ARTIFICIAL HUMAN LIMBS – A DESIGN APPROACH FOR MILITARY APPLICATION

R.Karthikeyan, Department of EIE, Veltech,Member IEEE,
rkarthiekeyan@gmail.com
Anitha Karthikeyan, Department of ECE,Meenakchi College of Engineering.
mrs.anithakarthikeyan@gmail.com

S.Sivaperumal, Department of ECE,Vel HIGHTECH SRS Engineering College.
sivaperumals@gmail.com

Abstract— The most essential automation is saving human life, saving their belongings, protecting their properties and making arrangements in a systematic way for automation. This paper deals with the design of a real time Human Limb which acts according to the design configurations of the prescribed datas as per the sensor calibrated. This research proposes to overcome current limitations using three axis optimal inertial sensors combined with an Embedded Controller on which the filter algorithm as well as analog to digital converter is implemented for correcting drift and angular motion through all orientations. The mechanical design will have miniature or hybrid stepper motors with associated mechanical elements to move the limbs on all the axis like up/down ,roll, elevation and azimuth.

I.INTRODUCTION

With the development of networked synthetic environments (SE) stand to revolutionize the fields of education, training, business, retailing and entertainment. They will fundamentally alter our societies and the way in which mankind views the world. In the educational field, synthetic environments will offer the ultimate in hands-on and visualization of difficult concepts. They will allow training to transpire in a place much like that in which the skills being practiced will be used without exposure to possible hazards and at less cost. In the workplace, employees will be able to work “side by side” even though they may be physically separated by hundreds or even thousands of miles. .[Durlach -1995].Using synthetic environments, corporations will obtain a safe, economical and efficient method of testing new concepts and systems. Retailers will create virtual department stores where consumers will be able to try out products to an unprecedented degree before actually buying them.

Using synthetic environments, the entertainment industry will be able to create entire worlds in which customers will be able to experience thrills and live out entire fantasy lives [Zyda-1997].The power of the synthetic environment lies in its ability to immerse users in a different world. The more complete the immersion, the more effective the synthetic environment. For complete immersion, the user should sense and interact with the synthetic environment in the same manner in which interaction with the natural world takes place. Interaction in the natural world results from body motion. Information regarding the surrounding environment is obtained through the five senses. Changes in body posture and position directly affect what is seen, heard, felt and smelled[Mavor-1995].

The parameters sensed in the environment are altered and manipulated by the actions of the body. Thus, in order for a user to interact with a synthetic environment in a natural way and have the synthetic environment present appropriate information to the senses, it is imperative that data regarding body motion and posture be obtained[Skopowski,1996]. Body posture and location data are also needed in multi-user environments to drive the animation of avatars which represent the actions of users of the environment to each other. At this time, there is no practical and intuitive interface that allows an individual human to be inserted into a SE in a fully immersive manner. [Badler, N,1993].

Numerous motion tracking technologies are currently in use, but each suffers from its own set of limitations. Depending on the technology, these limitations may include marginal accuracy, user encumbrance, restricted range, susceptibility to interference and noise, poor registration, occlusion difficulties and high latency. Due to these problems, real-time animations of avatars must be largely script-based using motion libraries. For the most part, only a single user may be tracked in a small working volume. Thus, none of the current technologies fulfills the need for wide-area tracking of multiple users. The ideal motion tracking technology must meet several requirements. It should have low latency, be tolerant to noise and other environmental interference, track multiple users and maintain both adequate accuracy and registration throughout a large working volume [MoletAubel-1999].

The primary reason current tracking systems fail to meet the requirements described above is the dependence of these systems on a generated “source” to determine orientation and location information. This source may be sent by transmitters to body-based receivers or it may be sent from body-based transmitters to receivers positioned at known locations throughout the working volume. Usually the effective range of this source is extremely limited or there may be compromises between resolution and range. Interference with or distortion of this source will at best result in erroneous orientation and position measurements.

II.MOTIVATION

Motion tracking technology currently fail to provide accurate wide area tracking of multiple users without interference and occulation problems. This research proposes to overcome current limitations using three axis optimal inertial sensors combined with an Embedded Controller on which the filter algorithm as well as analog to digital converter is implemented for correcting drift and angular motion through all orientations.The mechanical design will have miniature or hybrid stepper motors with associated mechanical elements to move the limbs on all the axis like up/down ,roll, elevation and azimuth. An appropriate electronic circuit is used for isolation between the stepper motors and an Embedded Controller in computers.

The electronic system will be suitable upto 5A for 70kg cm stepper motor but in this research the stepper motor used is only 7kg cm .Joint angle determination for robots with flexible links is difficult. Use of Bluetooth technology will enable sensors to wirelessly transmit data from body extremities to the wearable PC. Inertial orientation tracking combined with RF positioning are also tried to provide an accurate method for determining orientation and location. It describes a system designed to determine the posture of an articulated body in real time. Finally ,this work describes the design, implementation ,calibration algorithm for the sensors and testing of inertial tracking system of human limb segment.

IV.OBJECTIVES

Based on the above discussion, the objectives of the present research work are,

- Orientation tracking of human limb segments using three axis inertial sensors.
- Calibration of individual sensors without the use of any specialized equipment .
- Sufficient dynamic response and update rate (100 HZ or better) to capture faster human body limb motion.
- Ability to change the three stepper motors rotation according to the assigned threshold value.
- Three sensors are attached on the human limb, if the threshold value attains 360 and below, then the three stepper motors rotates in the forward direction. Finally , axis direction and three sensor data are also displayed graphically in the computer as per the limb movement of the human body.
- Three sensors are attached on the human limb, if the threshold value attains 400 and above then the three stepper motors rotates in the reverse direction. Finally , axis direction and three sensor data are also displayed graphically in the computer as per the limb movement of the human body.
- If the sensors are not attached on the human limb, ,the threshold value, rotation of stepper motors ,axis direction and the three sensor data all this parameter should lie in the initial condition.
- Automatic accounting for the peculiarities related to the mounting of a sensor on an associated limb segment.
- Creation of data files for recording data relating to limb as per the embedded software Filter.
- Use of Bluetooth technology will enable sensors to wirelessly transmit data from body extremities to the wearable PC.
- RF positioning are also tried to provide an accurate method for determining orientation and location .
- It describes the design, implementation ,calibration of the sensors and testing of inertial tracking system of human limb segment.

III.DESIGN EQUATIONS OF LIMB

Force Calculations of Joints

The point of doing force calculations is for motor selection. We must make sure that the motor we choose can not only support the weight of the robot arm, but also what the robot arm will carry (the blue ball in the image below).

The first step is to label the FBD(Diagram 1), with the robot arm stretched out to its maximum length.

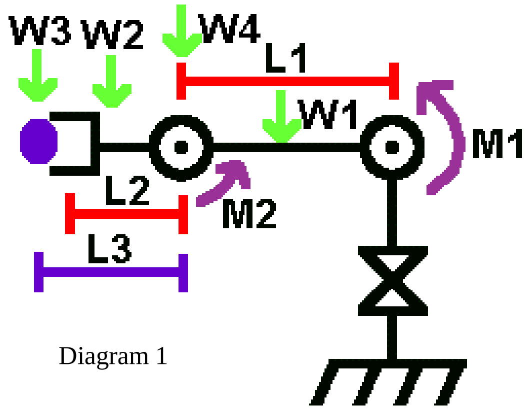


Diagram 1

Next we do a moment arm calculation, multiplying downward force times the linkage lengths. This calculation must be done for each lifting actuator. This particular design has just two DEGREE OF FREEDOM that requires lifting, and the center of mass of each linkage is assumed to be Length/2.

Torque About Joint 1:

$$M1 = L1/2 * W1 + L1 * W4 + (L1 + L2/2) * W2 + (L1 + L3) * W3$$

Torque About Joint 2:

M2 = L2/2 * W2 + L3 * W3

Forward Kinematics

Forward kinematics is the method for determining the orientation and position of the end effector, given the joint angles and link lengths of the robot arm. For our robot arm, here we calculate end effector location with given joint angles and link lengths.

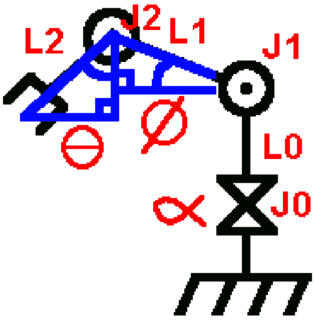


Diagram 2

Assume that the base is located at x=0 and y=0. The first step would be to locate x and y of each joint as shown in Diagram2

Joint 0 (with x and y at base equaling 0):

x0 = 0
y0 = L0

Joint 1 (with x and y at J1 equaling 0):

cos(psi) = x1/L1 => x1 = L1*cos(psi)
sin(psi) = y1/L1 => y1 = L1*sin(psi)

Joint 2 (with x and y at J2 equaling 0):

sin(theta) = x2/L2 => x2 = L2*sin(theta)
cos(theta) = y2/L2 => y2 = L2*cos(theta)

End Effector Location (make sure your signs are correct):

x0 + x1 + x2, or 0 + L1*cos(psi) + L2*sin(theta)
y0 + y1 + y2, or L0 + L1*sin(psi) + L2*cos(theta)
z equals alpha, in cylindrical coordinates

Inverse Kinematics

Inverse kinematics is the opposite of forward kinematics. This is when we have a desired end effector position, but need to know the joint angles required to achieve it. The robot sees a kitten and wants to grab it, what angles should each joint go to? Although way more useful than forward kinematics, this calculation is much more complicated too.

psi = arccos((x^2 + y^2 - L1^2 - L2^2) / (2 * L1 * L2))

theta = arcsin((y * (L1 + L2 * c2) - x * L2 * s2) / (x^2 + y^2))

where c2 = (x^2 + y^2 - L1^2 - L2^2) / (2 * L1 * L2); and s2 = sqrt(1 - c2^2);

There is the very likely possibility of **multiple, sometimes infinite, number of solutions** (as shown below). How would the arm choose which is optimal, based on torques, previous arm position, gripping angle, etc.? There is the possibility of **zero solutions**. Maybe the location is outside the workspace, or maybe the point within the workspace must be gripped at an impossible angle(Diagram 3).

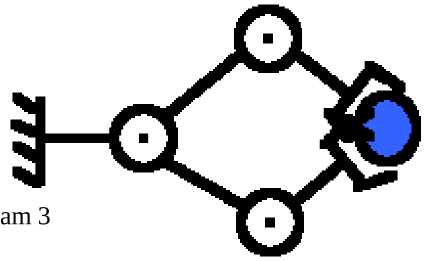


Diagram 3

Singularities, a place of infinite acceleration, can blow up equations and/or leave motors lagging behind (motors cant achieve infinite acceleration).

And lastly, exponential equations take forever to calculate on a microcontroller. No point in having advanced equations on a processor that cant keep up.

Motion Planning

Motion planning on a robot arm is fairly complex so I will just give you the basics.

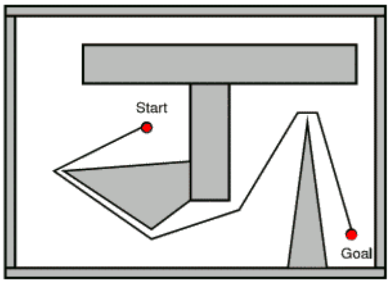


Diagram 4

Suppose the robot arm has **objects within its workspace** (Diagram 4), how does the arm move through the workspace to reach a certain point? To do this, assume the robot arm is just a simple mobile robot navigating in 3D space. The end effector will traverse the space just like a mobile robot, except now it must also make sure the other joints and links do not collide with anything too. This is extremely difficult to do . . .

What if you the robot end effector to **draw straight lines** with a pencil? Getting it to go from point A to point B in a straight line is relatively simple to solve. What the robot should do, by using inverse kinematics, is go to many points between point A and point B. The final motion will come out as a smooth straight line. We can not only do this method with straight lines, but curved ones too. On expensive professional robotic arms all we need to do is program two points, and tell the robot how to go between the two points (straight line, fast as possible, etc.).

Velocity (and more Motion Planning)

Calculating end effector velocity is mathematically complex, so we will go only into the basics. The simplest way to do it is assume the robot arm (held straight out) is a rotating wheel of L diameter. The joint rotates at Y rpm, so therefore the velocity is

Velocity of end effector on straight arm = 2 * pi * radius * rpm

However the end effector does not just rotate about the base, but can go in many directions. The end effector can follow a straight line, or curve, etc.

With robot arms, the quickest way between two points is often not a straight line. If two joints have two different motors, or carry different loads, then max velocity can vary between them. When we tell the end effector to go from one point to the next, we have two decisions. Have it follow a straight line between both points, or tell all the joints to go as fast as possible - leaving the end effector to possibly swing wildly between those points.

In the diagram 5 the end effector of the robot arm is moving from the blue point to the red point. In the top example, the end effector travels a straight line. This is the only possible motion this arm can perform to travel a straight line. In the bottom example, the arm is told to get to the red point as fast as possible. Given many different trajectories, the arm goes the method that allows the joints to rotate the fastest.

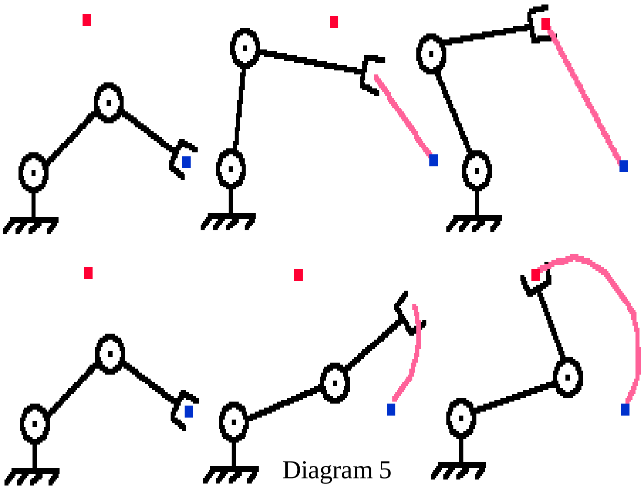


Diagram 5

There are many deciding factors to select the best method. Usually we want straight lines when the object the arm moves is really heavy, as it requires the momentum change for movement (momentum = mass * velocity). But for maximum speed (perhaps the arm isn't carrying anything, or just light objects) we would want maximum joint speeds.
Now suppose we want the robot arm to operate at a certain rotational velocity, how much torque would a joint need?

First, lets go back to our Functional Block Diagram (Diagram 6):

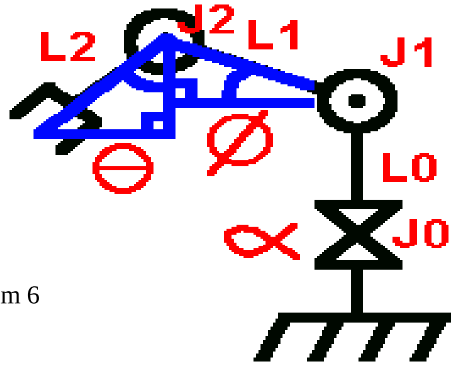


Diagram 6

Now lets suppose we want joint J0 to rotate 180 degrees in under 2 seconds, what torque does the J0 motor need? Well, J0 is not affected by gravity, so all we need to consider is momentum and inertia. Putting this in equation form we get this:

torque = moment_of_inertia * angular_acceleration

breaking that equation into sub components we get:

torque = (mass * distance^2) * (change_in_angular_velocity / change_in_time) and

change_in_angular_velocity = (angular_velocity1)-(angular_velocity0)

angular_velocity = change_in_angle / change_in_time

Now assuming at start time 0 that angular_velocity0 is zero, we get
torque = (mass * distance^2) * (angular_velocity / change_in_time)

where distance is defined as the distance from the rotation axis to the center of mass of the arm:
center of mass of the arm = distance = 1/2 * (arm_length)

(use arm mass)

but we also need to account for the object the arm holds:

center of mass of the object = distance = arm_length

(use object mass)

So we calculate torque for both the arm and then again for the object, then add the two torques together for the total:

torque(of_object) + torque(of_arm) = torque(for_motor)

And of course, if J0 was additionally affected by gravity, add the torque required to lift the arm to the torque required to reach the velocity needed.

V IMPLEMENTATION OF INERTIAL TRACKING OF HUMAN LIMB SEGMENTS

The implementation of inertial tracking of human limb segment is shown in figure 1. Three inertial sensors are mounted on the body of the human limb segment. The analog output from the limb for adults is 20mv (infants is 5 mv)and given to signal conditioner circuit here for better ADC accuracy it amplify the 20mv to 5v and the corresponding amplifier gain is 5000/20 250.

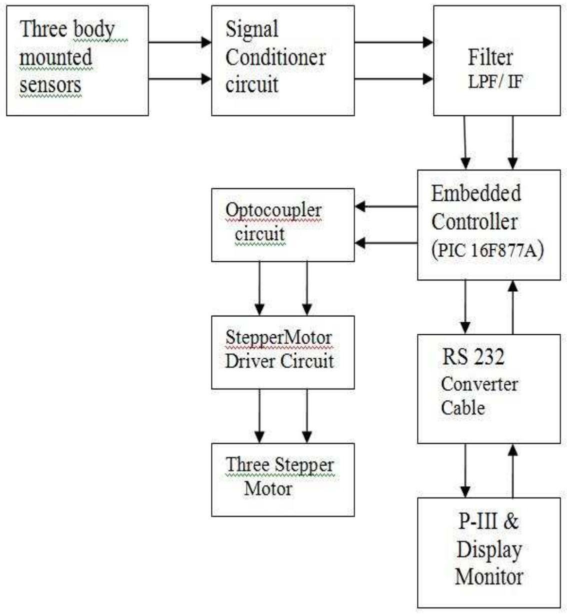


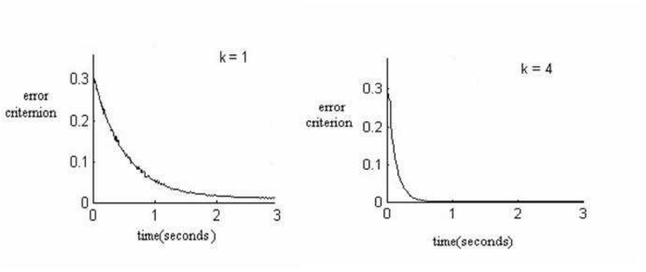
Figure 1 : Prototype Inertial Tracking Of Human Limb Segment

The 5v analog output is given to filter circuit which provides high speed noise filtering output with constant frequency approximately 100HZ and to Embedded Controller on which the software filter algorithm as well as analog to digital converter is implemented. The output is digitized by an associated inbuilt A\D converter .The digitized output from an Embedded Controller by a RS 232 converter is connected to the PC. All data processing and calculations are performed by software running on this single processor.

An appropriate electronic optocoupler circuit is used for isolation between the three stepper motors and an Embedded Controller. The driver circuit drives the three stepper motor in different direction. The electronic system will be suitable upto 5A for 70kg cm stepper motor but in this research the stepper motor used is only 7kg cm The rotation depends upon the human limb movement on all the axis like up/down ,roll, elevation and azimuth as per the assigned threshold value. The threshold value ,three stepper motor direction, axis movement and graphical representation and sensing system all this implemented data can be displayed on the monitor by means of using C programming language. The optimal filter theory to the filter software is done by Flash Embedded Controller and visual simulation software run on a single standard Pentium III processor in computers RH (Barnett,LO'Cull,2004).Use of Bluetooth technology will enable sensors to wirelessly transmit data from body extremities to the wearable PC. Inertial orientation tracking combined with RF positioning are also tried to provide an accurate method for determining orientation and location. Finally, the prototype sensor overall system hardware kit is shown in figure 2 .

Figure 2 Prototype Sensor Overall System Hardware Kit

Figure: 3 Error Convergence of 30 degree Transient Errors%.



Static Stability of the system

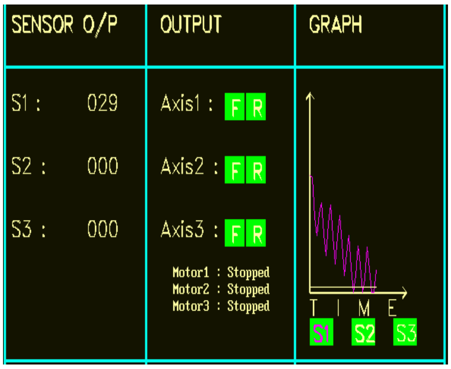
Figure 3 plots the magnitude of the quaternion filter criterion function versus time. The drift characteristics of the quaternion filter algorithm and the MARG sensor over extended periods were evaluated using static tests. Average total drift is about 1%. During the experiment shown, the filter gain, *k* was set to unity. It is expected that increasing the filter gain to 4.0 would reduce the drift error by a factor of four or to about 0.25 percent. Further experiments indicated that nearly all drift was due to bias in the rate sensors. Experiments are currently underway using improved sensors containing rate-sensor capacitive coupling conditioning circuitry designed to remove these biases.

Dynamic Response of the system

Preliminary experiments were conducted to establish the accuracy of the orientation estimates and the dynamic response of the system. The preliminary test procedure consisted of repeatedly cycling the sensor through various angles of roll, pitch and yaw at rates ranging from 10 to 30 deg./sec. Accuracy was measured to be better than one degree. The overall smoothness of the plot shows excellent dynamic response.

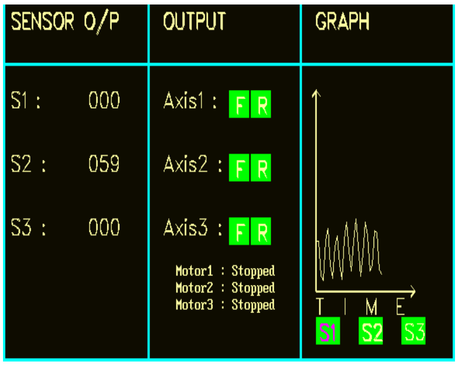
VI. EXPERIMENTAL TEST RESULTS OF HUMAN LIMB SEGMENT

Figure 4



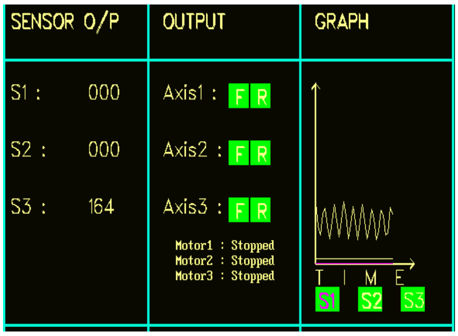
S1 : 029
S2 : 000
S3 : 000
Motor1 : Stopped
Motor2 : Stopped
Motor3 : Stopped

Figure 5



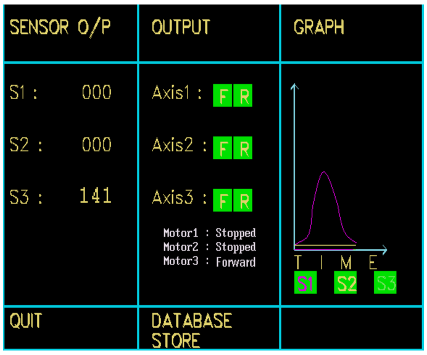
S1 : 000
S2 : 059
S3 : 000
Motor1 : Stopped
Motor2 : Stopped
Motor3 : Stopped

Figure 6



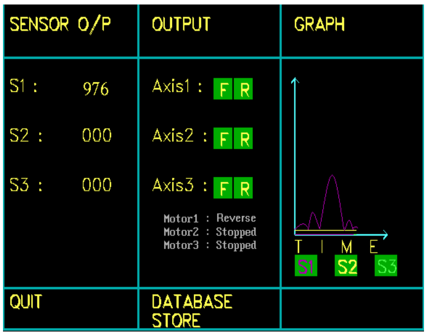
S1 : 000
S2 : 000
S3 : 164
Motor1 : Stopped
Motor2 : Stopped
Motor3 : Stopped

Figure 7



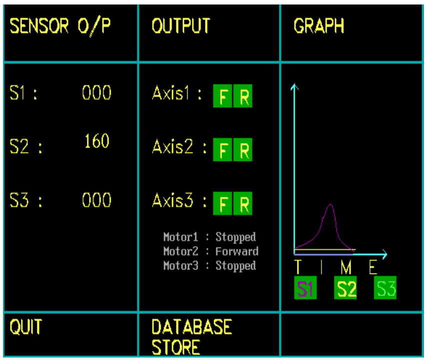
S1 : 000
S2 : 000
S3 : 141
Motor1 : Stopped
Motor2 : Stopped
Motor3 : Forward

Figure 8



S1 : 976
S2 : 000
S3 : 000
Motor1 : Reverse
Motor2 : Stopped
Motor3 : Stopped

Figure 9



S1 : 000
S2 : 160
S3 : 000
Motor1 : Stopped
Motor2 : Forward
Motor3 : Stopped

Table 1

Stepper Motor	Threshold Value		Axis Rotation	Sensors
	400 & above	360& below		
M_1	Reverse	Forward	Forward/ Reverse	S_1
M_2	Reverse	Forward	Forward/ Reverse	S_2
M_3	Reverse	Forward	Forward/ Reverse	S_3

Table 2 Simulation Results Of Human Limb Segment

Stepper Motor M$_1$	Stepper Motor M$_2$	Stepper Motor M$_3$	Sensor S$_1$	Sensor S$_2$	Sensor S$_3$	Axis Rotation	Threshold Value
Forward	Stopped	Stopped	**110**	000	000	Forward	360 and below
Stopped	**Forward**	Stopped	000	**160**	000	Forward	
Stopped	Stopped	**Forward**	000	000	**141**	Forward	
Reverse	Stopped	Stopped	**450**	000	000	Reverse	400 and above
Stopped	**Reverse**	Stopped	000	**580**	000	Reverse	
Stopped	Stopped	**Reverse**	000	000	**650**	Reverse	

The above results were obtained using the hardware and software to achieve an update rate of 100 Hz. The roll, pitch, and yaw test results are presented in Figures 4 , 5, 6 & 7 respectively. The smoothness of the graphs indicates excellent dynamic response. It is expected that adjusting the filter gain values that improves the overall accuracy and dynamic response. The transition times observed in the plots are around 4.5-5 seconds as expected for a 10-degree per second rotation rate to 45 degrees. In qualitative tests, the system was able to track the limb segment, including those in which pitch equaled 90 degrees the same orientations normally cause singularities in Euler angle filters. The qualitative tests also show that the system could easily be combined with a simulation program and track motion in real time.

Fig10 Roll Excursion for 45 deg

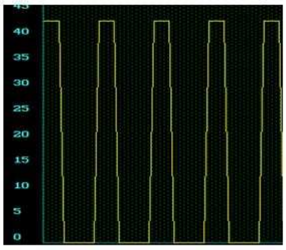
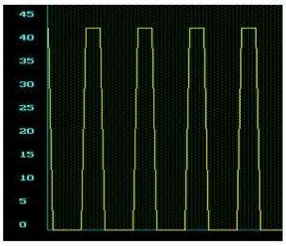


Fig11 Pitch Excursion for 45 deg



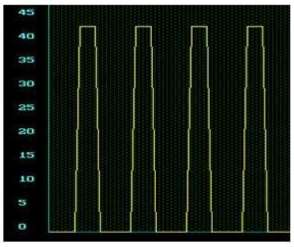


Fig12 Yaw Excursion for 45deg

The purpose of the human body tracking system is to estimate the orientation of multiple human limb segments and use the resulting estimates to set the posture of the human body model that is visually displayed. Numerous experiments were conducted to qualitatively evaluate and demonstrate this capability.

In each experiment three sensors where attached to the limb segments to be tracked. Due to the minimal number of sensors available, tracking was limited to a single arm or leg. In the case of arm and limb segments, sensor attachment was achieved through the use of elastic bandages. In most cases this method appeared to keep the sensors fixed relative to the limb. Body tracking was also performed using various gains.

VII.**CONCLUSIONS**

This research has demonstrated an alternative technology for tracking the posture of an articulated rigid body. High speed Embedded Controller avoids the electronic complexity , Bluetooth technology enables sensors to wirelessly transmit data from body to PC and the use of inertial sensors determine the orientation of link in the rigid body. RF positioning provides the source less capability of inertial sensing and enables tracking of multiple users over a wide area. At the core of the system is an efficient complementary filter that uses a quaternion representation of orientation and the filter can continuously track the orientation of human body limb segments (Robert B. McGhee2000).. Drift corrections are also made. This research overcomes the analysis and calculations used by the previous researchers by the technology of Embedded Controller. Embedded software filter process the data from three axis inertial sensors which is attached on the human limb segment. Sensor calibration is achieved without using any specialized equipment .Accurate calibration algorithm compensates the misalignment between sensor and limb segment co-ordinate axis. Hybrid stepper motors with associated mechanical elements is used to move the limbs on all the axis like up/down, roll, elevation and azimuth. The implemented system tracks human limb segments accurately with a 100 Hz update rate. Experimental results demonstrate the inertial orientation estimation is a practical method of tracking human body posture. With additional sensors, the architecture produced could be easily scaled for full body tracking. Due to its source less nature, tracking could overcome many of the limitations of motion tracking technologies currently in widespread use. It is potentially capable of providing wide area tracking of multiple users for synthetic environment and augmented reality applications.

VIII. REFERENCES

[1] An, K.N., Chao, E.Y., Cooney, W.P., and Linscheid, R.L., 1979, “Normative Model of Human Hand for Biomechanical Analysis,” J. Biomechanics, vol. 12, pp. 775-788.

[2] Bennet, D.J., Hollerbach, J.M., 1990, “Closed-loop Kinematic Calibration of the Utah-MIT Hand,” in Experimental Robotics I: The First International Symp., V. Hayward, O. Khatib, (eds.), Springer-Verlag, N.Y., pp. 539-552.

[3] Cooney, W.P., Lucca, M.J., Chao, E.Y.S., Linschied R.L., 1981, “The kinesiology of the thumb trapeziometacarpal joint,” J. Bone Joint Surg. 63A:1371-1381.

[4] Fischer, M., van der Smagt, P., Hirzinger, G., 1998, “Learning Techniques in a Dataglove Based Telemanipulation System for the DLR Hand,” 1998 IEEE ICRA, pp1603-1608.

[5] Hollister, A., Buford, W.L., Myers, L.M., Giurintano, D.J, Novick, A., 1992, “The Axes of Rotation of the Thumb Carpometacarpal Joint,” J. of Orthopaedic Res., vol. 10, pp. 454-460.

[6] Khatib, O., 1987, “Unified Approach for motion and force control of robot manipulators: the operational space formulation,” IEEE J. of Robotics and Automation, vol. 3, no. 1, pp. 43-53.

[7] Kramer, J.F., “Determination of Thumb Position Using Measurements of Abduction and Rotation,” U.S. Patent #5,482,056.

[8] Kuch, J.J., Huang, T.S., 1995, “Human Computer Interaction via the Human Hand: A Hand Model,” 1995 Asilomar Conf. on Signals, Systems. and Computers. pp. 1252-1256.

[9] Rohling, R.N., Hollerbach, J.M., 1993, “Calibrating the Human Hand for Haptic Interfaces,” Presence, vol. 2 no. 4, pp.281-296.

[10] Rohling, R.N, Hollerbach, J.M., Jacobsen, S.C., 1993, “Optimized Fingertip Mapping: A General Algorithm for Robotic Hand Teleoperation,” Presence, vol. 2 no. 3, pp. 203-220.

[11] Turner, M.L., Gomez, D.H. Tremblay, M.R. and Cutkosky, M.R., 1998, “Preliminary Tests of an Arm-Grounded Haptic Feedback Device in Telemanipulation,” 1998 ASME IMECE Symp. on Haptic Interfaces. pp.145-149.

[12] Turner, M.L., Findley, R.P., Griffin, W.B., Cutkosky, M.R., Gomez, D.H., 2000, “Development and Testing of a Telemanipulation System with Arm and Hand Motion,” Accepted to 2000 ASME IMECE Symp. on Haptic Interfaces.

[13] Wampler, C.W., Hollerbach, J.M., Arai, T., 1995, “An Implicit Loop Method for Kinematic Calibration and its Application to Closed-chain mechanisms,” IEEE Trans. Robotics and Automation, vol. 11, no. 5, pp. 710-724.

[14] Wright, A.K., Stanisic, M.M., 1990, “Kinematic Mapping between the EXOS Handmaster Exoskeleton and the Utah-MIT Dextrous Hand,” 1990 IEEE Int’l Conf. on Systems Engineering, pp. 809-811.

[15] www.societyofrobots.com /robot building ideas