A Sensor Skin System for Motion Control of Robot Arm Manipulators

Edward Cheung and Vladimir Lumelsky

Report No. 8915

December 1989

A Sensor Skin System for Motion Control of Robot Arm Manipulators*

Edward Cheung and Vladimir Lumelsky
Yale University, Department of Electrical Engineering
New Haven, Connecticut 06520

Abstract

This work addresses the hardware implementation issues of sensor-based motion planning system for a robot arm manipulator operating among unknown obstacles of arbitrary shape. In order to realize on-line planning algorithms while protecting the whole arm body from potential collisions with obstacles, the system includes infrared based proximity sensitive skin covering the arm body, computer hardware for signal processing and motion planning, and an interface between the planning and arm control systems. These components are described in detail, and their characteristics are discussed.

1. Introduction

Current research in robotic motion planning encompasses two major trends. In one approach complete information about the robot and its environment is assumed. A priori knowledge about the obstacles in the environment is represented by an algebraic description, such as a polyhedral representation.

Another approach, considered in this paper, assumes incomplete information about the environment. Such a situation takes place, for example, when the robot has no á priori knowledge about the environment, and is equipped with sensors that notify it of impending collision, or proximity to an obstacle. Realization of this approach in a robot system requires integration of four basic components: 1) physical sensing ability for detecting the presence of obstacles in the environment combined with adequate real-time signal processing and incorporated into an adequate system architecture, 2) local algorithms for guiding the robot in the vicinity of an obstacle, 3) global algorithms for controlling the general motion so as to avoid infinite cycles and guarantee convergence. This paper addresses the implementation issues that appear in realizing the first component. Issues related to the components (2) and (3)have been discussed elsewhere [1,2] and are outside of our current topic.

Specifically, this paper presents the system architecture and design of a prototype system

* Supported in part by the National Science Foundation Grants DMC-8712357 and IRC-8805943.

for sensor-based motion planning of a three-dimensional (3d) robot arm manipulator operating in workspace with unknown obstacles. Uses for robot arms equipped with such capabilities include applications for space exploration, work in hazardous, hostile or unhealthy environments, and unstructured factory work cells.

Assuming that, in general, every point of the arm body is subject to potential collisions, one approach to the sensing problem is to cover the surface of the whole arm with an array of proximity sensors. Obstacles appearing from any direction can then be detected, and appropriate action taken to avoid them. Such a sensor system based on the infrared light sensitive skin, together with sensor data interpretation algorithms, has been developed at the Yale University Robotics Laboratory [1]. The prototype system is based on an articulated industrial arm manipulator General Electric Model P5, Figure 1. Only the arm major linkage that includes the first three links and joints is covered by the motion planning system. In addition to the sensor hardware/software, the assembled system includes several microprocessors in a distributed processor arrangement, an IBM-AT computer as the user interface, and a MicroVax workstation as the graphical display device and data logger. Various data handling subroutines implemented in a parallel processing language operate asynchronously from each other to process the data from the sensors. The overall system block diagram is shown in Figure 2.

Ideally, the sensitive skin should include the ability to accurately detect the distance to any obstacle, regardless of surface texture, color and material. It should have a high noise immunity in order to prevent false alarms, and a fast response time to allow the fastest possible motion of the arm or objects in its environment. The entire arm should be covered so that obstacles approaching from all directions can be detected. The density of sensors along the arm should be sufficiently high to provide unambiguous data about the location of obstacles in the arm's path. To be resistant against wear and tear, wires that run across the joints of the arm should tolerate constant flexing. Although it is difficult to build a sensitive skin having all these properties, one can come close enough by careful design of the system.

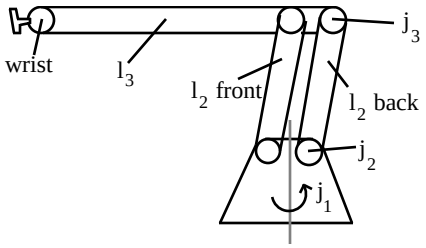


Figure 1. Sketch of the robot arm; j_1, j_2, j_3 are arm joints; l_2, l_3 - arm links; link l_1 is of zero length.

The sensitive skin described in detail below uses amplitude modulated infrared light as the sensing medium. The amount of reflected light is used for the proximity indication. More complicated methods such as time-of-flight and triangulation were considered too complex for integration into the sensitive skin. A more complete comparison of the various choices of sensor types can be found in [2]. In the current prototype system, an array of about 500 proximity sensors is integrated onto a flexible circuit board which forms the skin surface and is wrapped around the body of the arm.

To eliminate the possibility for interference between links, the frequencies of the light transmitted by the sensors on link l_3 and link l_2 are kept at a constant 2:1 ratio. Using the demodulation system in the sensor processor circuit, the light transmitted by one link's sensors is rejected by the receivers of the other link, allowing parallel sensor operation. In the current implementation, the sensor polling rate is such that the entire skin is polled once every one sixteenth of a second.

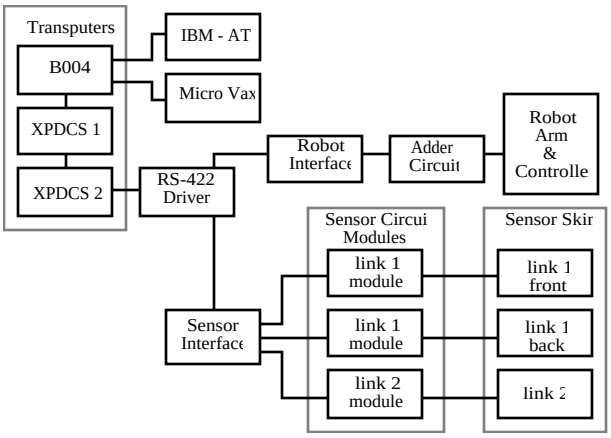


Figure 2. Overall system block diagram

The overall system shown in Figure 2 can roughly be divided into three major sections: computer hardware, robot arm control system, and the sensor system. Below, the computer hardware is discussed in Section 2, the robot arm control system in Section 3, and the sensor system - in Section 4.

2. Computer Hardware

The main source of on-line computational power comes from three transputer boards, labeled XPDCS 1, XPDCS 2, and B004 in Figure 2. In addition to the transputers, an IBM-

AT computer is used as the user interface, and a MicroVax workstation is used for real-time monitoring, documenting, and the graphical display.

The transputer board labeled "B004" is a product of the INMOS Corporation, manufacturer of the transputer chips. It contains 2 megabytes of computer memory and a T400 transputer rated at about 100 kiloflops. The development environment runs on this transputer board and communicates with the IBM-AT and the MicroVax. The other two transputer boards are a product of a Yale Robotics Laboratory affiliate [3]. They each contain 128 kbytes of computer memory and a T818 transputer, rated at about 1.2 megaflops. Each transputer can communicate with four high-speed direct memory access (DMA) serial links that allow the interconnection of transputers in different configurations. The high speed links are also connected to "link adaptors", also manufactured by INMOS, which convert the serial links into eight bit data buses. Data to and from the sensor and robot control systems are handled by these link adaptors.

The board XPDCS 2 performs the low level sensor and robot interface. Three main subroutines run asynchronously on this board. They handle the sensor polling, robot command refreshing, and information exchange to XPDCS 1 respectively. This latter board asynchronously runs the subroutines that filter the raw sensor data and the sensor interpretation (step planning) algorithm. The B004 does the high level motion planning in addition to interface to the development system (IBM-AT) and graphical display station (MicroVax).

The development environment, called Transputer Development System (TDS), is also by INMOS. It includes a text editor, compiler, debugger, as well as other utilities related to managing a network of parallel processors [4]. Programs that run on the transputers are written in OCCAM, a parallel processing language [5].

The MicroVax is connected via a serial connection to the B004 board, and charts the progress of experiments on its display console. A stand-alone software package written in C language, takes data from the serial port of the MicroVax and displays the current position of the robot, as well as the history of the experiment. This path and its various projections can be viewed from any direction using the utilities available in the software package. The display program also allows hard copy documentation of the experiments.

3. Robot interface

3.1 Interface architecture

The original robot controller in the GE P5 robot has no provisions for host computer interface, and programming the arm for a task can only be accomplished by point to point "teaching" of the manipulator using the teach pendant. To implement the sensor-based motion system, an interface between the host computer that houses the planning software and the robot controller has been built.

The connection between the host computer and the robot arm controller is via the 16-bit counters that keep track of the joint angle values of the arm. Signals that increment or decrement these counters are supplied by the incremental encoders mounted on the motor shafts. By inserting a binary adder in the feedback loop of each robot joint, a position command can then be given to the robot arm controller by an outside computer, see Figure 3.

Neglecting any time delay introduced by the binary adder A_2, Figure 3, stability and transient response of the altered system should be roughly equal to the original system. The total propagation time delay of the introduced electronics is less than 0.05 μsec. This eliminates the need for tuning of the control loop parameters such as the derivative and proportional gains.

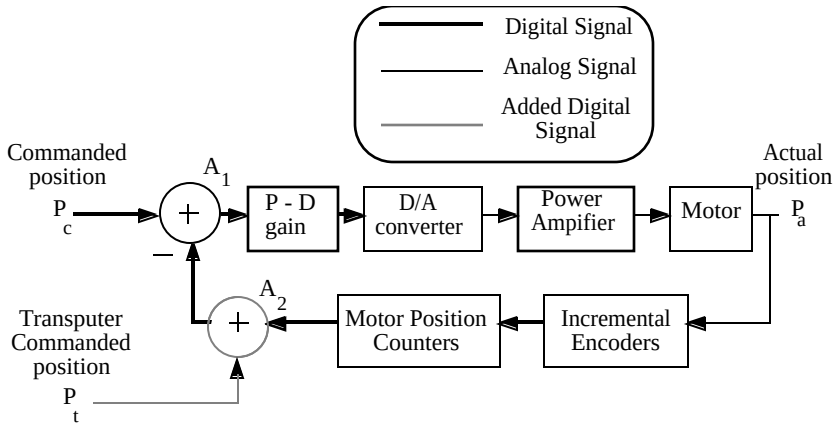


Figure 3. Low level motor control loop of each joint. Solid lines relate to the original controller, dotted lines indicate added circuits.

Apart from the above advantage of this particular interface scheme, all of the original hardware in the robot arm controller can be continued to be used, which simplifies significantly the interface. Its other advantage is "plug in" implementation, so that the controller could be restored to its original state in a matter of minutes. Furthermore, the added hardware could be made imperceptible using one toggle switch, allowing the controller to be used as if the new electronics were not installed. These features are especially useful at the development stage - e.g. for isolating the sources of malfunctions.

Since only the first three joints (Θ_1, Θ_2, Θ_3) are controlled by the motion planning algorithm, only these control loops are affected by the adder interface. The other arm joints

can be controlled by the regular means, such as the teach pendant interface. Note that as far as the transputers are concerned the position commands are given in an open loop fashion. No feedback of an actual robot position is returned to the transputer. It is assumed that the original robot control loop will position the joint to the commanded position after some settling time, and that motion velocities, $d\Theta_i/dt$ are relatively low. The adder interface thus effectively transforms the robot into an open loop positioning device. Joint velocities can then be commanded by writing a suitable software driver, which, as our preliminary 2d experiments showed [2], are sufficient for our tasks.

To use the interface, the robot is first placed by the robot controller in some known position P_c, see Figure 3. When a command P_t is given via the interface, it will cause the CPU to servo the arm to stabilize the control loop, causing the arm to move to the position P_a, where $P_a = P_c - P_t$. If P_c is kept fixed at the origin, $\Theta_1 = \Theta_2 = \Theta_3 = 0$, the arm can be given position commands so that $P_a = - P_t$.

3.2 Interface hardware

The purpose of the robot interface is to convert signals from the transputer XPDCS 2 into commands for the Robot Adder Circuit. The overall interface block diagram is shown in Figure 4. Commands from the transputers are communicated over twisted wire pairs connected to two link adaptors; these are eight bit bidirectional interface chips by INMOS Corp. One byte (eight bits) received from the transputer is interpreted as the data byte, and the other as the command byte. A sixteen bit Data Bus is generated by demultiplexing the data byte from the link adaptor. Each bit of the control byte, when asserted, triggers an appropriate action in the interface. The functions of each bit are listed in Table 1, and a bit by bit description of the functions of the command byte is given below.

Table 1. Bit functions in the command byte
(bit 0 is the least significant bit)

Bit	Command
0	Link adaptor send
1	Load high byte of Position Data Bus
2	Write teach pendant interface
3	Θ_1 position command
4	Θ_2 position command
5	Θ_3 position command
6	Emergency stop override
7	Not used (spare)

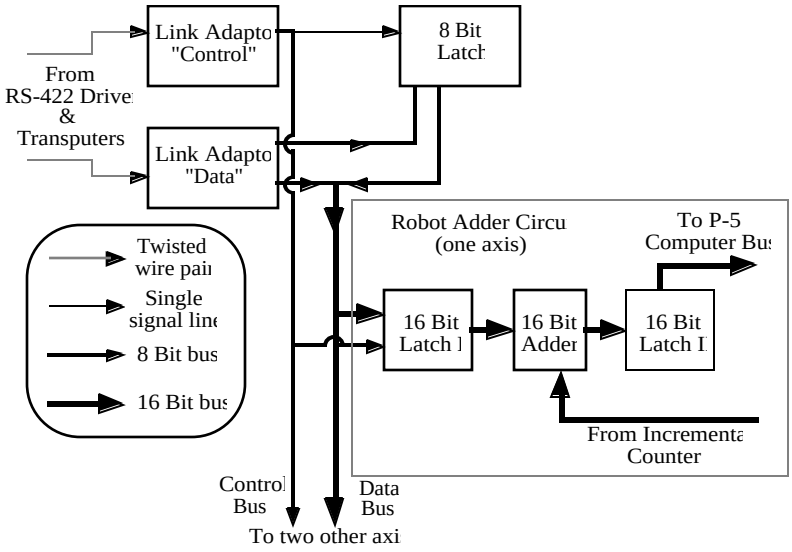


Figure 4. Overall robot interface block diagram

When asserted, bit 0 of the command byte causes the link adaptors to send the sixteen bits present at their collective input lines "Iin" to the transputer, see Figure 5. This is done by asserting the IValid input on IC3 and IC4. Although these lines are not used in the system, the input data bus was originally wired to implement sending data from the interface to the transputers.

The next bit, bit 1, of the command byte is used to demultiplex the data byte into the sixteen bit Position Data Bus of the robot interface. When this bit is a "1", the data byte from the link adaptor is loaded into the octal flip-flop IC7 [7]. The output of the octal flip-flop forms the most significant byte of the Position Data Bus. The least significant byte of the Position Data Bus is formed by the data byte itself. Data from the Position Data Bus can be loaded into the teach pendant interface by asserting bit 2 of the interface. This mode of operation is used to control the remaining arm joints if needed.

Bits 3 through 5 cause the data on the Position Data Bus to be loaded into the adders pertaining to joints Θ_1 through Θ_3 respectively, and causes the writing of a position command. The decimal number represented by the data on the Position Data Bus is the two's complement representation of an integer value proportional to the commanded joint angle in degrees. The proportionality constant depends on the encoder pulses per motor revolution and the gearing

between the motor shaft and the joint.

The next bit, bit 6, bypasses the automatic emergency stop circuit. When this bit is a "1", the so-called "watch dog circuit" is disabled. This system will be explained in more detail below.

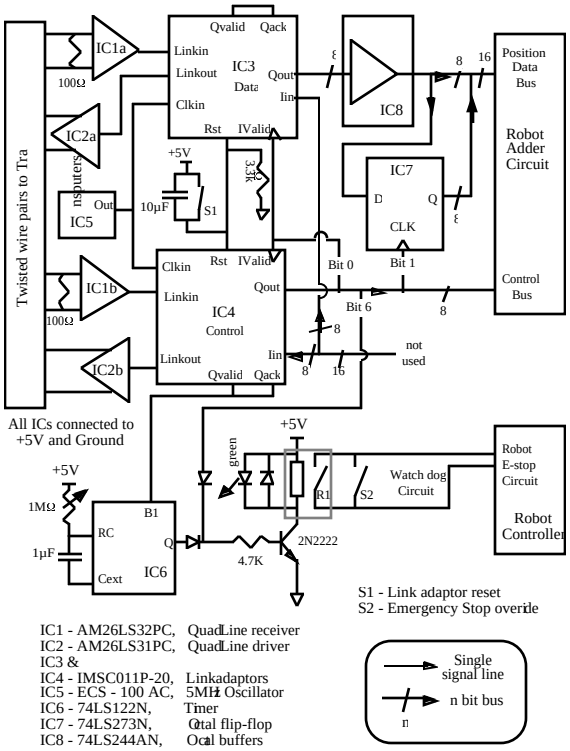


Figure 5. Robot interface circuit diagram.

The circuit that performs the robot interface is shown in Figure 5. The circuit is connected to the transputers by a twisted wire pair to increase the noise immunity. The main chips in the interface circuit, link adaptors IC3 and IC4, are connected to IC1 & IC2, which convert the signals from RS422 protocol to single ended TTL level signals. The output of IC3, which forms the data byte, is buffered by IC8, and forms the least significant byte of the Position Data Bus. It is also connected to the input of IC7, which latches the most significant byte of the Position Data Bus when bit 1 of the command byte is asserted. The output of IC4 forms the 8 bit command bus of the interface. Each bit of the command byte controls a particular

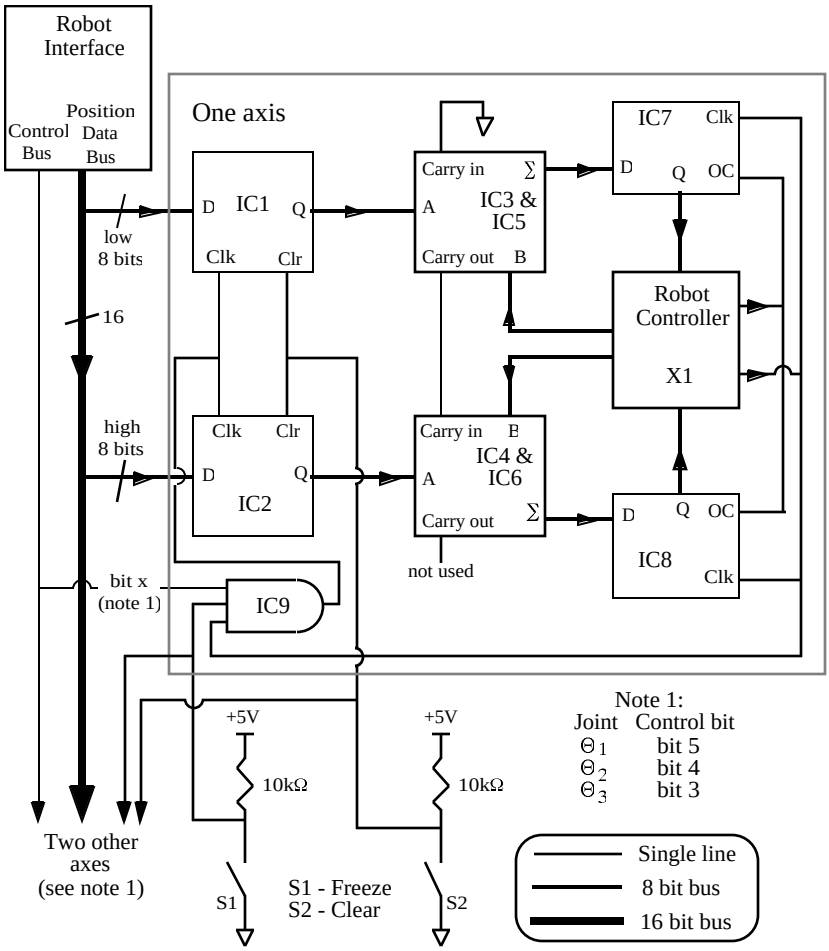
function as explained above.

Every 15 msec, the transputer XPDCS 2 writes a position command to the interface; thus all three joints receive a new position command at that interval. At the arrival of a new byte, the link adaptors assert their respective "QValid" signal. Using a suitably designed circuit, this signal can be utilized for the purpose of monitoring the status of the transputer. This "watch dog" circuit is formed by IC6, a retriggerable multivibrator. Its "Q" output stays high if pulses arrive at its "B1" input that are spaced sufficiently close in time. This minimum spacing is set by the variable resistor connected to its "RC" input. If the "Q" output of IC6 changes state to a low voltage level, relay R1 will interrupt current flow in the robot controller's emergency stop circuit, which causes the servos to switch off, and the brakes of the robot arm to be energized. The arm will stay in this disabled state until the emergency stop circuit is again closed, and a front panel reset switch is pushed. This feature can be bypassed in software by asserting bit 6 of the command byte. In this case, the relay will remain energized even if the transputer connected to the robot interface crashes. In series with R1 and S2 is the so-called "chicken switch" (not shown in the circuit diagram), which can be held in the hand of the robot operator. Actuating this switch interrupts current flow in the emergency stop circuit, and can be used for an emergency stop in the case of a malfunction.

Figure 6 shows the circuit diagram of the binary adder part of the robot interface. It converts commands from the Robot Interface Circuit to data which it inserts into the low level robot feedback loops. The procedure used to "break into" the robot controller began by removing the chips that contain the registers which keeps track of the joint values of the arm. The chips are on the circuit board inside the motor controller called the Counter Board. Cables were installed in place of the chips, which jumpered all the necessary signals from the Counter Board to the Adder Interface Circuit board. The original registers that were removed from the Counter Board were then placed on the Adder Interface board, with the binary adders inserted in series with the data path. The binary adders used were 4 bit full adders, thus four were used per joint (IC3 through IC6 on Figure 6). The registers that were removed from the Counter Board are IC7 through IC8 for the joint shown.

The following example illustrates how data is loaded into the binary adders, and thus given to the robot controller. The sequence shows the data (in Hex) on the data and command bus of the robot interface during a load of the number 325(Hex) into the adder for joint Θ_1. This corresponds to a value of about 5^o for Θ_1.

Data bus	Command bus	Operation
03	02	Load "03" into high byte register.
25	10	Load Position Data Bus into Θ_1 adder.



IC1 & IC2 - 74LS273, Octal flip-flop
IC3 thru IC6 - 74LS283, 4 bit adder
IC7 & IC8 - 74LS374, Octal flip-flop
IC9 - 74LS11, Three input AND gate

X1 - 74LS374 Sockets on Counter Board in Robot Controller

IC3 & IC5, IC4 & IC6 cascaded to form an 8 bit adder

Figure 6. Adder circuit diagram.

The switches S1 and S2 control overall operation of the interface. The data path from the Position Data Bus can be broken by closing S1, which causes the robot arm to halt at its current position. In addition, the adders can be cleared by closing S2, which functionally restores the robot controller to its original state, without the adders.

4.Sensor skin and sensor interface

4.1 Sensor interface

The purpose of the sensor interface circuit is to allow computer access to the sensor. Its two major components are the analog to digital converter, and the one-shots that control the addressing of a particular sensor. The overall sensor interface block diagram is shown in Figure 7, and the circuit that implements the sensor interface is shown in Figures 8 and 9. Points labeled "A", "B", and "C" on Figure 8 are connected to the corresponding points in Figure 9.

Communication to the sensor system from XPDCS 2 is handled by one link adaptor, which is connected with twisted wire pairs for noise immunity, Figure 8. The output byte of the link adaptor at "Qout" is used as the command byte. Similar to the robot interface's command byte, asserting a particular bit of the command byte triggers an action in the sensor interface, which in turn can cause an action in the sensor circuit module. The function of each bit of the command byte is shown in Table 2.

Table 2. Bit functions of the command byte
(bit 0 is the least significant bit)

Bit	Function
0	Link adaptor send
1	Analog channel select
2	Analog channel select
3	Analog channel select
4	Analog to digital converter trigger
5	Increment l_3 sensor
6	Increment l_2 back sensor
7	Increment l_2 front sensor

The frequency of the transmitted signal by the sensor skin is generated on the sensor interface circuit board by op-amp IC5a. This signal is distributed to the three sensor processing circuit modules, which are mounted on the surface of the arm, underneath the sensitive skin. This signal is also distributed to other parts of the sensor interface circuit.

The analog to digital converter (ADC) converts the analog signal at its input "In" to a digital number available at the "Dout" output, triggered by a high to low transition on its "Trig" input. During the conversion, which takes about 5 μsec, the "Busy" output remains low. This signal

controls the sample and hold (IC8), and the "IValid" signal on IC3. This latter data path allows IC9 to trigger IC3 directly as soon as the conversion is finished, and eliminates the need for the transputer to constantly poll the interface. Note that IC9 is a 12 bit ADC, the eight least significant bits of the digitized data are sent as soon as the conversion is finished. The most significant four bits are multiplexed on the eight bit data bus via a buffer internal to IC9. This data is placed on the output data bus "Dout" when the "Trig" input is high.

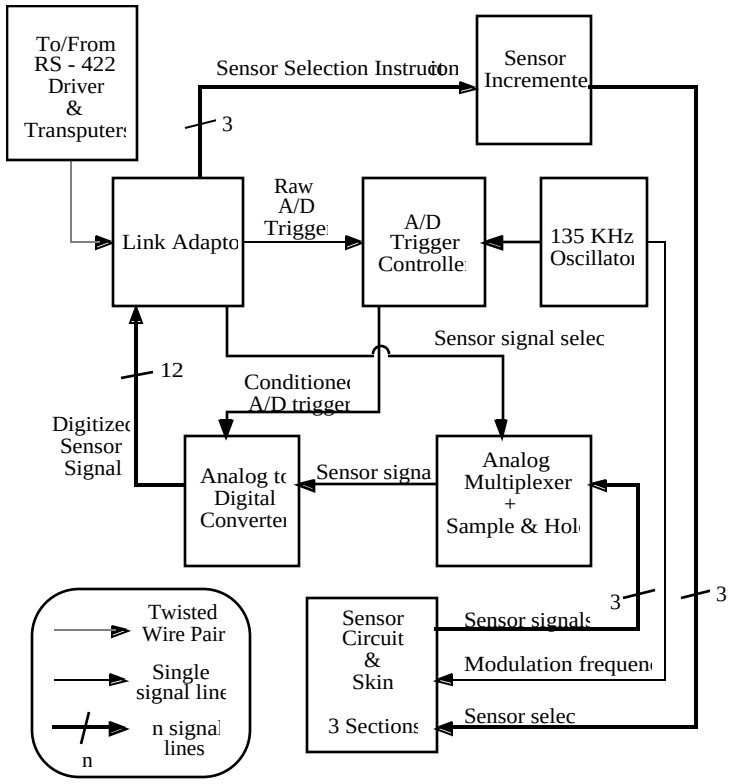
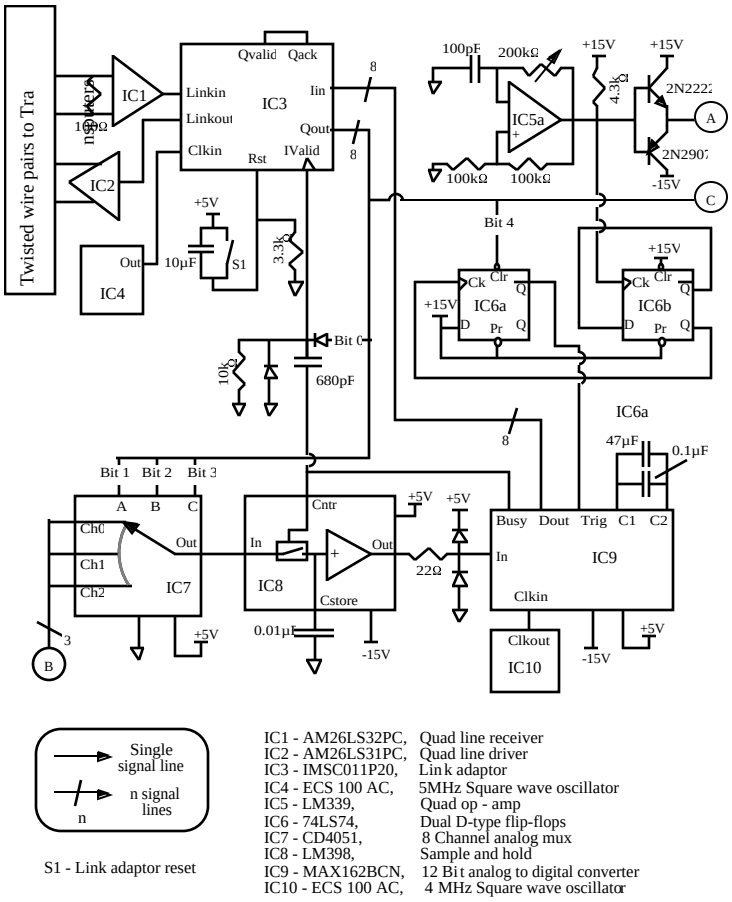


Figure 7. Sensor interface block diagram



Single signal line

n signal lines

S1 - Link adaptor reset

IC1 - AM26LS32PC, Quad line receiver
IC2 - AM26LS31PC, Quad line driver
IC3 - IMSC011P20, Link adaptor
IC4 - ECS 100 AC, 5MHz Square wave oscillator
IC5 - LM339, Quad op - amp
IC6 - 74LS74, Dual D-type flip-flops
IC7 - CD4051, 8 Channel analog mux
IC8 - LM398, Sample and hold
IC9 - MAX162BCN, 12 Bit analog to digital converter
IC10 - ECS 100 AC, 4 MHz Square wave oscillator

Figure 8. Sensor interface circuit, part 1. Points A, B, C refer to the corresponding points on Figure 9.

The analog sensor signal that is digitized by the ADC is switched by the analog multiplexer IC7 [8]. Bits 1 through 3 of the command byte select which output of the three sensor circuit modules is digitized. The binary number represented by the previously mentioned bits corresponds to the number of the channel that is selected, with bit 1 representing the least significant bit of the channel number. For example, if bit 1=0, bit 2=1, and bit 3=0, then channel "Ch2" is selected.

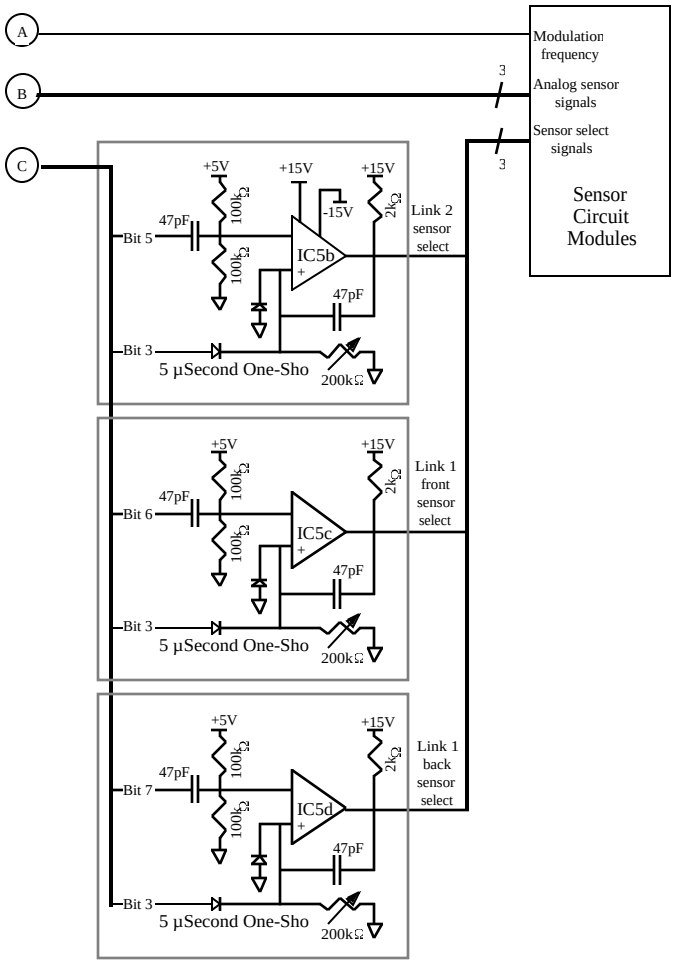


Figure 9. Sensor interface circuit, part 2

The triggering of the ADC is initiated by bit 4 of the command byte. In principle, this signal can be connected directly to the "Trig" input of IC9, but the trigger signal is preconditioned by IC6 to reduce the noise level of the digitized signal. The origin of this noise is as follows. The sensor interface circuit is located a few feet from the sensor skin, at the base of the robot. Some of the signal from the 135KHz oscillator formed by IC5a is capacitively

coupled to the analog sensor signal lines. In addition, the 135KHz signal, and another signal at 67.5 KHz is used for the modulation and demodulation of signals on the sensor circuit module. The result is that there exists a small amount of the modulation frequencies' signal on the analog sensor signal lines. This noise could have been removed by using a high-pass filter, but it is rejected in a different way. First, IC6b divides the frequency of the 135KHz signal by two to 67.5KHz, this signal appears at the chip's "Q" output. The trigger for the ADC is synchronized by IC6a to a rising edge of the 67.5Khz signal. Thus when bit 4 of the command bus is asserted by the transputer, the ADC is not triggered until the next rising edge of the 67.5 KHz signal. By triggering the ADC at the same point with regard to the phase of the 67.5KHz signal, signals that are integer multiples of the 67.5KHz signal are filtered from the digitized data.

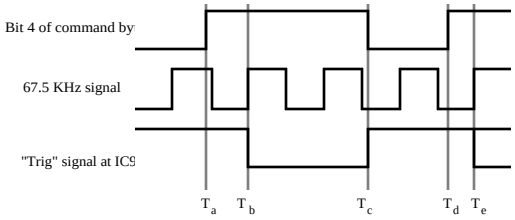


Figure 10. Triggering the ADC in phase with the 67.5KHz signal.

Referring to Figure 10, at point T_a bit 4 of the command byte is brought high, but the ADC is not triggered until at point T_b. When bit 4 returns low at T_c, the trigger signal of IC9 is reset for another sample, which occurs again at points T_d and T_e.

A slight drawback of this triggering scheme is that the trigger command will be delayed by at most 14.8 µsec, however this short amount of time can be considered negligible in the application described.

The sensors are addressed in a serial fashion by a short pulse on the Sensor Select inputs of the sensor circuit module, and are reset by a long pulse on this same signal line. The short pulse is delivered by the One-Shots formed by IC5b, IC5c, and IC5d. Each sensor section is pulsed by its own One-Shot as indicated on Figure 9. The exact pulse length is adjustable by the 200kΩ variable resistor. The long high pulse required for the reset function is provided by bit 3 of the command byte. The Sensor Select signal remains high as long as bit 3 is a "1". By asserting bit 3, the entire skin is reset together, synchronizing the address counters on the three sensor circuit modules.

4.2 Sensor circuit modules

A sensor circuit module contains the majority of the electronics necessary to implement the sensing function. There are three such modules in the current version of the system, each connected to a different part of the skin. The module is fastened to the arm using Velcro fasteners, and the sensor skin is then wrapped over it.

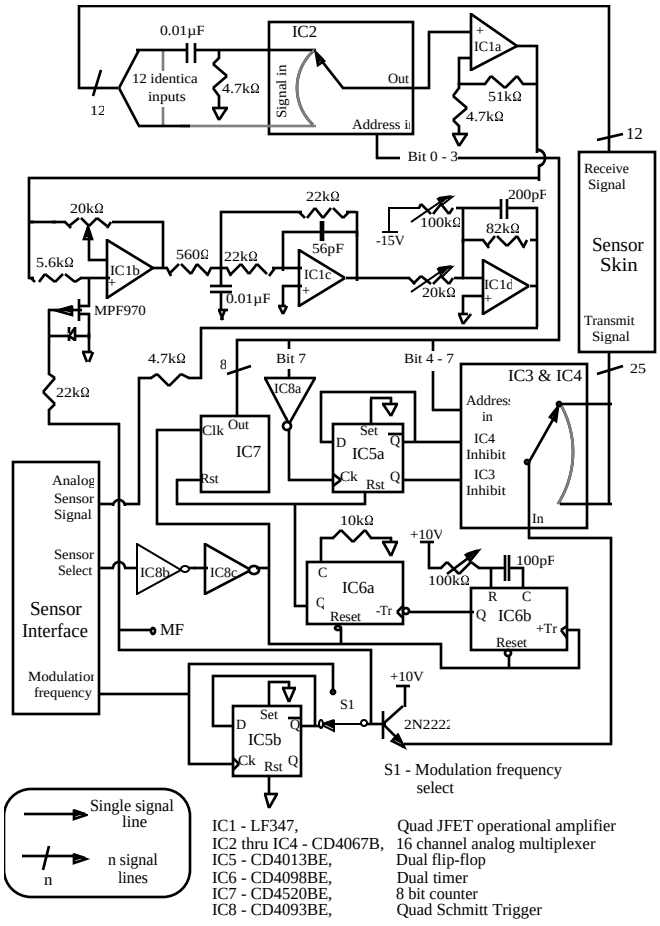


Figure 11. Sensor Circuit Module.

The sensor circuit module can be divided into two parts, see Figure 11. The first part is the sensor addressing circuit, which decodes which sensor is being addressed currently, the second part is the sensor detection circuit which amplifies and filters the signals from the light detectors.

To increase the reliability of the sensor system, it is desirable to minimize the amount of interconnecting wires that run across the joints of the robot. There is a conflicting requirement, however - it is also desirable to be able to address each sensor individually, so that the position of obstacles with respect to the arm can be determined. If a parallel addressing scheme is used, similar to how computer memory is accessed, $log_2\ n$ lines will be needed, where n is the total number of sensors. By using a serial addressing scheme, the sensors can be addressed using just one signal line.

The addressing scheme is organized as follows. Each sensor circuit module has a counter which keeps track of which sensor is currently being addressed. The counter is incremented on every rising edge (low to high transition) of the serial clock line, which causes a new sensor to be selected. Thus pulses of any length, including short ones, will increment the counter. The counter is reset to zero with a long pulse by using a pulse discriminator on the sensor circuit module. In the implemented system, pulses that are "high" for longer than 10 µsec are considered reset pulses; pulses shorter than 10 µsec increment the counter. This addressing scheme allows just one signal line to address a potentially infinite number of sensors.

An obvious drawback of this scheme is that random addressing is not possible. All sensors with addresses lower than a desired sensor will have to be selected when picking a particular sensor. By keeping the pulses as short as possible, one can quickly address a particular sensor, and minimize the amount of wasted time. Note, however, that in general in such a sensor system there will be a need to poll <u>all</u> sensors periodically. In this case the order of addressing is immaterial, and hence the advantages of the serial addressing scheme outweigh its disadvantages.

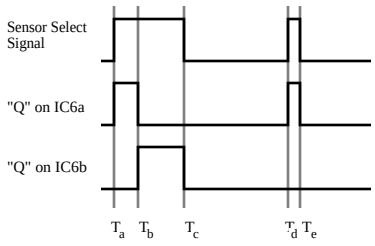


Figure 12. Pulse Discriminator wave forms

Referring to Figure 11, the Sensor Select signal from the Sensor Interface is first "cleaned up" by Schmitt triggers IC8b and IC8c, and is then connected to the "Clk" input of the eight bit counter, which keeps track of which sensor is currently being selected. The pulse discriminator, which decides when the counter should be reset is composed of the dual one-shot IC6.

The pulse discriminator operates as follows. While the Sensor Select line is low, the one-shots' outputs "Q" are low, and the eight bit counter is not reset. On the arrival of a pulse on the Sensor Select line at T_a, one-shot IC6a is triggered by the positive edge, and its output "Q" goes high. If the Sensor Select line stays high longer than 10 µsec, it will allow IC6a to time out, causing its output to drop low at T_b. This triggers IC6b, and its output "Q" goes high, which resets the counter. Thus pulses longer than 10 µsec cause the counter IC7 to be reset. On the other hand, if the Sensor Select signal goes low before IC6a times out, such as at T_d and T_e, Figure 12, no reset pulse will be generated, and the counter increments normally.

The light transmitted by the Infra Red Emitting Diode (IRED) is amplitude modulated, and then synchronously detected to increase the immunity to other light sources. It also allows operation on several "channels", since light transmitted by one sensor can be rejected by another detector which is demodulating at a different frequency. This will be explained in Section 4.3 below.

The output byte of IC7 at its output "Out" controls the analog multiplexers that switch the appropriate optical components to the sensor circuit. The least significant four bits are connected to the analog multiplexer (mux) IC2, which selects among the 12 signals from the preamplifiers on the sensor skin. The signal from the analog switch is first high-pass filtered by IC1a to remove components due to room lighting from the received signal [9]. It is then connected to the synchronous detector ICb, which demodulates the transmitted signal. Operation of the detector is explained below. After demodulation, the signal is then low-pass filtered by the three pole Butterworth filter composed of IC1c and IC1d. The cut-off frequency of the Butterworth filter is at 10KHz. The output of IC1d is then connected to one of the input channels on the Sensor Interface Board via a 4.7kΩ resistor which provides short circuit protection for IC1d's output stage.

The settling time of the Butterworth filter, which determines overall sensor response time, is approximately 0.25 msec. A higher bandwidth filter would settle in less time, but would also make the circuit more noise prone. Since the two links of the arm are polled in parallel, the limiting factor in the sensor update rate is the polling of the larger of the two sensor skin sections, which is link l_3. In the current implementation, the sensor polling rate is such that the entire skin is polled once every sixteenth of a second.

The switch S1 selects the operating frequency of a particular Sensor Circuit Module. By selecting the output of IC5b, the circuit operates at 67.5 KHz, or one half the frequency of the

signal "Modulation frequency", supplied by the Sensor Interface Circuit. The lower frequency (67.5KHz) is used for sensors on link l_2, and the higher frequency (135KHz) is used for sensors on link l_3.

4.3 Operation of the demodulator

Instead of using a full-fledged synchronous demodulator, which requires an analog multiplier, a more simple way of demodulating the signal is used. The demodulator IC1b operates by alternately amplifying the signal at the output of IC1a by +1 and -1 in step with the digital signal at the gate of the MPF970 transistor. This signal is also the same one that is transmitted by the selected IRED, thus the flashing of the IRED is in step with the operation of IC1b. The demodulator operates as follows. Assuming that the 20kΩ variable resistor is at its center, the circuit can be broken into two versions. One version, shown in Figure 13a, is the approx. equiv. circuit when the signal "MF" is high; the other, shown in Figure 13b is when that signal is low [10]. The gain of the demodulator stage thus alternates between +1 and -1 depending on the logic level of the signal "MF".

(a) MF is logic high, gain is -1 (b) MF is logic low, gain is +1.

Figure 13. Approximate circuits for the demodulator depending on the state of signal "MF".

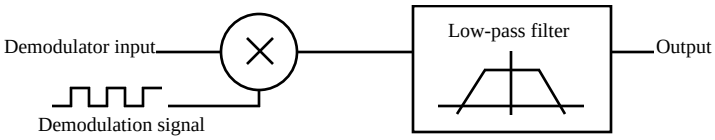


Figure 14. Model of the demodulator and low-pass filter.

To analyze the effect of the demodulator on its input signal, we model it as an analog multiplier with one input connected to a square wave of frequency ω_s and its output connected to a low-pass filter.

A true demodulator uses a sine wave as its demodulation signal. By using the described

circuit, a square wave becomes the demodulation signal. The effect of the demodulator on its input signal is analyzed in two ways, in the frequency domain and in the time domain.

The operation of the demodulator is analyzed by applying a sine wave of known frequency ω_t to the input of the demodulator. The magnitude of the spectrum of the demodulation signal (square wave) is shown in Figure 15.

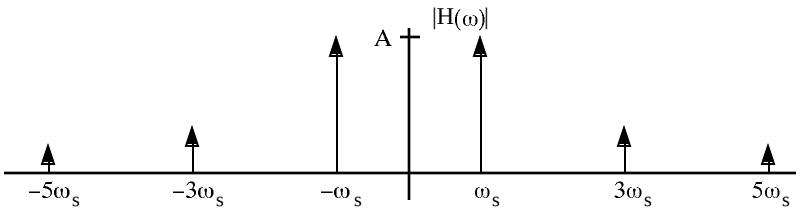


Figure 15. Sketch of the magnitude of the spectrum of a square wave at angular frequency ω_s.

The multiplication of the two signals at the input of the multiplier produces signals whose frequencies are the sum and difference of the frequencies of the original signals, and is equivalent to translation of the spectrum of the input signal [11]. Thus the spectrum of the signal at the output of the demodulator is that at the input shifted by $\omega = \pm \omega_s, \pm 3\ \omega_s, \pm 5\ \omega_s$,...., etc. The output of the demodulator is fed to a low-pass filter to remove unwanted components. If a component of the input signal is shifted into the pass band of the low-pass filter, it will be passed unattenuated. This will happen when the input signal has a component at $\omega_t = n\ \omega_s$, where n = 1, 3, 5 .....etc. More formally stated, the frequency response of the switching synchronous detector is the convolution of the spectrum of the square wave and the response of the low-pass filter, Figure 16.

To gain a more intuitive understanding of the operation of the demodulator, let us examine its operation in the time domain. In Figure 17, the input signals to the analog multiplier are shown on the left, and the output signals of the low-pass filter on the far right. The output signal of the demodulator is filtered by the low-pass filter, 14.

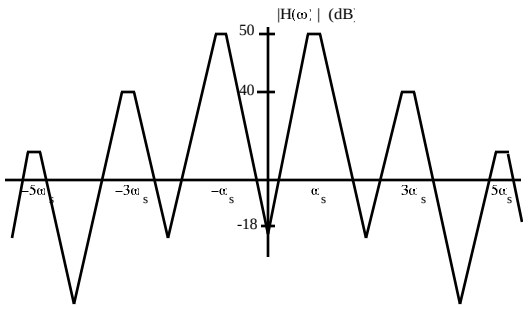


Figure 16. Frequency response of the switching synchronous detector; ω_s is the frequency of the demodulating square wave.

Let us consider three examples, first, assume that the demodulation signal and Signal 1 are in phase, and the signal swing is between ±1 Volt, Figure 17a. The resulting output signal is a constant 1 Volt because the inversion operation of the multiplier is in phase with the polarity of Signal 1.

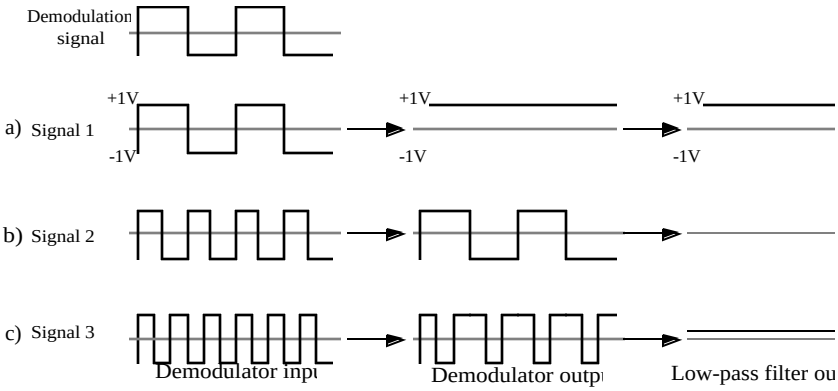


Figure 17. Waveforms for the demodulator examples.

In the second example, the input signal is Signal 2. The output of the demodulator, shown to the right of Signal 2, is filtered by the low-pass filter, Figure 17b. Since this signal has an average (DC) value of 0 Volts, it is blocked by the low-pass filter, and there is no signal at the output of the low-pass filter. Thus signals at twice the frequency of the demodulation

frequency are highly rejected by this system. It can similarly be shown that signals at one half the frequency of the demodulation frequency are also highly rejected. This response is the reason why the operating frequency of the sensors of link l_2 and l_3 are at a 2:1 ratio. Light emitted by one link's transmitter does not affect the operation of a receiver on the other link.

In the third example, the frequency of Signal 3 at the input of the demodulator is three times the demodulation frequency, and its phase is as shown in Figure 17c. In this case, the output of the demodulator has a DC value of 0.3333 Volts. Thus the output of the low-pass filter is 0.3333 Volts. By continuing this example for signals of higher frequencies, it can be shown that signals at odd multiples of the demodulation frequency are passed. More precisely stated, if:

the voltage swing of the input signal $= \pm A$ Volts

the frequency of the demodulation signal $= \omega_s$,

the frequency of the input signal $= \omega_t$, where $\omega_t = n\,\omega_s \quad n = 1,3,5,.....$etc.

Then the DC value of the output of the low-pass filter $< A/n$ Volts

The inequality in the last expression is due to the uncertainty of the phase between the input signal and the demodulation signal. The results of this time domain analysis agree with the results of the frequency domain analysis. Signals at odd multiples of the demodulation frequencies are passed, while signals at even multiples of the demodulation frequency are rejected.

The drawback of the simpler demodulation circuit used in the sensor system is its sensitivity to signals at the odd multiples of the demodulation signal frequency. It is unlikely, however, that there will be stray optical radiation modulated at such a high frequency in the robot environment. Furthermore, for the incident light to produce a false reading, it must have a strong component in the infrared region since the optical components are molded from a specially colored plastic. In addition, high frequency components are attenuated by the *1/n* factor mentioned in the time domain analysis. Even flashing light sources such as cathode ray tubes operate in a much lower frequency band (<16KHz).

The advantage of our switching synchronous demodulator over a multiplying synchronous demodulator is its simplicity and compactness. By using the switching detector, the need for a pure sine wave and an analog multiplier is eliminated. Since the transmitted signal is a square wave, it can easily be used as the control signal for the switching detector. The detection function can be performed using a single op-amp, and the need for circuits that can fit in a small space can be important in these applications.

4.4 Performance of the sensor circuit

The sensor response was assessed experimentally using a randomly selected sensor pair and a 3" x 3" piece of white paper as the test obstacle. The response is measured as a function of the distance D from the skin to the test obstacle, and the angle Φ between the perpendicular to the test obstacle and the sensing axis, see Figure 18.

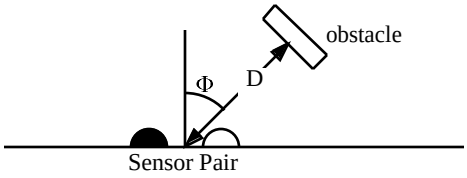


Figure 18. Test setup for measurement of sensor response.

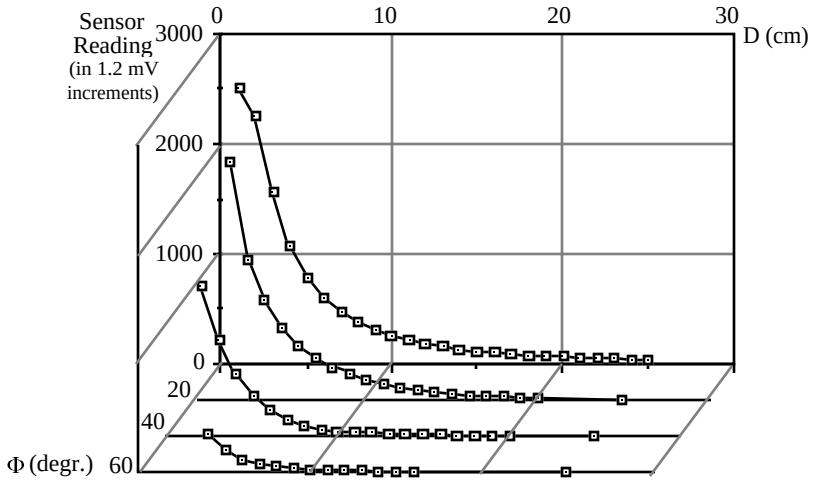


Figure 19. Sensor output for various values of the distance D to the obstacle, and for four values of the angle Φ between the obstacle and the sensing axis.

The output of the sensor, where each unit represents 1.2 mVolts (thus 1000 = 1.2 Volt), is shown in Figure 19 for four values of the angle Φ, Φ= 0°, 20°, 40°, and 60°. Twenty samples were taken for each position of the obstacle, and the results were then averaged. Note a decrease in response for non-zero

a decrease in response for non-zero Φ values due to the beam width of the IRED.

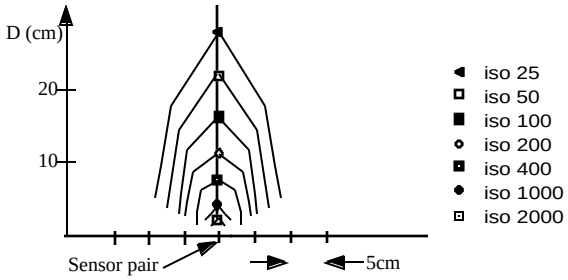


Figure 20. Directional response of one sensor pair.
Iso X refers to the line of constant sensor output of X.

In Figure 20, the sensor data is plotted to show the sensitivity cone for one sensor pair. Lines of constant sensor reading are shown; e.g., "iso 25" refers to the line of a constant sensor reading of 25. The tick marks on the horizontal axis indicate the locations of sensor pairs. Note that if the skin is placed flat, adjacent sensitivity cones overlap when an obstacle similar in color and size to the test obstacle is being detected. Where the skin is curved to fit the contour of the arm, the sensor spacing is reduced to ensure overlap of the sensitivity cones, see Figure 26.

Because of the sufficient density of sensor pairs along the sensitive skin (less than or equal to 5cm between the neighboring sensor pairs), the neighboring sensor beams overlap, and so no obstacle that is at least as detectable in terms of its size and texture as the test obstacle can approach the arm undetected. In fact, if the test obstacle were the smallest possible obstacle in the work space of the arm, and the skin were placed flat, the minimum sensor spacing could be increased to 11 cm. However, if the sensor spacing is increased, or if the sensor skin is bent over a sharp corner, the resulting sensitive area around the arm will have a very irregular or "bumpy" shape. This may lead to rough motion while following the contour of sharp obstacles, since the motion planning system servos around a constant sensor reading. Therefore, sharp corners in the skin material should be avoided and corners should be rounded gradually, see Figure 26.

4.5 The Sensor Skin

The sensor skin is manufactured of a Kapton based material, only 0.0085" thick. Both sides of the material are copper cladded, resembling a regular printed circuit board. The material is etched and drilled using standard printed circuit techniques. After processing, the

circuit board provides for both structural support and electrical interconnection for the optical components. Availability of this material has drastically reduced the problem of mounting the optical components. It allows the placement, with great precision, of the optical devices with little labor.

There were three major challenges faced when covering the arm with a sensitive skin. The first relates to covering link l_2 of the robot arm. As Figure 1 shows, link l_2 of our arm is composed of a trapezoidal closed kinematic chain whose width varies with the change in the joint value Θ_2. Human skin stretches and flexes while maintaining its sensing function, an ability that is difficult to duplicate in electronic devices. To solve this problem, each of the two sections of link l_2 is covered by its own sensor sheet which moves independent of the other. The second challenge was covering the wrist of the arm. Although only the first three degrees of freedom of the arm are directly controlled, to provide collision-free gross motion, we also wish to maintain utility and safety of the wrist. The solution used here was to cover the wrist with a domed section, see Figure 21, that has a slot, narrow enough to assure sensing beam overlap at neighboring sensor pairs but sufficiently wide to allow for the wrist motion. The third challenge relates to covering the joint j_3. To avoid wear and tear of the skin around the joint, the solution chosen was to have the section of sensitive skin covering link l_2 "slide" under the section covering link l_3, see Figure 22.

Altogether, three sections of the skin are used to cover the robot arm, two sections for link l_2 , and one section for link l_3. To save labor and production costs, a single large "standard" skin section was designed. One copy of the standard section was used to cover link l_3; the two smaller skin sections covering link l_2 are cut out of another copy of the standard section, see Figure 23. This design standardization necessitated some small compromises in the overall placement of sensor pairs. Alternatively, the circuitry on the Sensor Circuit Module could probably have been incorporated onto the sensor skin together with the optical components, but it would have necessitated interrupting the regular pattern of sensor pairs.

The overall size of the resulting section covering link l_3 is 61cm x 122cm (2' x 4'), while the smaller sections covering link l_2 are 51cm x 51cm (20" x 20"). One challenge during the production was the shear size of the artwork - several companies specializing in printed circuit boards were not able to accommodate such a large pattern. The resolution of the circuit production equipment is also important because lines as narrow as 0.05" have to be reproduced in some skin sections.

Figure 21. Front view of the arm with installed sensitive skin.
Note the skin dome covering the robot wrist.

Figure 22. Back view of the arm with sensitive skin.
Note the sliding section of the skin around the joint j_3.

Out of the few copies of the standard sheet in the manufacturing batch, three were used for a "dry run": the sensor sheets were cut, drilled, and mounted on the arm without optical components. This allowed final adjustments of the placement of mounting posts and mounting holes and, more importantly, allowed fine-tune planning of the wrist coverage. As mentioned before, the latter includes slits cut into the end of the artwork and joined into a dome shaped enclosure around the wrist. A large opening was then cut into the resulting dome, to allow the wrist motion, see Figure 21. This necessitated transplanting several sensor pairs to new locations.

4.6 Sensor skin circuitry

To reduce the amount of wiring on the arm, the optical components are arranged in an irregularly spaced row - column format. On the (standard) section of sensor skin covering link l_3, there are 25 rows of IREDs, each containing 12 emitting diodes, see Figure 23. The PIN diodes are arranged in 12 columns, each containing 25 light detectors. All the PIN diodes in a column are wired in parallel, and then placed in a reverse bias configuration. As a result of this wiring configuration, the output of all the PIN diodes are summed, and the signal from each of the 12 columns is preamplified by an op-amp operating in transresistance mode. The output of the op-amp is then connected via a connector to the Sensor Circuit Module, Figure 24. The IREDs common to a row are wired in series. The resistor R_i limits the pulsing current to a 20 mAmp peak. This arrangement causes all the IREDs in a row to pulse in unison.

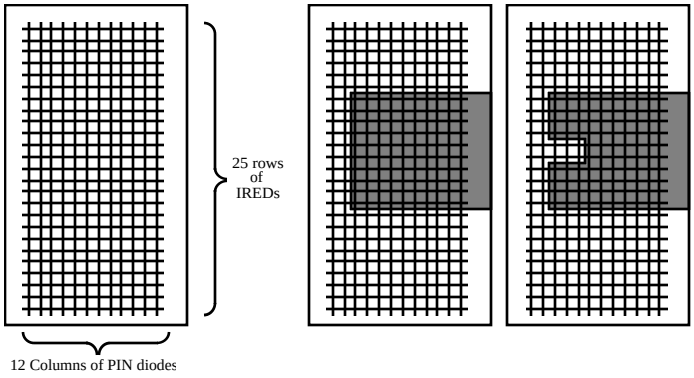


Figure 23. Sheets of the sensitive skin. a) The standard sheet as used for link l_3 (12 x 25) sensor pairs. b) Piece of the sheet (shaded area) cut for the back part of link l_2; contains (10x10) sensor pairs. c) Piece of the sheet cut for the front part of link l_2; contains (10x10 - 6) sensor pairs.

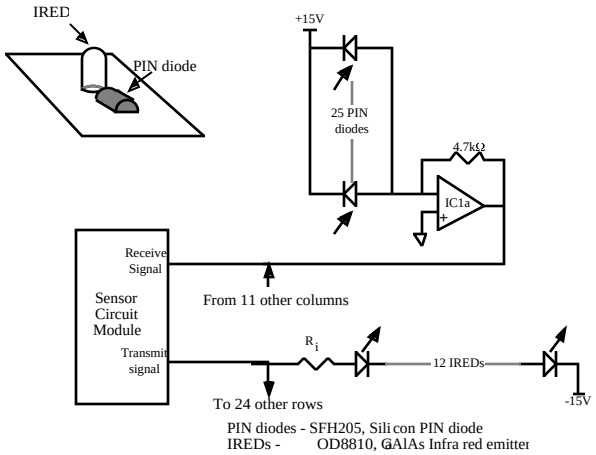


Figure 24. Sensor skin circuit

Two major criteria were considered during the selection of optical components: electrical and optical properties. Cost was less of a factor because in general the prices of these components do not vary widely. As far as electrical properties are concerned, the most efficient and powerful devices available were selected. Selecting the optical properties required more thought, however. In order to reduce the effect of obstacle size on the amount of reflected light (and thus sensor reading), it is desirable to reduce the beam width of the transmitted signal as much as possible. This comes from the fact that at distances at which the sensor operates the emitted light of the IRED illuminates only a spot on the obstacle, causing the rest of the obstacle to be invisible to the sensor. A compromise must be used, however: if the beam widths of the emitters are too narrow, small obstacles may be missed if they can fit between two sensors' beams. Ultimately, the limited choices available between the beam widths led to selecting an IRED with a half-power angle of 10^{o} [12].

As far as the PIN diodes are concerned, to reduce the affect of room lighting, the material of its enclosure should double as an ambient light filter. The last element to select is the kind of lens provided with the PIN diode, which determines its directional sensitivity. It turns out that by a proper choice of the lens the sensor insensitivity to the surface reflectance property of the obstacle can be improved. The wavelength of infrared light is in the range of hundreds of nanometers, scattering light equally in all directions, and tending obstacles to appear matte [6]. Many obstacles, however, do reflect light specularly to some degree, causing light to be reflected away from the transmitter. This problem can be partially solved using the fact that the

outputs of all the PIN diodes in one column are summed. Thus, the light beam emitted from one sensor pair can be picked up by a PIN diode at another location along the arm. This effectively results in a receiver that is physically distributed over the arm. Light reflected to another sensor pair - for example from Sensor pair 4 to Sensor pair 3, in Figure 25 - will be hitting the PIN diode at a large angle Φ. For this light to be received, the PIN diode should have a concave lens, or no lens at all. Since the former type is not available, PIN diodes with no lenses were selected.

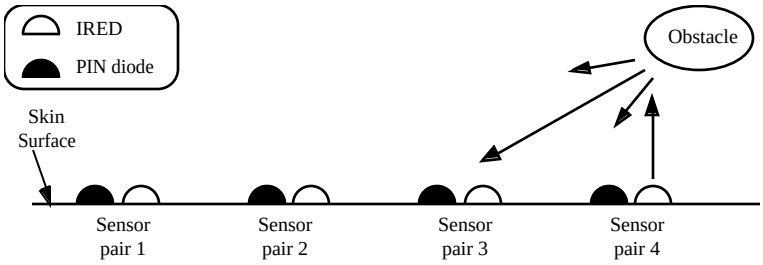


Figure 25. Light from the emitter in one sensor pair
can be received by the detector in another sensor pair.

4.7 Mounting the skin on the arm

The sensor skin is mounted on the robot arm using several fasteners distributed over the surface of the arm. A cross section view of link l_3 and the sensitive skin is shown in Figure 26. Each fastener presents a bolt attached at an angle of 10^o to the surface of an acrylic foot. The foot is a 1" x 1" piece of acrylic glued to the surface of the arm. The 10^o angle accommodates the curvature of the sensor skin at the mounting point. Two nuts threaded onto the bolt clamp the sensor skin down, and allow continuous adjustment of the cross section shape of the skin. The two skin sections covering link l_2 are fastened using a similar technique.

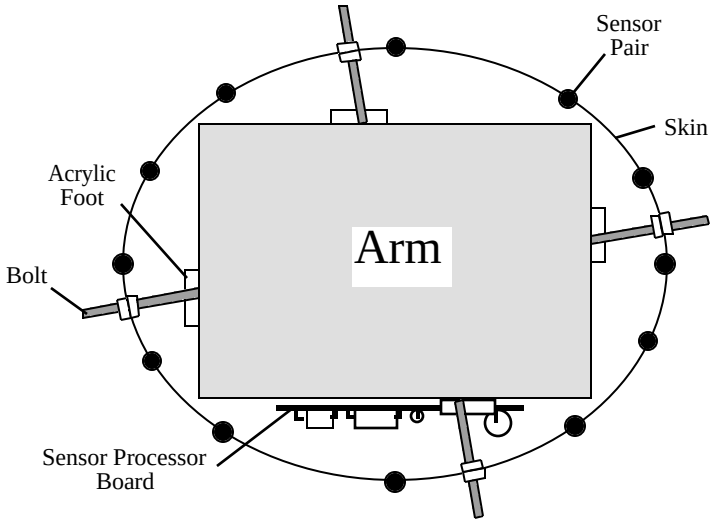


Figure 26. Cross-sectional view of the arm with sensitive skin. Note the gradual curve in the skin and the variable sensor spacing - the distance between sensors is smaller at the areas of greater curvature of the skin.

References

1. E. Cheung and V. Lumelsky, Development of Sensitive Skin for a 3d Robot Arm Operating in an Uncertain Environment, Proc. 1989 IEEE Conference on Robotics and Automation, Scottsdale, AZ, May 1989.

2. E. Cheung and V. Lumelsky, Proximity Sensing for Robot Manipulator Motion Planning: System and Implementation Issues, IEEE Transactions on Robotics & Automation, December 1989.

3. M. Buehler, L. Whitcomb, F.Levin, D. Koditschek, A Distributed Message Passing Computational and I/O Engine for Real-time Motion Control, Proc. American Control Conference, American Control Society, Pittsburgh, PA, June 1989.

4. Transputer Development System, Prentice-Hall, New York 1988.

5. C.A. Hoare, OCCAM 2 Reference Manual, Prentice-Hall, New York 1988.

6. D. Ballard, C. Brown, Computer Vision, Prentice-Hall, New Jersey, 1982.

7. The TTL Data Book, Texas Instruments, 1981.

8. GE Solid State Data Book High Reliability CMOS ICs, General Electric, Somerville New Jersey, 1988.

9. P. Horowitz, W. Hill, The Art of Electronics, Cambridge University Press, Cambridge 1980.

10. Linear Data Book, National Semiconductor Corporation, 1982.

11. B. Lathi, Signals, Systems and Communication, John Wiley & Sons, New York 1965.

12. Opto-Diode Data Book, Opto-Diode Corporation, 1988.