N94- 32472

TECHNOLOGY TRANSFER OF OPERATOR-IN-THE-LOOP SIMULATION

K. H. Yae
H. C. Lin
T. C. Lin
Simulation and Design Optimization of Mechanical Systems
Department of Mechanical Engineering
The University of Iowa
Iowa City, Iowa 52242
and
H. P. Frisch
NASA Goddard Space Flight Center
Greenbelt, MD 20771

October 7, 1993

Abstract

The technology developed for operator-in-the-loop simulation in space teleoperation has been applied to the Caterpillar's backhoe and wheel loader, and off-highway truck. On a SGI workstation, the simulation integrates computer modeling of kinematics and dynamics, real-time computation and visualization, and an interface with the operator through the operator's console. The console is interfaced with the workstation through an IBM-PC in which the operator's commands were digitized and sent through a RS 232 serial port. The simulation gave visual feedback adequate for the operator in the loop, with the camera's field of vision projected on a large screen in multiple view windows. The view control can emulate either stationary or moving cameras. This simulator created an innovative engineering design environment by integrating computer software and hardware with the human operator's interactions. The backhoe simulation has been adopted by Caterpillar in building a virtual reality tool for backhoe design.

Key Words: Teleoperation, redundant manipulator, simulation, recursive dynamics

1. Introduction

Teleoperation [1,2] involves a human operator, a hand controller, and a manipulator, in which the manipulator under the operator's supervision completes tasks ranging from simple trajectory following to pick-and-put operation. The operator's interaction with the manipulator becomes a key issue, because of the intrinsic difference between a human operator and a robotic manipulator; that is, a human operator works more efficiently in Cartesian space [3] whereas most manipulators are designed with joint servo control.

For off-line trajectory analysis, the inverse position and orientation problem can be solved iteratively using inverse velocity/angular velocity solutions [4]. In teleoperation, however, the desired configuration of the end-effector is known only after the operator has commanded through the mini-master. This means that both Jacobian construction and inverse kinematic analysis must be completed on-line in real time. Unlike the Jacobian defined for the pre-determined end-effector's trajectory, the constraint Jacobian can be constructed on-line as the operator controls the manipulator's end-effector through the mini-master.

The constraint Jacobian is derived from the six constraints [5] that are imposed between the current and the desired end-effector's Cartesian position and orientation. The desired Cartesian position and orientation of the end-effector is viewed as the target that the current position and orientation will eventually have to assume. Constraining the two sets of position and orientation yields six constraints from which the constraint Jacobian is constructed [6]. This Jacobian and its pseudoinverse are used in iterations to yield the joint command angles necessary for the joint controllers. The procedure is illustrated with a Kraft 6-d.o.f. mini-master and a 7-d.o.f. redundant manipulator [7] and also with an operator's console and a backhoe.

Since the manipulator's dynamics involves kinematic redundancy and intermittent kinematic loop closure, the dynamic model is constructed in the recursive Newton-Euler dynamic formulation adapted for high-speed simulation of general constrained mechanisms [8,9,10,11,12]. According to d'Alembert's principle, this formulation starts out with the virtual work of all the links and the cut joint, where the cut joint is expressed in the Lagrange multiplier

The next goal for the project include improving the software and hardware for generating VR images, developing prescribed fly-throughs and incorporating multi-media into the VR fly-throughs. Other anatomical imaging data will be obtained from CT scans, MRI and Cadavers to develop VR imaging of anatomical regions that contain different tissues with different data densities

REFERENCES

1. Brooks, F. P. 1988. "Grasping Reality Through Illusion—Interactive Graphics Serving Science." *Proceedings AMC SIGCHI.*

2. Fuchs, H., Levoy, M., and Pizer, S. M.. 1989. "Interactive Visualization of 3D Medical Data." *IEEE Computer*, August.

3. Furness, T. 1987. "Designing in Virtual Space." Chapter in W.B. Rouse and K.R. Boff, eds., *System Design: Behavioral Perspectives on Designers, Tools, and Organizations. North Holland.*

C1-6.

and the Cartesian coordinates that include both independent and dependent coordinates. The dependent coordinates are then replaced with independent joint coordinates. The replacement is recursive and systematic and it has been automated [8,10]. Consequently, this method facilitates modeling by allowing formulation in Cartesian coordinates, expresses the model in terms of independent joint variables, and improves computational efficiency. As a result, this formulation can efficiently generate the equations of motion of a constrained and/or unconstrained multibody system for a single processor computer [8] and for a multi-processor computer [10]. The manipulator's dynamic equations of motion are then combined with the control system. This model of dynamics and control is then teleoperated by an operator through the mini-master.

2. Dynamics and Control of Manipulators

Being driven interactively by the operator-in-the-loop, the simulation requires that dynamics and control computation and graphics rendering be fast enough to provide the operator with adequate visual cues without too much delay. For high-speed computation, a recursive Newton-Euler dynamic formulation has been used to model the manipulator and vehicle dynamics. The recursive formulation derived in [8,9,10,11,12] facilitates modeling a general constrained mechanical system and improves computational efficiency.

2.1. Recursive dynamic formulation in spatial vectors

The basic procedure for deriving the equations of motion based on the Newton-Euler formulation can be understood if we view it as a way to calculate the joint driving force and torque necessary to realize a given trajectory of the joint coordinates.

For illustration, let us consider a tree-like open chain of rigid bodies with their joint constraints known. Suppose that the current values of joint displacements, velocities, and accelerations are known and that the force and torque exerted on the end link by the environment and a grasped object are given. First, we calculate the angular velocity, the angular acceleration, the linear velocity, and the linear acceleration of a link with respect to the reference frame. Second, using Newton's and Euler's equations, we calculate the force and torque that must be applied to the center of mass of a link to realize such a motion. Third, we calculate the force and torque that must be applied at a joint to produce the corresponding force and torque, starting from the end link and moving inward to the base, with the given values of the force and torque at the end link. Finally, we calculate the joint driving force (and/or torque) at each joint [13].

The kinematics of two contiguous bodies is defined using the spatial velocity vector [14], also called the velocity state vector in [8]. The use of spatial vectors simplifies the derivation and improves computational efficiency in dynamic simulation. Each link has two body-reference frames located not necessarily at the center of gravity but at two joints, proximal and distal. Each joint connects a proximal (inboard) link and a distal (outboard) link. Starting from the base body, the recursive kinematics is developed from a proximal link to a distal link. After the kinematics has been defined, starting from a tree-end body, the recursive dynamics is developed in the opposite direction, i.e., from a distal link to a proximal link.

The spatial velocity vector represents the velocity of a point that is moving with the body but is instantaneously coincident with the origin of the global inertial coordinate [14]. An interesting property of this vector is that it remains invariant no matter where the body reference frame is located within the body [10]. The spatial velocity vector is then the sum of that of its inboard body's spatial velocity vector and the angular velocity of the joint.

The virtual work of the manipulator is first expressed in the variations of spatial vectors [8,10,12,15]. When there exists a kinematic loop in the system, the cut-joint technique [16] is used. This equation is then converted to the recursive equations of motion in joint variables. The conversion proceeds from the outmost body to the base body.

2.2. The power train model of 950F wheel loader

The wheel loader dynamics requires manipulator dynamics for the loader and vehicle dynamics for the tractor. The engine's driving torque is transmitted through a gear train to the four tires. The wheel loader is then driven by the tire-ground reaction forces.

The gear train consists of torque converter, clutch, brake, and final reduction gears. The computation of the power train starts from the torque converter model. The input and output torque of the torque converter are determined first

based on the input speed (engine RPM) and the output speed (clutch speed or vehicle speed). The input torque to the torque converter is viewed as the engine load. The converter output torque is the input to the rest of the power train. This output torque is transformed into wheel driving torque through the gear train. Once the wheel velocity is computed, the driving force and torque acting on the chassis is then computed through the tire-ground interaction. During the simulation, the wheel speed is updated by integrating the accelerations of the engine and the wheels (tires).

Engine model: A simplified empirical engine model is developed for real-time simulation. The engine's torque generation is determined by the engine speed and the throttle. The net torque applied to the equation of motion is equal to the sum of the engine torque and the torque reflected by the torque converter. This net torque yields the engine acceleration, which is then integrated to give the speed for the next time step. The generation of engine torque can be characterized by three curves: power, friction, and governor curve as shown in Figure 1. The output torque curve consists of three segments that are determined by the throttle percentage. Once the torque curve is known, the actual output torque is determined by the engine's running speed. These curves are constructed based on the experimental data.

Tire and brake model: For real-time simulation, no differential gear is modeled in dynamic analysis. The output torque from the transmission is equally distributed to the four wheels. In addition, each wheel can rotate freely with no coupling with other wheels. A simplified model is developed for computing the tire-ground reaction forces, based on two assumptions: 1) the ground is flat and 2) the tire contacts the ground vertically at a point with an effective radius (deformed radius). While a tire is rolling, the translational direction of the chassis may not align with the tire's longitudinal direction. The angle between the translational and the longitudinal direction is called the slip angle. The slip index that represents the longitudinal deformation of a tire is used to compute the tire-ground interaction force in traction (longitudinal) and cornering (lateral) components. These two components give the driving force to the wheel loader.

2.3. Actuators and Control Systems

The control and actuation system generates the driving force and torque that is fed into the dynamic model. As for the telerobotic manipulator, the controller has been modified for real-time simulation and validated with the original controller's model through actual experiments conducted in the Robotics Laboratory at NASA Goddard Space Flight Center. Finally the simplified controller and the dynamics are put together. The nonlinear dynamic model feeds back joint position and velocity to the joint controller, and the joint controller generates driving torque that drives the manipulator.

In the wheel loader shown in Figure 1, four hydraulic systems drive the swing, plate, and Z-bar linkage. Each system includes a spool valve and a cylinder. The hydraulic actuator is modeled as a five-port system, with the assumptions of constant supply and drain pressure, compressible fluid, and zero-lapped spool valves.

A mathematical model of the hydraulic actuator has been first developed and then modified to include static friction force and damping force. The model consists of a set of differential equations that relate the time derivatives of two pressures in one cylinder with the flow rates and the piston velocity. The cylinder pressures are then obtained by integrating the differential equations. The pressure difference yields actuating force in a cylinder.

3. Interface with the Kraft mini-master and an operator's console

A situation unique to teleoperation is that the target position and orientation is entered on-line by the operator. Consequently, the control algorithm must first translate incremental Cartesian coordinates into incremental joint angle change.

As for the wheel loader, the operator's inputs are accelerator pedal, brakes, gear shift, and bucket lift and tilt. Some of them are interpreted as spool valve displacements. They all are merged into dynamics, control, and hydraulic actuator models without any need for inverse dynamics.

3.1 Cartesian Space Control of a Telerobotic Manipulator with the Kraft Mini-Master

The telerobotic manipulator used in this research is a 7 d.o.f. device with shoulder roll and pitch, elbow roll and pitch, wrist roll and pitch, and toolplate roll. The mini-master in Figure 3 is a 6 d.o.f. force-reflective device

446

kinematically similar to a human arm, with shoulder yaw and pitch, elbow pitch, and wrist yaw, pitch, and roll. Even with one joint of the 7 d.o.f. manipulator being locked, the one-to-one mapping of joint angles between these two simply confuses the operator. It is, therefore, necessary to translate the mini-master's joint angle readings into the joint angle commands to the manipulator's joint controllers.

For the interface with the mini-master, we found it easier to first build the mini-master's kinematic model and expand the mini-master's workspace to the manipulator's workspace as closely as possible. The mini-master's kinematic model converts three joint angles (shoulder yaw and pitch and elbow pitch) into the hand grip's Cartesian position in the mini-master's workspace; then, this Cartesian position is transformed into the Cartesian position of the manipulator's end-effector, which is then converted into four (lower) joint angles of the manipulator through a generalized inverse method. Similarly for orientation, the other three (upper) joint angles of the mini-master (wrist yaw, pitch, and roll) are mapped onto manipulator's three joint angles (wrist roll and pitch and toolplate roll).

The control algorithm must first translate incremental Cartesian coordinates into incremental joint angle change [4], and then the incremental joint angle change into joint torque. This translation requires inverse kinematic analysis of a manipulator as described in the following.

The end-effector's Cartesian position and orientation is defined by the joint angles through a kinematic relation. For known Cartesian coordinates, instead of seeking an explicit expression for the joint angles, we view the kinematic relation as constraint equations. From their variations the constraint Jacobian is defined.

When the telerobotic manipulator is kinematically redundant, a single Cartesian position of the end-effector may correspond to multiple sets of associated joint angles. If an additional condition is added, such as a minimization of the Euclidean norm of angular velocity [4,7,17], a unique set of joint angles can be identified. This method is called the pseudoinverse method, or Moore-Penrose generalized inverse method [18]. Although using the pseudoinverse for the derivatives does not yield an inverse function between the variables themselves [19], the pseudoinverse method turns out to be useful for real-time on-line computation as is required in teleoperation.

Teleoperation is initiated by the operator's command input through the mini-master. The operator's input motion is divided into a series of increments in Cartesian space. These Cartesian increments are transformed into incremental angles by the Jacobian. The incremental joint angles are then input to the joint servo controllers, which in turn produce necessary joint torque. The resulting position and orientation is displayed on a graphics workstation to provide the operator with visual feedback.

The Kraft mini-master (Figure 3) is kinematically similar to a human arm so that the human operator can ``wear'' it comfortably for use in Cartesian space with minimal cognitive learning [20]. The manipulator of seven degrees of freedom, on the other hand, is joint-controlled. Therefore, the task is to relate the mini-master's six joint angles to the seven joint angles of the manipulator in the way that the manipulator's end-effector follows the mini-master's grip.

The host computer, an IRIS 4D/320 VGX, provides enough computational power for high-speed simulation of the dynamic and control system, as well as high-speed graphics rendering and data communication between the host computer and the mini-master. Through a RS 422 serial port, the Kraft mini-master is capable of sending out the readings of its own joint angles and receiving feedback joint torque for tactile feedback. When the sensed joint angles reach the host computer, they are converted into the joint input commands. The feedback joint torque received can represent a payload or reaction force and torque for the end-effector in contact with the environment.

3.2. Interface between operator console and simulation

The setup of simulation is shown in Figures 4 and 5. During the simulation the operator receives visual feedback and controls the wheel loader through the operator's console. The console generates three types of output: the analog signal from the rotational potentiometers, digital signal through digital I/O port, and the encoder output signal. All these inputs are consolidated on a PC and sent to the host computer, IRIS 4D/320 VGX, through RS 232 ports, as shown in Figure 6. The PC digitizes analog signal, receives digital data through digital I/O, counts pulses of the encoder output, and transmits data through RS 232 serial ports to the host computer. In the host computer, the dynamics and the graphics program run concurrently and share data through shared-memory interface.

447

An incremental encoder is used for detecting the operator's steering inputs. The spool displacement of hydraulically-actuated steering is proportional to the turning rate of the steering wheel, instead of its rotational displacement. Thus a high-resolution encoder is used to give high-resolution rotational displacement from which a turning rate is obtained. The encoder generates 1200 pulses per revolution. The encoder has two outputs, between which there exists a 90-degree phase shift, that can tell the direction of rotation and can also quadruple the resolution. Thus, the total resolution of the encoder is 4800 pulses per revolution.

Five rotational potentiometers are used to detect the gas pedal, brake, clutch, lift, and tilt levels. The output from these sensors is digitized through an 8-channel 12-bit A/D converter. The output from the gear switching box is a 7-bit digital signal and is decoded through a digital I/O port.

4. Applications to Telerobotic Simulation and Wheel Loader Operation

Now that the dynamics, control, and actuator models are ready and the interface with the Kraft mini-master and the operator's console have been defined, they must all be integrated with computer graphics. When the computer graphics give visual feedback to the operator, its fidelity becomes important. The graphics should include a general perspective view for depth perception, as well as proper rendering attributes such as color, surface texture, shading, and lighting. In addition, the graphics rendering speed should be fast enough not to show jerky motion.

For the interactive simulator, the Visualization of Dynamic Systems (VDS) [21] was modified [22,23]. This graphics software needs three data files to display a manipulator in motion. Two of them, created before the simulation, define the geometry and the rendering attributes. The geometry of each link can be generated by MOVIE.BYU [24] or other compatible software. The third file (or data stream), created during the simulation and updated at each frame, contains the latest position and orientation data computed from dynamics and control.

In Figure 7 three views from the on-board cameras on the tele-robot and one view from a camera on the space shuttle (lower right in Figure 7). The first view point is located right on the RMS end-effector and looks straightforward (upper right window in Figure 7), where the ball indicates the location of the telerobot's end-effector. The second one is located 2.5 ft behind, 1 ft left, and 1 ft up from the first one (upper left window in Figure 7). The third one is 3 ft away from the first one to the right and looks toward the first one (lower left window in Figure 7). The RMS is operated by a pair of joysticks and the tele-robot is operated by the force-reflective mini-master, as shown in Figure 3.

Figure 8 shows the overall view of the wheel loader simulation. When the simulation starts, the position analysis computes the position and orientation of each object based on the system properties and the initial conditions. The position and orientation is then sent to the computer graphics package, and combined with the prepared geometric model data, for animation. With a minor modification, the wheel loader simulation is extended to the simulation of an off-highway truck.

On a two-processor IRIS workstation, the dynamic and control analysis is executed on one processor and graphics rendering on the other. Based on the position information, the joint velocities are computed for use in control analysis, which computes the control driving torque. The joint torque is then sent to acceleration analysis. Next, the computed acceleration is integrated to update velocity and position. This update is then sent back to position analysis for the next time step. In Figure 3, the Kraft mini-master is shown in the foreground and the simulated manipulator in the background on the graphics screen. In Figure 9, the operator's view is displayed with two side mirrors' view, as seen from the off-highway truck's console in Figure 5.

5. Conclusion

The dynamics and control model of a manipulator has been teleoperated through a 6 d.o.f. mini-master. A high-speed operator-machine interactive simulation includes the manipulator's recursive dynamics, control, high-speed graphics, and interface with the mini-master and the operator's console. This technology developed for operator-in-the-loop simulation in space teleoperation has been applied to the Caterpillar's backhoe, wheel loader, and off-highway truck. On a SGI workstation, the simulation integrates computer modeling of kinematics and dynamics, real-time computation and visualization, and an interface with the operator through the operator's console. The console is interfaced with the workstation through an IBM-PC in which the operator's commands were digitized and sent through a RS 232 serial port. The simulation gave visual feedback adequate for the operator in the loop, with the camera's field of vision projected on a large screen in multiple view windows. This simulator created an

innovative engineering design environment by integrating computer software and hardware with the human operator's interactions.

[1] J. Vertut and P. Coiffet. Teleoperations and robotics: evolution and development. In Robot Technology. Prentice-Hall, 1986.

[2] T. B. Sheridan. Telerobotics. Automatica, 25(4):487-507, 1984.

[3] L. Stark. Telerobotics: Display, control, and communication problem. IEEE J. Robotics and Automation, RA-3(1):67-75, 1987.

[4] C. W. Wampler. Manipulator inversekinematic solutions based on vector formulations and damped least-squares methods. IEEE Trans. Systems, Man, and Cybernetics, pages 93-101, 1986.

[5] E. J. Haug. Computer Aided Kinematics and Dynamics of Mechanical Systems, Volume I: Basic Methods. Allyn Bacon, 1989.

[6] S. T. P. Chern, K. H. Yae, and T. C. Lin. An application of the constraint J acobian to teleoperation of a redundant manipulator. Int. J. Robotics Automation, 1993. to appear.

[7] D. N. Nenchev. Redundancy resolution through local optimization: A review. J. Robotic Systems, 6(6):769-798, 1989.

[8] D. S. Bae and E. J. Haug. A recursive formulation for constrained mechanical system dynamics: Part I -open loop systems, Part II-closed loop system. Mechanics of Structures and Machines, 15(3,4):359-382 and 481-506, 1987.

[9] D. S. Bae, R. S. Hwang, and E. J. Haug. A recursive formulation for real-time dynamic simulation of mechanical systems. ASME J. Mechanical Design, 113:158-166, 1991.

[10] F. F. Tsai and E. J. Haug. Real-time multibody system dynamic simulation Part I: A modified recursive formulation and topological analysis, Part II: A parallel algorithm and numerical results. Mechanics of Structures and Machines, 19:99-127 and 129-162, 1991.

[11] T. C. Lin and K. H. Yae. The effects of harmonic drive gears on robotic dynamics. In ASME 1991 Advances in Design Automation, volume DE-Vol. 32-2, pages 515-522, 1991.

[12] T. C. Lin and K. H. Yae. Recursive dynamic formulation of a manipulator driven by harmonic drives. Mechanics of Structures and Machines, 1993. to appear.

[13] T. Yoshikawa. Analysis and control of robot manipulators with redundancy. Robotics Research: The First International Symposium. The MIT Press. 1984.

[14] R. Featherstone. Robot Dynamics Algorithms. Kluwer Academic Publishers, 1987.

[16] J. Wittenburg. Dynamics of Systems of Rigid Bodies. B. G. Teubner Stuttgart, 1977.

[15] E. J. Haug and M. K. McCullough. A variational-vector calculus approach to machine dynamics. ASME J. Mechanisms, Transmissions, and Automation in Design, 108:25-30, 1986.

[17] H. Asada and J.-J. E. Slotine. Robot Analysis and Control.John Wiley Sons, Inc., 1986.

449

[18] C. R. Rao and S. K. Mitra. Generalized Inverse of Matrices and its Applications. John Wiley Sons, 1971.

[19] C. A. Klein and C. H. Huang. Review of pseudoinverse control for use with kinematically redundant manipulators. IEEE Trans. Systems, Man, and Cybernetics, SMC-13(3):245-250, 1983.

[20] B. Hannaford and R. Anderson. Experimental and simulation studies of hard contact in force reflecting teleoperation. In IEEE Conference on Robotics and Automation, 1988.

[21] M. W. Dubetz, J. G. Kuhl, and E. J. Haug. A network implementation of real-time dynamics simulation with interactive animated graphics. In Proceedings of the 12th ASME Design Automation Conference, pages 509-518, 1988.

[22] J. L. Chang and S. S. Kim. A low-cost real-time man-in-the-loop simulation for multibody systems. In Proceedings of the 12th ASME Design Automation Conference, pages 95-99, 1989.

[23] J. L. Chang, T. C. Lin, and K. H. Yae. Man-in-the-control-loop simulation for manipulators. In Proceedings of The 3rd Annual Conference on Aerospace Computational Control, pages 688-699, 1989.

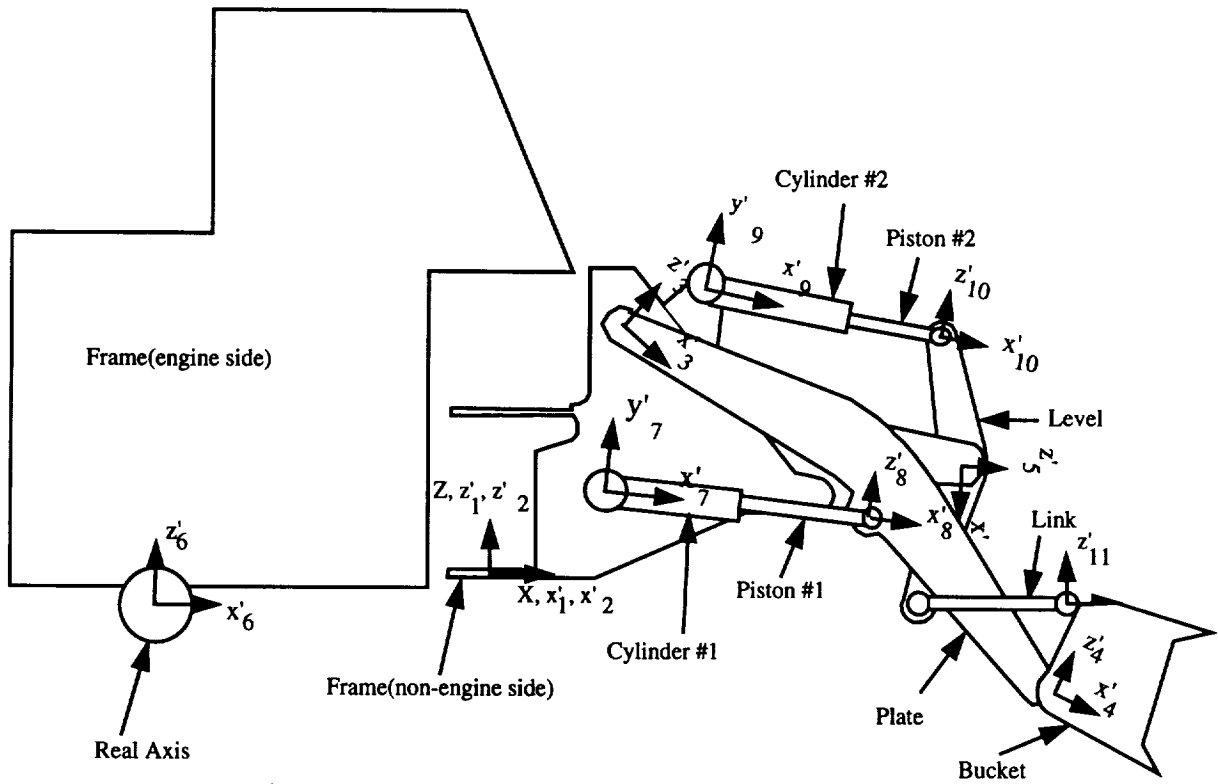[24] H. N. Christiansen. MOVIE.BYU Training Text, 1987.
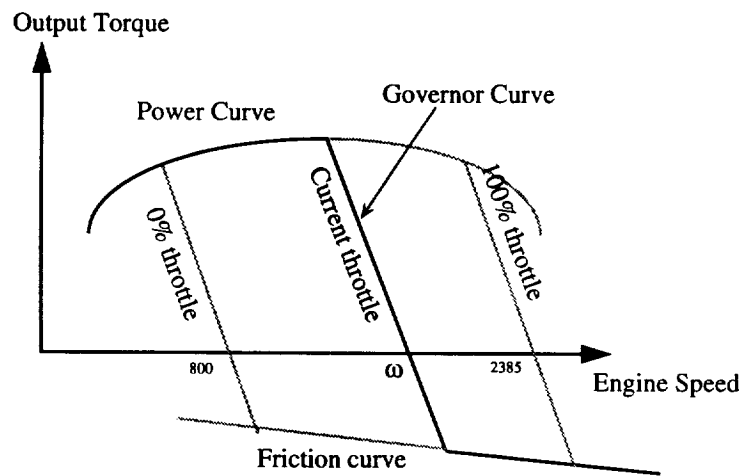
Figure 1: Linkages of Wheel Loader
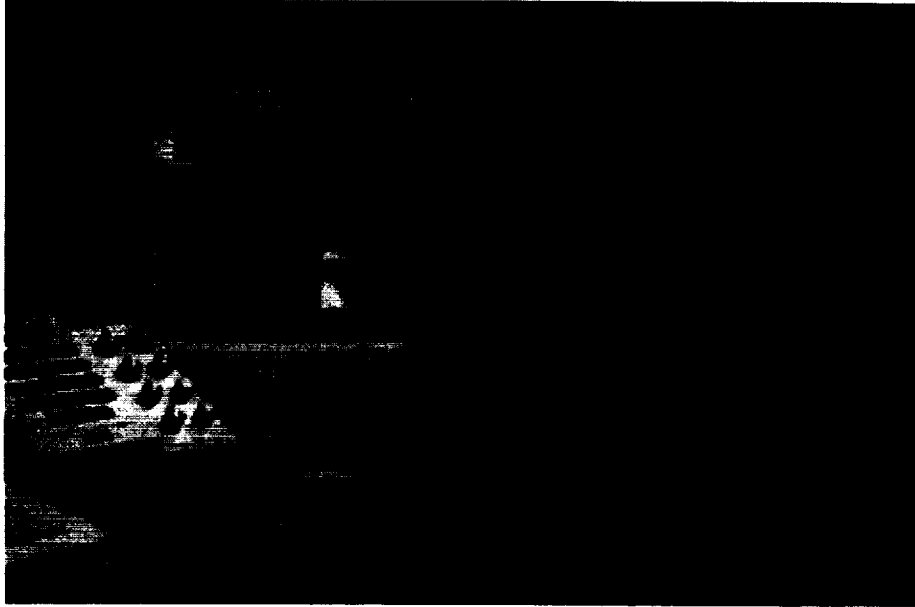

Figure 2: Engine torque output

Figure 3: A Kraft mini–master in the foreground and the simulated manipulator in the background on the graphics screen
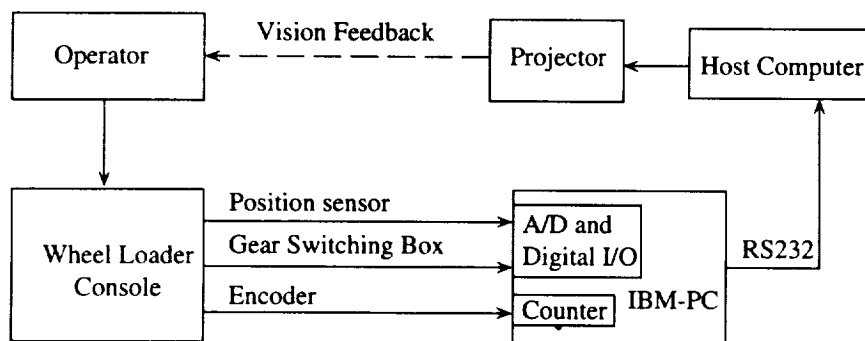

Figure 4: Simulation setup

Figure 5: The operator's console and the projection screen

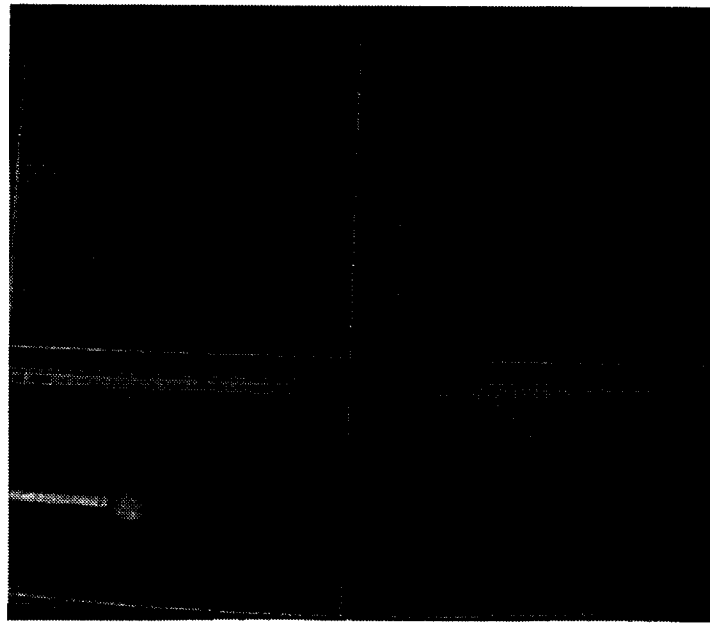Figure 6: Rotational potentiometers, optical encoders, and their interface wiring

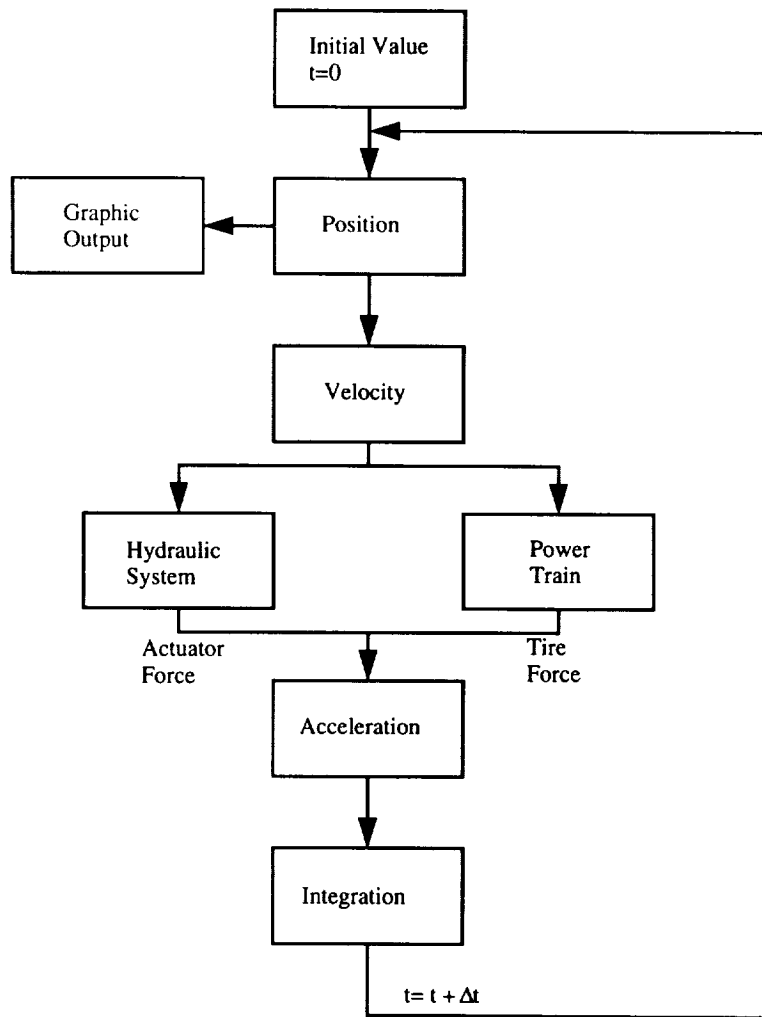Figure 7: Emulation of camera views on an IRIS graphics workstation: three on the RMS and one on the space shuttle
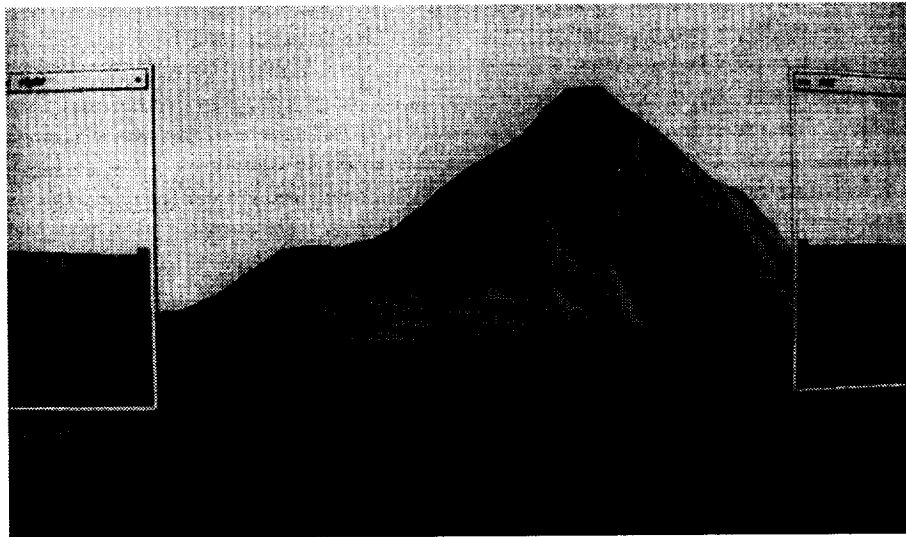

Figure 8: Computation flow

Figure 9: The operator's view displayed with two side mirrors' view

HIGH PERFORMANCE REAL-TIME FLIGHT SIMULATION AT NASA LANGLEY

Jeff I. Cleveland II
Project Engineer

National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23681-0001

ABSTRACT

In order to meet the stringent time-critical requirements for real-time man-in-the-loop flight simulation, computer processing operations must be deterministic and be completed in as short a time as possible. This includes simulation mathematical model computation and data input/output to the simulators. In 1986, in response to increased demands for flight simulation performance, personnel at NASA's Langley Research Center (LaRC), working with the contractor, developed extensions to a standard input/output system to provide for high bandwidth, low latency data acquisition and distribution. The Computer Automated Measurement and Control technology (IEEE standard 595) was extended to meet the performance requirements for real-time simulation. This technology extension increased the effective bandwidth by a factor of ten and increased the performance of modules necessary for simulator communication. This technology is being used by more than 80 leading technological developers in the United States, Canada, and Europe. Included among the commercial applications of this technology are nuclear process control, power grid analysis, process monitoring, real-time simulation, and radar data acquisition. Personnel at LaRC have completed the development of the use of supercomputers for simulation mathematical model computation to support real-time flight simulation. This includes the development of a real-time operating system and the development of specialized software and hardware for the CAMAC simulator network. This work, coupled with the use of an open systems software architecture, has advanced the state-of-the-art in real-time flight simulation. This paper describes the data acquisition technology innovation and experience with recent developments in this technology.

INTRODUCTION

NASA's Langley Research Center (LaRC) has used real-time flight simulation to support aerodynamic, space, and hardware research for over forty years. In the mid-1960s LaRC pioneered the first practical, real-time, digital, flight simulation system with Control Data Corporation (CDC) 6600 computers. In 1976, the 6600 computers were replaced with CDC CYBER 175 computers. In 1987, the analog-based simulation input/output system was replaced with a high performance, fiber-optic-based, digital network. The installation of supercomputers for simulation model computation was completed in 1992.

The digital data distribution and signal conversion system, referred to as the Advanced Real-Time Simulation System (ARTSS) is a state-of-the-art, high-speed, fiber-optic-based, ring network system. This system, using the Computer Automated Measurement and Control (CAMAC) technology, replaced two twenty year old analog-based systems. The ARTSS is described in detail in references [1] through [6].

An unpublished survey of flight simulation users at LaRC conducted in 1987 projected that computing power requirements would increase by a factor of eight in the coming decade. Although general growth was indicated, the pacing discipline was the design testing of high performance fighter aircraft. Factors influencing growth included: 1) active control of increased flexibility, 2) less static stability requiring more complex automatic attitude control and augmentation, 3) more complex avionics, 4) more sophisticated weapons systems, and 5) multiple aircraft interaction, the so called "n on m" problems.

Finding no alternatives to using large-scale general-purpose digital computers, LaRC issued a Request for Proposals in May, 1989 and subsequently awarded a contract to Convex Computer Corporation in December of that year. As a result of this action, two Convex supercomputers are used to support flight simulation. The