N93-29737

WALKING ROBOT: A DESIGN PROJECT FOR UNDERGRADUATE STUDENTS

S25-37
160601
P. 4

UNIVERSITY OF MARYLAND, MECHANICAL ENGINEERING

The objective of the University of Maryland walking robot project was to design, analyze, assemble, and test an intelligent, mobile, and terrain-adaptive system. The robot incorporates existing technologies in novel ways. The legs emulate the walking path of a human by an innovative modification of a crank-and-rocker mechanism. The body consists of two tripod frames connected by a turning mechanism. The two sets of three legs are mounted so as to allow the robot to walk with stability in its own footsteps. The computer uses a modular hardware design and distributed processing. Dual-port RAM is used to allow communication between a supervisory personal computer and seven microcontrollers. The microcontrollers provide low-level control for the motors and relieve the processing burden on the PC.

INTRODUCTION

Special-purpose robots are commonplace today in manufacturing and other controlled environments. Current technology is sufficient to build walking machines capable of traversing rugged terrain. However, few robots of this type have been built. The University of Maryland's walking machine, PredaTerp, is a six-legged robot designed to be able to walk over uneven ground, clear small obstacles, climb stairs, and autonomously navigate any desired path.

The robot's design evolved to be a novel application of existing technologies. The six legs were designed by combining well-understood mechanisms and optimized for performance, flexibility, and simplicity. The body design used two tripods for walking stability and ease of turning. The electrical hardware design employed modularity and distributed processing to drive the many motors. The software design used feedback to coordinate the system.

The robot is designed to be easily enhanced. Minor modifications would enable the machine to perform useful tasks with high precision and reliability. The walking machine may be easily adapted to hostile environments such as high radiation zones and alien terrain.

LEG DESIGN

It was desired to design a leg with an ovoid walking path to minimize the "slamming" effect caused by a robot's inertial forces during normal walking. This effect is highly pronounced in designs employing a circular kinematic path. The leg consists of three mechanisms. The ovoid path is generated by a modified crank-and-rocker mechanism; it is magnified by a pantograph mechanism and can be raised and lowered by a leg-lift mechanism.

The four-bar mechanism consists of a crank link, coupler link, rocker link, and fixed (ground) link. In this design, the traditional straight bar coupler was replaced with an oblique triangular link. The internal angles of the modified coupler can be varied to create an array of continuous, oblique circular paths at the disjointed vertex of the triangle. The constraints required the leg to have an ovoid path in order to prevent the inertial "slamming" effect during walking motion. In addition, the constraint on the number of motors required this motion to be carried out by one motor.

The four-bar, crank-and-rocker mechanism (Fig. 1) is defined by links AP (crank), BQ (rocker), ABC (coupler), and PQ (ground). The motor turns the crank through a worm gear combination. As the crank rotates, a pendulum path is created by the rocker link. The stride length of the path at point C, (Fig. 1) is 3.5 cm, with a stride height of 0.68 cm.

The desired stride length was 15 cm. In order to achieve this, the 7.5-cm path produced by the crank-and-rocker mechanism is amplified by a pantograph mechanism (the remaining four links of Fig. 1). The ovoid path at point C has been translated, inverted, and magnified by a factor of 2 at the foot (point H).

The leg-lift mechanism is capable of changing the leg height as well as the stride length. The leg-lift mechanism is the pinion gear and lifter gear-link attached to point F (Fig. 1). The lifter motor rotates the lifter gear, causing the pantograph mechanism to compress or expand. The leg was designed to extend and compress 7.5 cm from the position of the foot during normal walking. This results in a total lift range of 15 cm, sufficient to clear small obstacles and climb stairs.

The crank-and-rocker, pantograph, and leg-lift mechanisms are supported between two rectangular plates. These plates provide the ground attachments for the crank-and-rocker at points P and Q, and also for the lifter mechanism at points R and S. The plates also provide a convenient means for mounting the entire leg assembly to the robot body, and protect the leg links from external objects that could damage or bind the moving links during operation. The motor and gearbox combinations of the lifter and four-bar mechanisms are mounted outside the plates to avoid mechanical interference. Motors and gearboxes can be mounted on either of the two plates, depending on their orientation on the robot body. Three legs have a right-hand orientation and the remaining three have a left-hand orientation for this design.

Dynamic engineering analysis using DADS computer software provided insight into link forces, torques, displacements, velocities, and accelerations during normal walking as well as during

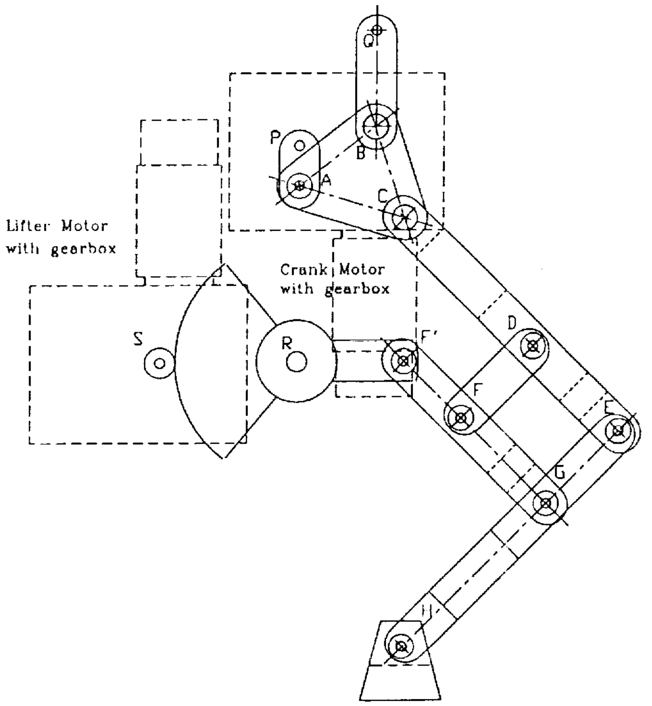

Fig. 1. Leg mechanism.

stair climbing. These data were used to design against stress failure and to select appropriate motors and bearings for the legs. This analysis also prompted an idea to connect a torsion spring to point Q on the rocker link. During the DADS analysis it was discovered that a torsion spring attached to the rocker would lower the maximum required torque by approximately 40%. This resulted in a smoother torque vs. time curve (smoother walking motion) and allowed the use of a smaller and lighter motor.

All the links for the leg mechanisms were machined by numerical control (NC). An NC machining center was programmed to cut the aluminum and steel leg links and modify the lifter gear. This highly precise machine cut complex shapes and drilled critical holes quickly while maintaining tolerances within 0.0002" (0.00051 cm). The aluminum support plates were punched by an NC turret punch. It was important to maintain tight tolerances for the hole positions, relative to one another, in the support plates because they provide the ground link for the lifter and four-bar mechanisms. The machine holds tolerances to within 0.001" (0.0025 cm).

BODY DESIGN

The main goal of the body design was to produce a frame capable of providing maximum flexibility and agility for the walking robot. The general design is of two concentric triangles, 180° out of phase, but able to rotate with respect to one another. The frame was designed to be inexpensive, lightweight, and to accommodate the electrical hardware and leg assemblies. It also needed to allow a minimum of 30° free rotation between body planes. It was decided that the foot paths must be 15 cm long, creating a stride of 30 cm. The front-to-back distance between adjacent foot centers on opposite frames is 15 cm, and the front-to-back distance between feet on the same frame is 45 cm. This causes the robot to walk in its own footsteps.

The main part of the frame was made of 1" (2.5 cm) steel electrical conduit (Fig. 2). Cold rolled 0.125" × 1" (0.3 cm × 2.5 cm) steel bar was bent into U shapes and welded between the two side pieces of each frame to hold them in place. The bars also serve as attachment points for the front and back leg assemblies. Supporting arcs were welded to the side pieces to hold them apart, and to provide rigidity and resistance to the moment of the side legs. An 18-gauge steel sheet was riveted to both sides of each frame. The sheet added rigidity to the frame, but its main purpose was to provide a place to install the hardware, and to improve the appearance of the machine.

The robot body employed a turning mechanism to rotate one body plane relative to the other. Plates of 3/16" (0.5 cm) steel were cut to fit between the support arcs of each frame to hold up the turning column. This was necessary because the turning column supports half the body weight at all times. Two tapered roller bearings were used in the turning mechanism. Four gussets were welded to the bottom of the turning shaft to add resistance to moment. An aluminum bearing hub was

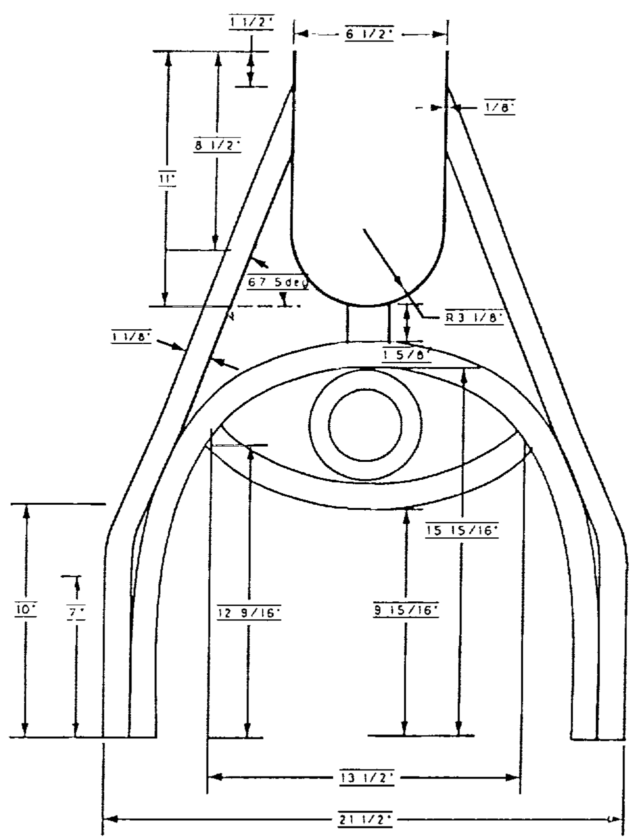

Fig. 2. Body frame.

constructed to separate the two bearings on the shaft. Bearings were press-fit into the hub. The topmost part of the turning shaft was threaded and a retaining nut was used to hold the frame assembly together and constrain the bearings. The motor torque was magnified through a worm gear combination. This was mounted rigidly to the upper body frame.

ELECTRICAL HARDWARE DESIGN

Although the mechanical hardware is quite flexible, the number of motors it requires leads to a quite formidable problem: the simultaneous monitoring and controlling of 13 motors. The electrical hardware performs this task by having seven identical microcontroller circuits, each controlling one leg or the turning motor and an IBM PC-compatible computer as the supervisory processor (Fig. 3). All the subsystems of the computer control system of the robot contain identical elements. This modularity simplifies construction and troubleshooting. Once a prototype module is perfected, it only need be replicated to complete the entire subsystem.

Each microcontroller circuit is designed around an 87C196KB, a 16-bit embedded controller with many on-chip peripherals, including 8 kbytes of EPROM, pulse-width modulation, and high-speed IO functions.

Communication between the microcontrollers and the PC is accomplished with dual-port static RAMs. Each microcontroller shares a 2-k block with the PC. Each microcontroller will control, at most, two motors at a time. These microcontrollers are supervised by a personal computer, which oversees and coordinates the individual processors. The supervisory processor provides long-term system control over the navigational functions, as well as periodically checking up on each leg as needs dictate.

The motor power circuitry is able to handle a very wide current range very efficiently. All the power transistors are generously heat-sinked to quickly dissipate heat and thus maintain a safe operating temperature range. Motor speed is controlled by pulse-width modulation, in which peak current remains constant and average voltage is adjusted. This is an efficient method of speed control, ideally suited to computer control. Power to the motor is switched on and off rapidly, the duty cycle determined by the processor.

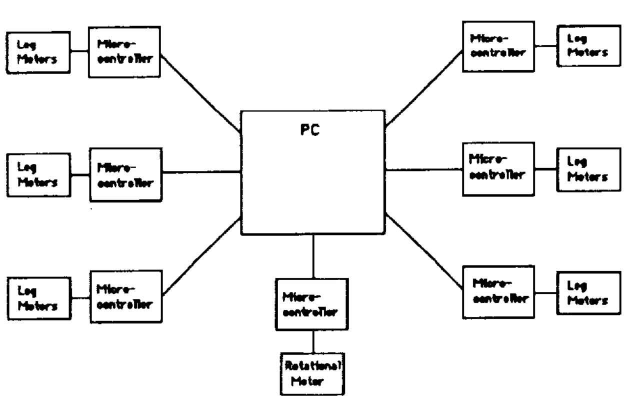

Fig. 3. Electrical hardware design.

Onboard batteries power the robot, with a switching voltage regulator controlling the voltage supplied to the computer. In order to maximize the robot's length of service per charge, the most efficient available regulator was used. The chosen switching regulator rapidly switches the unregulated power through a coil, efficiently providing a constant voltage.

All motors incorporate optical encoders that provide precise data concerning the position of the motor shaft. Simple circuits were used to decode phase data from the encoders into a form easily utilized by the microcontrollers. This provides the microcontroller with the direction and velocity of the motors.

The robot must protect itself from damage. In certain situations, if a motor does not turn off, the various mechanisms may be destroyed. Fail-safe switches were incorporated in the design to prevent such occurrences. If a motor reaches a certain point, the power driving the motor in that direction is shut off, and a signal is sent to the microcontroller. The microcontroller can still turn the motor in the opposite direction. Fuses are located throughout the robot to avoid electrically overstressing components and risking fires. Each motor has its own fuse. If a fuse should blow, an LED would provide a visual indication of the bad fuse.

SOFTWARE DESIGN

The primary design goal of PredaTerp's software was to implement an optimal solution to control the mechanical and electrical systems. The software engineering was split into three separate but interdependent tasks. The high-level code, written on the supervisory PC, coordinates the leg and turning motor actions to complete the events. The low-level task is that of driving the motors with the 80C196KB microcontrollers. The communications protocol provides the means of communication between the PC and the microcontrollers through the dual-port RAM.

The use of the PC allowed the utilization of established programming tools and techniques to write autonomous programs. The use of the PC also afforded a simple and flexible solution for tether design: a standard keyboard.

The PC is also able to assure that the commands passed to the legs and body are performed as desired. The PC does not necessarily oversee operations continually while in tether control mode, but the ability to do so is vital for autonomous operation. Rather than being required to repeatedly look at positional information, the PC is able to view this data only as the need arises. This ensures that the PC is kept free for decision making tasks, and only concerns itself with the microcontrollers if problems occur. Examples of such problems are leg timing during walking and turning and ensuring that a minimum of three legs are on the ground for stability. The PC also monitors the "environmental" sensors that return feedback from the robot's surroundings. Presently, these consist of embedded foot switches and a simple voice recognition circuit. The foot switches sense the presence of solid footing or the lack thereof.

The PC can generate commands autonomously or it can recieve them through the tether. The commands available are "Stop," "Reset," "Quit," "Update Display," "Set Velocity," "Autonomous Program," "Walk," "Set Height," and "Turn." The "Stop" subroutine executes a code that halts any actions being performed by the microcontrollers. "Reset" executes a code that sets the robot to a known initial state. "Quit" halts all program execution. "Update Display" rewrites the graphics screen to the monitor. "Set Velocity" specifies a walking or turning speed. The "Autonomous Program" command puts the robot into a state in which it is able to operate on its own in an unknown enviroment. "Walk" executes a code that causes the microcontrollers to engage the walking motors, and "Height" engages the lifter motors for the stair event. "Turn" specifies the angle and direction required of the turning motor. The user specifies which leg a specific command refers to via a sequential leg number protocol.

Upon initially receiving power, the microcontroller sits idle until it receives a "Reset" command. The "Reset" commands are "XReset," "YReset," and "TurnReset." Their purpose is to bring the mechanical systems to a known state, a "home" position, and thus gain control over the machine.

In addition to executing commands from the PC, the '196 is responsible for maintaining motor control within error limits deemed acceptable by the PC. To accomplish this, the '196s execute a proportional integral differential (PID) control algorithm, which is set to run on an interrupt to ensure execution at precise time intervals. The PID routine receives feedback from the encoders and determines the power necessary to keep the motor at the desired position and velocity. This value is passed to the pulse-width modulation hardware.

Commands and data are passed between the PC and the '196 microcontrollers by the dual-port RAM (DPRAM). The commands sent are "Walk," "Set Velocity," "Turn," "XReset," "YReset," "Height," "Stop," and "TurnReset." Data passed to execute these commands are velocity, direction, height, step number, and turn angle. The microcontrollers send a signal back to the PC after completing a given command. This signal causes an interrupt.

DISCUSSION AND CONCLUSION

The design and manufacture of a walking machine was completed by 40 students in 7 months. The responsibilities were divided into leg, body, hardware, and software tasks.

The leg design combined a modified crank-and-rocker mechanism with pantograph and leg-lift mechanisms. The six legs each operate with two degrees of freedom, providing great flexibility. Structural integrity was maintained through computer engineering analysis and numerical control machinery.

The body design provided an additional degree of freedom for the robot. This was achieved with a turning mechanism. This mechanism controls the relative position of the two body frames. The rigid tripod frames provide a means to mount the six legs.

The electrical hardware design employed distributed processing and modular components to control and power the walking machine. A supervisory personal computer accepts commands, oversees control, and runs autonomous programs. Microprocessors were used to directly control the 13 motors. Communication between the PC and microprocessors is performed with dual-port RAM.

The software design coordinates the robot's actions. A low-level code written to the microcontrollers controls the motor positions. A high-level code written to the PC processes programs and commands. A communications code breaks down PC commands into smaller microcontroller tasks and coordinates timing of data.

The machine is easily adapted to almost any terrain because of the design's flexibility. The leg design emulates a human stride, allowing a modified system to serve in functions hazardous to humans. The feedback control design allows the robot to be adapted to perform repeatable precision tasks. The PC-based master processor allows for easy expansion of hardware and software capabilities in the future.