International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-2 Issue-5, June 2013

Efficient Design and Implementation of 4-Degree of Freedom Robotic Arm

Ankit Gupta, Mridul Gupta, Neelakshi Bajpai, Pooja Gupta, Prashant Singh

***Abstract**— A robotic arm is a mechanical arm which is designed to perform a function similarly as a human arm does. It is usually programmable and arm may be a sum total of the mechanism or can be part of a complex robot. To design a robot, various links are being connected by joints allowing either a translational (linear) motion or a rotational motion in various planes. In many cases sensors are used in the arm that usually indicates the controller about the hardness by which the gripping is done by arm or directs the arm in directions in which it should move to perform the task or it tells the system about presence of object in front of it.*

Various aspects that are kept in mind while designing a robotic arm are torque calculation for each motor used, concept of inverse kinematics, interfacing to remote controller, ways for noise reduction in ADC (analog-to-digital converter).We have designed this robotic arm using servos, ATmega32 microcontroller with interfacing analog joysticks controller. We have made some improvements in design by reducing the noises that were generated due to mechanical construction of the joystick which continuously varies and gives some noises. And also we have used the fact the mechanical system has larger response time as compared to associated electronics. To overcome this we went for some software filters and algorithms following the ADC. Before this, we have introduced inverse kinematics with its difference from forward kinematics and various calculations related to it.

***Index Terms**— Averaging Filter, DOF (Degree of Freedom), End effectors, Inverse kinematics, IIR Filter.*

I. INTRODUCTION

A robot is an electro-mechanical system which can perform any of the operation commanded by the user with the help of input devices like remote controller or any type of sensor, which are interfaced with some computer and electronic programming. Certain algorithms are the controlling medium created by the user so that it performs the specified functions. Majorly any robot can be of two types: semi-autonomous and autonomous.

Manuscript published on 30 June 2013.
* Correspondence Author (s)
Ankit Gupta, DESE, Indian Institute of Science, Bangalore, India.
Mridul Gupta, Electronics & Communication, Kanpur Institute of Technology, GBTU University, Kanpur, India.
Neelakshi Bajpai, Electronics & Communication, Kanpur Institute of Technology, GBTU University, Kanpur, India.
Pooja Gupta, Electronics & Communication, Kanpur Institute of Technology, GBTU University, Kanpur, India.
Prashant Singh, Electronics & Communication, Kanpur Institute of Technology, GBTU University, Kanpur, India.

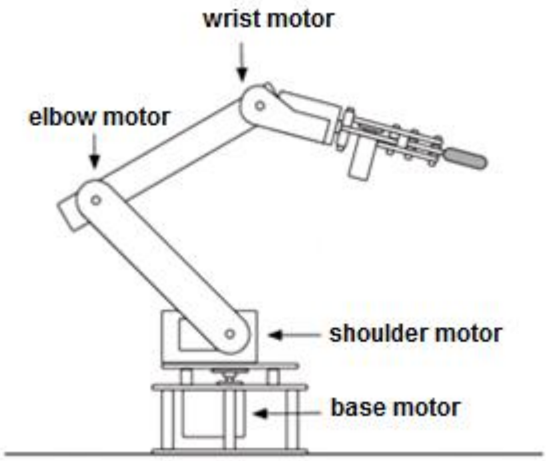

Fig 1: General Structure of Robotic Arm

Out of above mentioned types we have gone with semi-autonomous joystick controlled robotic arm. After link-joints relation, DOF considerations and torque calculations we proceeded for the overview of inverse kinematics and calculations.

II. INVERSE KINEMATICS

Kinematics is the study of motion, describing the motion of the structure. It is usually studied under two basic parts: *forward kinematics* and *inverse kinematics*.
Forward kinematics manipulates with the structure which is done by changes of the joint angles inside the controlled structure (Fig 2a).

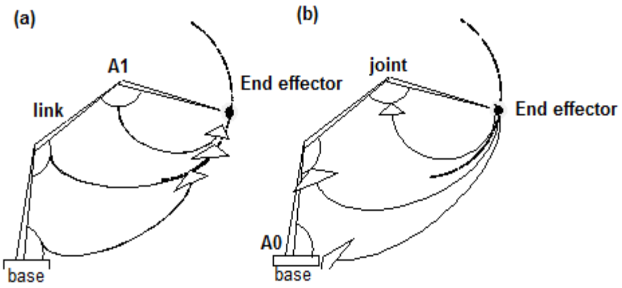

Fig 2: Structure manipulation a) *forward* b) *inverse kinematics.*

But if we want to know how the upper joint structure would keep on moving when it is necessary that the end effector should move towards its goal position. In that case direct calculations are carried out in the structure so that joint angle can be determined as the changes are done in the end effector (Fig 2b).

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Therefore we can say that the concept of inverse kinematics provides joint parameter values when we are known to the desired position of the end effector with the help of kinematic equations of robot.

III. MATHEMATICAL ANALYSIS OF INVERSE KINEMATICS [1]

There could be many possible solutions if only the coordinates of the end effector are given as the input. So to avoid redundancy we are taking one more gripper angle w.r.t. horizontal as input. Following are specified as the input to the Robotic Arm:

- The *x, y, z* coordinate of the end effector.
- The angle of the grip from horizontal (ɵ).

The following values are known:

- *Base length* (a), *shoulder length* (b), *arm length* (c), *gripper length* (d)
- *Gripper angle* (ɵ)
- *Radial length* (r)

For convenience, we change the calculations of 3-D inverse kinematics into the 2-D form. For this two planes are being considered to be perpendicular with each other- a z-plane and x, y-plane in horizontal as shown in below Fig 3.

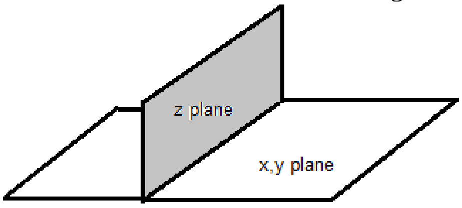

Fig 3: x, y, z plane

A. *The x, y plane*

Calculations in the x, y plane use coordinates x and y.

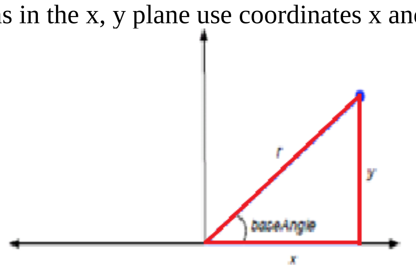

Fig 4: x, y plane

The position of the z-plan is decided with the help of rotating motor. It passes through the coordinates (0,0,0) and the radial distance (r) is given at (x,y,0) coordinates.

The base angle and the radial distance calculations are shown.

B. *Calculations*

$$base\ angle = tan^{-1}(y/x)$$

$$r = \sqrt{x^2 + y^2}$$

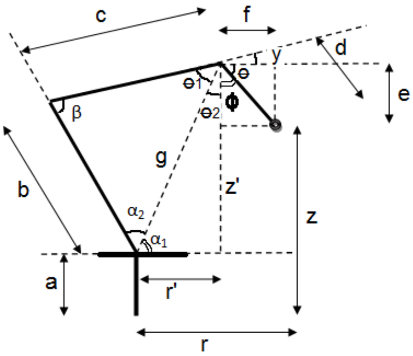

Fig 5: Various Parameters of arm

Consider the triangle made by wrist

$$e = d \times sin(\theta) \qquad (1)$$

$$f = d \times cos(\theta) \qquad (2)$$

$$\therefore r' = (r - f) = r - d \times cos(\theta) \qquad (3)$$

$$\because z' = z + d \times sin(\theta) - a \qquad (4)$$

Therefore, in triangle where r' and z' are known

$$g = \sqrt{[(r')^2 + (z')^2]} \qquad (5)$$

Now in triangle where all sides g, r' and z' are known, is considered

$$\alpha_1 = cos^{-1}\{[g^2 + (r')^2 - (z')^2] / (2 \times g \times r')\} \qquad (6)$$

Triangle where g, b and c sides are known

$$\alpha_2 = cos^{-1}\{[g^2 + b^2 - c^2] / (2 \times g \times b)\} \qquad (7)$$

Refer (6) and (7)

$$shoulder\ angle,\ \alpha = \alpha_1 + \alpha_2 \qquad (8)$$

In same triangle,

$$\beta = cos^{-1}\{[c^2 + b^2 - g^2] / (2 \times c \times b)\} \qquad (9)$$

$$\therefore arm\ angle = 180 - \beta$$

$$\because \theta_1 = 180 - (\beta + \alpha_2) \qquad (10)$$

$$\because \theta_2 = 180 - (\alpha_1 + 90) \qquad (11)$$

$$\phi = 90 - \theta \qquad (12)$$

$$\because wrist\ angle = y + \theta$$

$$wrist\ angle = 180 - (\theta_1 + \theta_2 + \phi) \qquad (13)$$

Hence from above calculations we obtain different side's angles when gripper angle and side's lengths are set as input by user. This is how inverse kinematics calculations are being carried out for deriving joint angles from changes of the end of the structure.

IV. IMPLEMENTATION OF ARM WITH SERVOS

Use The ARM is built with Servo Motors as it gives precise control over the angles. And also it gives added advantage of in-built feedback mechanism.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

We are not required to know the previous values of angle for the upcoming values. The angles computed earlier converts the PWM wave of required width and fed to the respective servo to move the end effector to the desired location.

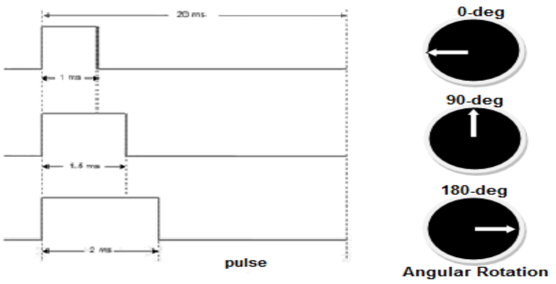

Figure 6: Control Pulses required for Servo.

V. NOISE SUPPRESSION ALGORITHMS IN A.D.C (ANALOG TO DIGITAL CONVERTORS) [2, 3]

Coordinates are given by the Analog Joysticks available in market with USB Game Controllers. Their position values are sampled by the ADC of the microcontroller. Let us take a Atmega32 microcontroller for its 10-bit ADC. We built up system and although the ADC gives 10 bits of resolution, the lower couple of bits are unreliable. There are many reasons

1) Noise in the Supply.
2) Due to the high frequency noise of switching circuit.
3) The Joystick is not of very good quality therefore the analog value keeps on changing.

So every sample from ADC is the combination of signal (S) and noise (N). So for reducing this noise from upcoming signal we make use of two different software filters. The main goal of these filters is to reduce the noise without impacting the signal. Since noise has incoherent nature and it adds in a root sum of squares (RSS) pattern. This can be reduced with a simple averaging of several samples.

$$y(n) = (1/L)\sum(m = 0 \text{ to } L-1)\ x(n - m)$$

Output y(n) is the average of latest coming L samples. After above calculations it is found that signal is not changed but there is some noise reduction. This reduction in noise is inversely proportion to the square root of the number of samples. Each time as the number of averaged sample is doubled, the noise reduces by $(1/\sqrt{2})$. The filters using finite number of samples without feedback mechanism are called *FIR (finite impulse response)* filters. In this, 20 values are accumulated and then divided by 20. This is easy way to implement. But if the filtered values are needed for update after each sample, then "n" previous samples are needed to be saved.

In other filter, for calculating the next value, the last output is used. Therefore main functionality includes averaging of a part of the input with (n-1) part of the last calculated output value.

In this filter every input is changed by the value "n" and each time whenever the output is given back to system as a feedback then it changes by a factor (n-1)/n. This means that for computation through this filter only one addition, subtraction and only one division is required.

Again similar results were seen through these filters which were found in the previous one, excepting the amount of noise reduction. The amount of reduction is almost √2 more in this filter when compared with previous filter. This is IIR filter or infinite impulse response filter because each input effect on the output never vanishes. This can also be proved from the statement given by Graham Bell that the sound levels only attenuate and can never be fully disappeared.

The notable thing from the above discussion is that although the IIR filter reduces more noise, but it surely reduces the signal bandwidth. Therefore if signal bandwidth is to be reserved then we should use FIR filter otherwise IIR filter where lesser RAM is required. IIR filter only requires storage of the result while the FIR filter needs data storage for each of the previous "n" samples.

VI. ALGORITHM

A. Averaging Out

Filter out the noise by taking 20 consecutive samples and averaging them out. This will eliminate any noise present in the joystick and thus resulting in any continuous variation in the values. Noise can cause continuous variation or jitter in the servos and will affect the performance of the system.

This noise is introduced due to the mechanical construction of the Joystick itself which continuously varies and gives some noise. Since we are averaging out the samples and each one is taken at 32ms so taking 20 samples will give enough time for servos to stabilize.

B. IIR Filter

Second problem with the system is that the joystick used is normal USB game controller so there is always a problem of large swinging values with even small change. To minimize this effect we have put a small IIR filter in which the new value depends on the previous values as well as the current sampled value. And in this also the overall variation is controlled to 33% with the new values

$$newvalue = \left(\begin{array}{c} previous\ value \times 2 + \\ current\ sampledvalue \end{array}\right) / 3$$

The Algorithm for this works as follows:-

- Sample the new value coming from the Filter defined in the step one
- Now check whether this new sampled value is close to previous sampled value or not

$$newvalue(\pm)\ 2 = previousvalue$$

- If yes then apply the equation shown above otherwise assign the new values directly.

VII. CONCLUSION

From above we can figure out one of methods to acquire calculative aspect of inverse kinematics for deriving joint angles from changes of the end of the structure. Hence precise calculations can lead to accurate functioning of our servos with changing coordinates. Next output from above is one of the methods that can be precisely used to reduce the noise which is created by simple analog controller for better operation.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

REFERENCES

[1] White Paper "Controlling Lynx6 Robotic Arm Micro mega"

[2] http://electronicdesign.com/analog/use-software-filters-reduce-adc-no is

[3] Barnett, Cox and O'Cull "Embedded C Programming and the Atmel AVR " 2nd Edition 2007.

[4] http://www.lynxmotion.com/images/html/proj057.htm

[5] www.engineersgarage.com/articles/servo-motor?page=1

Ankit Gupta has done M.Tech in Electronics Design from IISc Bangalore and B.E. from Bharati Vidyapeeth Pune. His interest lies in design and development of systems inspired by nature that helps to study and conserve it. His major has been development of Camera Traps for wild life researchers.

Mridul Gupta is final year student of B.Tech from Kanpur Institute of Technology. He is passionate about embedded systems and animation. He loves to paint and write in his free time.

Neelakshi Bajpai is final year student of B.Tech from Kanpur Institute of Technology. She is passionate about mathematical analysis related to signal and controlling systems. She has a great interest in dancing and excels in the same field.

Pooja Gupta is final year student of B.Tech from Kanpur Institute of Technology. She is passionate about learning new things and has good knowledge in signaling and system.

Prashant Singh is final year student of B.Tech from Kanpur Institute of Technology. He is passionate about embedded systems and logistic tasks.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)