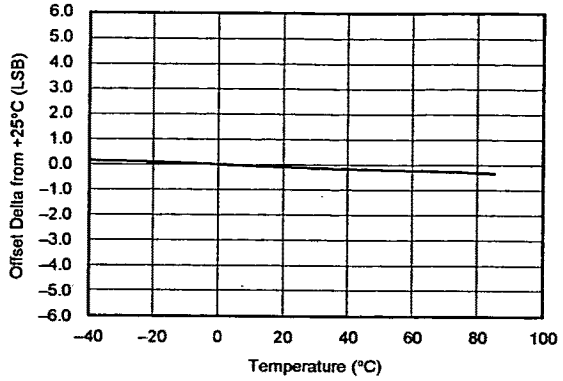
EXHIBIT 3.14

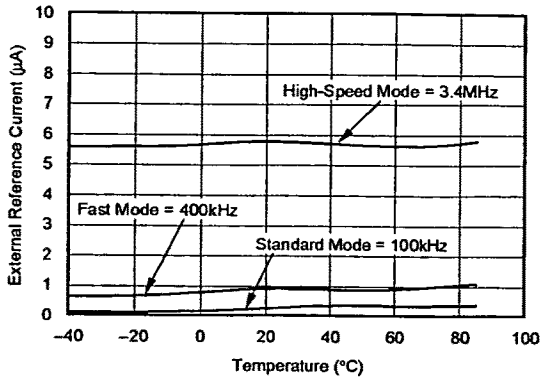
TYPICAL CHARACTERISTICS: +2.7V (Cont.)

At T_A = +25°C, $+V_{DD}$ = +2.7V, V_{REF} = External +2.5V, I²C bus frequency = 3.4MHz, PD1 = PD0 =0, unless otherwise noted.

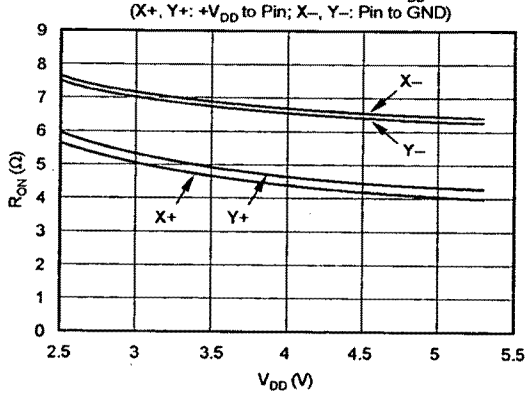
CHANGE IN OFFSET vs TEMPERATURE



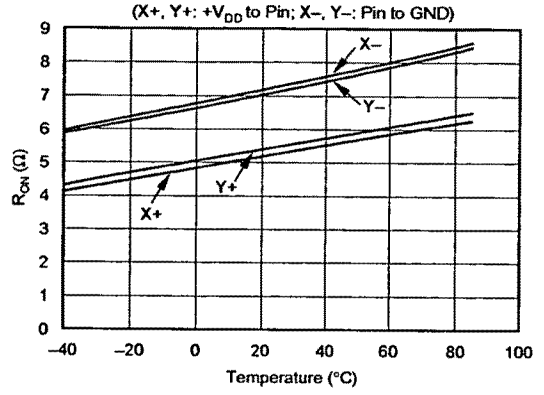
EXTERNAL REFERENCE CURRENT vs TEMPERATURE



SWITCH-ON RESISTANCE vs V_{DD} ($X+$, $Y+$: $+V_{DD}$ to Pin; $X–$, $Y–$: Pin to GND)

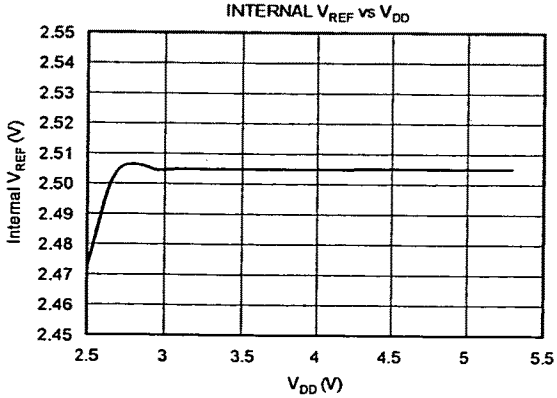


SWITCH-ON RESISTANCE vs TEMPERATURE ($X+$, $Y+$: $+V_{DD}$ to Pin; $X–$, $Y–$: Pin to GND)



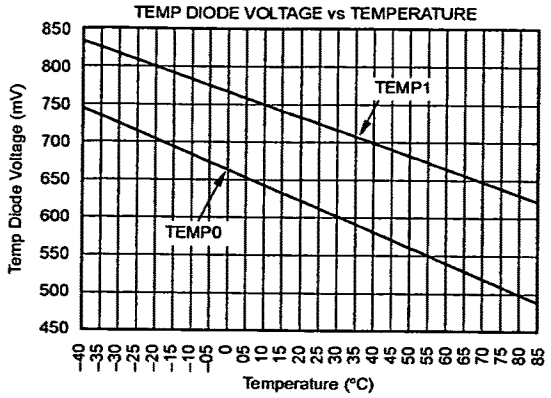
INTERNAL V_{REF} vs TEMPERATURE

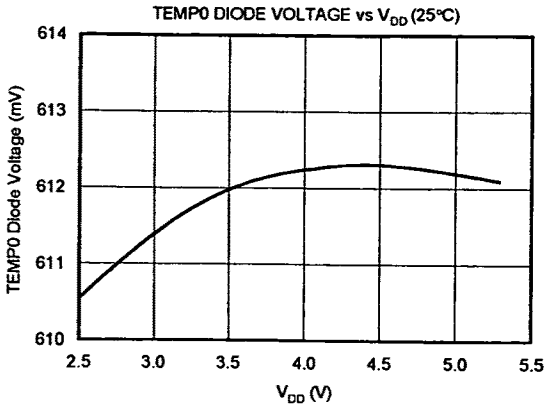
INTERNAL V_{REF} vs V_{DD}

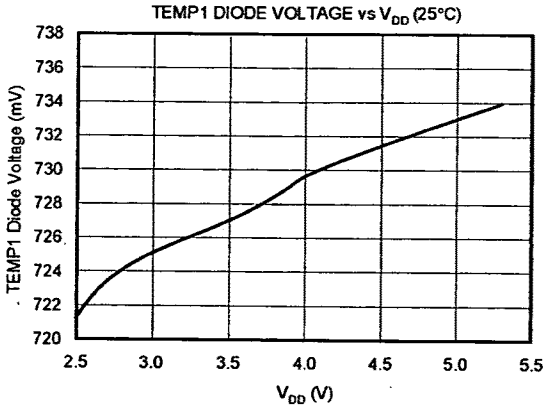


TYPICAL CHARACTERISTICS: +2.7V (Cont.)

At T_A = +25°C, $+V_{DD}$ = +2.7V, V_{REF} = External +2.5V, I²C bus frequency = 3.4MHz, PD1 = PD0 =0, unless otherwise noted.







THEORY OF OPERATION

The TSC2003 is a classic Successive Approximation Register (SAR) Analog-to-Digital (A/D) converter. The architecture is based on capacitive redistribution which inherently includes a sample-and-hold function. The converter is fabricated on a 0.6µ CMOS process.

The basic operation of the TSC2003 is shown in Figure 1. The device features an internal 2.5V reference and an internal clock. Operation is maintained from a single supply of 2.7V to 5.25V. The internal reference can be overdriven with an external, low-impedance source between 2V and $+V_{DD}$. The value of the reference voltage directly sets the input range of the converter.

The analog input (X, Y, and Z parallel coordinates, auxiliary inputs, battery voltage, and chip temperature) to the converter is provided via a multiplexer. A unique configuration of low on-resistance switches allows an unselected A/D converter input channel to provide power, and an accompanying pin to provide ground for an external device. By maintaining a differential input to the converter, and a differential reference architecture, it is possible to negate the switch's on-resistance error (should this be a source of error for the particular measurement).

ANALOG INPUT

See Figure 2 for a block diagram of the input multiplexer on the TSC2003, the differential input of the A/D converter, and the converter's differential reference.

When the converter enters the Hold mode, the voltage difference between the +IN and –IN inputs (see Figure 2) is captured on the internal capacitor array. The input current on the analog inputs depends on the conversion rate of the device. During the sample period, the source must charge the internal sampling capacitor (typically 25pF). After the capacitor has been fully charged, there is no further input current. The amount of charge transfer from the analog source to the converter is a function of conversion rate.

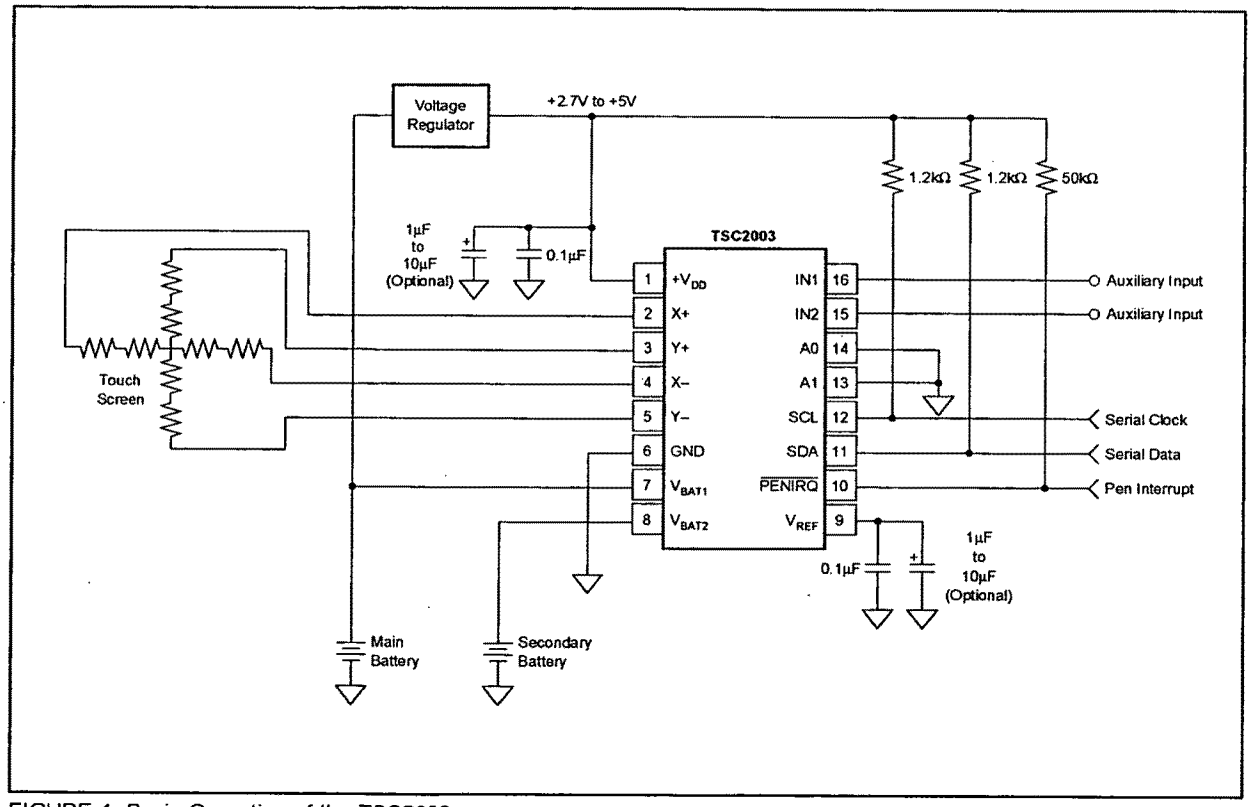


FIGURE 1. Basic Operation of the TSC2003.

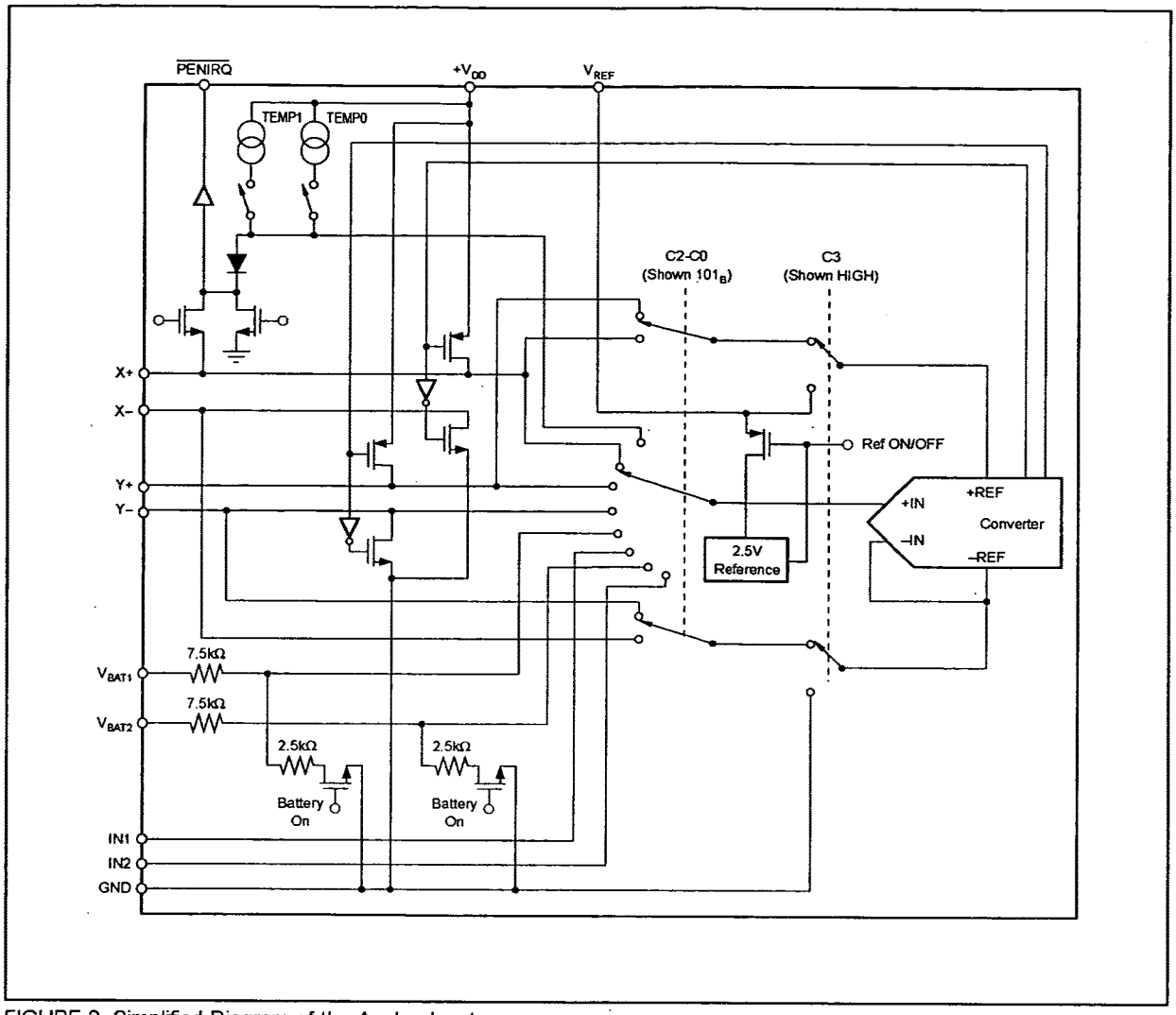


FIGURE 2. Simplified Diagram of the Analog Input.

INTERNAL REFERENCE

The TSC2003 has an internal 2.5V voltage reference that can be turned ON or OFF with the power-down control bits, PD0 and PD1 (see Table II and Figure 3). The internal reference is powered down when power is first applied to the device.

The internal reference voltage is only used in the single-ended reference mode for battery monitoring, temperature measurement, and for measuring the auxiliary input. Optimal touch screen performance is achieved when using a ratiometric conversion; thus, all touch screen measurements are done automatically in the differential mode.

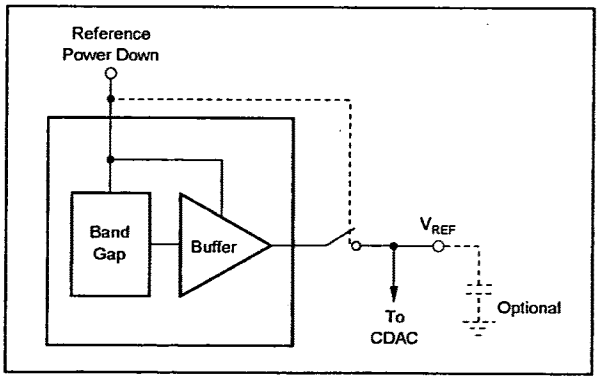


FIGURE 3. Simplified Diagram of the Internal Reference.

REFERENCE INPUT

The voltage difference between +REF and –REF (see Figure 2) sets the analog input range. The TSC2003 will operate with a reference in the range of 2V to $+V_{DD}$. There are several critical items concerning the reference input and its wide-voltage range. As the reference voltage is reduced, the analog voltage weight of each digital output code is also reduced. This is often referred to as the LSB (Least Significant Bit) size, and is equal to the reference voltage divided by 4096 (256 if in 8-bit mode). Any Offset or Gain error inherent in the A/D converter will appear to increase, in terms of LSB size, as the reference voltage is reduced. For example, if the offset of a given converter is 2LSBs with a 2.5V reference, it will typically be 2.5LSBs with a 2V reference. In each case, the actual offset of the device is the same, 1.22mV. With a lower reference voltage, more care must be taken to provide a clean layout including adequate bypassing, a clean (low-noise, low-ripple) power supply, a low-noise reference (if an external reference is used), and a low-noise input signal.

The voltage into the V_{REF} input is not buffered, and directly drives the Capacitor Digital-to-Analog Converter (CDAC) portion of the TSC2003. Therefore, the input current is very low, typically < 6µA.

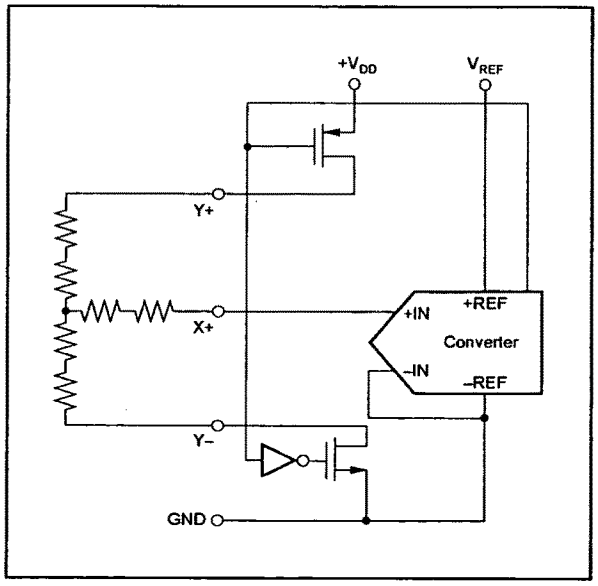


FIGURE 4. Simplified Diagram of Single-Ended Reference.

REFERENCE MODE

There is a critical item regarding the reference when making measurements while the switch drivers are ON. For this discussion, it is useful to consider the basic operation of the TSC2003 (see Figure 1). This particular application shows the device being used to digitize a resistive touch screen. A measurement of the current Y position of the pointing device is made by connecting the X+ input to the A/D converter, turning on the Y+ and Y– drivers, and digitizing the voltage on X+, as shown in Figure 4. For this measurement, the resistance in the X+ lead does not affect the conversion; it does, however, affect the settling time, but the resistance is usually small enough that this is not a concern. However, since the resistance between Y+ and Y– is fairly low, the on-resistance of the Y drivers does make a small difference. Under the situation outlined so far, it would not be possible to achieve a 0V input or a full-scale input regardless of where the pointing device is on the touch screen because some voltage is lost across the internal switches. In addition, the internal switch resistance is unlikely to track the resistance of the touch screen, providing an additional source of error.

This situation is remedied, as shown in Figure 5, by using the differential mode: the +REF and –REF inputs are connected directly to Y+ and Y–, respectively. This makes the A/D converter ratiometric. The result of the conversion is always a percentage of the external reference, regardless of how it changes in relation to the on-resistance of the internal switches.

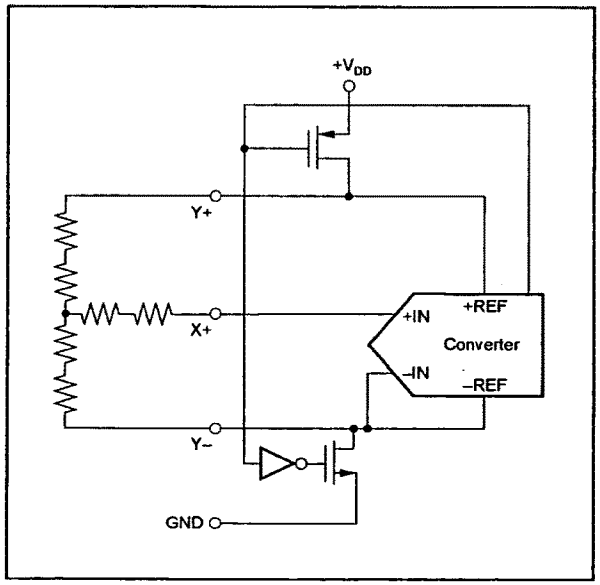


FIGURE 5. Simplified Diagram of Differential Reference (Y Switches Enabled, X+ is Analog Input).

Differential reference mode always uses the supply voltage, through the drivers, as the reference voltage for the A/D converter. V_{REF} cannot be used as the reference voltage in differential mode.

It is possible to use a high-precision reference on V_{REF} in single-ended reference mode for measurements which do not need to be ratiometric (i.e., battery voltage, temperature measurement, etc.). In some cases, it could be possible to power the converter directly from a precision reference. Most references can provide enough power for the TSC2003, but they might not be able to supply enough current for the external load, such as a resistive touch screen.

TOUCH SCREEN SETTLING

In some applications, external capacitors may be required across the touch screen for filtering noise picked up by the touch screen (i.e., noise generated by the LCD panel or backlight circuitry). These capacitors will provide a low-pass filter to reduce the noise, but they will also cause a settling time requirement when the panel is touched. The settling time will typically show up as a gain error. The problem is that the input and/or reference has not settled to its final steady-state value prior to the A/D converter sampling the input(s), and providing the digital output. Additionally, the reference voltage may still be changing during the measurement cycle.

To resolve these settling time problems, the TSC2003 can be commanded to turn on the drivers only without performing a conversion (see Table I). Time can then be allowed before the command is issued to perform a conversion. Generally, the time it takes to communicate the conversion command over the I²C bus is adequate for the touch screen to settle.

TEMPERATURE MEASUREMENT

In some applications, such as battery recharging, a measurement of ambient temperature is required. The temperature measurement technique used in the TSC2003 relies on the characteristics of a semiconductor junction operating at a fixed current level to provide a measurement of the temperature of the TSC2003 chip. The forward diode voltage (V_{BE}) has a well-defined characteristic versus temperature. The temperature can be predicted in applications by knowing the 25°C value of the V_{BE} voltage and then monitoring the delta of that voltage as the temperature changes. The TSC2003 offers two modes of temperature measurement.

The first mode requires calibrations at a known temperature, but only requires a single reading to predict the ambient temperature. A diode is used during this measurement cycle. The voltage across the diode is connected through the MUX for digitizing the diode forward bias voltage by the A/D converter with an address of C3 = 0, C2 = 0, C1 = 0, and C0 = 0 (see Table I and Figure 6 for details). This voltage is typically 600mV at +25°C, with a 20µA current through it. The absolute value of this diode voltage can vary a few millivolts; the Temperature Coefficient (TC) of this voltage is very consistent at –2.1mV/°C. During the final test of the end product, the diode voltage would be stored at a known room temperature, in memory, for calibration purposes by the user. The result is an equivalent temperature measurement resolution of 0.3°C/LSB.

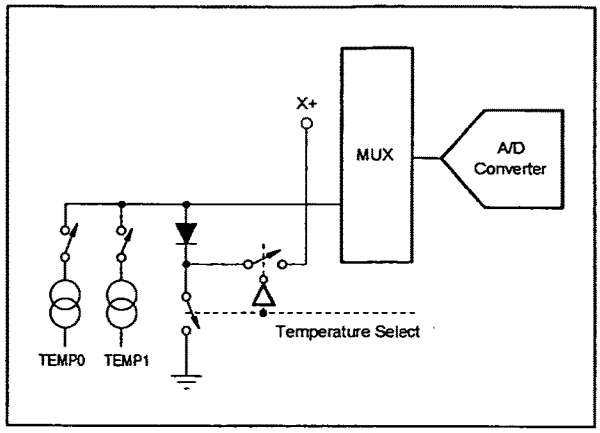


FIGURE 6. Functional Block Diagram of Temperature Measurement Mode.

The second mode does not require a test temperature calibration, but uses a two-measurement method to eliminate the need for absolute temperature calibration and for achieving 2°C/LSB accuracy. This mode requires a second conversion with an address of C3 = 0, C2 = 1, C1 = 0, and C0 = 0, with an 91 times larger current. The voltage difference between the first and second conversion using 91 times the bias current will be represented by kT/q • 1n (N), where N is the current ratio = 91, k = Boltzmann's constant ($1.38054 \cdot 10^{-23}$ electrons volts/degrees Kelvin), q = the electron charge ($1.602189 \cdot 10^{-19}$ C), and T = the temperature in degrees Kelvin. This mode can provide improved absolute temperature measurement over the first mode, but at the cost of less resolution (1.6°C/LSB). The equation to solve for °K is:

$$°K = \frac{q \bullet \Delta V}{k \bullet 1n(N)} \quad (1)$$

where:

$$\Delta V = V(I_{91}) - V(I_1) \text{ (in mV)}$$

$$\therefore °K = 2.573 \Delta V °K/mV$$

$$°C = 2.573 \bullet \Delta V(mV) - 273°K$$

NOTE: The bias current for each diode temperature measurement is only turned ON during the acquisition mode, and, therefore, does not add any noticeable increase in power, especially if the temperature measurement only occurs occasionally.

BATTERY MEASUREMENT

An added feature of the TSC2003 is the ability to monitor the battery voltage on the other side of the voltage regulator (DC/DC converter), as shown in Figure 7. The battery voltage can vary from 0.5V to 6V, while the voltage regulator maintains the voltage to the TSC2003 at 2.7V, 3.3V, etc. The input voltage (V_{BAT1} or V_{BAT2}) is divided down by 4 so that a 6.0V battery voltage is represented as 1.5V to the A/D converter. The simplifies the multiplexer and control logic. In order to minimize the power consumption, the divider is only ON during the sample period which occurs after control bits C3 = 0, C2 = 0, C1 = 0, and C0 = 1 (V_{BAT1}) or C3 = 0, C2 = 1, C1 = 0, and C0 = 1 (V_{BAT2}) are received. See Tables I and II for the relationship between the control bits and configuration of the TSC2003.

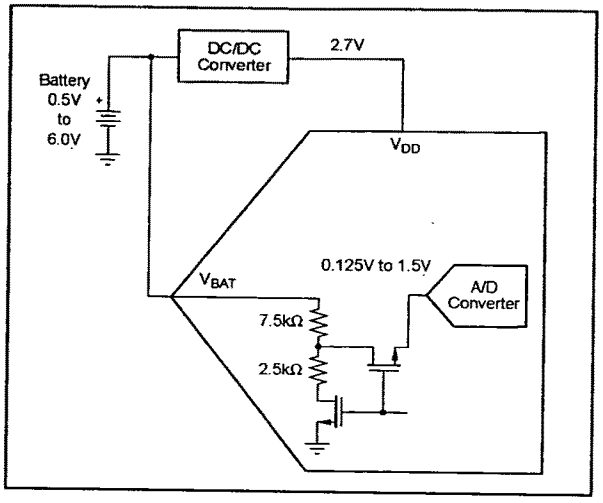


FIGURE 7. Battery Measurement Functional Block Diagram.

PRESSURE MEASUREMENT

Measuring touch pressure can also be done with the TSC2003. To determine pen or finger touch, the pressure of the "touch" needs to be determined. Generally, it is not necessary to have high accuracy for this test, therefore, the 8-bit resolution mode is recommended. However, calculations will be shown with the 12-bit resolution mode. There are several different ways of performing this measurement—the TSC2003 supports two methods.

The first method requires knowing the X-Plate resistance, measurement of the X-Position, and two additional cross-panel measurements (Z_2 and Z_1) of the touch screen, as shown in Figure 8. Using Equation 2 will calculate the touch resistance:

$$R_{TOUCH} = R_{X-Plate} \bullet \frac{X-Position}{4096} \left(\frac{Z_2}{Z_1} - 1 \right) \quad (2)$$

The second method requires knowing both the X-Plate and Y-Plate resistance, measurement of X-Position and Y-Position, and Z_1. Equation 3 calculates the touch resistance using the second method:

$$R_{TOUCH} = \frac{R_{X-Plate} \bullet X-Position}{4096} \left(\frac{4096}{Z_1} - 1 \right) - R_{Y-Plate} \bullet \left(1 - \frac{Y-Position}{4096} \right) \quad (3)$$

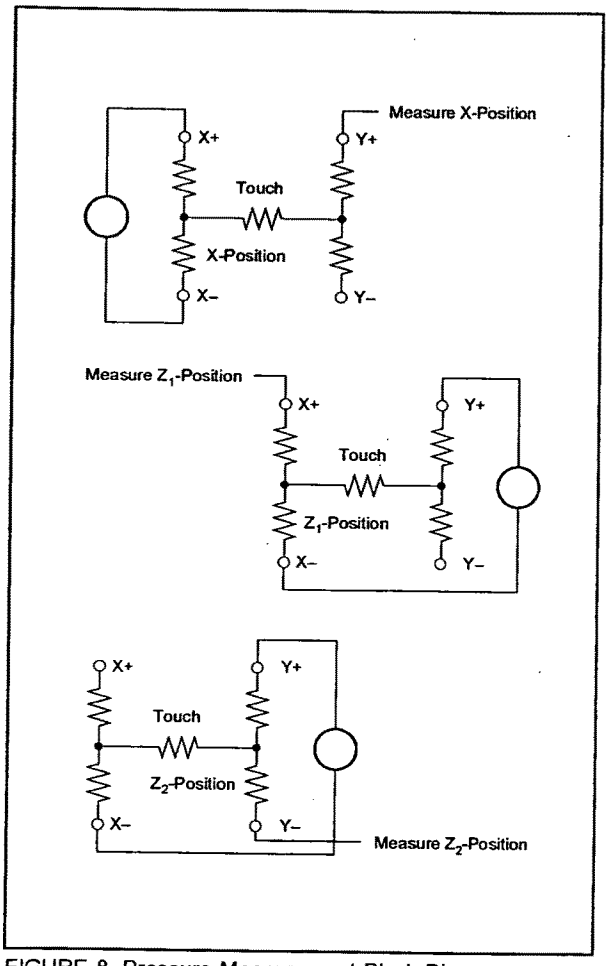


FIGURE 8. Pressure Measurement Block Diagrams.

DIGITAL INTERFACE

The TSC2003 supports the I²C serial bus and data transmission protocol in all three defined modes: standard, fast, and high-speed. A device that sends data onto the bus is defined as a transmitter, and a device receiving data as a receiver. The device that controls the message is called a *master*. The devices that are controlled by the master are *slaves*. The bus must be controlled by a master device which generates the

serial clock (SCL), controls the bus access, and generates the START and STOP conditions. The TSC2003 operates as a slave on the I²C bus. Connections to the bus are made via the open-drain I/O lines SDA and SDL.

The following bus protocol has been defined, as shown in Figure 9:

- Data transfer may be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is HIGH. Changes in the data line while the clock line is HIGH will be interpreted as control signals.

Accordingly, the following bus conditions have been defined:

Bus Not Busy: Both data and clock lines remain HIGH.

Start Data Transfer: A change in the state of the data line, from HIGH to LOW, while the clock is HIGH defines a START condition.

Stop Data Transfer: A change in the state of the data line, from LOW to HIGH, while the clock line is HIGH defines a STOP condition.

Data Valid: The state of the data line represents valid data when, after a START condition, the data line is stable for the duration of the HIGH period of the clock signal. There is one clock pulse per bit of data.

Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of data bytes transferred between START and STOP conditions is not limited, and is determined by the master device. The information is transferred byte-wise, and each receiver acknowledges with a ninth-bit.

Within the I²C bus specifications, a standard mode (100kHz clock rate), a fast mode (400kHz clock rate), and a high-speed mode (3.4MHz clock rate) are defined. The TSC2003 works in all three modes.

Acknowledge: Each receiving device, when accessed, is obliged to generate an acknowledge after the reception of each byte. The master device must generate an extra clock pulse, which is associated with this acknowledge bit.

A device that acknowledges must pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable LOW during the HIGH period of the acknowledge clock pulse. Of course, setup and hold times must be taken into account. A master must signal an end of data to the slave by not generating an acknowledge bit on the last byte that has been clocked out of the slave. In this case, the slave must leave the data line HIGH to enable the master to generate the STOP condition.

Figure 9 details how data transfer is accomplished on the I²C bus. Depending upon the state of the R/$\overline{W}$ bit, two types of data transfer are possible:

- **Data transfer from a master transmitter to a slave receiver.** The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after the slave address and each received byte.
- **Data transfer from a slave transmitter to a master receiver.** The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. Next, a number of data bytes are transmitted by the slave to the master. The master returns an acknowledge bit after all received bytes other than the last one. At the end of the last received byte, a 'not acknowledge' is returned.

The master device generates all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the bus will not be released.

The TSC2003 may operate in the following two modes:

- **Slave Receiver Mode:** Serial data and clock are received through SDA and SCL. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and direction bit.
- **Slave Transmitter Mode:** The first byte (the slave address) is received and handled as in the slave receiver mode. However, in this mode the direction bit will indicate that the transfer direction is reversed. Serial data is transmitted on SDA by the TSC2003 while the serial clock is input on SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer.

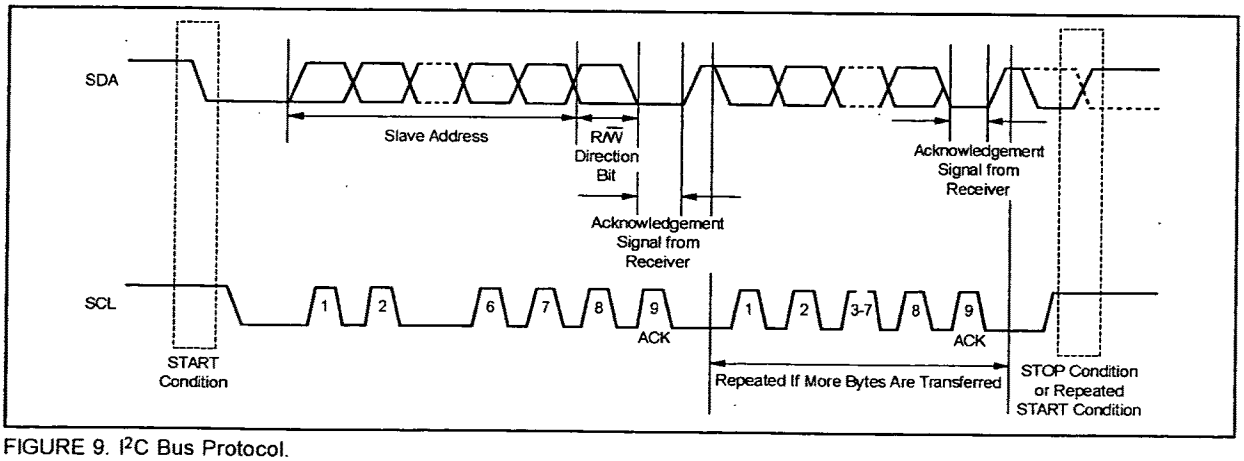


FIGURE 9. I²C Bus Protocol.

Address Byte

The address byte, as shown in Figure 10, is the first byte received following the START condition from the master device. The first five bits (MSBs) of the slave address are factory preset to 10010. The next two bits of the address byte are the device select bits: A1 and A0. Input pins (A1-A0) on the TSC2003 determine these two bits of the device address for a particular TSC2003. Therefore, a maximum of four devices with the same preset code can be connected on the same bus at one time.

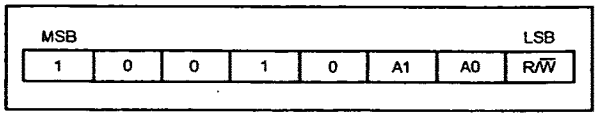


FIGURE 10. Address Byte.

The A1-A0 Address Inputs can be connected to V_{DD} or digital ground. The last bit of the address byte ($R/\overline{W}$) defines the operation to be performed. When set to a "1", a read operation is selected; when set to a "0", a write operation is selected. Following the START condition, the TSC2003 monitors the SDA bus and checks the device type identifier being transmitted. Upon receiving the 10010 code, the appropriate device select bits, and the $R/\overline{W}$ bit, the slave device outputs an acknowledge signal on the SDA line.

Command Byte

The TSC2003's operating mode is determined by a command byte, which is shown in Figure 11.

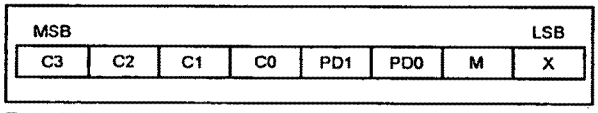


FIGURE 11. Command Byte.

The bits in the device command byte are defined as follows:

- **C3-C0:** Configuration bits. These bits set the input multiplexer address and functions that the TSC2003 will perform, as shown in Table I.
- **PD1-PD0:** Power-down bits. These two bits select the power-down mode that the TSC2003 will be in after the current command completes, as shown in Table II.

The internal reference voltage can be turned ON or OFF independently of the A/D converter. This can allow extra time for the internal reference voltage to settle to its final value prior to making a conversion. Make sure to allow this extra wake-up time if the internal reference was powered down. Also note that the status of the internal reference power down is latched into the part (internally) when a STOP or repeated START occurs at the end of a command byte (see Figures 12 and 14). Therefore, in order to turn the internal reference OFF, an additional write to the TSC2003, with PD1 = 0, is required after the channel has been converted.

It is recommended to set PD0 = 0 in each command byte to get the lowest power consumption possible. If multiple X-, Y-, and Z-position measurements will be done one right after another, such as when averaging, PD0 =1 will leave the touch screen drivers on at the end of each conversion cycle.

- **M:** Mode bit. If M is 0, the TSC2003 is in 12-bit mode. If M is 1, 8-bit mode is selected.
- **X:** Don't care.

PD1	PD0	$\overline{PENIRQ}$	DESCRIPTION
0	0	Enabled	Power-Down Between Conversions
0	1	Disabled	Internal reference OFF, ADC(1) ON
1	0	Enabled	Internal reference ON, ADC(1) OFF
1	1	Disabled	Internal reference ON, ADC(1) ON

NOTE: (1) ADC = Analog-to Digital Converter.

TABLE II. Power-Down Bit Functions.

When the TSC2003 powers up, the power-down mode bits need to be written to ensure that the part is placed into the desired mode to achieve lowest power. Therefore, immediately after power-up, a command byte should be sent which sets PD1 = PD0 = 0, so that the device will be in the lowest power mode, powering down between conversions.

Start A Conversion/Write Cycle

A Conversion/Write Cycle begins when the master issues the address byte containing the slave address of the TSC2003, with the eighth bit equal to a 0 ($R/\overline{W}$ = 0), as shown in Figure 10. Once the eighth bit has been received, and the address matches the A1-A0 address input pin setting, the TSC2003 issues an acknowledge.

C3	C2	C1	C0	FUNCTION	INPUT to ADC	X-DRIVERS	Y-DRIVERS	REFERENCE MODE
0	0	0	0	Measure TEMP0	TEMP0	OFF	OFF	Single-Ended
0	0	0	1	Measure V_{BAT1}	V_{BAT1}	OFF	OFF	Single-Ended
0	0	1	0	Measure IN1	IN1	OFF	OFF	Single-Ended
0	0	1	1	Reserved	–	–	–	Single-Ended
0	1	0	0	Measure TEMP1	TEMP1	OFF	OFF	Single-Ended
0	1	0	1	Measure V_{BAT2}	V_{BAT2}	OFF	OFF	Single-Ended
0	1	1	0	Measure IN2	IN2	OFF	OFF	Single-Ended
0	1	1	1	Reserved	–	–	–	Single-Ended
1	0	0	0	Activate X– Drivers	–	ON	OFF	Differential
1	0	0	1	Activate Y– Drivers	–	OFF	ON	Differential
1	0	1	0	Activate Y+, X– Drivers	–	X– ON	Y+ ON	Differential
1	0	1	1	Reserved	–	–	–	Differential
1	1	0	0	Measure X Position	Y+	ON	OFF	Differential
1	1	0	1	Measure Y Position	X+	OFF	ON	Differential
1	1	1	0	Measure Z_1 Position	X+	X– ON	Y+ ON	Differential
1	1	1	1	Measure Z_2 Position	Y–	X– ON	Y+ ON	Differential

TABLE I. Possible Input Configurations.

Once the master receives the acknowledge bit from the TSC2003, the master writes the command byte to the slave (see Figure 11). After the command byte is received by the slave, the slave issues another acknowledge bit. The master then ends the Write Cycle by issuing a repeated START or a STOP condition, as shown in Figure 12.

If the master sends additional command bytes after the initial byte, before sending a STOP or repeated START condition, the TSC2003 will not acknowledge those bytes.

The input multiplexer for the A/D converter has its channel selected when bits C3 through C0 are clocked in. If the selected channel is an X-,Y-, or Z-position measurement, the appropriate drivers will turn on once the acquisition period begins.

When R/$\overline{W}$ = 0, the input sample acquisition period starts on the falling edge of SCL once the C0 bit of the command byte has been latched, and ends when a STOP or repeated START condition has been issued. A/D conversion starts immediately after the acquisition period. The multiplexer inputs to the A/D converter are disabled once the conversion period starts. However, if an X-, Y-, or Z-position is being measured, the respective touch screen drivers remain on during the conversion period. A complete Write Cycle is shown in Figure 12.

Read A Conversion/Read Cycle

For best performance, the I²C bus should remain in an idle state while an A/D conversion is taking place. This prevents digital clock noise from affecting the bit decisions being made by the TSC2003. The master should wait for at least 10µs before attempting to read data from the TSC2003 to realize this best performance. However, the master does not need to wait for a completed conversion before beginning a read from the slave, if full 12-bit performance is not necessary.

Data access begins with the master issuing a START condition followed by the address byte (see Figure 10) with R/$\overline{W}$ = 1. Once the eighth bit has been received, and the address matches, the slave issues an acknowledge. The first byte of serial data will follow (D11-D4, MSB first).

After the first byte has been sent by the slave, it releases the SDA line for the master to issue an acknowledge. The slave responds with the second byte of serial data upon receiving the acknowledge from the master (D3-D0, followed by four 0 bits). The second byte is followed by a NOT acknowledge bit (ACK = 1) from the master to indicate that the last data byte has been received. If the master acknowledges the second data byte, then the data will repeat on subsequent reads with ACKs between bytes. This is true in both 12-bit and 8-bit mode. The master will then issue a STOP condition, which ends the Read Cycle, as shown in Figure 13.

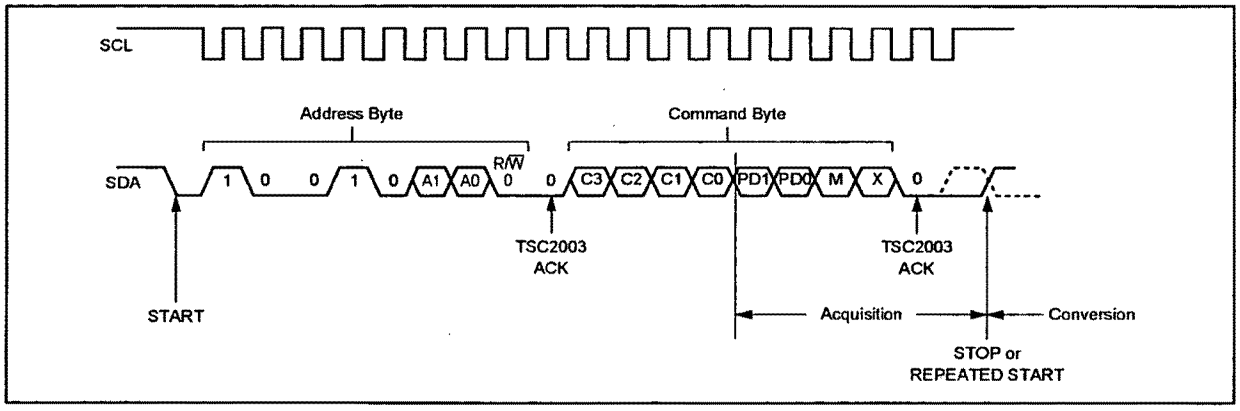


FIGURE 12. Complete I²C Serial Write Transmission.

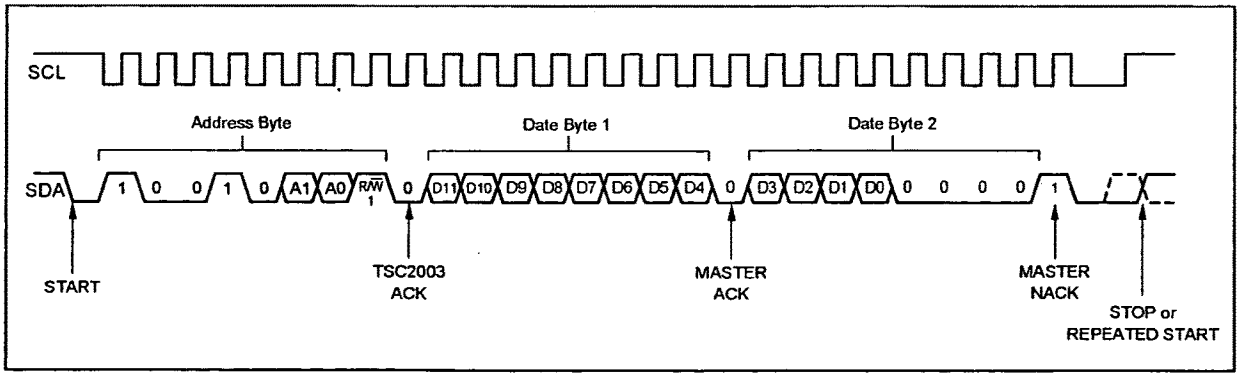


FIGURE 13. Complete I²C Serial Read Transmission.

I²C High-Speed Operation

The TSC2003 can operate with high-speed I²C masters. To do so, the simple resistor pull-up on SCL must be changed to the active pull-up, as recommended in the I²C specification.

The I²C bus will be operating in standard or fast mode initially. Following a START condition, the master will send the code 00001xxx, which the slave will not acknowledge. At this point, the bus is now operating in high-speed mode. The bus will remain in high-speed mode until a STOP condition occurs. Therefore, to maximize throughput only repeated STARTs should be used to separate transactions.

Since the TSC2003 may not have completed a conversion before a read to the part can be requested, the TSC2003 is capable of stretching the clock until the converted data is stored in its internal shift register. Once the data is latched, the TSC2003 will release the clock line so that the master can receive the converted data. A complete high-speed Conversion Cycle is shown in Figure 14.

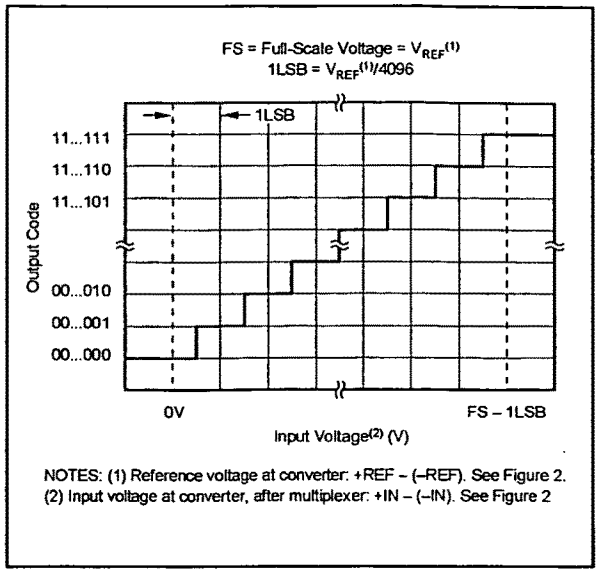
Data Format

The TSC2003 output data is in Straight Binary format, as shown in Figure 15. This shows the ideal output code for the given input voltage, and does not include the effects of offset, gain, or noise.

8-Bit Conversion

The TSC2003 provides an 8-bit conversion mode (M = 1) that can be used when faster throughput is needed, and the digital result is not as critical (for example, measuring pressure). By switching to the 8-bit mode, a conversion result can be read by transferring only one data byte.

This shortens each conversion by four bits and reduces data transfer time which results in fewer clock cycles and provides lower power consumption.



NOTES: (1) Reference voltage at converter: +REF – (–REF). See Figure 2. (2) Input voltage at converter, after multiplexer: +IN – (–IN). See Figure 2

FIGURE 15. Ideal Input Voltages and Output Codes.

LAYOUT

The following layout suggestions should provide optimum performance from the TSC2003. However, many portable applications have conflicting requirements concerning power, cost, size, and weight. In general, most portable devices have fairly "clean" power and grounds because most of the internal components are very low power. This situation would mean less bypassing for the converter's power, and less concern regarding grounding. Still, each situation is unique, and the following suggestions should be reviewed carefully.

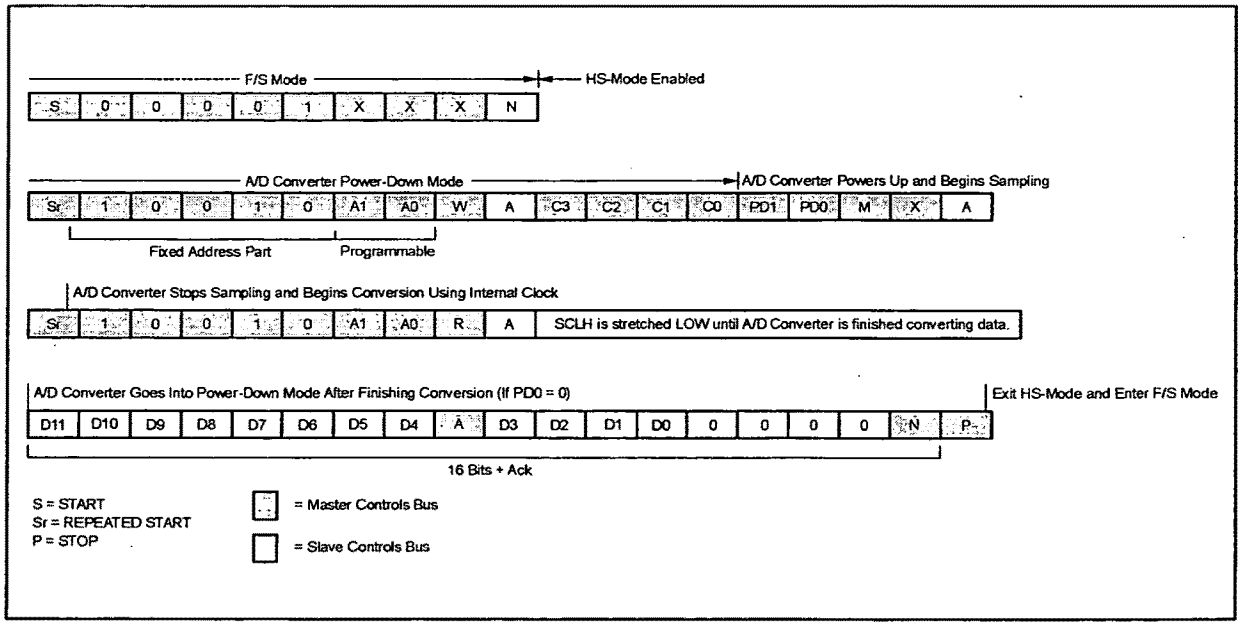


FIGURE 14. High-Speed I²C Mode Conversion Cycle.

For optimum performance, care should be taken with the physical layout of the TSC2003 circuitry. The basic SAR architecture is sensitive to glitches or sudden changes on the power supply, reference, ground connections, and digital inputs that occur just prior to latching the output of the analog comparator. Therefore, during any single conversion for an n-bit SAR converter, there are n "windows" in which large external transient voltages can easily affect the conversion result. Such glitches might originate from switching power supplies, nearby digital logic, and high-power devices. The degree of error in the digital output depends on the reference voltage, layout, and the exact timing of the external event. The error can change if the external event changes in time with respect to the SCL input.

With this in mind, power to the TSC2003 should be clean and well bypassed. A 0.1µF ceramic bypass capacitor should be placed as close to the device as possible. In addition, a 1µF to 10µF capacitor may also be needed if the impedance of the connection between $+V_{DD}$ and the power supply is high.

A bypass capacitor is generally not needed on the V_{REF} pin because the internal reference is buffered by an internal op amp. If an external reference voltage originates from an op amp, make sure that it can drive any bypass capacitor that is used without oscillation.

The TSC2003 architecture offers no inherent rejection of noise or voltage variation in regards to using an external reference input. This is of particular concern when the reference input is tied to the power supply. Any noise and ripple from the supply will appear directly in the digital results. While high-frequency noise can be filtered out, voltage variation due to line frequency (50Hz or 60Hz) can be difficult to remove.

The GND pin should be connected to a clean ground point. In many cases, this will be the "analog" ground. Avoid connections which are too near the grounding point of a microcontroller or digital signal processor. If needed, run a ground trace directly from the converter to the power-supply entry point. The ideal layout will include an analog ground plane dedicated to the converter and associated analog circuitry.

In the specific case of use with a resistive touch screen, care should be taken with the connection between the converter and the touch screen. Since resistive touch screens have fairly low resistance, the interconnection should be as short and robust as possible. Longer connections will be a source of error, much like the on-resistance of the internal switches. Likewise, loose connections can be a source of error when the contact resistance changes with flexing or vibrations.

As indicated previously, noise can be a major source of error in touch screen applications (e.g., applications that require a backlit LCD panel). This EMI noise can be coupled through the LCD panel to the touch screen and cause "flickering" of the converted data. Several things can be done to reduce this error, such as utilizing a touch screen with a bottom-side metal layer connected to ground. This will couple the majority of noise to ground. Additionally, filtering capacitors from Y+, Y–, X+, and X– to ground can also help.

$\overline{\text{PENIRQ}}$ OUTPUT

The pen-interrupt output function is shown in Figure 16. By connecting a pull-up resistor to V_{DD} (typically 100kΩ), the $\overline{\text{PENIRQ}}$ output is HIGH. While in the power-down mode, with PD0 = 0, the Y– driver is ON and connected to GND, and the $\overline{\text{PENIRQ}}$ output is connected to the X+ input. When the panel is touched, the X+ input is pulled to ground through the touch screen, and $\overline{\text{PENIRQ}}$ output goes LOW due to the current path through the panel to GND, initiating an interrupt to the processor. During the measurement cycle for X-, Y-, and Z-Position, the X+ input will be disconnected from the $\overline{\text{PENIRQ}}$ pull-down transistor to eliminate any leakage current from the pull-up resistor to flow through the touch screen, thus causing no errors.

In addition to the measurement cycles for X-, Y-, and Z-position, commands which activate the X-drivers, Y-drivers, Y+ and X-drivers without performing a measurement also disconnect the X+ input from the $\overline{\text{PENIRQ}}$ pull-down transistor and disable the pen-interrupt output function regardless of the value of the PD0 bit. Under these conditions, the $\overline{\text{PENIRQ}}$ output will be forced LOW. Furthermore, if the last command byte written to the TSC2003 contains PD0 = 1, the pen-interrupt output function will be disabled and will not be able to detect when the panel is touched. In order to re-enable the pen-interrupt output function under these circumstances, a command byte needs to be written to the TSC2003 with PD0 = 0.

Once the bus master sends the address byte with $R/\overline{W} = 0$ (see Figure 10) and the TSC2003 sends an acknowledge, the pen-interrupt function is disabled. If the command which follows the address byte has PD0 = 0, then the pen-interrupt function will be enabled at the end of a conversion. This is approximately 10µs (12-bit mode) or 7µs (8-bit mode) after the TSC2003 receives a STOP/START condition following the reception of a command byte (see Figures 12 and 14 for further details of when the conversion cycle begins).

In both cases listed above, it is recommended that the master processor mask the interrupt which the $\overline{\text{PENIRQ}}$ is associated with whenever the host writes to the TSC2003. This will prevent false triggering of interrupts when the $\overline{\text{PENIRQ}}$ line is disabled in the cases listed above.

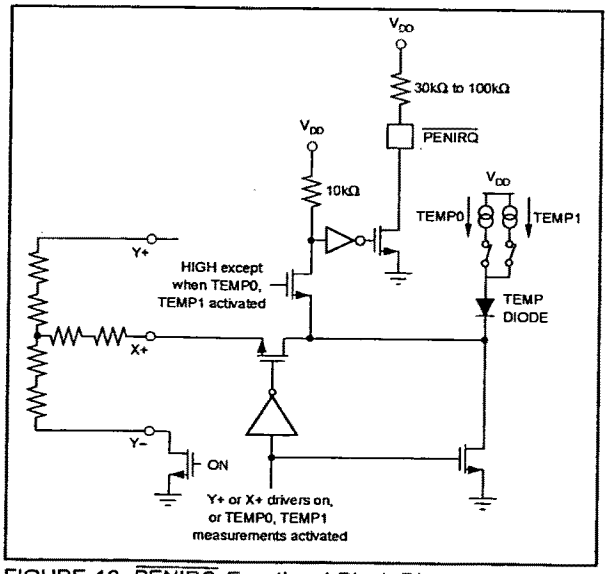


FIGURE 16. $\overline{\text{PENIRQ}}$ Functional Block Diagram.

PACKAGING INFORMATION

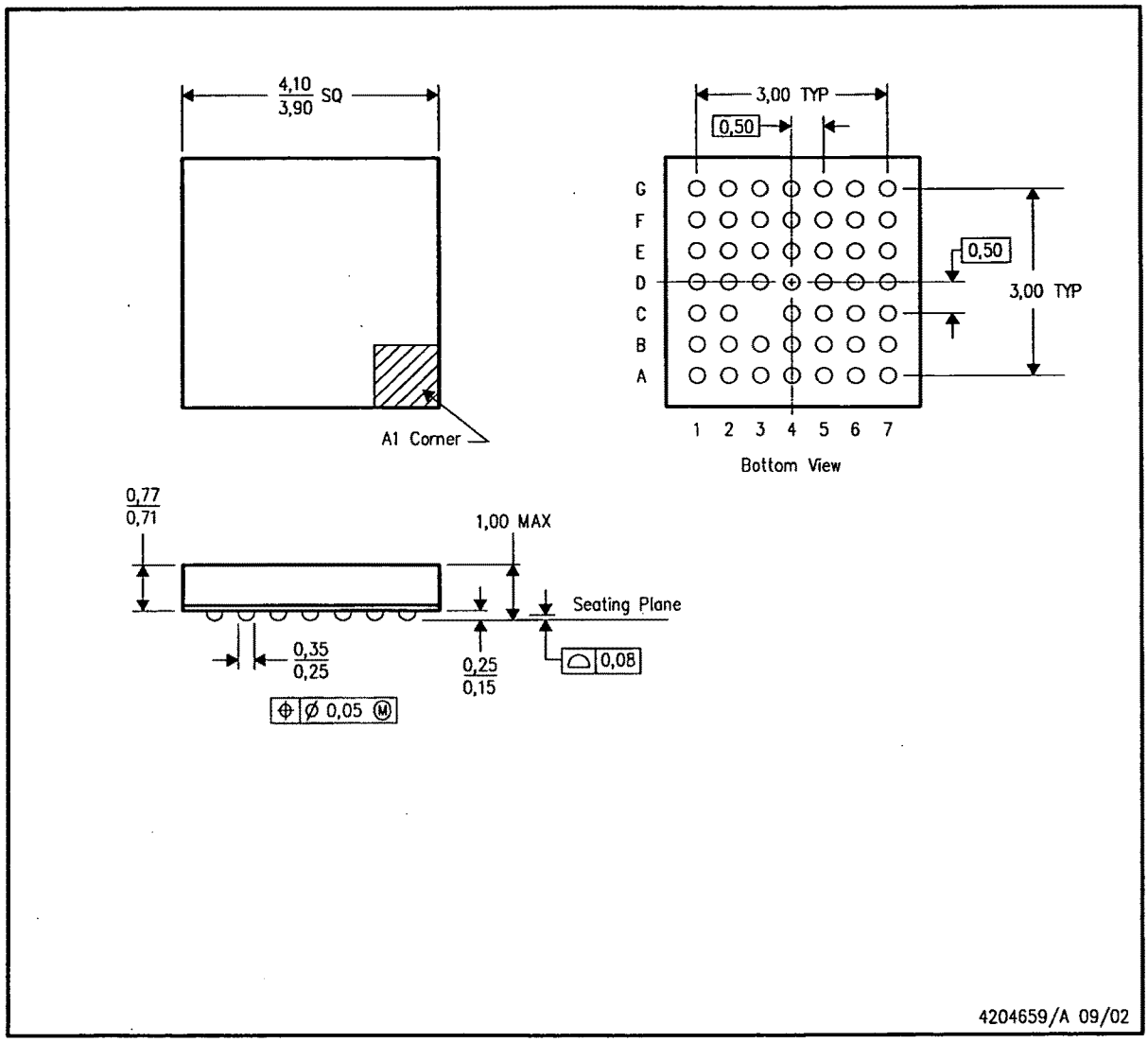
Orderable Device	Status [1]	Package Type	Package Drawing	Pins	Package Qty	Eco Plan [2]	Lead/Ball Finish	MSL Peak Temp [3]
TSC2003IPW	ACTIVE	TSSOP	PW	16	94	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
TSC2003IPWG4	ACTIVE	TSSOP	PW	16	94	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
TSC2003IPWR	ACTIVE	TSSOP	PW	16	2500	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
TSC2003IPWRG4	ACTIVE	TSSOP	PW	16	2500	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
TSC2003IPWT	PREVIEW	TSSOP	PW	48		TBD	Call TI	Call TI
TSC2003IZQCR	ACTIVE	BGA MICROSTAR JUNIOR	ZQC	48	2500	Pb-Free (RoHS)	SNAGCU	Level-3-260C-168 HR
TSC2003IZQCT	PREVIEW	BGA MICROSTAR JUNIOR	ZQC	48		TBD	Call TI	Call TI

[1] The marketing status values are defined as follows:
ACTIVE: Product device recommended for new designs.
LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.
NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.
PREVIEW: Device has been announced but is not in production. Samples may or may not be available.
OBSOLETE: TI has discontinued the production of the device.

[2] Eco Plan - The planned eco-friendly classification: Pb-Free (RoHS), Pb-Free (RoHS Exempt), or Green (RoHS & no Sb/Br) - please check http://www.ti.com/productcontent for the latest availability information and additional product content details.
TBD: The Pb-Free/Green conversion plan has not been defined.
Pb-Free (RoHS): TI's terms "Lead-Free" or "Pb-Free" mean semiconductor products that are compatible with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, TI Pb-Free products are suitable for use in specified lead-free processes.
Pb-Free (RoHS Exempt): This component has a RoHS exemption for either 1) lead-based flip-chip solder bumps used between the die and package, or 2) lead-based die adhesive used between the die and leadframe. The component is otherwise considered Pb-Free (RoHS compatible) as defined above.
Green (RoHS & no Sb/Br): TI defines "Green" to mean Pb-Free (RoHS compatible), and free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material)

[3] MSL, Peak Temp. -- The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

ZQC (S–PBGA–N48) PLASTIC BALL GRID ARRAY



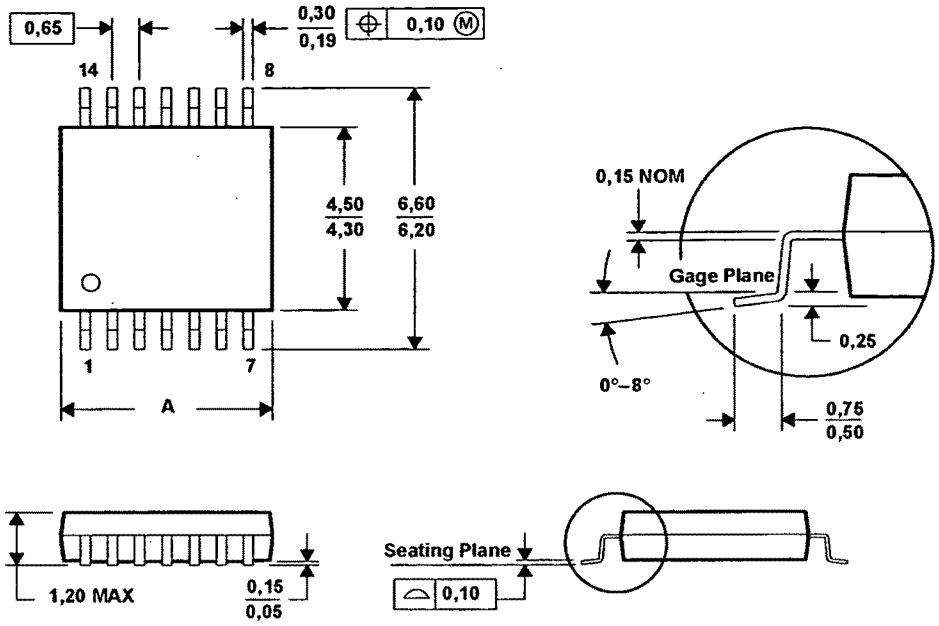
NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. MicroStar Junior™ BGA configuration
D. Falls within JEDEC MO–225
E. This package is lead–free.

MicroStar Junior is a trademark of Texas Instruments.

MTSS001C – JANUARY 1995 – REVISED FEBRUARY 1999

PW (R-PDSO-G)** **PLASTIC SMALL-OUTLINE PACKAGE**

14 PINS SHOWN



DIM \ PINS **	8	14	16	20	24	28
A MAX	3,10	5,10	5,10	6,60	7,90	9,80
A MIN	2,90	4,90	4,90	6,40	7,70	9,60

4040064/F 01/97

NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion not to exceed 0,15.
D. Falls within JEDEC MO-153

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
Low Power Wireless	www.ti.com/lpw	Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265

Integrality and Separability of Input Devices

Robert J.K. Jacob
Linda E. Sibert
Daniel C. McFarlane
M. Preston Mullen, Jr.

Human-Computer Interaction Lab
Naval Research Laboratory
Washington, D.C.

ABSTRACT

Current input device taxonomies and other frameworks typically emphasize the mechanical structure of input devices. We suggest that selecting an appropriate input device for an interactive task requires looking beyond the physical structure of devices to the deeper perceptual structure of the task, the device, and the interrelationship between the perceptual structure of the task and the control properties of the device. We affirm that perception is key to understanding performance of multidimensional input devices on multidimensional tasks. We have therefore extended the theory of processing of perceptual structure to graphical interactive tasks and to the control structure of input devices. This allows us to predict task and device combinations that lead to better performance and hypothesize that performance is improved when the perceptual structure of the task matches the control structure of the device. We conducted an experiment in which subjects performed two tasks with different perceptual structures, using two input devices with correspondingly different control structures, a three-dimensional tracker and a mouse. We analyzed both speed and accuracy, as well as the trajectories generated by subjects as they used the unconstrained three-dimensional tracker to perform each task. The results support our hypothesis and confirm the importance of matching the perceptual structure of the task and the control structure of the input device.

Keywords: Input devices, interaction techniques, gesture input, Polhemus tracker, perceptual space, integrality, separability.

INTRODUCTION

A mechanical input device encodes motion into a signal that can be read by the computer. Designing workable input devices requires that the motion clearly convey the intent of the user and complement his or her physical capabilities. The design of current input devices and their interaction techniques has been driven more by what is technologically feasible than from an understanding of human performance. Their success relies, in part, on the well-documented

human ability to adapt. To design and select more effective input devices and interaction techniques, we need to use a deeper understanding of task, device, and the interrelationship between task and device from the perspective of the user.

Such understanding may come from the intuition and judgment of designers and, perhaps, from empirical studies of specific new devices. However, greater leverage is available by reasoning from a more general predictive theoretical framework, rather than finding an *ad hoc* answer to each such question. The present study provides one example of the development and use of such a theoretical framework to the problem of three-dimensional input devices (as, for example, Card, Moran, and Newell [6] have done for two-dimensional pointing devices).

To do this, we extend the theory of processing of perceptual structure [9, 10], first developed with fixed images, to interactive graphical manipulation tasks. According to this theory, the attributes of objects in multidimensional spaces can have different dominant perceptual structures, *integral* or *separable,* as described in more detail below. The nature of that structure, that is, the way in which the dimensions of the space combine perceptually, affects how an observer perceives an object. We posit that this distinction between perceptual structures is a key to performance of multidimensional input devices on multidimensional tasks. Hence two three-dimensional tasks, such as those in our experiment, may seem equivalent, but if they involve different types of perceptual spaces, they should be assigned to correspondingly different input devices.

Three-Dimensional Tracker

A three-dimensional tracker, such as the Polhemus 3SPACE or Ascension Bird, is a three-dimensional absolute-position locator. In contrast, a mouse is a two-dimensional relative-position locator. The three-dimensional tracker reports its position in three-space relative to a user-defined origin. (In fact, the Polhemus and Ascension devices also report their rotational orientation but we have focused on position only.) The device allows a user to input three

coordinates or data values simultaneously and to input changes that cut across all three coordinate axes in a single operation. Such a device is obviously useful for pointing in three-space, but it is also applicable in any other situation that involves changing three values simultaneously. A mouse, in comparison, requires two operations to manipulate three variables. One commonly-used design for mapping three variables (such as x, y, and z for zooming and panning) onto a mouse allows two of the variables (x and y) to be input simultaneously in normal operational mode and the third (z) to be controlled through a mode change button that temporarily turns the mouse into a one-dimensional slider. A technological view of these two alternatives suggests that the three-dimensional tracker is logically a superset of the two-dimensional mouse (provided they are both used as absolute devices, i.e., the mouse is not lifted from its pad), since it provides the same two outputs plus a third. Thus the three-dimensional tracker should always be used in place of a mouse since it is always at least as good and sometimes better. This also assumes ideal devices (that do not exist) with equal cost and equal accuracy and the absence of other ergonomic differences in such parameters as control-display ratio, non-linearities, instabilities, size, shape, and weight. Nevertheless, our intuition tells us that this is unlikely to be true–but why? Our goal is to base such judgments on a firmer theoretical foundation.

History of Input Device Frameworks

From the inception of interactive computer graphics, researchers have proposed a number of frameworks that organize knowledge about input devices to simplify the job of selecting an appropriate device for a task. The earliest works focused on equating the physical properties of different input devices to minimize the programming effort in substituting one device for another. Later systems added knowledge about device pragmatics and task requirements to better capture their qualities and use.

An early abstraction is that of logical device found in device-independent graphics packages developed from standards such as ACM's Core Graphics System [11]. Devices were assigned to

logical device classes based on how they performed fundamental user actions. Foley, Wallace, and Chan's taxonomy of interaction techniques improved upon this concept by adding a middle layer that made explicit the fact that more than one interaction technique may perform a given elementary task [8]. However, both approaches considered, for example, selecting with a joystick or trackball equivalent. They hid the crucial pragmatic aspects of haptic input by treating devices that output the same information as equivalent, despite the different subjective qualities they present to the user.

Buxton recognized the importance of these qualitative differences which he called pragmatic attributes [5]. He developed a taxonomy that organized continuous input devices by property (position, motion, pressure) and the number of dimensions. A tablet, light pen, and two-dimensional joystick have two-dimensions and sense position, but they differ from a two-dimensional trackball because it senses motion. While this approach can point out that substituting a trackball for a joystick is not equivalent, it does not explain why.

Mackinlay, Card, and Robertson expanded Buxton's taxonomy to include most input devices and complex controls [7, 15]. One feature of their approach is an evaluation technique for comparing alternative designs in terms of expressiveness (is the meaning exactly conveyed) and effectiveness (is the meaning conveyed with felicity). Their approach both furthers our understanding of the structure of input device space and recognizes that human performance issues are important to understanding how a device actually works in a given situation. Although some relatively straightforward human factors issues are handled formally, such as matching the size of the domain and range of a value, the more subtle pragmatics of input device usage and task characteristics are still handled by a set of specific rules.

Bleser developed a device taxonomy and input model that explicitly incorporated the physical attributes of input devices, including the notion of the physical separability of input degrees of freedom, and knowledge about task requirements [4]. The taxonomy and model were

used in an interactive design tool to suggest one or more natural interaction techniques based on a description of an interaction task [3]. The model introduces a set of heuristic rules and a pattern matching procedure rather than a more general, theoretical framework, but it highlights the need for such information in the design process.

Therefore, while current frameworks include knowledge about physical structure, pragmatics, and some task requirements, it is not enough to explain why the three-dimensional tracker is not always better. We suggest that what is also needed is to move from an *ad hoc* understanding of input devices and task requirements to the added leverage of incorporating a predictive theoretical framework that allows reasoning about the utility of a device for a particular task. To help put the study of multidimensional input devices on a firmer theoretical footing, we extend the theory of processing of perceptual structure to interactive graphical manipulation [9, 10]. Our research hypothesis is that performance improves when the perceptual structure of the task matches the control structure of the device. To test our hypothesis, we conducted an experiment in which subjects performed two tasks that have different perceptual structures, using two input devices with correspondingly different control structures, a three-dimensional tracker and a mouse. The data analysis examined several aspects of performance: task completion time, time to reach an absolute accuracy criterion, and differences between the trajectories generated by subjects using the three-dimensional tracker as they completed each task. The results converge to confirm the utility of matching the perceptual structure of the task and the control structure of the input device.

BACKGROUND

This work merges two separate threads of research. One, described above, is the understanding of input devices in human-computer interaction, from the logical device concept through more recent taxonomies and tools. The other gives insight into task and performance issues; this is the theory of the processing of perceptual structure of a multidimensional space

formed by the attributes of an object.

A multidimensional object is characterized by its attributes. A small red circle has size, color, shape, and location. Investigations into spatial structures that describe the perception of such attributes [2, 16] led Garner [10] to observe the attributes of some visual objects combine perceptually to form a unitary whole, while those of other objects remain more distinct and identifiable. Attributes that combine perceptually are said to be *integral;* those that remain distinct are *separable.* For example, value (lightness) and chroma (saturation) of a color are perceived integrally, while size and lightness of an object are perceived separably [12]. The horizontal and vertical positions of a single dot in the middle of an outline square are integral [10], but color and shape are separable [13].

Integral and separable define two classes of perceptual structure that mark the endpoints of a continuum rather than forming a sharp dichotomy. Where an object falls can be determined by two operational methods. First, integral and separable objects can be distinguished by direct similarity scaling, a technique that measures the perceived similarity among objects. In this procedure, subjects are asked to compare pairs of objects, and derive a value that indicates how alike they are overall. These measures are then compared to the actual Euclidean and city-block distances in the attribute space. Integral objects are those related by the Euclidean metric (distance in Euclidean space); subjects' judgements are based on the overall sameness of the objects. Separable objects are related by the city-block metric which is the sum of the distances measured parallel to the coordinate axes (the distance one would have to travel in a city in which all streets run in an x or y direction). Their distance is the *sum* of distance in each attribute. Second, when asked to form classes of objects that have the same set of attributes (for example, that vary in shape and color), observers will partition an integral set into clusters based on their overall similarity, but a separable set will be partitioned by attribute. Integral objects are said to possess similarity structure, and separable objects dimensional structure. With similarity

structure, individual attributes do not dominate; with dimensional structure, the attributes cannot be ignored.

Perceptual Structure Extended to Interaction

The notion of integral and separable can be extended to interactive tasks by observing that manipulating a graphical interaction object is simply changing the values of its attributes. Since the attributes of an object define a perceptual space, changing these values is the same as moving in real-time within the perceptual space of the object. Because integral objects have a similarity structure and follow the Euclidean metric, and separable objects have a dimensional structure and obey the city-block metric, the nature of the interaction path can be predicted: movement in an integral space should be Euclidean, straight-line distance between two points; movement in a separable space should be city-block and run parallel to the axes. In other words, graphical interactive tasks have a perceptual structure that influences how they are completed. Moreover, the type of perceptual structure is the same as the type of perceptual structure of the underlying object.

Control Structure of Input Devices

The control spaces of input devices have similar characteristics. An input device with more than one degree of freedom can be characterized as integral or separable based on whether it is natural (or possible) to move *diagonally* across dimensions. With an integral device, movement is in Euclidean space and cuts across all the dimensions of control. A separable device constrains movement to a stair-step, city-block pattern; movement occurs along one dimension at a time.

METHOD

Our research hypothesis states that performance improves when the structure of the perceptual space of an interaction task mirrors that of the control space of the input device. To test its validity, we examined two interactive tasks, one set within an integral space and one in a

separable one, and two devices, one with integral control and one, separable. This yields a two by two experiment with four conditions. We predict performance on each task will be superior in the condition where the device matches the task in integrality/separability. That is, the interaction effect between task and device should far exceed the main effects of task or device alone. We also predict that the perceptual structure of the task determines how the subject manipulates the three-dimensional tracker; the path to target should be more nearly Euclidean with the integral task than the separable one.

We chose two tasks that require adjusting three variables but concentrated on task spaces that do not directly map to three-dimensional physical space and on manipulation rather than pointing tasks, in order to frame a balanced comparison. The integral three-attribute task required changing the *x-y* location and the size of an object to match a target. The separable task required changing *x-y* location and the color (lightness or darkness of greyscale). Studies of non-interactive stimuli suggested that position and size are perceived as integral attributes, while position and color are perceived as separable [10]. For the integral device, we used the Polhemus tracker which permits input of three integral values. For the separable device, we used a conventional mouse which permits two integral values, to which we added a mode change to enable input of a third, separable, value. Our research hypothesis asserts that the three degree of freedom input device (Polhemus) will be superior to the two degree of freedom plus mode change device (mouse) when the task involves three integral attributes (location/size) rather than two integral plus one separable attribute (location/greyscale). Conversely, the separable mouse plus mode change device will be superior for the separable (location/greyscale) task.

Design

Each subject performed both tasks using both devices. constituting a repeated measures design. Forty subjects (26 men and 14 women) were randomly assigned to four presentation orders. In each order, one task was completed with both devices before the second task was

presented to control for practice and fatigue with a device. Each order had a unique sequencing and together the four orders formed a latin square design so that each device and task combination appeared only once at each position in the sequence. Subjects were clerical, administrative and technical personnel from the Information Technology Division of the Naval Research Laboratory who volunteered to participate without compensation. All but two were right-handed. Only two had tried a Polhemus previously. Thirty-seven reported using a mouse daily, two weekly, and one monthly.

Stimulus and Apparatus

On each trial, the subject adjusted a user-controllable object until it matched the location and either size or greyscale level of a target. For the size task, the user-controllable object was a square whose greyscale level was 50% of the total greyscale range and whose location and size could be adjusted (see Figure 1). The target position and size were represented by a black outline square. The range of sizes was 0.7 to 6.2 inches on a side. For the greyscale task, the user-controllable object was a square of size 2.8 inches (the midpoint of the size range) that contained an embedded circle 1.5 inches in diameter (see Figure 2). The greyscale level of the circle was under user control and the target greyscale level was displayed in the outer region of the user-controllable object to facilitate matching. The target position was a black outline square. In both tasks, the user-controllable object was translucent so it never obscured the position target, the background was white to maximize the range of remaining available intensities that the human vision system can distinguish, and the monitor gamma correction was set at a level that enhanced the greyscale range of the monitor.

The maximum size of the space of possible three-variable targets is a cube 13.75 inches on a side, the width of the monitor. The restrictions we imposed on size and greyscale cut down this space differently for each task. We required that the entire target square fit within the 13.75 inch by 11 inch screen display area and that a target not be closer than 5% of the screen height from

the edge of the screen to allow for overshooting the target and to eliminate any conflict between the edge of the target and the edge of the screen, and we reserved the greyscale levels near black and white for the target position outline and screen background, cutting the range of allowable intensities by 20%. After taking these restrictions into account, the size target space became a rectangular frustrum with more small than large targets; and the greyscale space, a balanced rectangular prism. We made further adjustments to the two spaces so that they had equal volumes with equal number of targets (giving each target the same probability of begin selected), and so that each had a reasonable central value (the greyscale midpoint is the middle of the intensity range, the size midpoint is the centroid of the unbalanced frustrum), stimulus range, and equal range of hand movement.

A different stimulus set was randomly generated for each of the four conditions but constrained by a script so that the total three-dimensional distance between one stimulus and the next was the same across conditions. Corresponding trials thus had equal distances but different absolute locations. We also required each stimulus to differ from the preceding one by at least 0.5 inches in each dimension to avoid degenerate trials that could be completed without exercising all three dimensions of motion. Each stimulus trial was represented in the computer program as a three-dimensional point *(x,y,z)* and the software driving the experiment was identical except for how the *z* dimension was displayed on the monitor. The *x* and *y* dimensions of both tasks were mapped as position, but the *z* was mapped as either size or greyscale level.

The three-dimensional tracker used in the experiment was the Polhemus 3SPACE magnetic tracker. It consists of a transmitter that generates electromagnetic fields and a wand housing three orthogonal coils that sense the fields. The position and orientation of the wand is transmitted to the host computer. In this experiment, only the three position values were used. The Polhemus was configured as an absolute device with the origin at what would be the forward end of the arm of the chair. The source was permanently fixed under the subject's chair. The control-display

ratio was one inch of device movement to one inch of screen movement. To improve the performance of the Polhemus, the raw position data was smoothed with a small moving average filter, and we eliminated as much metal as possible from the surrounding area to reduce interference with the electromagnetic fields. The experiment was located away from electrical wiring and metal poles, the furniture was wooden, and the metal mouse pad was removed when not in use. The Polhemus was initialized from one of two precalibrated files (one for left-handed use, one for right) so that the operating area was consistent across subjects and its axes were orthogonal. Movement of the wand in a plane parallel to the screen moved the user-controllable object in x and y. Moving the wand toward the screen made the object either bigger or darker, away made it smaller or lighter.

The mouse was a standard relative-position optical mouse with its control-display ratio set to minimize the need for stroking (picking up and repositioning the mouse while moving the cursor over long distances) in order to make its behavior close to that of an absolute position device like the Polhemus. Two of the three input values needed were the standard x and y mouse coordinates. The third was a mode change into a one-dimensional slider, in which the mouse was moved along the vertical axis of the pad while any of the three mouse-buttons was depressed. We chose this strategy because it is similar to those currently used in applications that require mouse input of more than two variables. Movement over the optical pad corresponded to moving the user-controllable object in x and y. Movement while holding the mouse button corresponded to changing the size or lightness of the user-controllable object; movement toward the top of the optical pad (toward the display screen) made the object either bigger or darker, toward the bottom made it smaller or lighter.

The computer was a two-processor (16 MHz) Silicon Graphics Iris Workstation, model 4D/120G. The program was divided into two processes, which ran concurrently on the two processors. Most other system processes and all network daemons were eliminated. One of our

two processes continuously monitored the Polhemus over a serial port and fed data into an event queue in shared memory, while the other drew the images and supervised the experiment. This architecture was effective in greatly reducing the often-observed lag in response to movements of the Polhemus because handling of the heavy serial port data transfer from the Polhemus was performed by a separate processor, communicating via shared memory. Position data from the mouse or Polhemus were recorded continuously throughout the experiment, approximately every 20 milliseconds. The monitor was a 19-inch Hitachi and the gamma correction was set at 2.7.

Procedure

The arrangement of the apparatus is shown to scale in Figure 3. The subject was seated on a straight-backed chair in front of a monitor that was placed on a standard office desk. The subject was approximately 56 inches from the monitor, a distance that gave the subject ample room to manipulate the Polhemus and reduced the interference between the electromagnetic fields of the monitor and the Polhemus. The mouse pad was placed on a 27.5 inch high table located under the subject's preferred hand. The indicator button was located on another 27.75 inch high table under the subject's other hand. The tables and chair were located in the same position for each subject. The subject, however, was allowed to adjust the position of the mouse pad and the button upon the table tops for comfort. The experimenter sat at a terminal to the left rear of the subject. The experiment was conducted in a special purpose laboratory designed for such work [1].

Subjects were required to designate a preferred hand and use it to manipulate both devices. Information about handedness and choice of preferred hand was recorded, as was gender of the subject. The information was used to balance the assignment of subjects to experimental conditions to control for variation caused by gender differences and handedness.

The experiment was divided into four presentation orders, one for each device and task combination. Within each order, the subject was first given 33 practice trials to stabilize

performance before beginning 88 data trials. The data trials were subdivided into sets of 11, with the first in each set not scored because it measured the time to home the user-controllable object from an unknown starting position. The home position was located in the middle of the screen with the object being either mid-sized or mid-colored. Each trial required that the subject change the location and either size or greyscale color of the user-controllable object using one of the two devices until it matched the position and size or color of the target. The subject indicated that a match was complete by pushing the indicator button. It would have been easy to have the computer terminate each trial automatically, as soon as a match of sufficient accuracy was achieved. However, as discussed below, we wanted to collect data that would allow us to investigate a wide range of different accuracy criteria through a retrospective analysis procedure, and thus we did not want the experimental procedure to constrain the choice of the criterion in advance.

Within a set of trials, as soon as a subject pushed the indicator button, that trial was ended and another target presented for the next trial. An instruction screen separated the sets of trials, and the subject determined when to start a new set. Subjects were instructed to rest as long as they needed between sets of trials. Subjects were encouraged to ask questions during practice but not during the data trials. They were instructed that accuracy and speed were of equal importance. Subjects were asked to complete a short questionnaire at the conclusion of the experiment to learn their opinions about the devices and tasks and about their prior experience with selected input devices. Each subject took approximately 1.5 hours to finish the experiment.

RESULTS AND DISCUSSION

The data analysis is in three parts. The first looks at the time and accuracy when the subject ended the trial by pressing the indicator button, that is, when the subject thought he or she was satisfied with the match. The second examines time to reach a fixed absolute criterion (before the very end of the trial); this combines time and accuracy into a single measure. The last analyzes

the difference between the trajectories that subjects followed with the Polhemus as they completed each task. To facilitate the latter two analyses, we recorded the position of the mouse or Polhemus approximately every 20 ms. during each trial. All processing and aggregation was conducted after the data collection part of the experiment was completed. The analysis of questionnaire data can be found in [14].

Time to End of Trial

The first analysis looked at time and accuracy when a subject ended a trial. Time was measured from the appearance of the target until the subject pressed the indicator button. Accuracy was total Euclidean distance across all three dimensions of the stimulus space (x, y, and size or lightness) from the center of the user-controllable object to the center of the target. The research hypothesis predicts a strong interaction effect between task and device; the two task and device combinations in which both are integral or separable should give superior performance.

Average (mean) time to button press, shown in Figure 4 and graphed in Figure 5, suggests that, as predicted, neither task nor device alone produced as large an effect as the interaction of the two. These observations were evaluated with a repeated-measures analysis of variance. Order of presentation was not significant ($F(3,39) = 1.09$, $p > .30$), allowing aggregation of the four orders. There were significant effects for both task ($F(1,39) = 7.40$, $p < .01$) and device ($F(1,39) = 7.80$, $p < .01$) and a highly significant effect for interaction between task and device ($F(1,39) = 69.34$, $p < .0001$). An omega squared analysis indicated that 20% of the total variance was accounted for by this interaction; this is high given the large variation from individual differences.

The accuracy results, shown in Figure 6, exhibit a strong task difference, with subjects less accurate on the greyscale task. (The small device effect is confounded by a task and device interaction and so was not considered.) These observations were also evaluated with a repeated-measures analysis of variance. Order of presentation was again not significant ($F(3,36) = 2.29$, p

> .09), allowing aggregation. The task difference was highly significant ($F(1,39) = 117.11$, $p < .0001$). Both device ($F(1,39) = 11.32$, $p < .01$) and a task and device interaction ($F(1,39) = 6.52$, $p < .05$) were present. Although the mean scores differed by task, the variances of their accuracies are approximately the same, indicating a stability in a subject's underlying judgement process.

Since the button press indicates when a subject thinks the match is good enough, performance time is linked to the subject's own maximum accuracy criterion. The data showed that although the subjects selected a wide range of criteria, they used a consistent stopping criterion across tasks, as indicated by the standard deviations of the accuracy results. The strong interaction effect in performance times and the high percentage of the variance accounted for by the interaction strongly support the conclusion that matching integrality or separability of the device to that of the task provides an advantage. A common denominator to improving performance is matching the structure of the device and task.

Time to Absolute Criterion

The second analysis measured the time required by subjects to reach a fixed accuracy criterion on each trial. This can be viewed as simulating an experiment in which the subject was required to reach a certain accuracy, at which point the trial was terminated automatically. This approach combines speed and accuracy into a single measure and removes the effect of an individual subject's *personal* accuracy criterion for terminating trials. It also allows removing the effect of a speed-accuracy tradeoff, since performance to a fixed accuracy criterion is required. Accuracy was, again, the overall Euclidean distance to the target in the three-dimensional space of stimuli.

The raw position data, recorded approximately every 20 ms. during each trial, were transformed using linear interpolation into a time series with a 10 ms. period. A retroactive data analysis algorithm then calculated the *last* time a subject reached a given criterion on each trial.

Simply setting an accuracy cut-off would underestimate trial completion time by terminating a trial when a subject just nicks a target distance unintentionally while overshooting the desired accuracy. With our retroactive analysis, however, time is measured until the subject reached our criterion for the last time during a trial, a better approximation of performance to that distance from the target.

Retroactive analysis also allowed us to produce a series of simulated experiments, each having a different accuracy criterion. Each such simulated experiment is a snapshot of performance to a selected distance from the target. The results from each experiment are the average performance times for the four conditions. Two hundred evenly-spaced distances from the target were chosen in order to investigate how performance changed as the distance to the target decreased (i.e., as accuracy increased). These 200 experiments cover the full range of behavior, from distances that were too far from the target to be considered even a crude match, to distances that required the match to be very good. As noted, we could have run any *one* of these experiments by terminating trials automatically at a fixed accuracy criterion. The retrospective analysis, however, allows us to simulate many different experiments from the same data. (The simulation is not entirely precise, since subjects' behavior would have been somewhat different under different stopping criteria.)

The analysis involves two stages. First, the average performance of all subjects was computed for each task and device combination at each criterion distance from target. A plot of these for criteria near the target is shown in Figure 7.

Next, the four averages were ranked according to completion time. The bar drawn below, in Figure 7, highlights the region in which the rankings follow our predictions, that is, for the size task, where Polhemus always beats mouse and for the greyscale task, where mouse always beats Polhemus. This region contains all the accuracies that one would consider successful completions of the task, from very inaccurate task performance at the distance 0.24 inches

(Figure 8 illustrates a 0.24 inch match) down to an almost perfect match. The results from several criteria were analyzed further to evaluate the significance of these findings, and the results were all similar. We present results for 0.099 inches, an accuracy reached on 70% of all trials. These results are shown in Figure 9 and graphed in Figure 10. These observations were evaluated with a repeated-measures analysis of variance. Neither task ($F(1,39) = 2.48$, $p > .12$) nor device ($F(1,39) = 3.18$, $p > .08$) was significance but the interaction of task and device ($F(1,39) = 52.46$, $p < .0001$) was highly significant as predicted. Omega squared indicated that 22% of the variance was accounted for by this interaction. In contrast, task (1%) and device (2%) accounted for very little. The highly significant interaction supported by the high percentage of variance accounted for by the interaction again supports the hypothesis that matching task and device in integrality or separability leads to better performance.

The rankings also contain a second region of interest, further from the target. Although the distance to target in this region is too great to be considered a match, the results shed light on the process of completing a match. This region begins at approximately 3.65 inches from target and ends at the 0.24 inch mark (the distance at the right end of the bar in Figure 8). In this early behavior, for each task, the Polhemus always performed better than the mouse (although still better on the integral size task than the separable greyscale one). That is, for very crude, *quick-and-dirty* matches, the Polhemus was always faster. Again, several points in the range were examined and the data for 0.987 inches is presented here. The results are shown in Figure 11 and graphed in Figure 12. A repeated-measures analysis of variance was again conducted. Both task ($F(1,39) = 143.23$, $p < .0001$) and device ($F(1,39) = 97.60$, $p < .0001$) were highly significant, and there was no interaction ($F(1,39) = 0.83$, $p > .37$). An omega squared analysis showed that 35% of the variance was accounted for by task and 32% by device in this region.

Trajectory Analysis

Our research hypothesis implies that the nature of the task influences how it is perceived which, in turn, alters how the task is performed. To examine this notion, we investigated whether a plot of the trajectory taken by the subject performing a match reflects the perceptual structure of the task. The theory of perceptual structure suggests the shape of the trajectory for each task type; that is, integral stimuli exhibit an Euclidean pattern (subjects' movement cuts diagonally across the dimensions) and separable stimuli, a city-block pattern (movement occurs along one dimension at a time). Specifically, in our tasks, if z (size or greyscale) is traversed along with x and/or y (position), then the trajectory follows an Euclidean pattern. If z is manipulated separately from x and y in a stair-step manner, then it follows a city-block pattern.

To demonstrate this requires two conditions, both of which were present in the two Polhemus conditions of this experiment. First, the subject's range of motion must not be constrained; both tasks must be performed with a device that allows freedom to move in any direction. The Polhemus meets this requirement. In contrast, the combination of two-dimensional mouse movements and one-dimensional slider does not because it constrains the subject to city-block movement by not allowing all dimensions to be manipulated concurrently. Second, the tasks should differ only minimally to isolate key differences. In this experiment, the software driving the two tasks was identical except for how one parameter was visually presented. In both tasks, the x and y dimensions were displayed as position coordinates. The difference lay solely in how the z dimension was displayed, as either size or greyscale.

The pictures in Figures 13 and 14 show the average performance of all subjects on one selected trial out of the 80 total scored trials. Like-numbered trials in each condition had the same distance and so could be compared. The trial presented is one of the longer ones (5.885 inches out of the range 2.946 inches to 6.875 inches) and one of the last completed in a session (number 74 out of 80). This trial was selected because it clearly illustrates the observed

difference in subject performance between tasks. The Polhemus trajectory for the size task on this trial (Figure 13) suggests that the subject was cutting across all three axes. In contrast, the greyscale task (Figure 14) was completed first by resolving the discrepancy in x and y followed by the discrepancy in z. In other words, we do not see simultaneous motion in the x-y plane and along the z axis. These tasks differed only in how z was displayed. The perceptual structures of the tasks were thus strong enough to alter how subjects performed them physically. Other trials that were sampled exhibited a similar but less pronounced effect, that is, smooth, free-form, integral characteristics with the size task and stair-step, separable characteristics with the greyscale task.

While Figures 13 and 14 visually suggest the effect of perceptual space on trajectory for a selected trial, we wish to measure it in a more objective way, over all trials. We therefore developed a technique to quantify how a subject manipulated the Polhemus during a trial. The idea is to divide a trajectory into small segments, ask if the movement in each segment is Euclidean or city-block, and then tally the proportion of Euclidean versus city-block segments. The procedure involves four steps:

- First, the raw event data series was transformed into a time series with a 10 ms. period. It was then smoothed using a 15-point low-pass filter to remove high-frequency equipment noise and hand tremor.
- Second, the data were truncated to isolate the area of interest. A subject's behavior on a trial typically divides into two parts: strong, quick movements toward the target followed by back-and-forth fine-tuning behavior. The latter can be seen in the small *knot* at the left in Figure 13; the view angle in Figure 14 obscures this behavior. We truncated each trial at 0.3 inches from the target to remove this highly variable end part. (As discussed below, the sensitivity of our results to this specific choice of cutoff value was very small; other choices yielded similar results.)

- Third, the data for a trajectory were segmented, and each trajectory was passed through a classification algorithm that labeled the dimensions as having changed location (movement) or not (no-movement). A position change of more than 0.0008 inches in one 10 ms. time step was considered movement. (Again, as discussed below, the sensitivity of the results to the choice of this threshold value was small.)
- Fourth, a recognition algorithm computed the amount of Euclidean and city-block movement used to complete that trial. A segment was classified as Euclidean if its trajectory showed movement in more than one dimension (except for movement in the *x*-*y* plane, since the two tasks are both integral in this respect); it was classified as city-block if it showed movement in any single axis or in the *x*-*y* plane. The ratio of Euclidean to city-block segments was then calculated for each of the experimental conditions.

For the selected criterion of 0.3 inches and threshold of 0.008 inches, the average ratio of Euclidean to city-block behavior was 1.408 for the size task versus 1.234 for the greyscale task. A one-tailed paired comparison t-test showed a highly significant difference ($t(39) = 4.297$, $p < .0001$), supporting our hypothesis that the size task is completed in a more integral manner than the greyscale task. We performed a sensitivity analysis to give confidence in our choice of the two parameters, criterion from the truncation step and threshold from classification by computing a t-value for a range of criterion and threshold pairs. A series of one-tailed paired comparison t-tests of these pairs was significant throughout the range 0.217 to 0.38 (criteria) and 0.003 to 0.015 (threshold).

The results confirm that the way in which a user physically manipulates the Polhemus to complete a task depends on the perceptual structure of the task. The integral task induces more motion that cuts across all three dimensions. The separable task alternates more between changes to location and changes to greyscale.

APPLICATION

An example application of this work would be in designing controls for zooming and

panning of a geographic display. Zooming and panning, taken together, involve three degrees of freedom. The most common design uses a mouse or trackball for two-dimensional panning and a separate control for zooming. We claim that a user typically does not really think of zooming or panning operations separably, but thinks rather of integral operations like "focus in on *that* area over there." The space is thus Euclidean, like that of the size task in the experiment, and, therefore, making the user do the two separately violates perceptual compatibility. It would be more natural to permit a user to make a gesture that performs the overall operation he or she had in mind, using an integral three-dimensional input device. The user moves the puck around in a volume directly in front of the display screen. Moving it in the x or y direction parallel to the display surface causes panning; moving it perpendicular to the display (directly toward or away from it) causes zooming. The user typically moves the puck in all three dimensions simultaneously, resulting in some combination of zooming and panning and directly reaches the view of interest. We have demonstrated a mockup of this application.

CONCLUSIONS

Our research hypothesis is that performance improves when the structure of the perceptual space of a graphical interaction task mirrors that of the control space of the input device. We have presented an experiment to test this hypothesis and three different analyses in support. The first looked at time and accuracy at button press. Button press captures a subject's behavior when he or she is satisfied with the match. These outcome measures describe terminal actions and give insight into performance at a subject's maximum accuracy criterion. The results support our hypothesis that completing the integral size task with the integral Polhemus and the separable greyscale task with the separable mouse plus mode change lead to faster performance. The second approach removes the variation caused by individual differences in accuracy as well as speed-accuracy tradeoff and allows us to simulate a range of different experiments retroactively. We studied performance to a series of fixed criteria that combine time and accuracy into one

measure. This approach allowed us to investigate performance at a number of criteria to determine the stability of performance over time. We found that within the range of plausible matches, completion times are faster when the structure of the device and task match rather than for one device or one task uniformly. The third analysis examined the trajectory of the Polhemus on both tasks. The research hypothesis suggests that if the device used by the subject is not restrictive, the path taken will be influenced by the structure of the perceptual space of the task. Our analysis confirms that the perceptual structure of the task drives how an input device is used.

The overall conclusion confirms our prediction that choosing an input device for a task requires looking at the deeper perceptual structure of the task, the device, and the interrelationship between task and device. We have shown that perceptual structure determines how a user approaches an interactive manipulation task. Tasks in which the perceptual structure is integral operate by similarity and follow the Euclidean metric. The attributes of an integral object do not dominate and are viewed as a unitary whole. They are manipulated as a unit. Tasks set in a separable space operate by dimensional structure and obey the city-block metric. The attributes of a separable object cannot be ignored and are manipulated along each attribute in turn. If the input device supports the type of motion required by the task, then the task can be performed in an efficient manner. If the device limits necessary motion or does not restrict motion appropriately, efficiency can decrease. The interplay between task and device was more important in determining performance than either task or device alone.

Current input device taxonomies and other frameworks developed to aid device selection have typically started from the point of view of input device structure. They recognize the need for what Buxton calls pragmatics, Mackinlay, Card, and Robertson term expressiveness and effectiveness ratings, and Bleser includes in her input model, but they relegate these crucial pragmatic aspects of haptic input to rules based primarily on *ad hoc* testing or expert judgement. We suggest incorporating a predictive theoretical framework to allow formal reasoning about

selecting an input device. We offer the approach of extending a perceptual theory and provide as an example our extension of the processing of perceptual structure to interactive stimuli.

ACKNOWLEDGMENTS

We thank Jim Ballas, Susan Kirschenbaum, and Astrid Schmidt-Nielsen for help and advice, particularly in experimental design and data analysis; Lisa Achille, Jeff Brown, Robert Carter, Dave Heide, Connie Heitmeyer, John Sibert, and Stan Wilson for all kinds of help with this research; our NRL colleagues who took time from their own work to serve as experimental subjects; and Stuart Card and the anonymous *TOCHI* referees for their thoughtful reading and helpful comments on this paper. This work was sponsored by the Office of Naval Research.

REFERENCES

1. L.B. Achille, "Considerations in the Design and Development of a Human Computer Interaction Laboratory," NRL Report 9279, Naval Research Laboratory, Washington, D.C. (1990).

2. F. Attneave, "Dimensions of Similarity," *American Journal of Psychology* **63** pp. 516-556 (1950).

3. T.W. Bleser and J.L. Sibert, "Toto: A Tool for Selecting Interaction Techniques," *Proc. ACM UIST'90 Symposium on User Interface Software and Technology* pp. 135-142, Addison-Wesley/ACM Press, Snowbird, Utah (1990).

4. T.W. Bleser, "An Input Device Model of Interactive Systems Design," Doctoral Dissertation, The George Washington University (May 1991).

5. W. Buxton, "There's More to Interaction Than Meets the Eye: Some Issues in Manual Input," pp. 319-337 in *User Centered System Design: New Perspectives on Human-computer Interaction*, ed. D.A. Norman and S.W. Draper, Lawrence Erlbaum, Hillsdale,

N.J. (1986).

6. S.K. Card, T.P. Moran, and A. Newell, *The Psychology of Human-Computer Interaction*, Lawrence Erlbaum, Hillsdale, N.J. (1983).

7. S.K. Card, J.D. Mackinlay, and G.G. Robertson, "A Morphological Analysis of the Design Space of Input Devices," *ACM Transactions on Information Systems* 9 pp. 99-122 (1991).

8. J.D. Foley, V.L. Wallace, and P. Chan, "The Human Factors of Computer Graphics Interaction Techniques," *IEEE Computer Graphics and Applications* 4(11) pp. 13-48 (1984).

9. W.R. Garner and G.L. Felfoldy, "Integrality of Stimulus Dimensions in Various Types of Information Processing," *Cognitive Psychology* 1 pp. 225-241 (1970).

10. W.R. Garner, *The Processing of Information and Structure*, Lawrence Erlbaum, Potomac, Md. (1974).

11. GSPC, "Status Report of the Graphics Standards Planning Committee," *Computer Graphics* 11 (1977).

12. S. Handel and S. Imai, "The Free Classification of Analyzable and Unanalyzable Stimuli," *Perception and Psychophysics* 12 pp. 108-116 (1972).

13. S. Imai and W.R. Garner, "Structure in Perceptual Classification," *Psychonomic Monograph Supplements* 2(9) (1968). Whole No. 25.

14. R.J.K. Jacob and L.E. Sibert, "The Perceptual Structure of Multidimensional Input Device Selection," *Proc. ACM CHI'92 Human Factors in Computing Systems Conference* pp. 211-218, Addison-Wesley/ACM Press (1992).

15. J.D. Mackinlay, S.K. Card, and G.G. Robertson, "A Semantic Analysis of the Design Space of Input Devices," *Human-Computer Interaction* 5 pp. 145-190 (1990).

16. R.N. Shepard, "Attention and the Metric Structure of the Stimulus Space," *Journal of Mathematical Psychology* **1** pp. 54-87 (1964).

Figure 1. Stimulus for the integral (size) task. The outline square is the target. The user adjusts the location and size of the solid grey square to match the target.

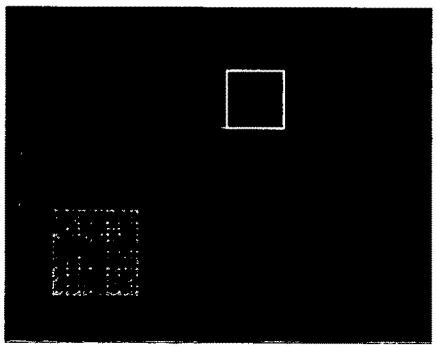


Figure 2. Stimulus for the separable (greyscale) task. The outline square gives the target location, and the outer area of the solid square gives the target color. The user adjusts the location of the solid grey square to match the target outline and the color of the inner circle on the grey square to match that of the outer area.

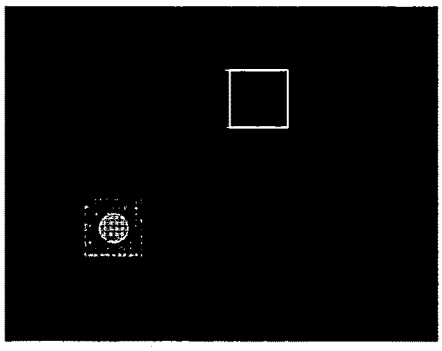
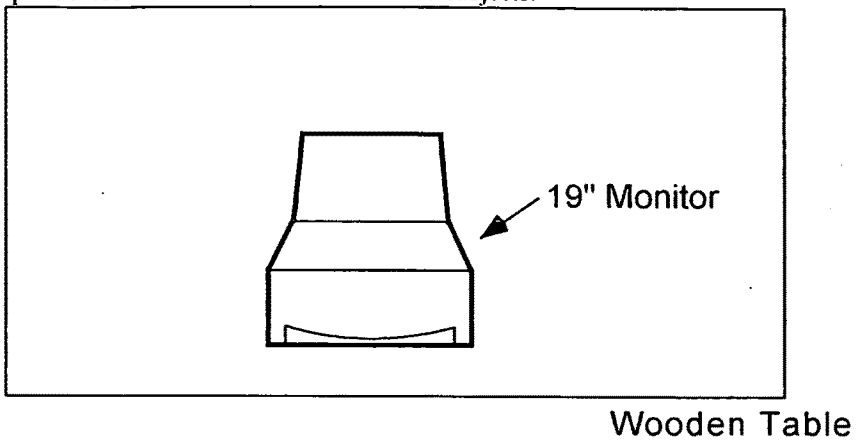


Figure 3. Room layout, precisely to scale; note how the subject sits 56 inches from the monitor. This illustration shows the mouse condition. For the Polhemus condition, the wooden table and mouse were removed, and the subject held the Polhemus wand in their right hand. The indicator button and input device were reversed for left-handed subjects.



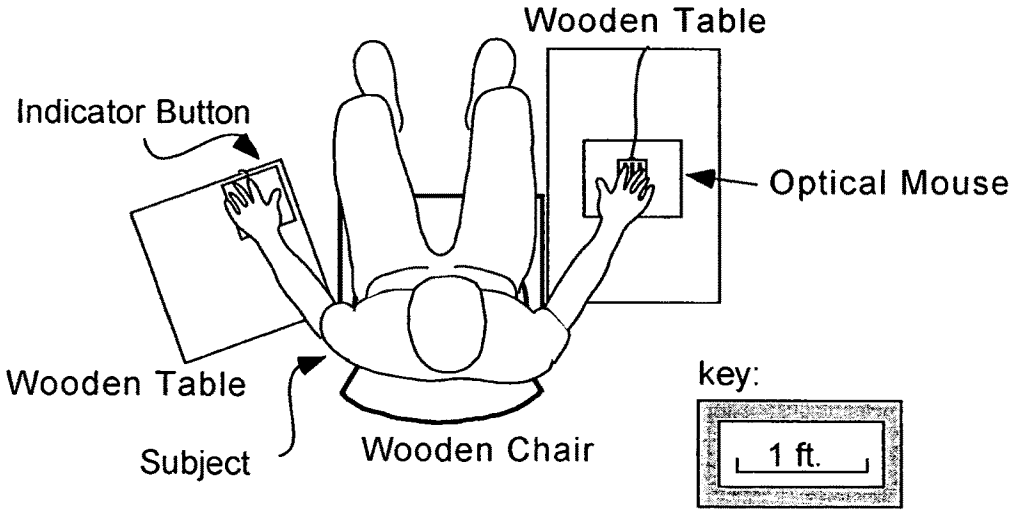


Figure 4. Time per trial in msec.

Task	Device			
	Integral (Polhemus)		Separable (Mouse)	
	Mean (ms.)	Std. dev.	Mean (ms.)	Std. dev.
Integral (Size)	4981	2065	6274	2518
Separable (Grey)	5357	1613	4838	1269

Figure 5. Graph of mean time per trial in msec., illustrating interaction effect. Line marked **S** shows performance on the integral (size) task; **G**, the separable (grey) task.

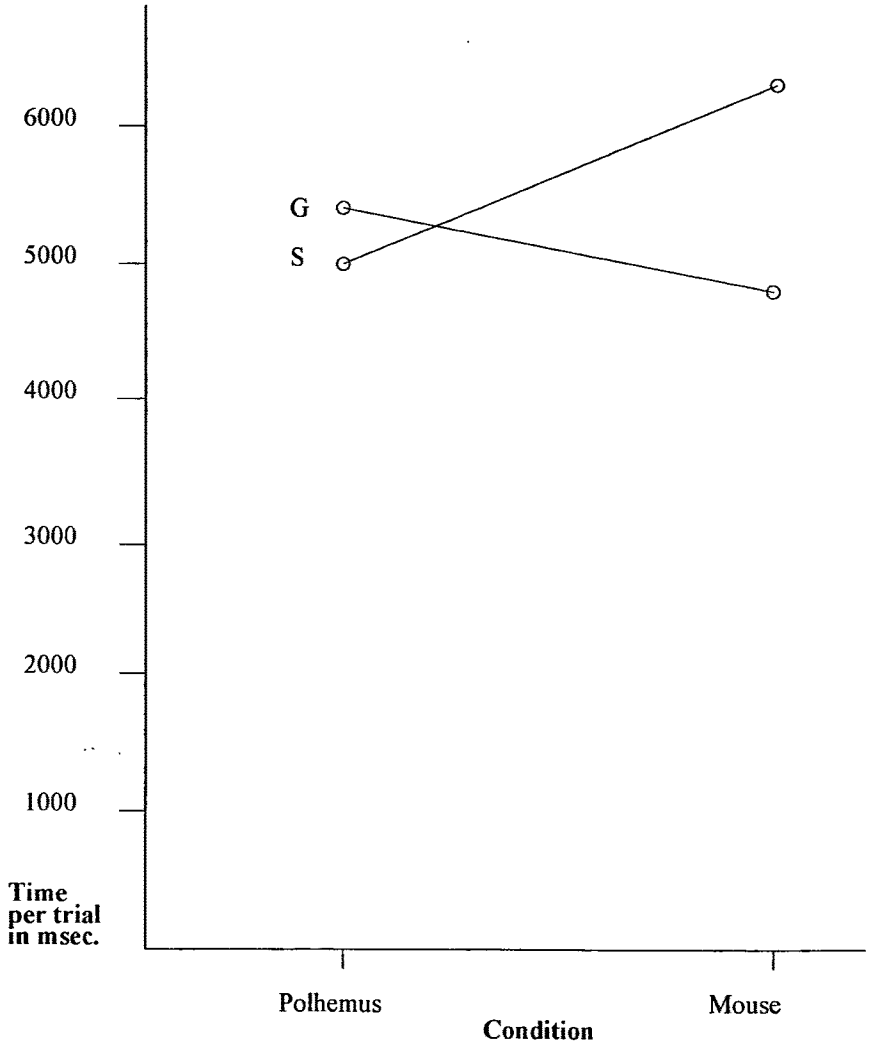


Figure 6. Accuracy in inches.

Task	Device			
	Integral (Polhemus)		Separable (Mouse)	
	Mean (ms.)	Std. dev.	Mean (ms.)	Std. dev.
Integral (Size)	0.0805	0.0308	0.0630	0.0334
Separable (Grey)	0.1088	0.0303	0.1034	0.0422

Figure 7. Average performance by condition over a range of stopping criteria covering 0.579 to 0.034 inches (Euclidean distance to target). The line marked **s/p** gives the data for the size task using the Polhemus; **s/m** is size task using mouse; **g/p** is the greyscale task using Polhemus; and **g/m** is greyscale task using mouse. The striped bar at the bottom indicates the criteria where performance time rankings followed our predictions. The value 0.24 inches at the right end of the bar corresponds to the criterion at which **g/m** and **g/p** cross.

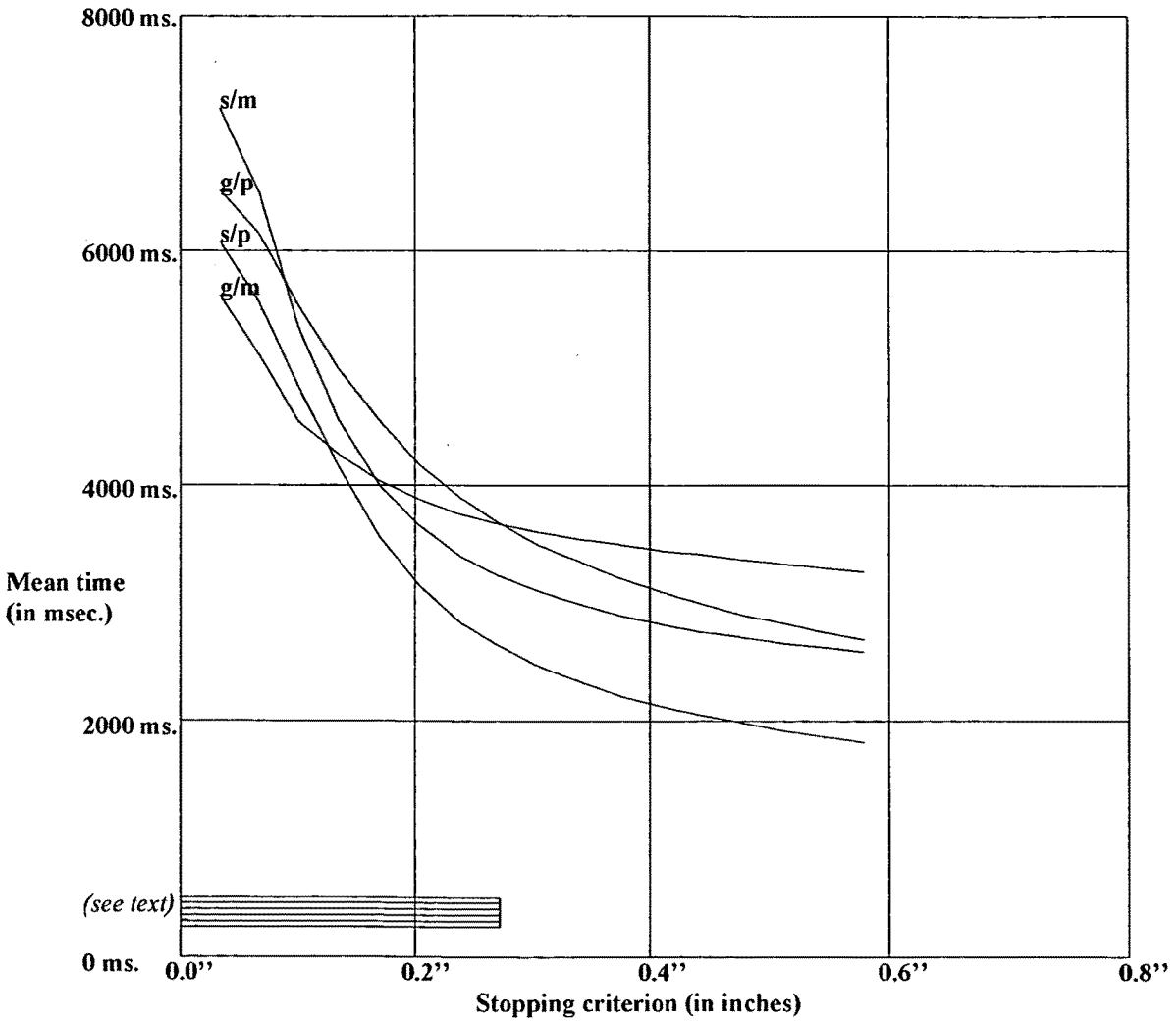


Figure 8. Illustration of the accuracy implied by the 0.24 inch criterion referenced in Figure 7, shown to scale for a 1.2 inch wide target and user-controllable object. Shaded square is 0.24 inches in Euclidean distance from the outline square in x and y only, in the stimulus space. This criterion distance is too large to be considered a match for most purposes.

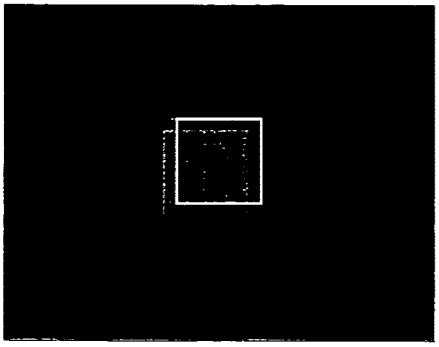


Figure 9. Time in msec. to reach 0.099 inch criterion accuracy.

Task	Device			
	Integral (Polhemus)		Separable (Mouse)	
	Mean (ms.)	Std. dev.	Mean (ms.)	Std. dev.
Integral (Size)	4062	1523	4501	1424
Separable (Grey)	4902	1427	4035	938

Figure 10. Graph of mean time to 0.099 inch criterion in msec., illustrating interaction effect. Line marked **S** shows performance on the integral (size) task; **G**, the separable (grey) task.

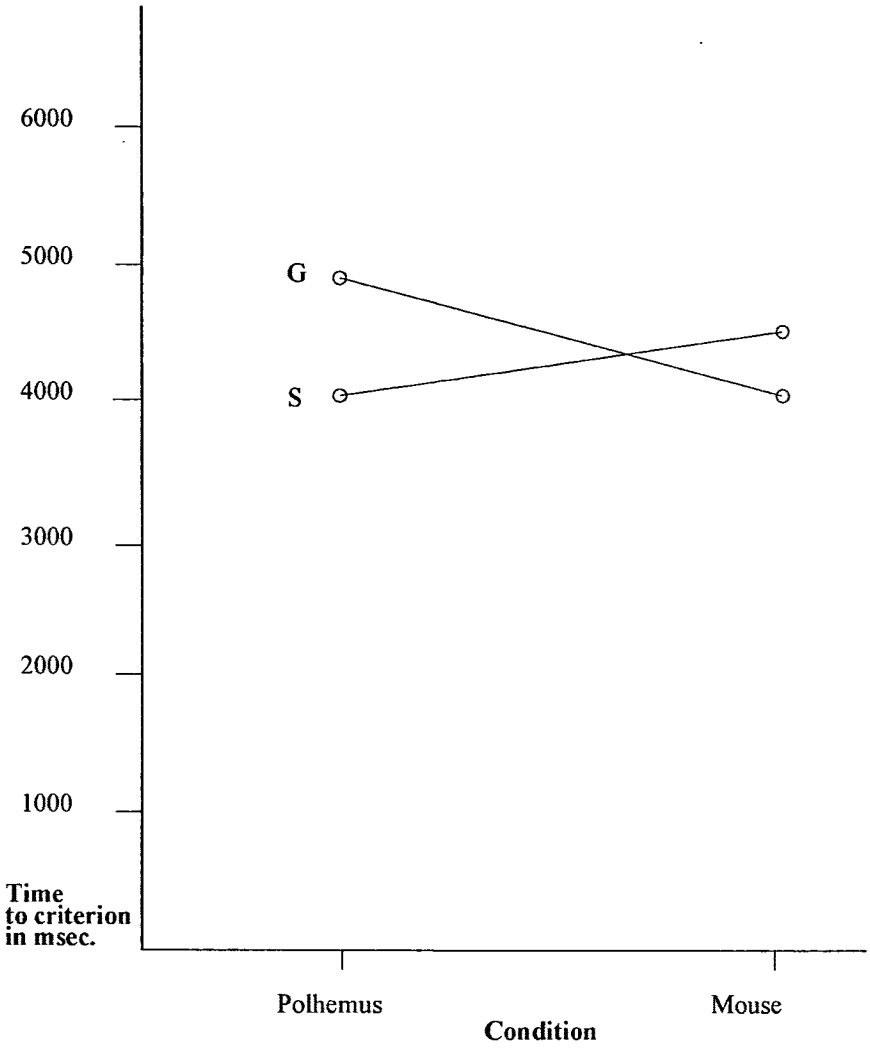


Figure 11. Time in msec. to reach 0.987 inch criterion accuracy.

Task	Device			
	Integral (Polhemus)		Separable (Mouse)	
	Mean (ms.)	Std. dev.	Mean (ms.)	Std. dev.
Integral (Size)	1442	438	2113	558
Separable (Grey)	2147	416	2929	688

Figure 12. Graph of mean time to 0.987 inch criterion in msec., illustrating lack of interaction effect. Line marked S shows performance on the integral (size) task; G, the separable (grey) task.

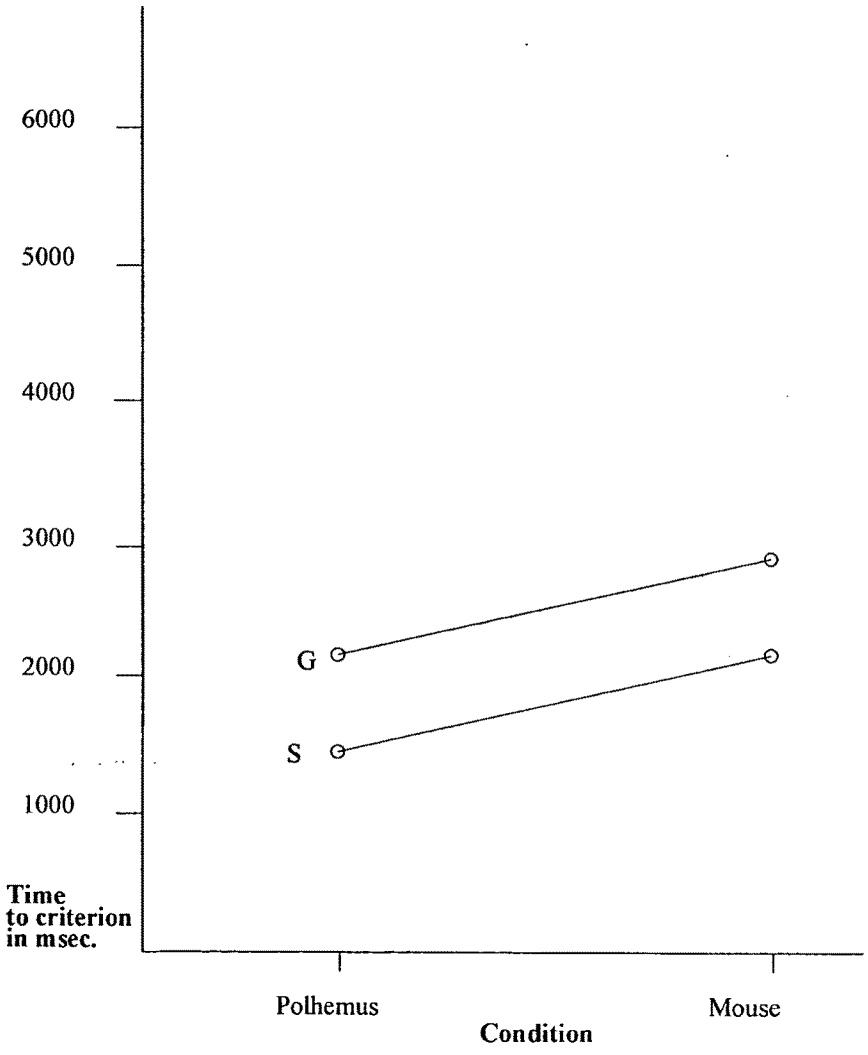


Figure 13. Picture of average Polhemus trajectory for all subjects on one trial in the integral (size) condition. Trajectory shows that subjects cut across all three axes in an integral fashion. The trial begins at the upper right corner. Fine-tuning is the small *knot* seen at the end of the trial (to the left).

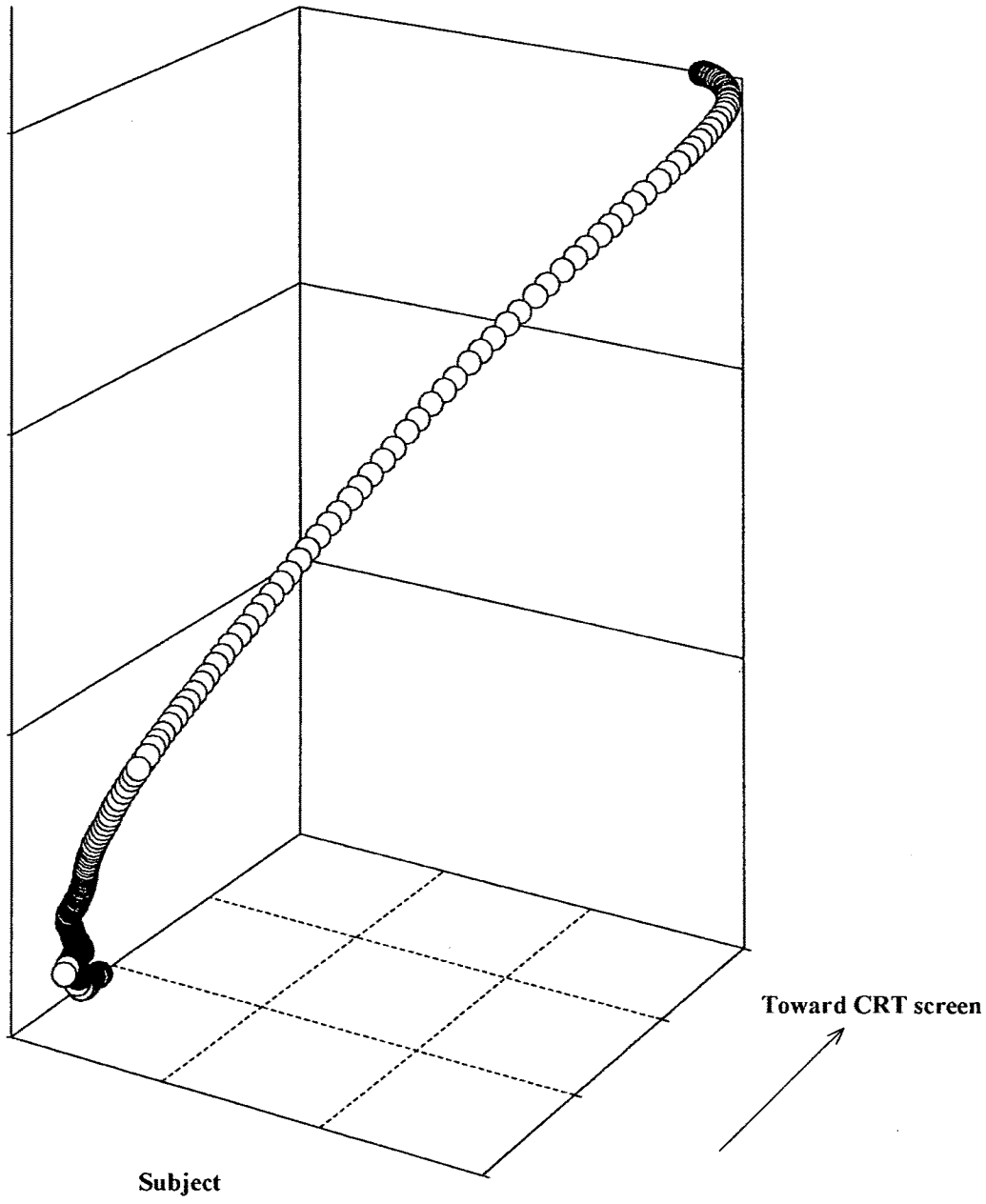
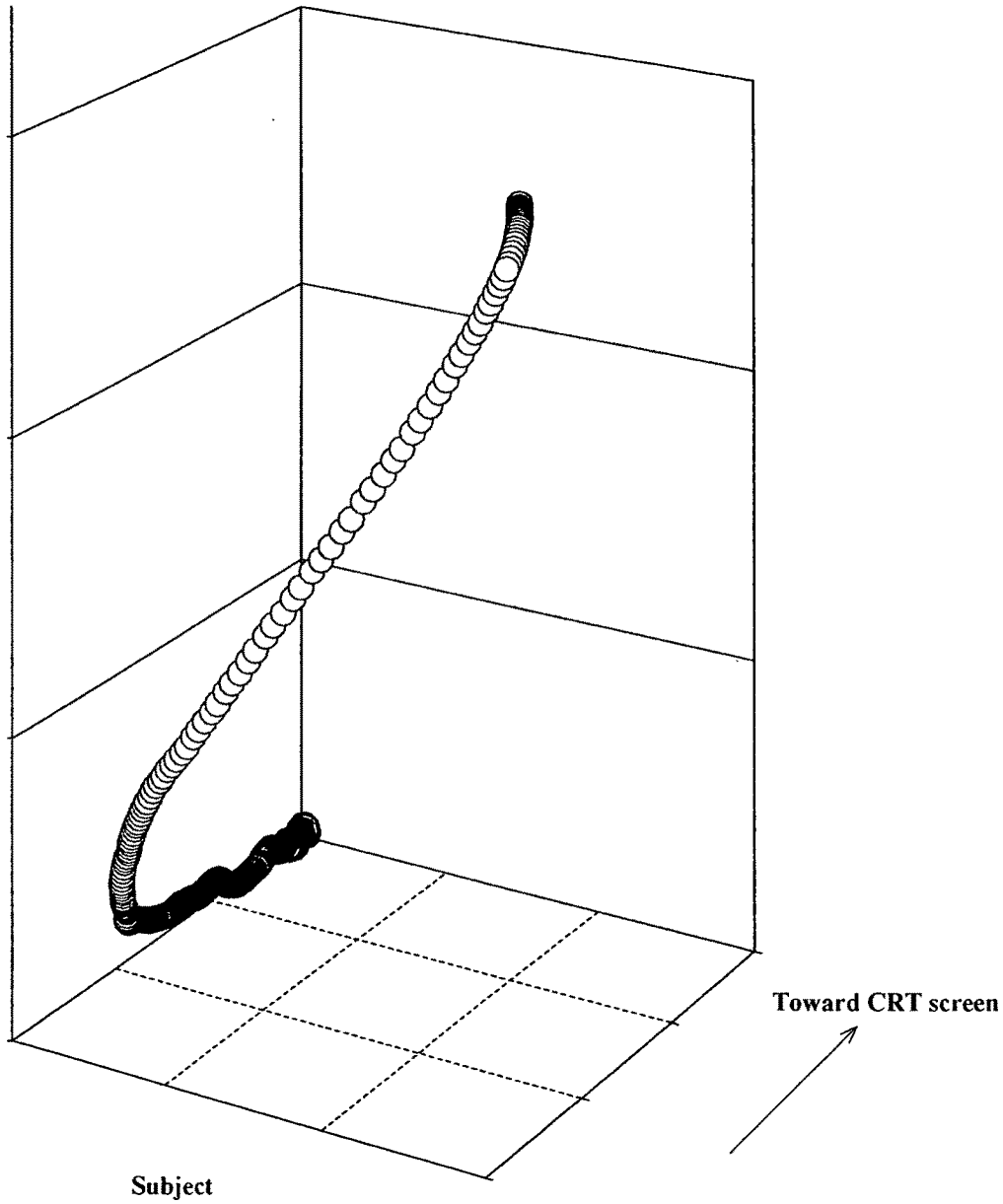


Figure 14. Picture of average trajectory for all subjects on one trial in the separable (greyscale) condition. Trajectory shows that subjects did not move simultaneously in the *x*-*y* plane and along the *z* axes, consistent with separable performance. The trial begins from the top-right and moves down to the left.



A FAST MULTIPLE-TOUCH-SENSITIVE INPUT DEVICE

SEONKYOO LEE

A Thesis Submitted in Conformity with
the Requirements for the Degree of
Master of Applied Science
in the
Department of Electrical Engineering
University of Toronto

October 1984

ABSTRACT

Touch sensitive input devices in various forms have appeared in increasing numbers in the market place. Available devices, however, are deficient in the number of parameters that can be obtained from a human gesture. Most provide only the location of a single finger tip. But there are more parameters of interest, the pressure of a single touch or the existence of multiple touches for example.

This thesis describes the design and implementation of a fast-scanning multiple-touch-sensitive input device. This device utilizes a capacitance sensing technique in conjunction with a binary scanning algorithm for both flexibility and speed.

The resolution of the sensor matrix provided is variable to allow a tradeoff between resolution and speed. Even though the apparent maximum hardware resolution is fixed to be 64 by 32, the resolution can be further increased through the application of various interpolation schemes.

TABLE OF CONTENTS

CHAPTER 2

DESIGN PROCESS

2.1 INTRODUCTION

A fruitful approach to developing a flexible multi-purpose input device is seen to lie in determining the amount of information that can be simply and conveniently provided through a single device. This is contrast to seeking ways of combining a number of existing devices.

The information that the user provides could be by means of physical gesture. To the user this implies repeatability such that the same gesture results in the same meaning, and dexterity in order that a variety of parameters can be produced with minimum sequence of expressions. The human hand can provide both required characteristics.

In particular, a simple natural communication technique is "TOUCH". Touch itself in general contains much more information than simply physical location. It could convey additional information such as, for example, the pressure or speed of touch of one or more moving fingers.

This chapter outlines the preliminary design processes leading to the development of a fast multiple-touch-

sensitive input device(FMTSID).

Section 2.2 presents five existing touch-sensitive devices with emphasis on their sensing mechanisms; section 2.3 discusses the shortcomings of the existing tablets; section 2.4 defines what must be done to the hardware and software to achieve FMTSID; and finally, section 2.5 concludes this chapter.

2.2 EXISTING TOUCH-SENSITIVE INPUT DEVICES

This section outlines existing touch tablet or screen sensing mechanisms emphasizing those using resistive, infrared, ultrasound, capacitive, and video techniques.

Resistive:

There are several approaches to the use of resistive sensors for pointing location on a tablet. Two specific and distinctive approaches are examined, namely those of the Elographics data tablet and a high resolution imaging touch sensor developed at M.I.T.

The Elographics data tablet [ref 2.1] uses a uniform resistive substrate with resistors attached to the edges in order to compensate non uniformity of the mapping between resistance and position. When the tablet is touched, pressure from the finger tip causes a deformable coversheet with

a conductive layer to contact the resistive substrate. A low voltage is applied alternatively along orthogonal axes to the edges so that each axis is exclusively selected to measure the voltage at the touch point which is a function of the distance from the selected edge. Figure 2.0 represents the functional models and construction of the Elographic tablet.

The M.I.T high resolution imaging touch sensor[ref 2.2] consists of a flexible printed circuit board, a sheet of unidirectionally uniform conductive silicone rubber and a separator to pull the conducting layer aprrt when pressure is released. The sensor arrays and electrical model of one row are shown in Fig. 2.1 and 2.2. The image of the touch is derived from the measurements of the impedances of all columns and rows in the resistor matrix. The individual impedance in a column [Fig 2.2] R[n], is obtained from successive measurements of the two port parameters of the sub-networks. The relevant equations are as follows:

$$Z[0] = 0$$

$$G[0] = 1$$

$$R[n] = \frac{V - I[mn] * G[n-1] * (Z[n-1] + Rl)}{I[n]}$$

$$Z[n] = R[n] * \frac{Z[n-1] + Rl}{Z[n-1] + R[n] + Rl}$$

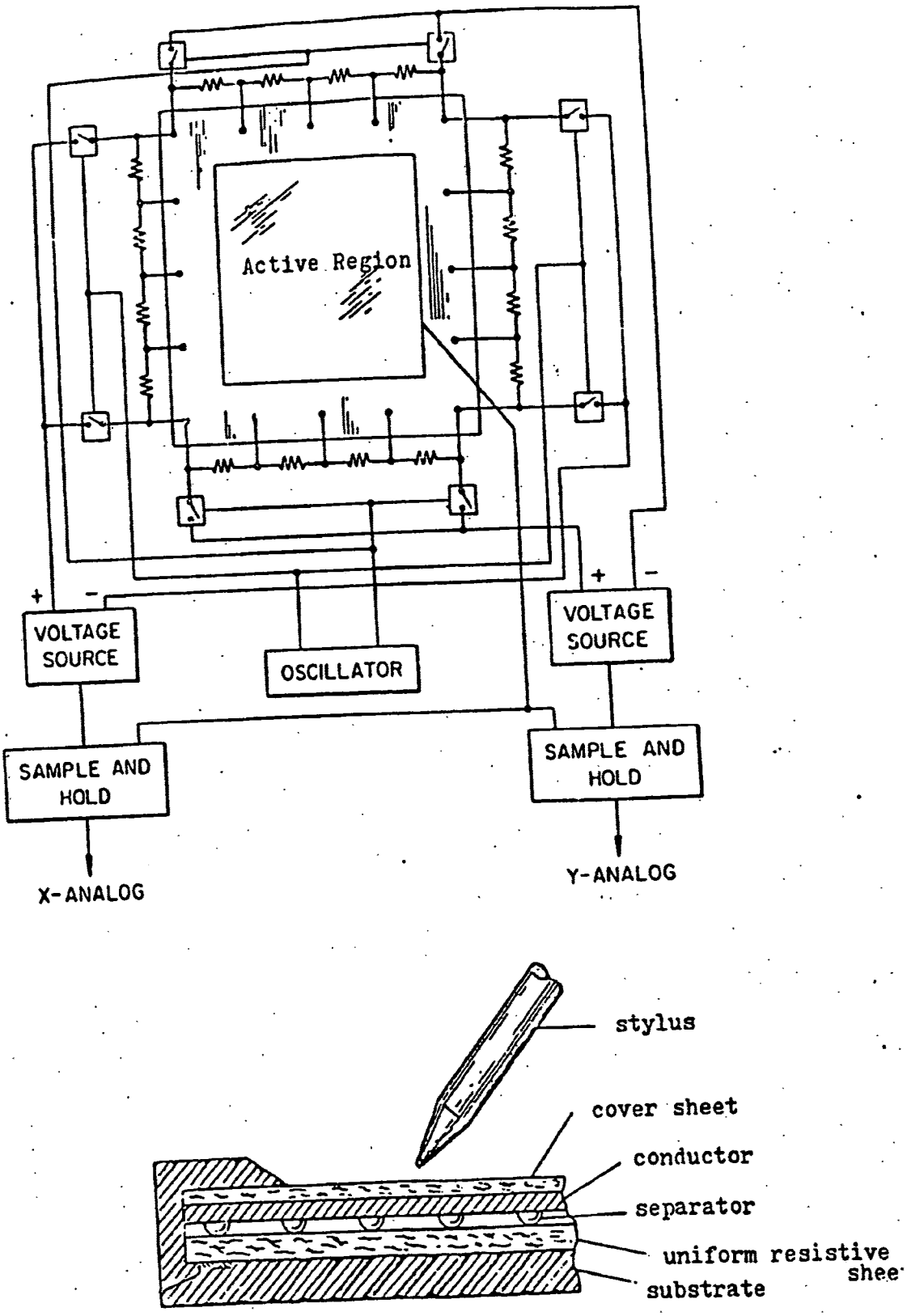


Fig. 2.0 Block Diagram and the Cross-section of the Elographics Data Tablet

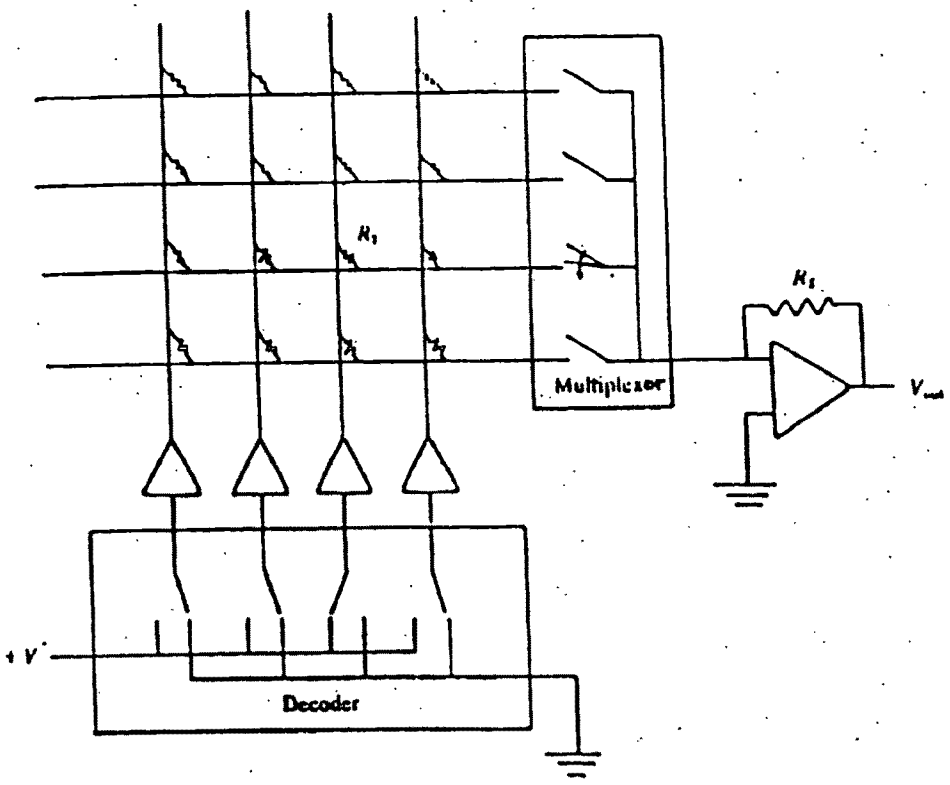


Fig. 2.1 Electrical Model of M.I.T. Imaging Touch Sensor

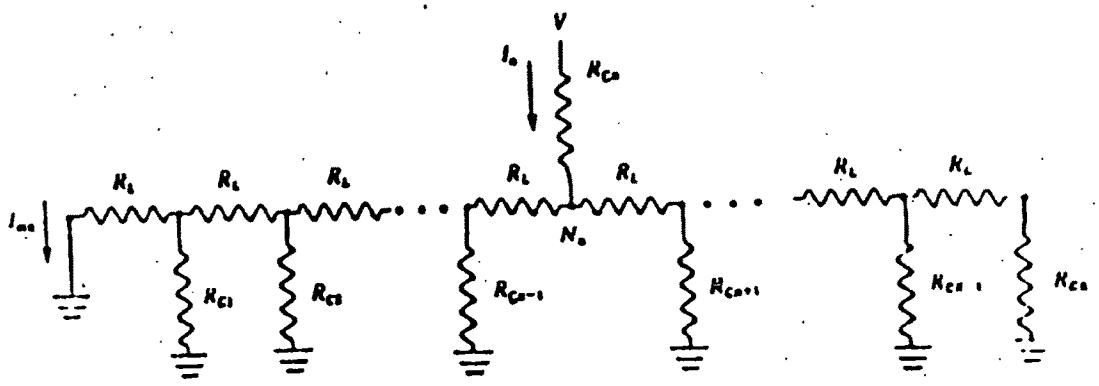


Fig. 2.2 Electrical Model of one Row of M.I.T Imaging Touch Sensor

Here V is the applied voltage, I[n] and I[mn] the measured currents, Z[n] the input impedance, G[n] the voltage transfer ratio of the network to the left of R[n], And Rl is the linear resistance of the unidirectional uniform resistive rubber sheet.

From these equations, the value of R[n] is obtained only when R[n-1] is known.

Infrared:

Infrared(IR) is one of the oldest techniques used in current sensing systems. A representative example in current use is that in the HP personal computer system.[ref 2.3] A number of the IR detectors and equal number of IR emitting diodes are mounted opposite to each other in horizontal and vertical directions such that the presence of a finger tip can be detected by the absence of the IR light at the horizontal and the vertical detectors.

Ultrasound:

The ultrasound sensor technique has been used in various areas. One of the applications in a touch sensitive input device employs piezo electric transducers mounted on horizontal and vertical edges of transparent glass screen. Here, the acoustic wave (Rayleigh wave) generated by the transducer travels along the free boundary of the glass, much like the ripple on a pond. A reflected wave or echo is