Displacement Detection Application of HDNS A2051 Optical Sensor for Controlling a 4-DOF Cylindrical Manipulator

Rawand E. J. Talabani

University of Kirkuk - College of Engineering - rawand_talabany@yahoo.com

Abstract

Parallel port interface circuit is designed, fabricated and tested for programmable control of electrical motor actuators used in a cylindrical robotic manipulator. The robotic arm built in – house has four degrees of freedom through the use of five electrical motors (four for joint actuating and one for end–effector). The interface circuit consists of relays as output devices, an octal bus transceiver, and a quad 2-input exclusive-XOR gate as input devices. An optical mouse – sensor (one for each joint) is used to indicate the actual position of the joint and uses it as a feedback signal. A menu driven QBASIC program is developed for controlling the robotic arm using the manipulator kinematics. The program contains a counter for counting the variation in digital signal (changing times of the state). The entire circuits, devices, and the program worked accurately. The result shows that the optical mouse sensors function properly in the controlling system and displacement measuring.

تطبيق حساب الإزاحةِ لمتحسّس بصري من نوع A2051 HDNS في السَيْطرَة على معالج إسطواني الحركة ذات 4 درجات من الحرية

روندإحسان جلال نصرالدين

جامعة كركوك- كلية الهندسة

الخلاصة

تم تصميم دائرة ربط توازي للسيطرة المبرمجة على مشغلات مؤازرة كهربائية والمستخدمة في معالج آلي اسطواني الحركة عن طريق أستخدام الحاسوب كمسيطر. المعالج الآلي المصنع محليا يمتلك اربع درجات من الحرية في حركته من خلال أستخدام خمس محركات كهربائية (اربعة محركات لتشغيل اربع مفاصل وواحدة لتشغيل نهايـة المعالج). دائرة الربط التوازي تحتوي على Relay للسيطرة على اختيار المشغل المؤازر وأتجاه حركته ويستخدم كأداة أخراج, كما ويحتوي الدائرة على Buffer و بوابة XOR كأدواة أدخال حيث تستعمل الأخيرة في تسهيل مهمة حساب مقدار قراءة المتحسس للأزاحة. تم أستعمال المتحسس البصري المستخدم في تطبيقات الحاسوب (الفـأرة) كمتحـسس للأزاحة الحقيقية التي يتحركها المفصل وأستخدامه كأشارة داخلة الى منظومة السيطرة. أستُعمِل برنامج QBASIC في عملية البرمجة والسيطرة على حركة الذراع الآلي وبأستخدام معادلات الكينماتيكية المشتقة بأعتمـاد شـكل الـذراع (المعالج). البرنامج يحتوي على عدّاد لحساب التغير الحاصل في الإشارة الرقمية الخارجة من المتحسس. أداء الدوائر والمعدات الألكترونية والبرنامج كانت عالية ودقيقة, ونتيجة البحث أظهرت أمكانية أستعمال المتحسس البصري(بشكله الموجود في الفأرة) في دوائر السيطرة وكجهاز لقياس الأزاحة.

1-Introduction:

Robotics is a relatively young field of modern technology that crosses traditional engineering boundaries. Robotics is a mixture of electrical engineering, mechanical engineering, system and industrial engineering, computer science, and mathematics. The science of robotics has grown tremendously over the past twenty years, fueled by rapid advances in computer and sensor technology as well as theoretical advances in control and computer vision. Typically, robots are rather electrically, hydraulically, or pneumatically powered Robotic system can be classified by control method into servo and non-servo systems, non-servo robots are essentially opened – loop devices whose movement is limited to predetermined mechanical or electrical switch stops, servo robots use closed – loop computer control to determine their motion and are thus capable of being truly multifunctional, reprogrammable devices. Most industrial manipulators have six or fewer degrees-of-freedom, which are usually classified kinematically on the basis of the first three joints of the arm. The majority of these manipulators fall into one of five geometric types: articulate (RRR), spherical (RRP), SCARA, (RRP), cylindrical (RPP), or Cartesian (PPP). In a manipulator using geometry of cylindrical, the first joint is revolute and produces a rotation about the base, while the second and third joints are prismatic. The joint variables are the cylindrical coordinates of the end effecter with respect to the base.

Control of a robot required knowledge of the position of each joint. There are a number of ways in which to obtain this information. Among the more recent and more accurate techniques are those which involve a particular kind of transducer, a calibrated rotary shaft called an optical shaft encoder, special – purpose hardware to interface the encoder to the computer and recently, the use of optical sensor

Optical mouse-sensor has been used by Hendriks[1] in registration a sheet of paper that has to be printed. The movement of the paper is indirectly encoded through rotation of motor-axes and PC-mouse is used to measured displacement of the paper. His work showed an error less than 200 µm and this error is quite small therefore it can be said that the PC-mouse sensor works quite well in control loop. Zhou, Lin, and Wang[2] used a HDNS2000 optical mouse chip to detect the actual motion state of a soccer robot which is used as the velocity feedback signal. The soccer robot match is a competitive and complicated activity, in which the accuracy of the system strategy is heavily depended on the motion performance of the robot, and to prevent the condition that the wheel of the robot skids when starts up, they presented a new velocity detection method by using of optical sensor, which can achieve the precision control of the robot. The system design of their robot had been easier and its performance had been improved from different aspects. The results of Zetterberg's[3] work showed that the hardware in the mice is already designed to function at top speeds of the optical sensors, that the mice defaults to a low resolution and needs special commands sent to them to make them work at there full potential. The purpose of his work was to take the currently implemented optical mice on the robocup robots in the AIRLab and increase the reading speed in a Linux environment by any means possible using current hardware and operating systems. His work included activities from reverse-engineering the sensor readers on the mice to creating a library for reading the mice and changing code on the current implementation.

Tim[4] utilized an optical sensor for the purposes of position tracking and object avoidance in the investigation of implementing a mobile robotics platform. The core element of his project is the implementation of a serial communications protocol between the host microcontroller and the optical sensor. By further examining the data being output from the optical sensor and feeding those results back into the motor control software, a simple proportional closed loop control system has been implemented to demonstrate the tracking ability of the sensor. Orienting an additional sensor in the direction of travel, Tim's project

also investigated the use of the optical sensor as an inexpensive camera-like device, by processing images and the available data from the forward looking sensor, object recognition and avoidance gives the robot the ability to navigate within dynamically changing real-world environments. His work showed that the optical mouse sensors are a highly viable alternative to more traditional odometry methods such as wheel encoders and the sensors can be configured to perform as a low-level image capture devices, in order to provide real time visual feedback of the operating environment.

Given an omni-directional mobile platform using four Mecanum wheels, it required further capability being programmed to achieve various motion behaviors and intelligence, Cooney, Xu, and Bright[5] describe a solution to the problem involving both hardware and software developments, two optical mice are attached to the front and rear of the robot giving positional feedback for closed-loop control and dead-reckoning for navigation and a microcontroller is interfaced to all TTL compatible devices and subsequently programmed to implement various robotic behaviors. A closed-loop control scheme using three independent PID controllers is implemented to follow three DOF motions. Their work found out that the optical mice are promising sensors to the dead-reckoning for the robot motion control.

In this paper, a 4-DOF robotic manipulator has been designed and manufactured with five electrical motors. Three of these motors are those used in satellite receiver antenna actuating and one motor for twist actuating and another for end-effecter. Ten relays are used as output devices, and one octal bus transceiver (buffer) for input signal. All output and input signals are interfaced using DB-25 parallel port of the PC. The manipulator involves four optical mouse-sensors as position sensors and feedback signal to the PC for closed-loop control system through quad 2-input exclusive-OR gate. A menu driven QBASIC program is developed for controlling the robotic arm using the manipulator forward and inverse kinematics. The program contains a counter for counting the input digital signal. The goal of this paper is the use of an optical mouse-sensor in a noncomplex manner in manipulation control systems and design a simple and inexpensive optical position sensor circuit, figure (1) shows the built in-house manipulator.

2- Kinematics of Manipulation:

- **Forward kinematics:**

The forward kinematics problem is concerned with the relationship between the individual joints of the robot manipulator and the position and orientation of the end-effector. Stated more formally, the forward kinematics problem is to find the position and orientation of the end-effector given the values for joint variables of the robot.

The joint variables are the angles between the links in the case of revolute or rotational joints, and the link extension in the case of prismatic or sliding joints. A large part of robot kinematics is concerned with the establishment of various coordinate systems to represent the positions and orientations of rigid objects and with transformations among these coordinate systems[6]. Homogeneous transformations combine the operations of rotation (R) and translations (P) into a single matrix multiplication. The fixed coordinate systems attached to the 4- link cylindrical manipulator linkages, which called the word or base frame are shown in figure(2). Five word frames are used to describe the position and orientation of the end-effector (word frame 4) with respect to manipulator base (word frame 0).
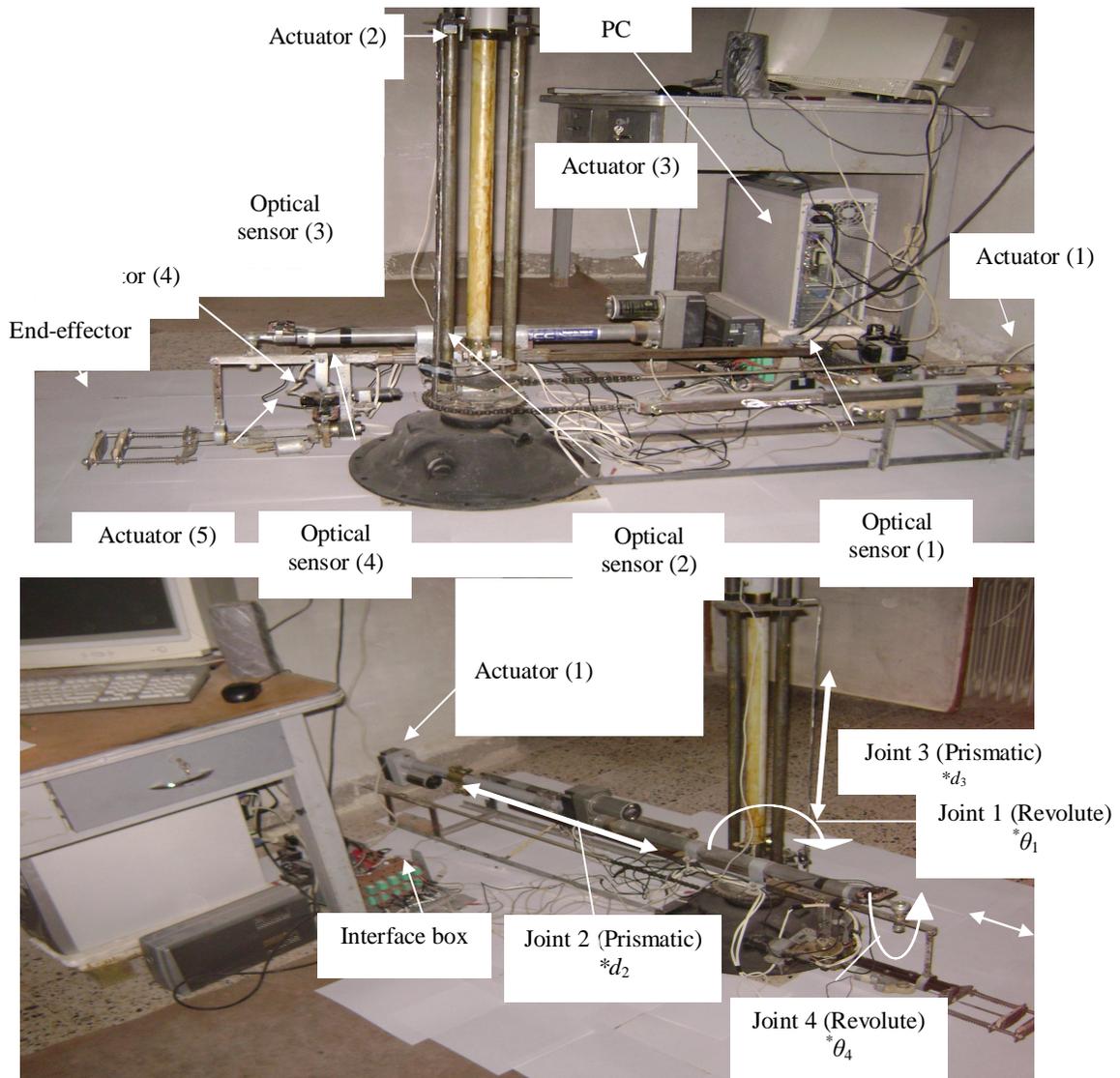
Figure (1): the built in-house 4-link, 4-DOF cylindrical manipulator
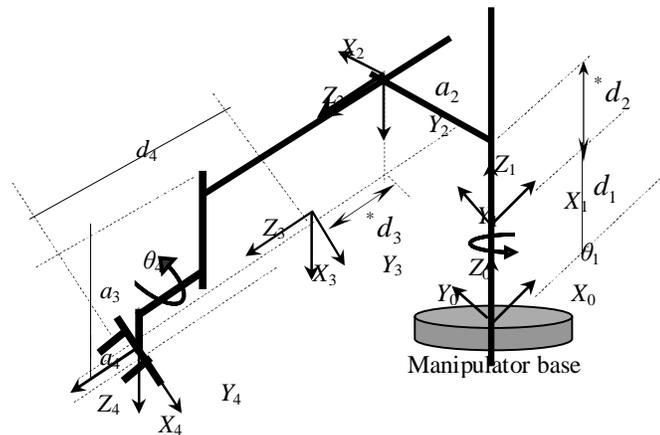
Figure (2): frame assignments for the 4-link cylindrical manipulator

The homogeneous transformations or Denavit-Hartenberg (DH) convention, which is used to simplify the transformation among the attached coordinate frames and finding the link parameters, for 4- link cylindrical manipulator, is: (for more details, see the appendix A)

$$T_4^0 = \begin{bmatrix} -C\theta_1 S\theta_4 & -C\theta_1 C\theta_4 & -S\theta_1 & -a_4 C\theta_1 S\theta_4 - d_4 S\theta_1 - {}^*d_3 S_1 + a_2 C\theta_1 \\ -S\theta_1 S\theta_4 & -S\theta_1 C\theta_4 & C\theta_1 & -a_4 S\theta_1 S\theta_4 + d_4 C\theta_1 + {}^*d_3 C\theta_1 + a_2 S\theta_1 \\ -C\theta_4 & S\theta_4 & 0 & -a_4 C\theta_4 - a_3 + d_1 + {}^*d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

Where $C\theta_i$ & $S\theta_i$ denote $\cos(\theta_i)$ and $\sin(\theta_i)$ respectively, and a_2, a_3, a_4, d_1, and d_4 are constants and depend on manipulator dimensions and geometry. The link parameters are given in table (1). The joint variables are the relative angle between links one and two θ_1, relative angle between links three and four θ_4, and links two and three extension ${}^*d_2$, & ${}^*d_3$.

Table (1): DH parameters for 4-link cylindrical manipulator

Link	a_i (mm)	α_i	d_i (mm)	θ_i
1	$a_1 = 0$	0^0	$d_1 = 125$	$*\theta_1$
2	$a_2 = 72.5$	-90^0	$*d_2$	0^0
3	$a_3 = 220.5$	0^0	$*d_3$	90^0
4	$a_4 = 28$	0^0	$d_4 = 532$	$*\theta_4$
* joint variable				

- **Inverse kinematics:**

The inverse kinematics problem concerned with finding the joints variables in terms of the end-effector position and orientation, and it is, in general, more difficult than forward kinematics problem. The more degrees of freedom that the manipulator may have, the more difficult inverse kinematics solution is. Because the current manipulator has 4-DOF, closed form solution that based on analytic expressions can be used[7].
Let:

$$H = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

be a (4×4) homogenous transformation, here H represents the desired position and orientation of the end-effector, and the task is to find the values for joint variables θ_1, θ_4, ${}^*d_2$, and ${}^*d_3$ so that $T_4^0 = H$. Therefore:

$$\left.\begin{array}{l} r_{13} = -S\theta_1 \Rightarrow S\theta_1 = -r_{13} \\ r_{23} = C\theta_1 \end{array}\right\} \theta_1 = A\tan 2(-r_{13}, r_{23}), \tag{3}$$

and

$$\left.\begin{array}{l} r_{32} = S\theta_4 \\ r_{31} = -C\theta_4 \Rightarrow C\theta_4 = -r_{31} \end{array}\right\} \theta_4 = A\tan 2(r_{32}, -r_{31}), \tag{4}$$

57

and for $^*d_2$, $^*d_3$

$$p_x = a_2 C_1 - {}^*d_3 S\theta_1 - d_4 S_1 - a_4 C\theta_1 S\theta_4$$
$$\Rightarrow {}^*d_3 = \left(a_2 C\theta_1 - d_4 S\theta_1 - a_4 C\theta_1 S\theta_4 - P_X\right)/S\theta_1 \tag{5}$$

or

$$p_y = a_2 S\theta_1 + {}^*d_3 C\theta_1 + d_4 C\theta_1 - a_4 S\theta_1 C\theta_4$$
$$\Rightarrow {}^*d_3 = \left(p_y - a_2 S\theta_1 - d_4 C\theta_1 + a_4 S\theta_1 C\theta_4\right)/C\theta_1 \tag{6}$$

and

$$p_z = d_1 + {}^*d_2 - a_3 - a_4 C\theta_4$$
$$\Rightarrow {}^*d_2 = p_z - d_1 + a_3 + a_4 C\theta_4 \tag{7}$$

   Equations (3) through (7) are the general solutions of inverse kinematics; the practical limitations of the joint variables because of manipulator workspace are shown in table (2).

Table (2): Operating voltage and joint displacement limitation

Joint actuator (electrical motor)	Operating voltage (V)	Joint displacement limitation
1	12	$0^0 \rightarrow 190^0$
2	36	$0 \rightarrow 431.8$ mm
3	36	$0 \rightarrow 584.2$ mm
4	24	$0^0 \rightarrow 300^0$

3- Optical Mouse- Sensor Units:

   The HDNS A2051 sensors implemented for this project was chosen due to the availability of the datasheet, as well as the sensors being available in computer mice at the start of the project. The sensors are manufactured with a small aperture on the underside of the chip, which is aligned to a small lens that rests between the sensor and the operating surface. This surface is illuminated by a high-intensity LED, in order to make the features of the surface more distinguishable. This layout can be seen in figure (3). The sensors are generally implemented with a red light source, that varies between a partially and fully on state, depending if movement is sensed under the device. The LED is placed into high intensity mode during movement, which is controlled directly by an output pin of the device. With the surface illuminated, the internal programming of the optical sensor processes the captured pixel data by examining highlights and particular components of the image. By marking particular features in one frame (each captured image is called a frame), and comparing this to additional frames captured after a small increment in time, movements can be calculated by examining the displacement of particular features between the frames. This process takes place within the optical sensor units, and is detailed to operate at 1500 frames per second (*fps*), with the default resolution of the sensor set to 400 counts per inch (*cpi*). This gives the sensor the ability to detect movements of up to 12 inches per second[8]. The output from the sensor represents the displacement of the sensor between the captured frames which is in the
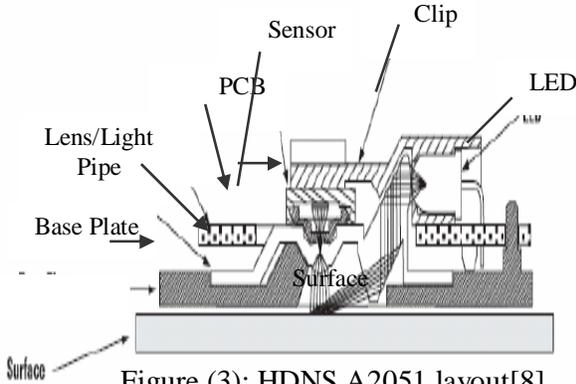
Figure (3): HDNS A2051 layout[8]
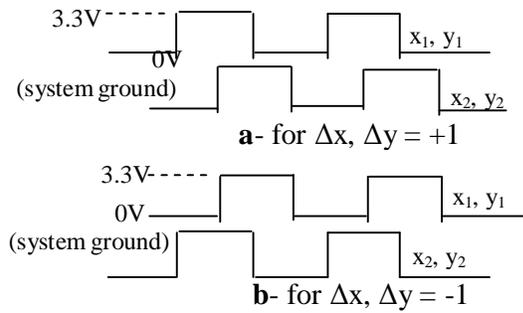
Figure (4): quadrature timing diagram[9]

form of quadrature format (figure 4) gives four output signals x1, x2, y1, y2; with voltage level from 0V (system ground) to 3.3V. The pair x1, x2 indicates the displacements in X-direction, and the pair y1, y2 indicates the displacements in Y-direction. The reference direction for the displacement results can be seen in figure (5), where movement to the right and upwards are both positive in magnitude, in the same manner as a Cartesian coordinate system. The initial state of quadrature outputs may be arbitrary. The condition of quadrature outputs is renovated one time at the end of frame. While displacement in and direction equals zero, corresponding pair of quadrature outputs does not change. If in each frame displacement is equal to +1 pixel, quadrature outputs will change in accordance with timing diagram in figure (4a), and when displacement is equal to –1 pixel, quadrature outputs will change in accordance with timing diagram in figure (4b)[9]. It has two state machines inside of HDNS A2051 which point to X direction and y direction respectively. Each state machine has four solid states, figure (6).
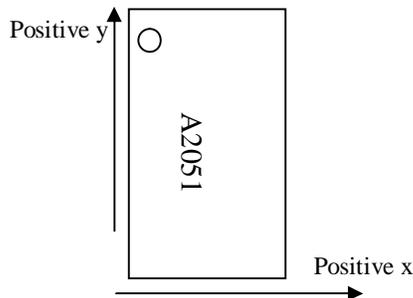
Figure (5): top view of ADNS2051 coordinate frames of reference[8]
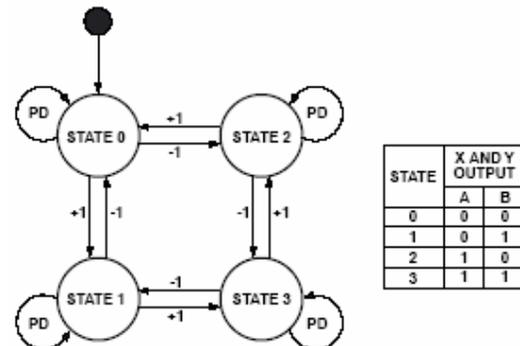
Figure (6): quadrature state machines[8]

State 0 is entered after a power up reset. When it detects the object moves in any direction by one unit, it transfers to the next state. So the motion amount of the object can be calculated if the continuous transformation of the state is detected. A corresponding logical calculus of the impulse signal can help to decide the distances between the current state and the pervious state and base on the above theory, the software and the hardware of the manipulator can be designed. Four optical sensor devices have been placed directly to each joint in the four links manipulator and calibrated to give the correct joint displacement. The sensors report the progressive movements as displacement measurements, and the values are reset after each displacement register read. The lens system provides a magnified view of the area immediately under the sensor, making it possible to track and report even the smallest of movements. Figure (7) shows the sensor chip pin configurations. The standard method for

connection of the mice to a PC is via the (USB) or PS/2 connection standards, in this project, the connection to a PC is made directly from the optical sensor chip (figure 8).
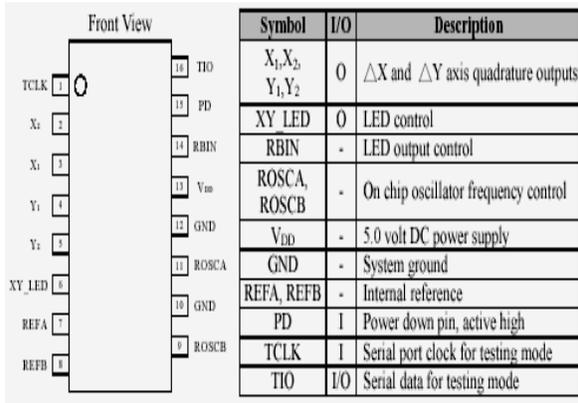
Figure (7): HDNS A2051 pin configurations and descriptions[8]
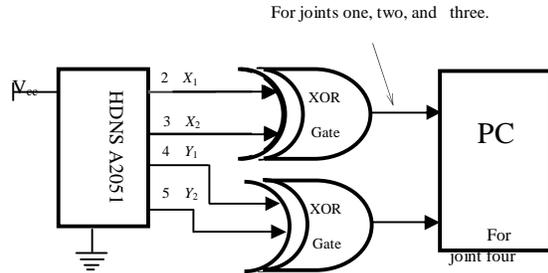
Figure (8): Pin connections of the optical chip

4- Circuit Design and Implementation:

- **System design:**

The parallel port interface circuit is designed using the DB-25 parallel port connector, shown in figure (9). The parallel port contains the eight data lines D0-D7 as output, five status lines as input, and four control lines as output. The 8 data line bits together with 2 bits of control lines are used for controlling the selection and direction of the actuating motors through ten electrical relays. Four bits of status lines are used to input position information for robotic arm through the use of four optical mouse-sensors and four quad 2-input exclusive-OR gate (74LS86) in the assigned special format as shown in figure (10). An Octal bus transceiver buffer (SN74LS245) is connected to the input digital signal. A power output circuit consisting of NPN transistor is required to translate the TTL level outputs from the digital control logic into required power level for the relays. A block diagram of the over-all circuit is shown in figure (11). Table (2) gives the details about the electrical motors used.
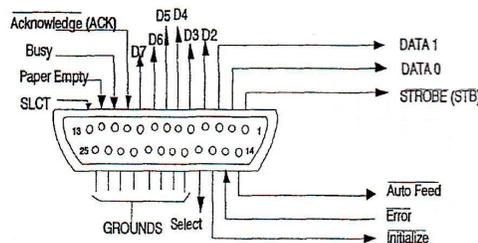
Figure (9): DB-25 connector for PCs

Data line digits (output)								Control line digits (output)				Status line digits (input)				
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	C_0	C_1	C_2	C_4	S_0	S_1	S_2	S_3	S_4

Joint one actuating (Revolute) 0 = C.W. 1 = C.C.W	Joint two actuating (Prismatic) 0 = forward 1 = backward	Joint three actuating (Prismatic) 0 = upward 1 = downward	Joint four actuating (Revolute) 0 = C.W. 1 = C.C.W	End-effector actuating 0 = opening 1 = closing	Not used	Position information input from optical sensor devices S_0 for joint one S_1 for joint two S_2 for joint three S_3 for joint four (inversed)	Not used

Figure (10): DB-25 digits format as used for controlling the manipulator operation
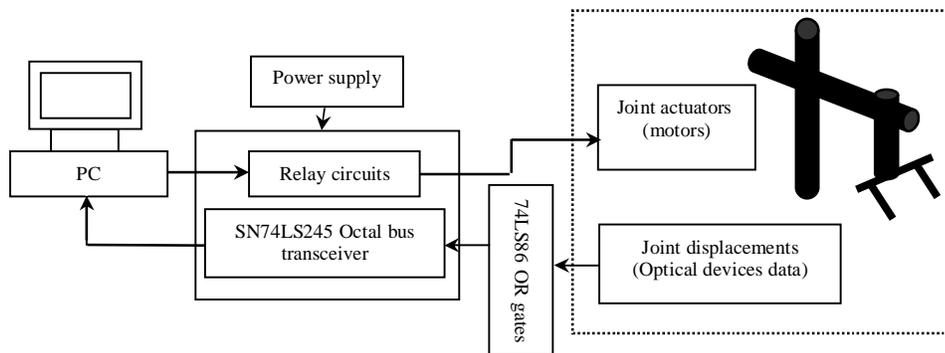
Figure (11): Block diagram of the digital motion control system

- **Hardware design of displacement detection using the optical mouse sensor:**

The capture velocity of the optical sensor chip is 1500 *fps*. As the time of each frame is 667 *µs* and the minimum state time is 133 *µs*, up to four states can exist in each frame[8]. The maximum motion distance of the optical sensor chip is 12 *ips* under the condition that no frame is lost. So the linear distance between a state and the consecutive state is:

$$d = \frac{P * 25.4}{F * N} = \frac{12 * 25.4}{1500 * 4} = 0.0508 \; mm / state , \qquad (8)$$

where:
P is the maximum motion distance between two consecutive states (*ips*),
F is the number of frames captured each second (*fps*),
N is the time of states changes (*state / frame*).

Equation (8) is used as the basis of distance detection. After the procession of information collected by the optical mouse chip, two pulse signals been output from X_1 and X_2 and the phase between them is 90 degrees. The distance detection can be realized by inquiring about the state changing times and calculating the moving distance. The logical XOR gate operates between X_1 and X_2 signals (for joints one, two, and three) and between Y_1 and Y_2 signals for joint four, the number of logical 1 is half of the changing times of the states (due to the quad 2-input exclusive-OR gate hardware specification). Impulse is counted by a software counter which is convenient to get the changing times of the states. The connection to the PC is made

directly from the optical sensor chip, pins 2 and 3 for joints one, two, and three and pins 4 and 5 for joint four, (figure 8).

5- Software Design:

In this project, the controlled software can be divided into two parts, the first part interests in controlling the input signals from the optical sensor devices and determine the measured displacements for each moving joint, while the second part is developed to control the manipulator operation. In both parts, QBASIC has been chosen for the software development. The PC that is used for system control has 2.4 GHz processor, 256 RAM, GIGABYTE motherboard with parallel port address (as LPT1) shown in table (3).

Table (3): Parallel port addresses

Parallel port	Data (base address)	Status (base address + 1)	Control (base address + 2)
LPT1	378H, 888D	379H, 889D	37AH, 890D

- **Optical sensor software:**

Basically the main step in software design for measuring the position of each joint is using a counter for counting the variation in the digital input signal. The input signal has a decimal form therefore it must be converted to a binary form so that the program knows which joint is moving (may two or more joints move at the same time). The next step is counting the variation in each binary number (between 0 and 1) which is equivalent to changing times of the states and using equations (9) (appendix B) to calculate the actual displacement for each joint.

$$
\left.\begin{array}{l}
{}^{*}\theta_1 \ (\text{deg}) = (d * (c.t.o.s)_1 \,/r_1) \, * (180^0 \,/ \pi) \\[4pt]
{}^{*}d_2 \ (mm) = \ d * (c.t.o.s)_2 \\[4pt]
{}^{*}d_3 \ (mm) = \ d * (c.t.o.s)_3 \\[4pt]
{}^{*}\theta_4 \ (\text{deg}) = (d * (c.t.o.s)_4 \,/r_4) \, * (180^0 \,/ \pi)
\end{array}\right\} \qquad (9)
$$

where
d is the linear distance between two states (*mm / state*),
$(c.t.o.s)_i$ is the corresponding changing times of the states,
$r_1 = 80.5$ *mm* and $r_4 = 38.5$ *mm* are the radii of joints one and four, respectively.

Using this procedure, the program can determine the displacement of each joint even if the four joints are actuated at the same time. The flow-chart is shown in figure (12).

- **Manipulator operation control software:**

Menu driven QBASIC program is developed to control the manipulator actuators. The desired end-effector position is input to the program, and then the joint variables are calculated using equations (3), (4), (5), and (7). The joint displacements, measured by the optical devices, are used as feedback signals and are compared with the calculated joint variables to determine the error signal. A simple program is developed for this operation because basically the project's goal is to provide further information about the usage of optical mouse-sensor in manipulation control systems and not to design effective control software. To test the software and the system, a task is chosen as follows: the arm should move from a zero reference point to a specific position, carry a bottle full with liquid, fill a glass with this liquid, bring the glass to the programmer hand, and finally go back to the zero

reference point. Figures (13) and (14) show flow-chart and manipulator operation during the above task. Table (4) shows the detected errors.

Optical-device Controller

$S = A_1 = A_2 = A_3 = A_4 = 0, S_1 = 1$

Input the status port data

Convert the decimal numbers into binary format and set it in array B(4)

B(1) = = S → Y

$S = B(1)$ ← N

$A_1 = A_1 + 1$

$Th_1 = (A_1*2*d/r_1)*(180^0/\pi)$

Joint 1 ($*\theta_1$)

B(2) = = S → Y

$S = B(2)$ ← N

$A_2 = A_2 + 1$

$d_2 = A_2*2*d$

Joint 2 ($*d_2$)

B(3) = = S → Y

$S = B(3)$ ← N

$A_3 = A_3 + 1$

$d_3 = A_3*2*d$

Joint 3 ($*d_3$)

B(4) = = S_1 → Y

$S_1 = B(4)$ ← N

$A_4 = A_4 + 1$

$Th_2 = (A_2*2*d/r_1)*(180^0/\pi)$

Joint 4 ($*\theta_4$)
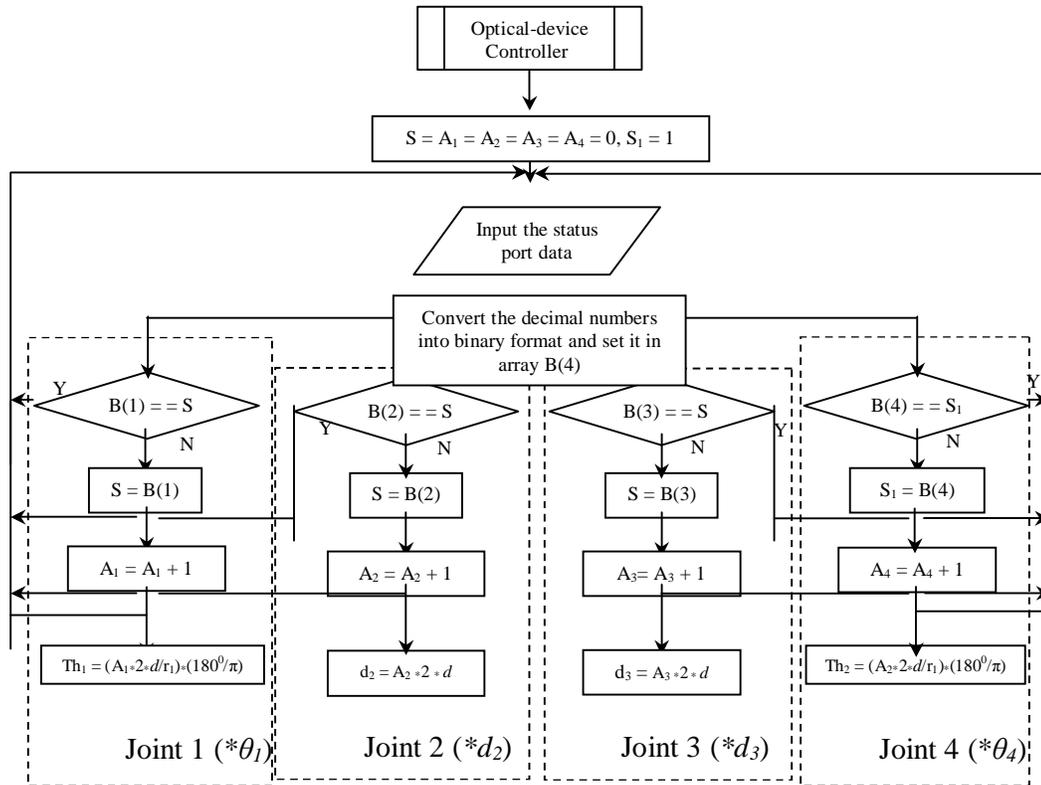
Figure (12): Flow chart for calculating joint displacements

Start

Input the desired end-effector positions

Calculate joint variables ($\theta_1, *d_2, *d_3, \theta_4$) from equations 3 through 7

Optical-device Controller

$P_1 \geq *\theta_1$
$P_2 \geq *d_2$
$P_3 \geq *d_3$
$P_4 \geq *\theta_4$

Y

N

Stop the corresponding joint actuator

Continue joint actuating

Stop
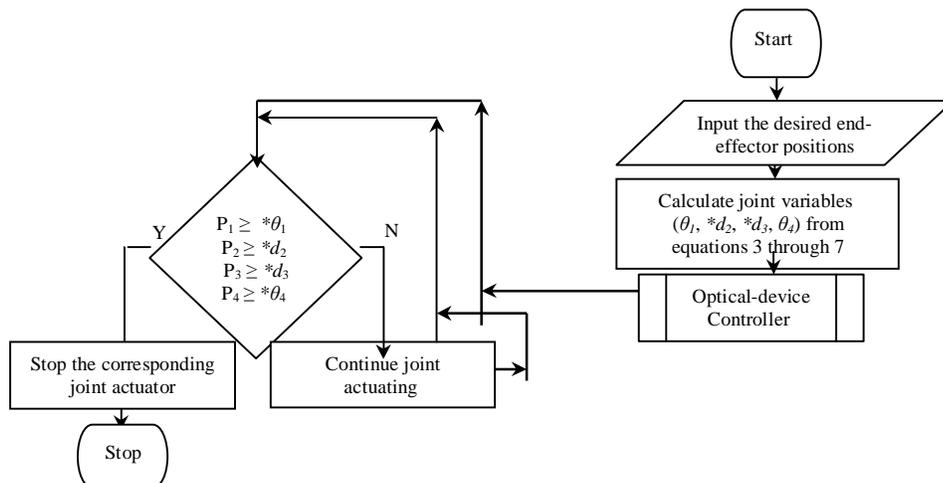
Figure (13): Controlling manipulator's operation flow chart
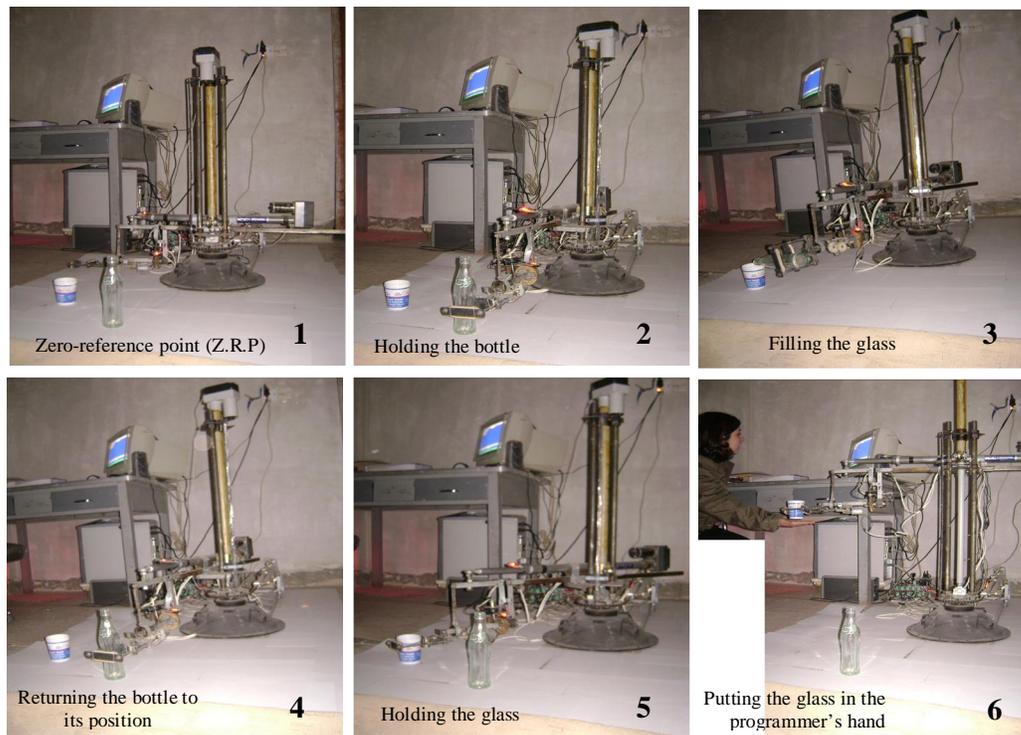
Figure (14): Manipulator operation

Table (4): Error detections

Task	Actuating joint	Movement direction	Joint variable calculated from Kinematics eqs. and path planning	Joint variable sensed by the optical sensor	Error %
1- Moving from Z.R.P. toward the bottle and holding it.	3	upward	$240.84mm$	$240.892mm$	0.021
	1	C.C.W.	55.38^0	55.372^0	0.014
	2	forward	$101.73mm$	$101.752mm$	0.021
	3	downward	$17.41mm$	$17.421mm$	0.063
2- Filling the glass and returning the bottle.	3	upward	$7.56mm$	$7.563mm$	0.039
	1	C.W.	15.68^0	15.692^0	0.076
	4	C.C.W.	116.73^0	116.724^0	0.005
	4	C.W,	116.73^0	116.719^0	0.009
	1	C.C.W.	15.68^0	15.678^0	0.012
	3	downward	$7.56mm$	$7.563mm$	0.039
3- Toward the glass and holding it.	3	upward	$7.56mm$	$7.558mm$	0.026
	1	C.W.	15.61^0	15.615^0	0.032
	3	downward	$9.57mm$	$9.568mm$	0.020
4- Moving toward the programmer and put the glass in his hand	3	upward	$403.64mm$	403.651^0	0.002
	1	C.W.	43.75^0	43.762^0	0.027
	2	forward	$200.52mm$	$200.518mm$	0.0009

6- conclusion:

A 4-DOF manipulator has been designed and four optical sensor devices have been used as feedback signal transducers. Simple software is developed for controlling the system and the work of the optical sensors using the PC printer port as an interface port. Although, a simple technique is used in connection of optical sensor chips with the PC (without using the standard USB or PS/2 technology), the entire system and software operated flawlessly, and the errors are quite small, therefore it can be said that the optical PC-mouse sensor works very well as a measurement transducer and as a feedback device. With a simple software and a relatively cheep and available device (optical PC mouse), one can get a high accuracy and reliable position sensor which can be used in closed-loop control systems, complex robotic systems, or to measure distance such as measuring material deformation or fluid flow.

Appendices

- **Appendix A**

The homogeneous transformations or Denavit-Hartenberg convention between the 4-linkages and equation (1) are driven using the procedure found in [10] and shown below:

- <u>Between links (0) & (1):</u> $T_1^0 = \begin{bmatrix} C\theta_1 & -S\theta_1 & 0 & 0 \\ S\theta_1 & C\theta_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$,

- Between links (1) & (2): $T_2^1 = \begin{bmatrix} 1 & 0 & 0 & a_2 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & {}^*d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$,

- Between links (2) & (3): $T_3^2 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & a_3 \\ 0 & 0 & 1 & {}^*d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$,

- Between links (3) & (4): $T_4^3 = \begin{bmatrix} C\theta_4 & -S\theta_4 & 0 & a_4C\theta_4 \\ S\theta_4 & C\theta_4 & 0 & a_4S\theta_4 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$.

and $T_4^0 = T_1^0 * T_2^1 * T_3^2 * T_4^3$

- **Appendix B**

The first and last terms in equation (9) is driven as follows:

$$S = r * \theta \Rightarrow \theta = S/r \quad \text{and} \quad S = d * (c.t.o.s)$$

$$\therefore \theta(rad) = (d * (c.t.o.s))/r \quad \text{or}$$

$$\theta(\deg) = [(d * (c.t.o.s))/r] * (180^0/\pi)$$
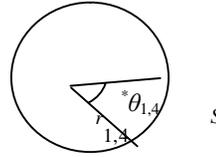
Where S is the arc length.

References:

[1] Hendriks, M.P., "Direct paper-motion control using an optical mouse-sensor", M.Sc. theses, University of Technology Eindhoven, Eindhoven, PP68., 2003.

[2] Zhou, J., Lin C., and Wang G., "Velocity detection application of HDNS2000 in robot soccer", college of mechanical & electrical engineering Hohai University. 2004.

[3] Zetterberg, J., "Optimization of optical mice for AI-robots", M.Sc. theses, Orebro University, Orebro, Sweden, PP56, 2005.

[4] Tim, M., "Optical mouse sensor for robot odometry and simple vision processing", M.Sc. theses, University of California, California, USA, PP83, 2005.

[5] Cooney, J. A., Xu, W. L., and Bright G., "Visual dead-reckoning for motion control of a mecanum-wheeled mobile robot", institute of technology and engineering, college of sciences, Massey University, New Zealand, 2003.

[6] Wang, J., Hu, Q., and Jiang, D., "A Lagrangian network for kinematic control of redundant robot manipulators". IEEE, 99, 1123-1132, 1999.

[7] Craig, J., "Introduction to robotics, mechanics and control", 2-nd edition, Addison-Wesley publishing company, USA, 449p, 1989.

[8] Jan, ADNS-2051 "Optical mouse sensor data sheet", Agilent technologies Inc, 2005.

[9] UIC1001, (2004), "Solid-state optical mouse sensor with quadrature outputs IC datasheet", Unique ICs, LLC, 2005.

[10] Spong, M. W., Hutchinson, S., and Vidyasagar, M., "Robot dynamics and control", 2-nd edition, Prentice Hall, USA, 343p, 2004.