The Servo-Control System for an Omnidirectional Mobile Robot

Dai Feng*, Mark B. Friedman, and Bruce H. Krogh*
The Robotics Institute
*Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

This paper describes the design and implementation of a motor drive control system for an omindirectional wheelbase, the Ilonator platform, developed at the Robotics Institute at Carnegie Mellon University. The experimental mobile platform is mounted on four omnidirectional wheels which permit a full three degrees of freedom (3 DOF) in the plane. The distributed servo-control system is coordinated by a Motorola 68000 on-board computer which converts 3 DOF acceleration commands into individual velocity commands for the four wheels. Four Hewlett-Packard HCTL-1000 motor control chips close the velocity control feedback loops for the wheels. The paper details the platform kinematics and motion command structure, servo hardware design and implementation, software for command and joystick control, and the experimental tuning of the controller parameters. The presentation emphasizes practical issues that need to be addressed in the actual implementation of a distributed servo-control system for a complex mobile platform.

1 Introduction

This paper concerns the analysis, design and implementation of a motor drive control system for the Ilonator, an experimental omnidirectional mobile platform developed in the Robotics Institute at Carnegie Mellon University. The four Ilonator wheels are omnidirectional, permitting a full three degrees of freedom (3 DOF) in the plane without conventional steering of the wheel axes [1]. In an earlier paper, we presented the basic kinematic equations of motion for the Ilonator and described the implementation of a single-computer open-loop controller which executed position-to-position commands[2] along kinematically simple paths [2]. Experimental results from the earlier implementation indicated that the motor torques required to execute a given trajectory varied significantly depending on the orientation of the Ilonator with respect to the path. These variations were attributed primarily to the frictional characteristics of the rollers on the omnidirectional wheels.

In the present paper we describe the design and implementation of a distributed servo-control system for the Ilonator which can execute arbitrary trajectories specified by a sequence of 3 DOF acceleration commands. Experimental results demonstrate the effectiveness of the distributed wheel control for overcoming the orientation-dependent torque requirements noted in our earlier work. The accuracy and repeatability of trajectory execution with the present control architecture makes the Ilonator an attractive wheelbase for research and applications of mobile robots in indoor environments where the floor surface is smooth and regular. Moreover, the servo-control software provides useful information for higher-level monitoring and control functions such as detection of wheel slip.

Figure 1 illustrates the level of control addressed in the present paper. The servo-controller accepts commands in terms of world-coordinates, indicating the desired trajectory for the mobile robot. The higher-level controller (human or otherwise) generates commands for the servo controller off-line or in real-time using sensory information to create and update a world model for path planning and obstacle avoidance. The software described in this paper could be integrated into a variety of higher-level control architectures. Currently, the Ilonator can be driven in one of two modes: real-time joystick control, where the servo commands are generated by sampling the voltages on a 3 DOF joystick; or command-file control, where the servo commands are down loaded into the on-board computer from an off-line trajectory planning program.

In the following section we review the basic kinematic equations of motion for the Ilonator which are used in the on-board computer software to generate the wheel velocity commands from the world-coordinate trajectory commands. Section 3 describes the servo-control hardware, particularly some of the practical considerations that needed to be addressed to make the conceptual design actually work. The control software is described in section 4, and experimental results are presented in section 5. In particular, we illustrate the significant improvement in trajectory tracking performance that was obtained by tuning the parameters in the four wheel control feedback loops. We also obtain limits on the dynamic response of the system which can be applied at the planning level to guarantee a prescribed level of path-tracking accuracy. The concluding section summarizes the results of this work and plans for future development and applications of the Ilonator.

2 Ilonator Kinematics

The Ilonator platform consists of a rectangular base with an omnidirectional wheel mounted on each corner, each wheel being driven by a geared, permanent magnet DC motor. The axes of the

[2]Throughout this paper we use the term *position* to refer to all 3 planar degrees of freedom.

CH2750-8/89/0000/1566$01.00 © 1989 IEEE

motors are fixed and, due to the construction of the wheels, motion in any direction is possible without conventional steering of the wheels axes. It is possible to accomplish arbitrary translation in any direction, rotation, or a combination of rotation and translation by driving the wheels at proper velocities.

The omnidirectional wheels are described in [1], [3]. Because the rollers can be mounted with their axes aligned in two different ways, it is possible to have "left hand" and "right hand" wheels, where right- or left- handedness indicates the direction in which the wheel can freely move at 45° with respect to the wheel axis with only roller rotation. The inverse kinematics of the Ilonator map the translational and rotational velocities of the platform into the four angular velocities of the wheels. This mapping is required to transform a desired velocity trajectory for the platform into reference velocity trajectories for the four wheel controllers. The wheelbase is shown in figure 2 with the axes, angles, and wheels labeled using the following notation:

ω_i - angular velocity of wheel 'i', defined as positive when the top of the wheel turns toward the front of the platform.

x-y - the global inertial coordinate system (global frame).

x'-y' - the coordinate system local to the platform (local frame).

Θ - orientation of the platform in degrees (0° is defined as front facing directed along the x axis, with Θ increasing counter-clockwise.)

v - platform translational velocity vector.

D - distance from the centroid of the platform to the wheels' points of contact with the ground.

Ω - rotational velocity of the platform (defined positive if turning counter-clockwise).

ϕ - an angle describing the rectangular platform, formed by a line drawn form the center of the platform to one corner and the side of the platform. (in degrees)

r_w - wheel radius.

The inverse kinematics which map the platform translational and rotational velocities into the wheel angular velocities are given by [4], [2]

$$\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ \omega_4\end{bmatrix} = \frac{1}{r_w}\begin{bmatrix}1 & -1 & -2\,D\,\cos(45-\phi)\\ 1 & 1 & 2\,D\,\cos(45-\phi)\\ 1 & -1 & 2\,D\,\cos(45-\phi)\\ 1 & 1 & -2\,D\,\cos(45-\phi)\end{bmatrix}\begin{bmatrix}v_{x'}\\ v_{y'}\\ \Omega\end{bmatrix} \quad (1)$$

This transformation is the basis of the command structure and software for coordinating the wheel velocities described below in section 4.

3 Servo-Control Hardware

Closed loop control of the Ilonator wheels is based on the use of a Hewlett Packard HCTL-1000 general purpose motion control integrated circuit [5], [6] for each of the four geared DC motors. This IC is a ROM programmed microprocessor with algorithms to support four different control modes:

- Position Control
- Proportional Velocity Control
- Point to Point, Trapezoidal Profile Velocity Control
- Integral Velocity Control using Linear Acceleration Profiles

Functions are specified by writing to registers from a shared bus. The chip is capable of supporting both analog DC and pulse width modulation (PWM) motor torque control. PWM was selected for this project because of the simplicity of the hardware needed to implement high-current drives.

As illustrated in figure 3, power MOSFETs connected in an "H" bridge configuration are used for bidirectional control from a single polarity power supply consisting of deep discharge lead-acid storage batteries. An incremental optical encoder provides position feedback from each wheel to its HCTL-1000 controller. The controllers are optically isolated from the H-bridge drive logic (figure 3) to increase motor commutation induced noise immunity. Similarly, separate power supplies are used for the supervisory MC68000 control board, the HP servo IC board and the H-bridge drive circuitry to reduce the susceptibility of the control system to power surge induced noise.

The HCTL-1000 circuits are designed to accept 2-channel incremental digital position feedback signals. The circuits internally decode these quadrature signals to determine direction of motion and use all four edges of the 2 channel square-wave quadrature signal waveform to increase accuracy. Initially, our 400 quadrature-count optical encoders were coupled directly to the Ilonator wheels. Because steering the platform is based on differential wheel velocity, low wheel rotation rates are often required. Often, the ROM-encoded software gain functions on the HP motion control chip required higher rates of digital feedback than were provided by our position encoders on slowly turning wheels. We minimized this signal rate problem by coupling our position encoders to the high speed drive shafts of the geared wheel drive motors, giving over 23K quadrature-counts per wheel revolution.

Each motion control IC provides two digital outputs to its H-bridge drive, a sign bit and a PWM bit that respectively control direction and speed. After optical isolation, these two signals maybe digitally decoded into four possible motor drive modes: power for forward motion, power for reverse motion, coast, and dynamic braking. To minimize battery power dissipation during pulse transitions, MOSFET drive transistors were selected over bipolar power transistors. Two H-bridge drive circuits were tested, one based on all N-type transistors and one using both P- and N-types. The latter case is illustrated in figure 5. The all N-type implementation requires an extra power supply or transformer coupling to achieve the biasing required to turn on the N-type FETs on the positive side of the H-bridge. Because of this added complexity, we chose to use P-type FETs for the positive side of the bridge. The simple PWM amplifier circuit shown in figure 3 has proven to be highly reliable.

4 Servo-Control Software

The overall structure of the control software for the on board computer is illustrated in figure 4. This program has three basic functions, namely, initialization of the motor control chips, real-time

ELECTRONICS ENGINEERS, INC.

generation of the wheel velocity trajectories from either the joystick controller or an acceleration command list, and collection of wheel encoder data during the trajectory execution for post-experimental analysis. In this section we describe each of these functions.

As described in the previous section, each motor control chip is programmable, allowing selection of the desired control mode (position or velocity, proportional, integral, etc.) and specification of the pole and zero for a digital filter in the control algorithm. Before executing a trajectory, the control program must initialize these parameters in the motor control chips. We have chosen the velocity control mode with integral action to compensate for varying load conditions on the motor. The digital pole and zero values were determined experimentally, as described in the following section. As an additional function, the control program allows the user to query the individual motor control chips to read the encoder count registers and verify the initialization parameters.

After the motor control chips have been initialized, the user can select the source of trajectory commands to be executed by the control program. If joystick control is selected, the control program samples the three joystick voltages at a frequency of 333 Hz. These voltages are considered to be velocity commands (v_x, V_y, Ω) in the local coordinates of the cart. In-line assembly code for the Motorola 6800 implements the inverse kinematics to generate the desired wheel velocities ω_1, . . ., ω_4 which are distributed by the control program to the four motor controllers.

The other mode of operation is to execute a trajectory specified by a list of acceleration commands of the form:

$$[a_x \quad a_y \quad a_\Omega \quad T]$$

where $a_x=\dot{v}_x$, $a_y=\dot{v}_y$, $a_\Omega=k\dot{k}\Omega$ and T is the duration of the acceleration in msec. The control program implements the accelerations by generating incremental changes in the desired wheel velocities at a frequency of 333 Hz. Thus, for a given acceleration command, the velocity command to each wheel is incremented as:

$$\omega_i(k+1) = \omega_i(k) + \Delta T\, \Delta\omega_i$$

where ΔT = 3msec, and for i = 1, . . .,4, $\Delta\omega_i$ is simply the desired wheel acceleration obtained from the forward kinematics equation (5). For an acceleration command of duration T, the velocity is incremented T/3 times. There is currently no correction made for fractional increments.

To execute a commanded trajectory, a set of acceleration commands are down-loaded into the on-board computer from the supervisory computer (an IBM PC AT). The controller executes the commands in sequence, and, as a safety measure, brings the Ilonator to a stop immediately when the command sequence has been depleted.

The final function of the control program is to collect on-line data during the execution of a trajectory. The user can request that the contents of the encoder counter registers for all four wheels be stored at each command cycle(3 msec). Following the execution of a trajectory, these data can be uploaded to the supervisory computer for analysis. This is the source of the experimental data presented in the following section.

5 Experimental Results and Tuning of Controller Parameters

The performance of the distributed control system for the Ilonator depends on the control mode and gains selected for the four wheel control loops. Because of the orientation-dependent load characteristics, velocity servoing with integral control action was selected as the control mode. As shown in figure 5, there are four gains to be selected in this control mode, the integral gain, the loop gain and two filter gains which determine the pole and zero, respectively of a digital filter in the forward path of the control loop.

Initial experiments indicated that the steady-state (constant velocity) error was negligible under integral control. To tune the remaining controller parameters, we performed a set of experiments to evaluate the overall trajectory-tracking performance of the Ilonator. Rather than tune the wheel control loops independently, we collected data from the four wheels during the execution of a trajectory and then generated from these data a least squares estimate of the actual trajectory executed by the Ilonator. Based on the tracking performance in world (Cartesian) coordinates, we determined the best controller gain.

The experiments used to tune the controller gains were based on translational trajectories with constant orientation (Ω=0). The desired trajectories were generated using acceleration commands as described in section 3. From these commands the desired reference velocity was obtained as $\mathbf{v}_d(k)$, where the index k represents sampling increments of 3 msec.

The encoder data were collected at 3 msec intervals from all four wheels, from which estimates of the actual wheel velocities $\omega_a(k)$ were computed as

$$\omega_a(k) = (\theta_a(k) - \theta_a(k-1))/\Delta T$$

where $\theta_a(k)$ is the vector of encoder readings. To obtain an estimate of the actual velocity trajectory $\mathbf{v}_a(k)$ executed by the Ilonator the least-squares estimate for the inverse kinematics was used, which is given by

$$\mathbf{v}_a(k) = \mathbf{L}\, \omega_a(k)$$

where

$$\mathbf{L} = \frac{r_w}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \end{bmatrix}$$

Figures 6 through 7 show the results of four experimental runs, all generated with step (velocity) inputs as reference commands with v_y = 0 m/sec. These results illustrate the effects of varying the poles, zeros and loop gains of the wheel control loops. In all cases the filter pole was placed at b=0 to cancel the system zero at 0 due to sampling. The best performance (fast response with little overshoot) was obtained by placing the filter zero as near to 1 as possible. We note that the step responses shown above correlate well with what might be predicted from a simple discrete-time root-locus analysis of a single motor response. If the motor/load system has a positive pole at z=p < 1, reflecting the mechanical time

constants of the system, a fast, reasonably damped response can be obtained when the controller zero is placed between p and z=1. Based on the results in 7, we chose a loop gain of K=32, and filter zero at a=255/256.

Since the four-wheel Ilonator platform is a redundant system with respect to the 3 degrees of freedom in the plane, the wheel velocities computed from on-line data will not, in general, be consistent with the inverse kinematics which were based on a no-slip assumption. The least-squares estimate of the platform velocity vector can be interpreted as the projection of the measured wheel velocity vector into the 3-dimensional space of wheel velocities which are consistent with the no-slip kinematics. To evaluate the consistency of the wheel data with respect to the conditions for no-slip, the least-squares projection error is computed as:

$$||\mathbf{e}(k)||^2 = \frac{1}{2}||\mathbf{P}\omega_a(k)||^2$$

where

$$\mathbf{P} = \begin{matrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{matrix}$$

When the measured wheel velocities $\omega_a(k)$ are not in the range space of the inverse kinematics mapping the magnitude of the projection error $||\mathbf{e}(k)||^2$ indicates the relative inconsistency in the data. There will always be a small amount of error due to the encoder discretization of the wheel angle measurements. However, when the projection error is large, it it indicates that some wheels are slipping. This is illustrated by the large initial projection error in figure 8. As would be expected, the large initial acceleration leads to large projection errors at the beginning of the trajectory due to wheel slip. As the steady-state velocity is achieved, the projection error is greatly reduced, but still non-zero, because of the discretization noise due to the encoders.

To use the Ilonator in higher-level control experiments, it is desirable to limit the magnitudes of the acceleration commands so that wheel-slip is minimized. This will enable the servo-control system to faithfully track the desired velocity trajectory. Figures 8 and 9 show the projection errors incurred as the velocity is ramped using acceleration of 3.33 m/sec^2 and 1 m/sec^2, respectively. Considering the steady-state error magnitude to be the acceptable level of performance, we conclude that no-slip can be assumed when the requested acceleration is limited to be less than or equal to 1 m/sec^2.

Experiments similar to the ones presented above were performed for straight-line translation at a variety of angles with respect to the Ilonator orientation. These experiments indicated that the system response characteristics are essentially independent of the orientation. Thus, the integral control action and high-gain feedback have eliminated the effects of the orientation-dependent frictional characteristics observed in our earlier work [4].

6 Conclusions

This paper presents the design and implementation details for a functioning omnidirectional mobile robot, the Ilonator. We note that unlike some wheelbases which achieve 3 DOF motion only by stopping and pivoting the wheels, the Ilonator can accelerate both translationally and rotationally while it is moving. This makes the Ilonator an attractive wheelbase for research in autonomous navigation since its motion is not constrained by the complex nonholonomic constraints characteristic of conventional wheelbases [7], [8].

To use the Ilonator servo-controller in a hierarchical control system, it would be useful in some applications to be able to issue commands with respect to a fixed reference frame. The present command structure designates accelerations with respect to the *local* reference frame of the wheelbase. Thus, when the angular velocity is nonzero, the translational acceleration commands are with respect to a rotating reference frame. We are currently implementing software to accept trajectory commands with respect to a fixed reference frame for both joystick and command file control.

The omnidirectional nature of the Ilonator allows the direct application of higher-level feedback schemes based on simple point-mass models of a mobile platform. We are currently developing a software interface to drive the Ilonator from simulations of our feedback scheme for local navigation and obstacle avoidance.

Other research applications of the Ilonator currently being investigated include coordination of mobility and manipulation, coordinated dynamic control of mobility, and applications of omnidirectional mobility to issues in space robotics and health robotics.

Acknowledgments

The authors gratefully acknowledge the efforts of Edwin Hoch, Gerard Kim, Lee Wilfinger, Peter Stern, and Wayne Carriker, each of whom contributed to various aspects of the hardware, software and experimental results presented in this paper. This research was supported in part by the National Science Foundation under Grants ECS-8404607 and DMC-8451493, the Robotics Institute, Carnegie Mellon University, and The Rehabilitation Institute of Pittsburgh.

References

1. Ilon, Bengt Erland, "Wheels for a Course Stable Selfpropelling Vehicle Movable in any Desired Direction on the Ground or Some Other Base". Patent number 3,876,255

2. D.J. Daniel, B.H. Krogh and M.B. Friedman, "Kinematics and open-loop control of an ilonator-based mobile platform", *Proceedings 1985 IEEE International Conf. on Robotics and Automation*, St. Louis, MO, Mar 1985, pp. 346-353.

3. Adams, Russ, "Omnidirectional Vehicle", *Robotics Age*, Vol. 6, No. 2, 1984, pp. 21-2.

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

4. Daniel, David J., "Analysis, Design, and Implementation of Microprocessor Control of a Mobile Platform", Master's thesis, Carnegie-Mellon University, September 1984.

5. Hewlett-Packard, *Optoelectronics Designer's Guide*, 1988.

6. W.D. Fisher and M.S. Mujtaba, "Using the HCTL-1000 Motor Control Chip for Coordinated Robot Motions", *Proceedings 1988 American Control Conference*, Atlanta, GA, June 1988, pp. 663-664.

7. T.J. Graettinger and B.H. Krogh, "Evaluation and Time-Scaling of Trajectories for Wheeled Mobile Robots", *ASME Journal of Dynamic Systems, Measurement, and Control*, to appear.

8. J.C. Alexander and J.H. Maddocks, "On The Maneuvering of Vehicles", *SIAM J. App. Math.*, Vol. 48, No. 1, Feb. 1988, pp. 38-51.

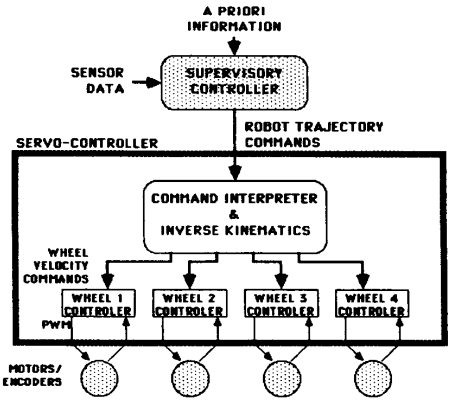


Figure 1: Multi-level vehicle control schemes.

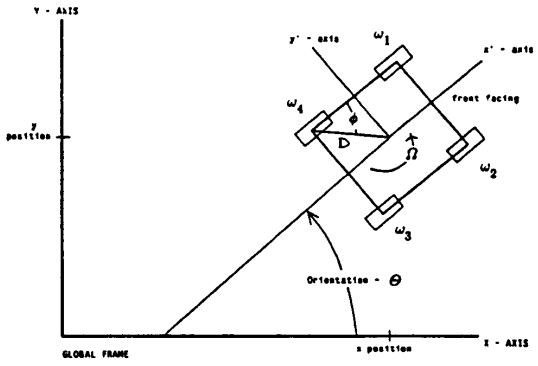


Figure 2: Definitions of local and global coordinate frames.

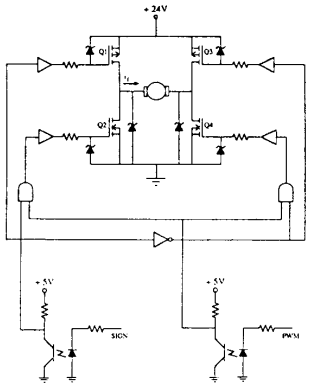


Figure 3: H bridge PWM power amplifier circuit.

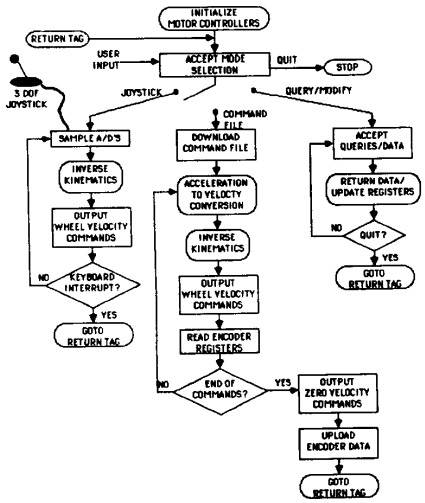


Figure 4: On-board Ilonator servo-control program.

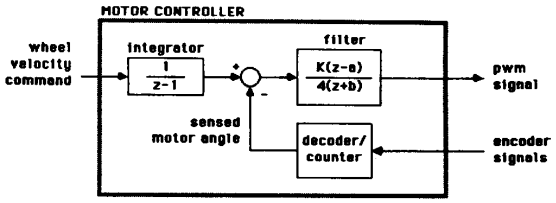


Figure 5: HP HCTL-1000 motor control feedback loop; velocity mode with integral action.

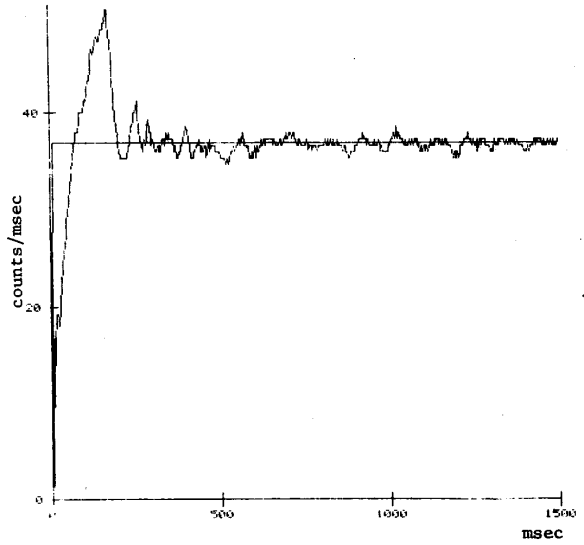


Figure 6: Ilonator velocity step response for K=16, a=192/256.

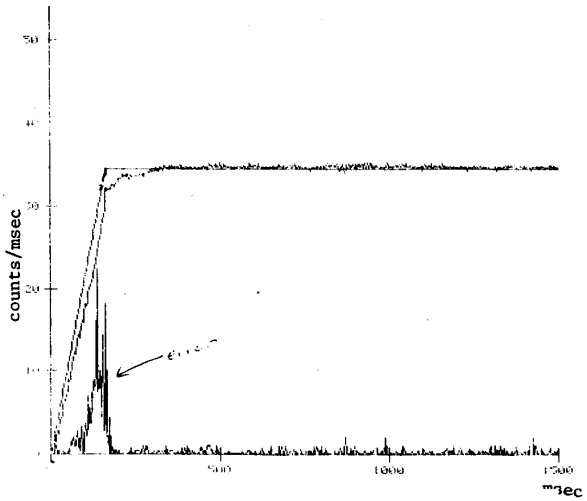


Figure 8: Velocity response and projection error for acceleration of 3.33 m/sec^2.

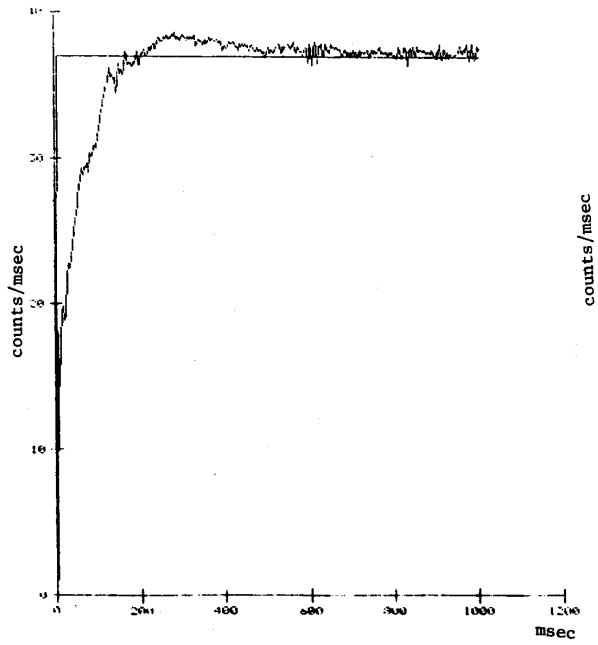


Figure 7: Ilonator velocity step response for K=32, a=255/256.

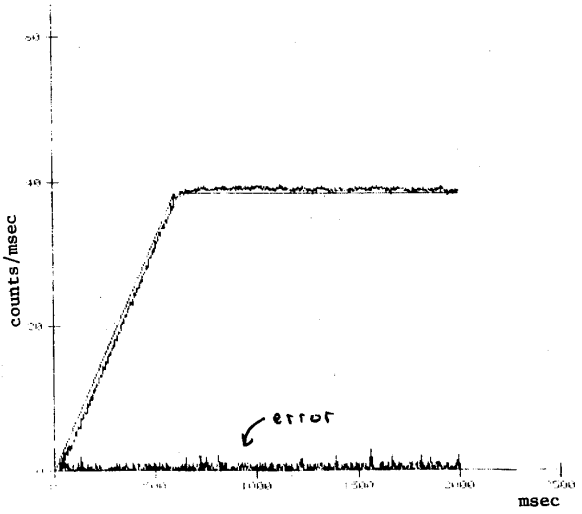


Figure 9: Velocity response and projection error for acceleration of 1.0 m/sec^2.

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.