maximum (870), then adjust the gain K_1 to give the desired steady-state velocity gain. For the x-axis, for example, the desired value is 1 m/sec for 10 volts on the signal 1 input. The time constant, T_p is then adjusted to give an acceptable transient response. For the x-axis test runs, the gain K_1 was automatically computed as a function of K_2 maintains the desired steady-state gain of 0.1 m/sec per volt. Linear system check is one of the program validation tests, where the nonlinear factors in the simulation program were removed (i.e. no saturation, static friction, or backlash and ideal gear coupling). The step response from the linearized simulation is compared with the step response obtained analytically from the transfer function of the linear system. Thus, the simulation results are agreed with the analytical results which is shown in fig. 3 for two values of time constant T_p. The lower plot indicates that the linear step response can be made to reach the steady-state value in about 0.03 seconds. However, the motor current required for this rapid response has a peak value more than 40 times the rated peak current for the motor. Thus, saturation limits will preclude this rapid, linear response. A second set of runs was made using the saturation, backlash, and other parameter values as given in the above list. The gain K_2 was set at 870, which required a value for K_1, of 1014 for the desired 1 rn/sec steady-state velocity. The time constant, T_p was then reduced in a series of steps from the maximum of 4.74 sec to a value of 0.04 sec., where a transient response similar to the desired response specified in the controller manual (1978) was obtained. Plots of the simulated x-axis position, velocity, motor angular velocity, current and voltage, and gear force, for T_p = 0.04 sec are shown in Figure 4.

It can be observed that the motor current is at the limit value for most of the transient part of the velocity step response, and that as the velocity reaches the 1 m/sec. set-velocity, the current transient changes to a more rapid "ringing" transient which decays to the proper steady-state value. For the system parameters used in these test runs, the "ringing" transient decays at an undesirably slow rate if T_p, is reduced below about 0.04 sec. The initial slope and time constant of the arm velocity for the exponential rise to the set velocity value are predominant governed by the saturation current level arm friction, respectively. The “ringing” transient in force during this portion of the response is predominantly governed by the gear stiffness and friction. By matching the arm velocity and acceleration output of the simulation program with velocity and acceleration data derived from an accelerometer mounted on the arm, reasonable estimates for these parameters were obtained. For example for the x-axis, the values which give a good match with Novel Dormand-Prince Sixth order Technique and RK-Sixth order technique when the motor current is not at the saturation limit, the response is affected by the overall closed-loop characteristics of the system components. Fig.5 shows the actual and simulated arm position and velocity patterns for a programmed x-axis movement of 1 meter and a set-velocity level of 0.9 rn/sec. The control parameters in the slave processor unit (see Fig. 2) have been adjusted in the simulation program to match those in the actual system. It can be seen that there are some minor differences in some portions of the response patterns, it appears that the simulation represents the basic characteristic of the robot arm movement.

Table 5. Various matching parameter values and % estimation

Parameters/ Symbols	Description	Novel Dormand-Prince Technique	RK-Sixth Order Technique
V_L	Controller Preamp. Limit	2.783 Volts	2.77 volts
I_M	Saturation Current (=K_iV_L)	8.125 amp	8.0 amp
B_a	Arm Friction Coefficient	211 N Sec/m	212 Nsec/m.
b_g	Gear Friction Coefficient	1500 N Sec/m	1500 N Sec/m
K_g	Gear Stiffness	5.264*10^6 N/m	5.275*10^6N/m

The influence of various step sizes on the relative percentage of the CPU time is examined for all the problems. It can be treated as a measure of the effort necessary to obtain higher accuracy solution from the less accurate step size. The difference between current CPU time and prior step sizes is expressed as a relative percentage which is given below:

$$\mathrm{Re}\,lative\,difference(in\ terms\ of\ CPU) = abs\left(\frac{CPU^{new} - CPU^{prior}}{CPU^{prior}}\right)*100$$

where CPU^{new} and CPU^{prior} represents the CPU time required for simulation of robot arm motion with specified step sizes. The percentage estimate is made for Novel Dormand and Prince Sixth order technique, RK-Sixth order algorithm under various matching parameter values as specified in table 5.

The relative differences in terms of CPU time for execution by Novel Dormand and Prince Sixth order technique is better in comparison with RK-Sixth order algorithm. In fig. 5, arm position and velocity time takes for the X-axis movement, predicted by numerical simulation of Novel Dormand and Prince Sixth order algorithm closely follows actual arm-movement obtained by the integrated accelerometer data. i.e. a linear model is sufficient for simulating robot arm movement. Therefore, it is observed that, for simulation studies of Robot arms, Novel Dormand and Prince Sixth order technique is efficient and takes less CPU time proving to be a better choice when compared to other schemes.

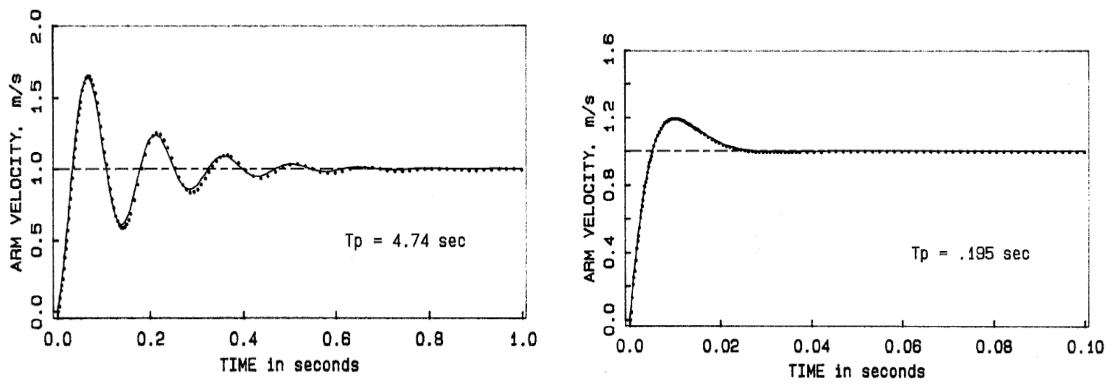

Figure 3. Comparison of simulation output (dots) and analytic solution (solid lines) for step response of linear system for two values of the compensator time constant T_p.

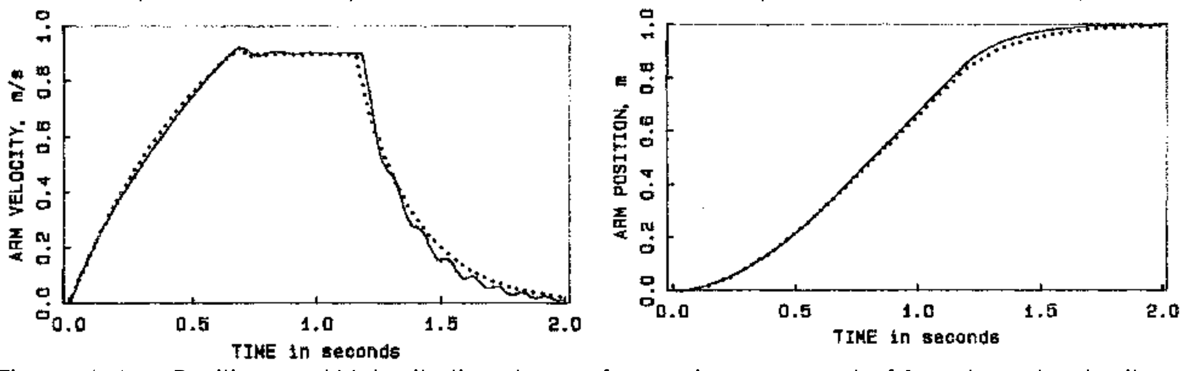

Figure 4. Arm Position and Velocity time traces for x-axis movement of 1 meter set-velocity = 1 meter/sec. Solid line is the integrated accelerometer data for actual arm movement; a dotted line shows the simulated output.

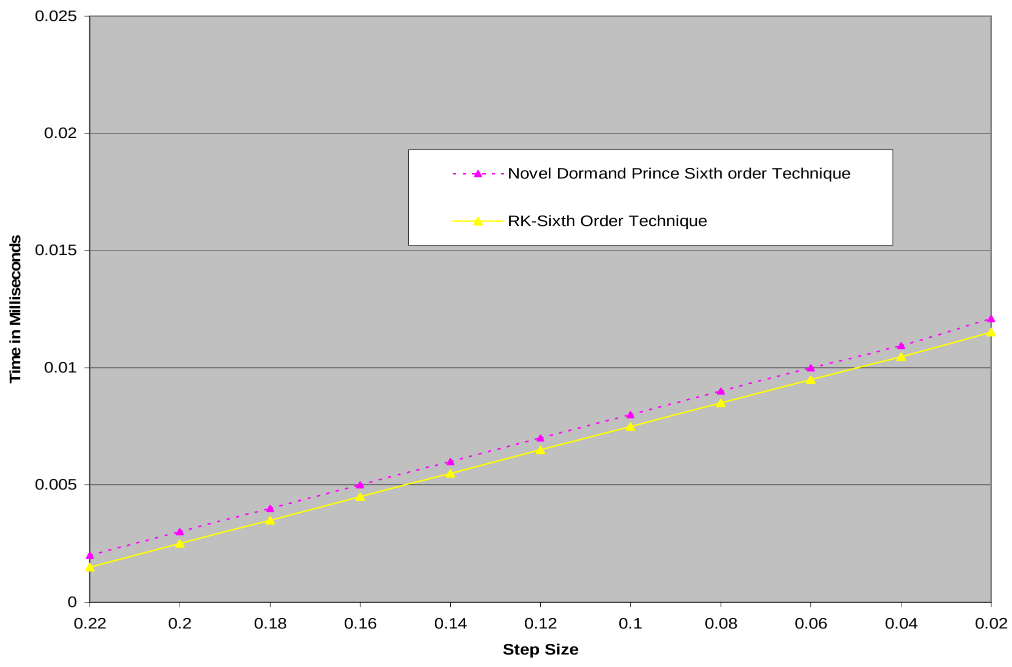

Figure. 5 CPU time of the Novel Dormand and Prince Sixth order technique and RK-Sixth order technique: A Comparison

5. DISCUSSIONS AND CONCLUSION

The present article sheds some light on two higher order numerical integration algorithms involved in Cartesian robot arm model problem. It is pertinent to pin-point out here that the obtained discrete solutions for the Robot Arm model problem using the RK-Sixth-order algorithm guarantee more accurate values compared to the RK-Fifth order technique. From the simulation results and from the graphical, it is observed that the so1ution obtained by the Novel Dormand and Prince Sixth order algorithm match well with the exact solutions of the Cartesian robot arm model problem in term of accuracy and less CPU time but the RK-Sixth order technique yields a little error because of the Taylors series. Hence, the Novel Dormand and Prince sixth order technique is more suitable for investigating the system of first order Cartesian robot arm model problem. When we make a work comparison with the, Novel Dormand and Prince Sixth order algorithm is efficient and takes less CPU time.

Furthermore, Novel Dormand and Prince Sixth order technique is stable because it is based on the exponential series algorithm. Hence, one can compute the numerical results for any length of time using Novel Dormand and Prince sixth order technique. This fact has motivated us to do further investigations in computing numerical solutions of industrial applicable problems of system of second order equations. The outcome of these results will form a subject of our forth coming paper.

ACKNOWLEDGEMENTS
This research work was fully supported as a part of Technical Quality Improvement Programme [TEQIP], sponsored by Govt. of India, National Institute of Technology, Tiruchirappalli-620 015, Tamilnadu, India. Telephone No: +91-0431-2501801/10; Fax No. : +91-0431-2500133; URL:http://www.nitt.edu.

REFERENCES
[1.] Astrom, K.J: 1983. Computer aided modeling, analysis and design of control systems-a perspective, IEEE Control Systems Magazine, Vol.3, No.2, pp.4-16, 1983.
[2.] Alexander, R.K and Coyle, J.J:1990.Runge-Kutta methods for differential-algebric systems, SIAM Journal of Numerical Analysis, Vol. 27, No.3, (1990), pp.736-752, 1990.
[3.] Bader, M: 1987. A comparative study of new truncation error estimates and intrinsic accuracies of some higher order Runge-Kutta algorithms, Computational Chemistry, Vol. 11, pp. 121-124, 1987.
[4.] Bader, M: 1998. A new technique for the early detection of stiffness in coupled differential equations and application to standard Runge-Kutta algorithms,Theoretical Chemistry Accounts, Vol. 99, pp. 215-219.
[5.] Bartoszewski. A and Jackiewic.Z:2007.Derivation of continuous explicit two-step Runge–Kutta methods of order three, Journal of Computational and Applied Mathematics, 205(2), pp.764-776, 2007.
[6.] Butcher, J.C:1964. On Runge proccesses of higher order, Journal of Australian Mathematical Society, Vol.4. pp.179, 1964.
[7.] Butcher. J.C: 1987.The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods, John Wiley & Sons, U.K,
[8.] Butcher, J.C: 1990.On order reduction for Runge-Kutta methods applied to differential-algebraic systems and to stiff systems of ODEs, SIAM J. of Numerical Analysis, Vol.27, pp. 447-456, 1990.
[9.] Dana Petcu: 2005: Speedup in solving differential equations on clusters of workstations, Int. J. of Computational Science and Engineering, Vol.1, No.2/3/4 pp. 134 -141, 2005.
[10.] Derby, S: 1983: Simulating motion elements of general-purpose robot arms, International Journal of. Robotics, Vol. 2, No.1, pp.3-12, 1983.
[11.] Evans, D.J: 1991. A new 4th Order Runge-Kutta technique for initial value problems with error control' Int. J. of Computer Mathematics, Vol.139, pp. 217-227, 1991.

[12.] Foroush Bastani, A and Mohammad Hosseini. A: 2007. A new adaptive Runge–Kutta technique for stochastic differential equations, Journal of Computational and Applied Mathematics, 206(2), pp.631-644, 2007.

[13.] Horowitz, E. Sahni, S and Rajsekar.S: (2004). Fundamental of Computer Algorithms, Galgotia publications, New Delhi, India.

[14.] Hung, C: 2000. Dissipativity of Runge-Kutta methods for dynamical systems with delays, IMA .J. of Numerical Analysis, Vol.20, pp. 153-166, 2000.

[15.] Jenson, O.C and Nelson, W.L:1983. Use of accelerometers for measurement and control of robot arm motion, Bell Labs Internal Document, 1983.

[16.] Jindong Tan, Ning Xi, Amit Goradia, Weihua Sheng: 2004. Coordination of multi-robot and human systems in a perceptive reference frame, Journal: Int. J. of Vehicle Autonomous Systems,Vol. 2, No.3/4 ,pp.201 – 216,2004.

[17.] John Butcher: 2007. General linear methods for ordinary differential equations', Mathematics and Computers in Simulation, In Press, Corrected Proof, Available online 25 February 2007.

[18.] Liu, M.Z, Hui Liang and Yang, Z.W:2007. Stability of Runge-Kutta methods in the numerical solution of linear impulsive differential equations Applied Mathematics and Computation, In Press, Accepted Manuscript, Available online 27 March 2007.

[19.] Mohsen Dadfarnia, Nader Jalili, Zeyu Liu and Darren M. Dawson: 2004. An observer-based piezoelectric control of flexible Cartesian robot arms: theory and experiment Control Engineering Practice, Vol. 12, Issue 8, pp.1041-1053, 2004.

[20.] Nelson, W.L and Ghang, J.D: 1984. Simulation of a Cartesian Robot Arm, Proceedings of the IEEE Intern. Conf. on Robotics and Automation, Vol.1, pp.212-219, 1984.

[21.] Orlandea, N and Berenyi, T: 1981.Dynamic continuous path synthesis of industrial robots using ADAMS computer program, Trans. of the ASME: J. Mech.Design, 103, pp.602-607, 1981.

[22.] Operating and service Manual: 1978. NG400 Series Servo Controller, Control Systems Research Division of Corporation, Pittsburgh, ST.

[23.] Shampine, L.F. and Gordon, M.K:1975. Computer solutions of. ordinary differential equations' W.H.Freeman, San Francisco. CA. p. 23, 1975.

[24.] Shampine, L.F. and Watts, H.A:1977. The art of a Runge-Kutta code. Part-I, Mathematical Software, Vol.3. pp. 257-275.

[25.] Walker,R.K, Gregory,C and Shah.S: 1984. MATRIXX A data analysis, system identification, control design and simulation package, IEEE Control Systems Magazine, Vol.2,No.4,1984.

[26.] Xinyuan Wu and Jianlin Xia: 2006.Extended Runge–Kutta-like formulae, Applied Numerical Mathematics, 56(12), pp.1584-1605, 2006.

[27.] Zanariah Abdul Majid, Mohamed B. Suleiman: 2006. Parallel block codes for solving large systems of ordinary differential equations, Journal: Int. J. of Simulation and Process Modelling, Vol. 2, No.1/2: pp. 98 –112, 2006.

[28.] J.R.Dormand and P.J.Prince: 1980. A family of embedded Runge-Kutta formula, Journal. Comput.Appl.Math, pp. 19-26, 1980.

[29.] B.Min,W.J..Lee and N.Park:2000. Efficient formulation of Raman amplifier propagation equations with average power analysis, IEEE Photonics Technol.Lett.12 (11), pp. 1486-1488, 2000.

[30.] X.M.Liu, M.D.Zhang: 2004. An effective method for two-point boundary value problems in Raman amplifier propagation equations, Opt.Communications,235,(1-3),75-82,2004.

[31.] T.G.Hodgkinson: 2004. Average power analysis technique for Erbiumdoped Fiber amplifiers using equally spaced pumps, J. Lightwave Technol. 22(6),pp. 1519-1522, 2004.

[32.] X.M.Liu, H.Y.Zhang and Y.L.Guo:2003.A novel method for Raman amplifier propagation equations', IEEE Photonics Technol. Lett. 15(3), pp. 392-394, 2003.

[33.] Xueming Liu: 2005. Effective Numerical algorithm for Fiber Amplifiers, Optical Engineering, 44(3), 035001(1-7), 2005.

[34.] R. Ponalagusamy and S. Senthilkumr: 2007. Time-Multiplexing CNN Simulation by an Efficient Numerical Integration Algorithm,JSSE,Communicated and Under Review.

[35.] Ponalagusamy, R and Senthilkumar. S: 2007. Multilayer Raster CNN Simulation: An Efficient Numerical Integration Algorithm, Journal of combinatorial mathematics and combinatorics computing, 2007