SOME NOTES ON THE RHINO XR-1 AND MINIMOVER 5

Jorg R. Jemelka

Digital Computer Laboratory, Room 29G

1304 West Springfield Avenue

Urbana, Illinois 61820

During the summer of 1981, the microcomputer laboratory at the Urbana-Champaign campus of the University of Illinois acquired a Minimover-5 instructional robot. The laboratory, headed by Mike Faiman, and the artificial intelligence group, under Donald Michie, undertook a joint project integrating elements of an ongoing vision project, a suitable task domain, a problem solver, and the Minimover-5. At the same time, the first Rhino XR-ls went into production locally. Two early models were acquired for inclusion in the project, one on loan, the other purchased.

The Minimover-5 and the Rhino XR-l are articulated (anthropomorphic) manipulators. In both, a change in the angular displacement of a joint does not influence the orientation of the other joints in workspace coordinates, but does affect their relative orientations. Joint transformations are thus performed in Cartesian space, using two triangles as a geometrical model (see figure 1). The first is the right triangle formed by the shoulder-wrist extension-wrist height (ABC), the other is the shoulder-elbow-wrist triangle (ADC). The angular displacements are translated into motor steps and passed to an assembly-language driver. The driver attends to the details of issuing individual motor commands.

The project world consists of a table setting with plate, knife, fork, spoon, and cup. It is scanned, its components are identified, their relative positions and orientations are ascertained, and a determination of *correctness* is made. *AflX* is generated for an incorrect scene, which is then executed by the manipulators. The world model in which the manipulators operate is essentially two-dimensional. All of the objects have a Z coordinate of 0, except the cup, which has a significant z. To operate in this world, approach, retreat, and rotation take place at shoulder height; grasping and releasing occur at object height. This pick-and-place task, analogous to those performed by larger machines in industry, is a suitable test of the manipulators' capabilities. A typical move sequence would be:

1 Raise hand to shoulder height,
2 Extend hand over object XY,
3 Open hand and align with object,
4 Descend to object Z and close hand,
5 Raise object to shoulder height, 6 Extend hand to destination
XY,
7 Rotate to destination orientation,
8 Lower to object Z,
9 Open hand,
10 Raise hand to shoulder height
at destination XY, close hand,
11 Park at "home" position.

The apparent similarity of the two machines, when seen from the task definition level of abstraction, masks significant differences that influence implementation strategies.

Rhino XR-l. The Rhino XR-l is powered by servomotors with feedback from incremental encoders that count steps. All communication from the host computer to the Rhino's Intel 8748 control microprocessor is serial. Motor control is performed using the Bang- Bang approach, full voltage is applied to a motor until its step count register is decremented to zero. There are six encoder steps per motor revolution. Dynamic braking is provided, locking the motor at the encoder position attained when its register is fully decremented. The 8748 reserves a memory location for each motor counter. Since the sign bit acts as a direction flag, individual commands are limited to 127 steps forward or reverse. Fifteen commands are required to move a typical joint 180 degrees.

Although the presence of a dedicated control processor holds the promise of reducing host processing time, the considerable range of motion required of each joint by our project application necessitated the incorporation of an assembly-language intermediate driver. This program receives as parameters the motor code (ASCII characters A through F) and the number of steps to be driven. The program issues repeated commands of 100 steps, first reading the feedback registers to ensure that it contains 20 or fewer steps, lest the addition of the new step sequence affect the direction bit and reverse the motor . Remaining steps fewer than 100 are transmitted as a final command, after which the intermediate driver returns control to the coordinate transform program.

While developing the driver routines, we discovered that turning on more than three motors simultaneously frequently resulted in a lowering of the power supply voltage to under the amount required to sustain the onboard logic. This voltage drop would cause uncontrolled acceleration of all joints, requiring a system shut-down to prevent damage to the motors. A separate 5 V supply was added to power the logic. The current production series of Rhino XR-ls is provided with an extra 12 V power supply for the 8748 and encoder LEDs, which should eliminate the problem.

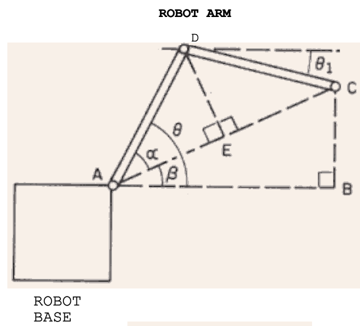
Moving the various Rhino XR-l joints through their full range of motion revealed a number of errors in the motor steps to angle conversions as stated in the manual. Tests were run using a construction level to align the elbow and hand horizontally and vertically. U sing this procedure, ::t: 2 steps deviation from the horizontal or vertical could be detected. The constants derived are:

Steps/Radian

Derived **In** Manual Hand 716.19785 750.57471 Elbow 725.74715 750.57471 Shoulder 500.38356 500.38356 Base 420.1694 359.69017

The rigidity of the chain driven shoulder joint accounts for the direct translation of motor steps through the various reductions to shoulder angle.

Our early series Rhino XR-1 came equipped with the "standard" hand. The finger assembly is operated by two concentric shafts. The outer shaft, on which the assembly is mounted, is used for rotation. The threaded, inner shaft opens and closes the fingers. Rotational accuracy for the standard hand is 15 degrees per encoder slot, or 3.81971 steps/radian. Closing or opening the fingers full throw requires approximately 7 seconds. Testing the finger assembly revealed that repeated open/



$$\beta = \arctan\left(\frac{BC}{AB}\right)$$

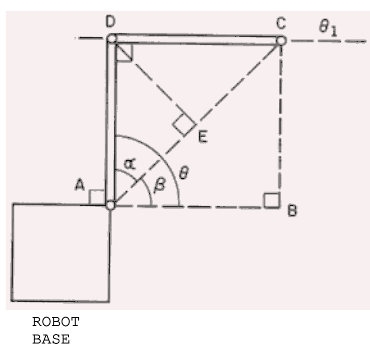
$$AC = \sqrt{AB^2 + BC^2}$$

$$DE = \sqrt{AD^2 - \frac{AC}{2}}$$

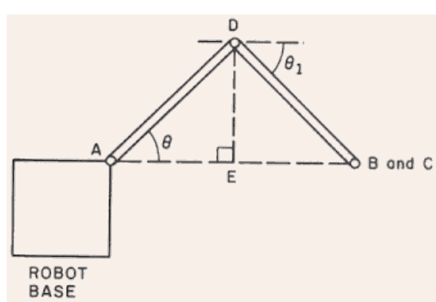
$$\alpha = \arctan\left(\frac{DE}{AC/2}\right)$$

$$\theta = \alpha + \beta$$

$$\theta_1 = \beta - \alpha$$



0: = 45°
.8 = 45°
8 = 90°
81= 0°



<l: .45°
,9 .0°
8. 45° 8!.-45°

Figure I: 8oll)e of the relationships between angles and joint placement in an articulated arm. All joint and manipulator placement calculations can be made using the simple trigonometric relationships defined in figure la. The lengths of arm segments AD and DC will remain constant for any specific arm. Figures Ib and lc show values for two specific cases.

close cycles dislodged the threaded actuating shaft from its mounting, a situation since remedied in later models. The open/close shaft is now dimpled to provide a secure seat for its set-screw. A further complication results from the fact that both the axis of rotation and the open/ close shaft are concentric. The hand tends to open or close further on rotation, since it rotates around the threaded, open/ close shaft. The optional "deluxe" hand is designed to decouple these axes.

Minimover-5. The Minimover-5 is powered by stepping motors without feedback. Control is maintained by the host computer through an assemblylanguage driver, full documentation for which is provided in the Minimover-5 manual. Communication is parallel, each byte containing a latching bit, a motor address, and a phase pattern. Eight successive phase patterns, each a 4-bit binary number, rotate the motor shaft 3.75 degrees. The problems encountered interfacing the Minimover-5 to our Northstar Horizon were minor . The early manual had misprints which incorrectly labeled the connector pins (since remedied). A timing problem with the latching pulse was cured by sending address and data out with the latch bit set to logical l , sending it out again with the latch bit 0, and then setting the latch pulse and sending the address and data a third time. The only change from the interfacing procedure as outlined in the manual is that the initial transmission is sent with the latch bit high.

The Minimover-5 driving software accepts 16-bit integers as commands. This allows straightforward command sequences to move all joints their full range. The Minimover-5 driver algorithm also simulates speed control, ensuring that all motors start and stop simultaneously. Implementation consists mainly of translating the Zilog mnemonics in the manual to Intel format.

High-Ievel Programming. The high level coordinate transform program to calculate the step sequence required in moving the various objects was written in BASIC. The similarity of the two machines simplified the programming task considerably. Aside from dimensional and motor constants, only the hand geometrics required different implementation strategies. The programming task was further simplified by the assumption, allowable in this case, that the hands always point straight down.

The Minimover-5 is designed so that its *home* position (shoulder fully back, elbow at rest on shoulder, hand normal to table surface), leaves its fingertips 35 mm ahead of and 10 mm below the shoulder joint. This provides values suitable for the calculation of initial and subsequent shoulder and elbow angles as well as providing fixed points for the end-of-travel of all joints. The Rhino XR-l 's similar home position leaves the fingertips slightly above and imperceptibly ahead of the shoulder joint. At- tempts to base calculations on these barely measurable initial values resulted in final position errors that varied with the joint extension. Consequently the defined home position was moved so that the fingertips were centered below and significantly ahead of the shoulder joint. This served to eliminate joint calculation error, with the drawback of leaving the home position in free space. The nearest limit switch cam mounting points are intermediate to this position and are not used as home position indicators, although they could, with additional software, be used to guide the arm to its parked position.

During the development of the high- level software, minor errors in the z- axis were compensated for by changing the fingertip to table height constants of both machines from the measured values. In this final development phase, collision with the table surface occurred occasionally, resulting in a loss of positional information, since motor steps, rather than joint angles, are controlled. The Minimover-5lost its position more easily because of its weaker stepping motor drive, while the Rhino XR-l, because of its rigid shoulder chain, suffered a positional error mostly in the elbow joint.

Reaching objects close to the base of the Rhino XR-l often resulted in unwanted collisions, since the shoulder and elbow joints are not coordinated to reach their destinations at the same time. In cases where the shoulder joints have few steps and the elbow many, the shoulder, reaching its goal first, provides a pivot for the elbow, low enough to cause the elbow to collide with the table while it is in the process of finishing its move. This could be compensated for by segmenting the moves into smaller subgoals to be attained en route to the final destination. The Rhino XR-l has sufficient reach, allowing us to keep its work space 100 mm out from the base without suffering a real penalty.

Testing. Both machines were given a move sequence to execute repeatedly under program control. The first test of twenty-five runs yielded timing and repeatability information. The overall cycle time to move both machines from home to 200,200 mm XY and 200,-200 mm XY was 1 minute 43.14 seconds. The Rhino XR-l required 1 minute 05.18 seconds, the Minimover-5 34.25 seconds, keeping in mind the 7-second hand open/close time. At cycle 23, the only error encountered in either machine was a 45 degree wrist rotational error, a result of round-off errors. A second test of 50 cycles resulted in the Rhino XR-l ' s slipping its shoulder drive belt on cycle 33 while parking. This overshoot error was in the direction of the shoulder joint, caused by the loss of a lOO-step command due to the absence of a serial communications protocol. Because of the shoulder's rigidity, this did not affect its performance on subsequent cycles. No errors were encountered with the Minimover-5 in either test.

With the completion of these tests, the hardware and software is fully operational. Project members anticipate further refinements with the development of C and Pascal software, the implementation of speed control, and a flexible driving routine for the Rhino XR-l's 8748 microcomputer. D

REFERENCES

Hill, J .W **."Introducing Minimover-5,"** ***Robotics Age,*** **Summer 1980, pages 18 to 27.**

Sandhu, H. "The Rhino XR-l: A **Hands-on Introduction to Robotics,"** ***Robotics Age,*** **March/April1982, pages 10to 18.**