International Journal of Humanoid Robotics

MOTION CAPTURE FROM INERTIAL SENSING FOR UNTETHERED HUMANOID TELEOPERATION

NATHAN MILLER

Department of Computer Science
California State University Bakersfield
Bakersfield, CA 93311
nathanlmiller@yahoo.com

ODEST CHADWICKE JENKINS
MARCELO KALLMANN
MAJA J MATARIĆ

Interaction Lab
Center for Robotics and Embedded Systems
941 W. 37th Place, SAL 300
Los Angeles, CA, USA 90089
kallmann,cjenkins,mataric@usc.edu

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

We describe the design of a modular system for untethered real-time kinematic motion capture using sensors with inertial measuring units (IMUs). Our system is comprised of a set of small and lightweight sensors. Each sensor provides its own global orientation (3 degrees of freedom) and is physically and computationally independent, requiring only external communication. Orientation information from sensors is communicated via wireless to host computer for processing. We present results of the real-time usage of our untethered motion capture system for teleoperating the NASA Robonaut[1]. We also discuss potential applications for untethered motion capture with respect to humanoid robotics.

Keywords: Humanoid Robotics, Robot Teleoperation, Motion Capture, Inertial Sensing.

1. Introduction

The ability to capture kinematic motion of human subjects is emerging as an invaluable tool in the control of robotic systems. Robotics research has begun to utilize motion capture for a variety of purposes, including teleoperation, implementing autonomous controllers, learning from demonstration, and human-robot interaction. The current state of motion capture systems, however, imposes considerable physical limitations (e.g., optical, exoskeleton, electro-magnetic systems). Available systems do not provide accuracy and reliability suitable for robot control (e.g., vision-based

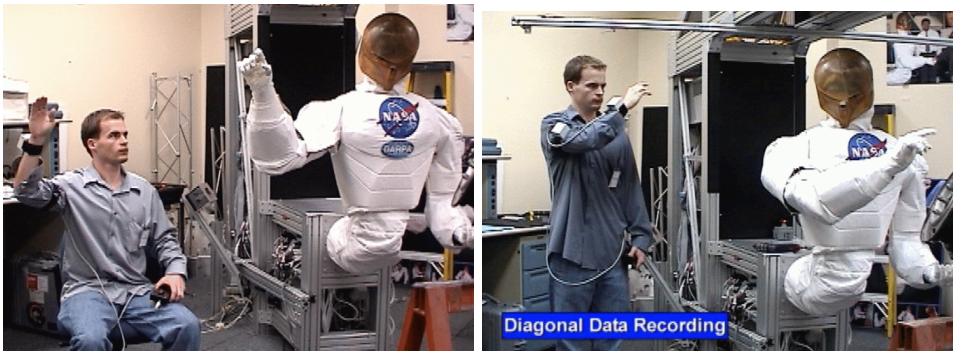

Fig. 1. A human teleoperating of the NASA Robonaut using our IMU-based motion capture system.

and fiberoptic systems), or cannot be used for real time applications (e.g., optical and vision-based systems). In addition, commercially available capture systems can be cost-prohibitive. Because of these limitations, human performers are greatly limited in their ability to behave naturally or act in a manner suitable for robot control. Informally, these limitations result in "mocap driving the person" rather than "the person driving the mocap".

In order to be truly effective for robotics, our perspective is that motion capture must be able to accurately perform in unstructured situations without sigificantly encumbering a performer. The development of such systems would effectively bring motion capture out of the lab and into ubiquitous usage. The benefits of untethered and uncumbersome motion capture are three-fold: 1) increased information for human-robot interaction, 2) more representative motion time-series data for robot learning, and 3) greater accessibility to robot control for non-technical people.

Towards these ends, we have developed a "motion suit" (Figure 1) comprised of small and lightweight independently functioning sensor units. Each sensor independently provides its global orientation using inertial measurment units (IMUs). Orientation information is communicated to a host computer to drive any desired kinematic model. The primary advantages in design of our IMU-based motion suit are that: 1) communication between sensors and a host computer does not represent a physical limitation, 2) the form factor for IMUs is small and progressively decreasing in size, and 3) scalability for adding, removing, or altering nodes in the sensor topology.

In this paper, we describe our design of an IMU-based motion suite and its application to teleoperating the NASA Robonaut[1]. We compare and contrast the suitability of our motion suit and other techniques for motion capture with respect to robot control.

2. Related Work

The impact of motion capture for work in humanoid robotics and robotics in general is quite significant. This impact can be evidenced by the wealth of work that has been generated by researchers developing and using capture systems towards robot control and interaction. A brief glance over the use of motion capture for humanoid robotics includes work for teleoperation[2,3,4,5], robot locomotion[6,7], evaluating robot control[8], learning from demonstration[9,10,11,12,13], and humanoid-robot interaction[14]. Additionally, work in human-robot interaction for non-humanoid robots, for purposes such as learning from demonstration[15], often requires structured perception modalities that are less accurate and reliable than motion capture. Also, many relevant techniques utilizing motion capture for kinematic character control and human motion understanding have been developed in computer animation and computer vision.[16,17,18,19,20,21,22]

In addition to being cost-prohibitive, the large majority of commercial motion capture systems for robotics suffers from one of two major shortcomings. The first shortcoming pertains to the overly restrictive physical limitations imposed by the motion capture system. Systems that are typically subject to this shortcoming are more traditional methods for motion capture, including exoskeletons[23,3], electro-magnetic systems[24], and optical systems[25,26]. Exoskeletons are typically built as a passive exterior skeleton consisting of potentiometers at joint locations connected by rigid structures (e.g., poles). When wearing the exoskeleton, the human user's kinematic configurations are obtained via reading values from the potentiometers. Potentiometer-like devices provide sensing local to each joint. Thus, each sensor is dependent upon having physical references on adjacent rigid structures and necessitates an additional overly encumbering physical skeleton. In contrast, typical electro-magnetic and optical systems are dependent upon equipment external to the human user. Electro-magnetic systems require the presence of a transmission source to provide a reference for sensors worn on the user. This transmission source must in close proximity to the user and is very sensitive to metallic interference. Optical systems consist of multiple (typically infrared) cameras that are used to triangulate markers worn on the user. These systems require an additional step for estimating kinematic configurations from marker positions. More importantly, such systems are sensitive to "marker dropout" due to occluded markers, requiring user to perform in highly structured environments.

Our motion suit avoids this problems by using inertial sensing to estimate orientation in global coordinates. Unlike optical and electro-magnetic systems, inertial sensors require only the Earth's gravitation force and magnetic field to function, providing sensing untethered to external equipment. Because each sensor estimates orientation independently, no physical linkage between adjacent limbs is necessary. Our motion suit does not require external physical structures, only attachment of the sensor with a small form factor to a rigid body.

The second shortcoming of existing motion capture systems is the lack of reliable

and accurate performance in various environments. These systems typically represent more cutting-edge techniques due to either the difficulty of the approach (e.g., vision-based capture[27,28,29,30,5]) or the immaturity of the underlying technology. In terms of the latter category, several new technologies for motion capture have appeared in recent years, including fiber-optic systems[31], smart textile systems[32], and other inertial sensing capture systems[33,34,35]. Unlike the majority of these, our design and algorithms are geared towards generally accessible motion capture through simple and cost-efficient implementations on standard hardware.

3. Estimating Rigid Body Orientation from Inertial Sensing

We present an algorithm for estimating the orientation of a rigid body using the accelerometer, magnetometer, and gyroscope readings provided by an IMU sensor. The algorithm presented here is similar to the Complementary Quaternion Attitude Filter presented by Bachmann et al.[36,34] However, our algorithm is different in several aspects, including the removal of the Gauss-Newton iteration method for execution on the Atmel Mega32 microcontroller. This microcontroller supports floating point numbers in software only and has a limited program storage size of 32 Kb.

Our basic approach is to integrate gyroscope signals for orientation estimates and drift errors with accelerometer and magnetometer readings. To do this, we must assume the following:

- The accelerometers provide a good approximation of the gravity vector. That is, only small accelerations are experienced that cause deviation from the true gravity vector. This assumption imposes a limit on the maximum acceleration that the capture subject can impart upon the sensor before errors in orientation will occur.
- The magnetometers provide a good approximation of the Earth's magnetic field vector. This requirement limits the effectiveness of the orientation sensor in highly distorted magnetic fields, or where the magnetic field fluctuates significantly from one point to the next.

These limitations can be significant and should be considered when determining if the target application is plausible for this system[35].

The Earth's gravity and magnetic field vectors are obtained by placing the sensor in a predefined orientation relative to gravity (in our application, the sensor communication ports pointing straight up in the opposite direction of the Earth's gravity) and cycling the suit power. Once cycled, the microcontroller samples the accelerometer and magnetometer sensors a predetermined number of times, calculates the means, normalizes the resulting vectors, and stores them in g_{ref} and m_{init}, respectively. In our application, g_{ref} and m_{init} represented by quaternions[37] with zero real part. m_{init} is further processed to form m_{ref} by correcting for the declination angle, the angle that the Earth's magnetic field makes with the horizontal.

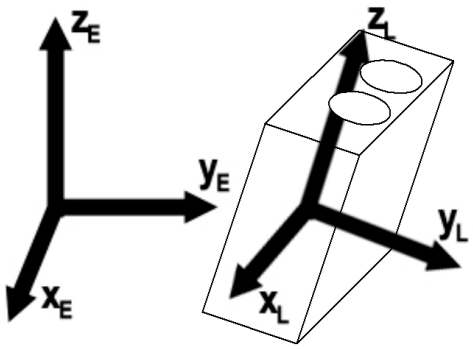

Fig. 2. Example of the difference between global frame E and local sensor frame L after initialization. The frame L is rigidly attached to the sensor body. Communication (minidin) ports are shown as circles on the sensor.

This correction is accomplished by taking the cross product between the g_{ref} and m_{init} to obtain an axis of rotation and determining the angle ϕ between the vectors. The correction angle Θ_{correct} is calculated to be $90 - \phi$ degrees, and m_{init} is rotated counterclockwise along the axis of rotation by Θ_{correct} degrees. g_{ref} and m_{ref} define the global Earth reference frame E.

The algorithm which resolves the sensor orientation relative to the Earth's frame E is a linear step-correct algorithm based on quaternion algebra. Quaternions are used because they solve the gimbal-lock problem inherent in Euler angles. A quaternion q consists of a real part r, along with three imaginary parts x, y, and z expressed as:

$$q = \begin{bmatrix} r \\ x \\ y \\ z \end{bmatrix} \tag{1}$$

or more formally expressed as $q = r + xi + yj + zk$.

The initial orientation of the sensor is determined by finding the difference between the local sensor frame L (rigidly attached to the sensor body) and the global frame E (Figure 2). This difference is used to construct the initial orientation quaternion q_{ref} relative to E, where $q_{\text{ref}} = q_t$ at time $t = 0$. To determine the orientation at time $t = t_i + \Delta_t$ we must integrate the gyroscope angular rate information. This is done by setting the angular rate quaternion ω as follows:

$$\omega = \begin{bmatrix} 0 \\ x'\Delta_t \\ y'\Delta_t \\ z'\Delta_t \end{bmatrix} \tag{2}$$

where x', y', and z' are the angular rates in the local frame provided by the gyroscopes (in degrees/sec). ω is multiplied by the current orientation quaternion q_t to obtain the change in orientation Δ_q. Δ_q is added to q_t to obtain the new orientation $q_{t+\Delta_t}$, as stated in the following step calculations:

$$\Delta_q = (1/2)q_t\omega \tag{3}$$

$$q_{t+\Delta_t} = q_t + \Delta_q \tag{4}$$

where quaternion multiplication is defined as:

$$q_1 q_2 = \begin{bmatrix} r_1r_2 - x_1x_2 - y_1y_2 - z_1z_2 \\ (r_1x_2 - x_1r_2 - y_1z_2 - z_1y_2)i \\ (r_1y_2 - x_1z_2 - y_1r_2 - z_1x_2)j \\ (r_1z_2 - x_1y_2 - y_1x_2 - z_1r_2)k \end{bmatrix} \tag{5}$$

This step equation would be sufficient if the gyroscopes were capable of providing rotation rate values without error or noise. Unfortunately, readings from gyroscopes incur significant noise and suffer from a variety of other errors which affect the signal. Therefore, we introduce an additional step for sensor correction, outlined in the following steps:

(1) g_{ref} and m_{ref} are converted to the local sensor frame by the following equations[36]:

$$g_{\text{local}} = q_t^* g_{\text{ref}} q_t \tag{6}$$

$$m_{\text{local}} = q_t^* m_{\text{ref}} q_t \tag{7}$$

where q^* is the complex conjugate of a quaternion defined as:

$$q^* = \begin{bmatrix} r \\ -x \\ -y \\ -z \end{bmatrix} \tag{8}$$

(2) g_{sensed} and m_{sensed} are the local sensor frame gravity and magnetic field vectors obtained from the current accelerometer and magnetometer readings.
(3) The error vectors g_{error} and m_{error} are the differences:

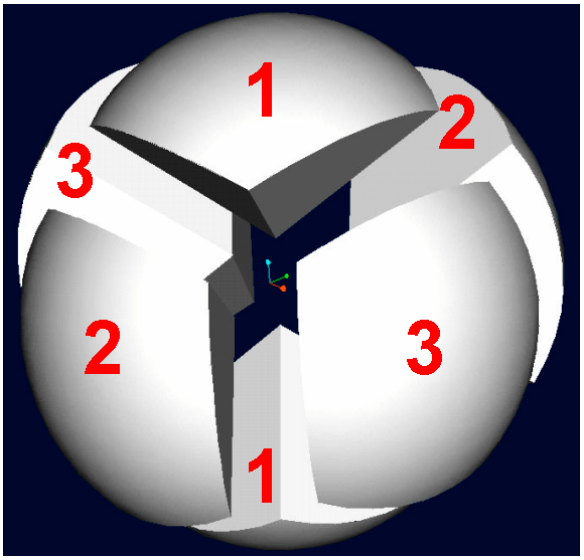

Fig. 3. Partitioning of space in the local sensor frame. g_{sensed} is checked against these partitions to determine which components of the gravity and magnetic field vectors are used to calculate the yaw, pitch, and roll angle corrections. Partition 1 lies along the z axis, partition 2 along the y axis, and partition 3 along the x axis.

$$g_{\text{error}} = g_{\text{local}} - g_{\text{sensed}} \tag{9}$$

$$m_{\text{error}} = m_{\text{local}} - m_{\text{sensed}} \tag{10}$$

Theoretically, if there is no error in orientation, then these two quaternions are zero. In practice, minor fluctuations in the sensor signals always results in small errors.

(4) g_{local}, g_{sensed}, and g_{error} are used with the law of cosines to determine the error angles ϕ (pitch) and Θ (roll) in all orientations, but are projected onto the local axis in the following manner:

- If g_{sensed} within partition 1: the angle ϕ is determined by the projection onto the x-z plane and the angle Θ is determined by the projection onto the y-z plane.
- If g_{sensed} within partition 2: the angle ϕ is determined by the projection onto the x-y plane and the angle Θ is determined by the projection onto the y-z plane.
- If g_{sensed} within partition 3: the angle ϕ is determined by the projection onto the x-z plane and the angle Θ is determined by the projection onto the x-y plane.

m_{local}, m_{sensed}, and m_{error} are used with the law of cosines to determine the

error angle μ (yaw) in all orientations, but are projected onto the local axis in the following manner:

- If g_{sensed} within partition 1: μ is determined by the projection onto the x-y plane.
- If g_{sensed} within partition 2: μ is determined by the projection onto the x-z plane.
- If g_{sensed} within partition 3: μ is determined by the projection onto the y-z plane.

(5) These angles are combined into the error angular rate quaternion ϵ that is used to obtain q_{correct}:

$$\epsilon = \begin{bmatrix} 0 \\ \Theta\Delta_t \\ \phi\Delta_t \\ \mu\Delta_t \end{bmatrix} \tag{11}$$

$$q_{\text{correct}} = (1/2)q_t\epsilon \tag{12}$$

The resulting step-correction computation is:

$$q_{t+\Delta_t} = q_t + \Delta_q - \alpha q_{\text{correct}} \tag{13}$$

where α is a scalar used to tune the algorithm for optimal performance.

(6) $q_{t+\Delta_t}$ is normalized as :

$$q_{t+\Delta_t} = \frac{q_{t+\Delta_t}}{\text{n}(q_{t+\Delta_t})} \tag{14}$$

The primary difference between our algorithm and Bachmann et al. is in the correction step. Bachmann et al. use the following equation as the correction step:

$$\Delta_q = [X^T X]^{-1} X^T \epsilon(q) \tag{15}$$

where $\epsilon(q)$ is defined as a 6x1 vector which contains the difference between the actual gravity and magnetic field vectors and computed gravity and magnetic field vectors respectively. The X matrix contains the partial derivatives of the computed measurement vector with respect to the components of the orientation quaternion.

Our correction step determines the yaw, pitch, and roll angles between the actual and sensed magnetic and gravity vectors and places these into an angular rate quaternion $\epsilon = [0, \Theta\Delta_t, \phi\Delta_t, \mu\Delta_t]^{-1}$. ϵ is multiplied by $(1/2) * q_t$ to obtain q_{correct}, which is multiplied by the scalar α, used to tune the algorithm for optimal performance.

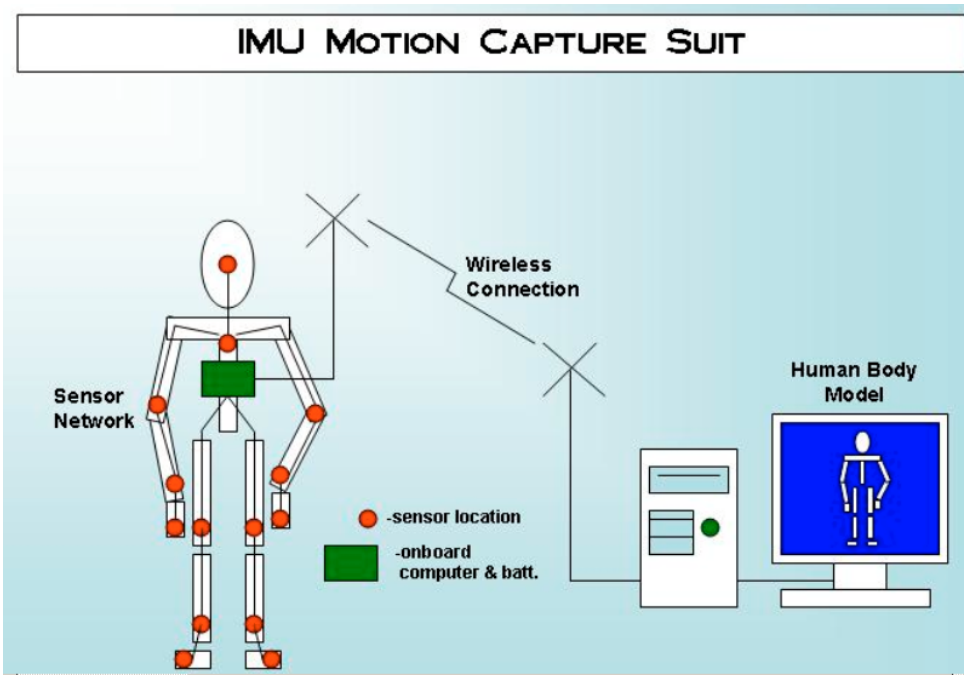

Fig. 4. Overview of our inertial sensor motion capture system.

4. Kinematic Motion Capture from Inertial Sensing Nodes

Our motion capture system consists of 3 hardware subsystems: the computer interface, the capture subject interface, and the individual sensor units. These subsystems are described below.

The computer interface consists of a single Atmel Mega32 microcontroller, a Linx Technologies TR-916-SC-PA transceiver, and a DLP Design DLP-USB245M USB FIFO adapter. The microcontroller provides the low-level control functionality and filters incoming packets for correct structure. In the current implementation, this subsystem acts as a relay between the transceiver and USB in receive mode only.

The capture subject interface consists of a single Mega32 microcontroller, a TR-916-SC-PA transceiver, 6v nicad battery, and 5 minidin ports in which sensors may be connected. The microcontroller acts as the Serial Peripheral Interface (SPI) master, requesting orientation packets from the connected sensors. Once a valid packet is buffered, it is sent over the transceiver which currently operates in send mode only. The radio subsystems operate reliably indoors to over 100 feet.

The inertial measurement units (IMUs) contain: a single Mega32 microcontroller; 12-bit, 11-channel Texas Instruments ADC; 3 axis magnetometer (consisting of 1 Honeywell HMC1051 and 1 Honeywell HMC1052); 3 axis accelerometer (con-

sisting of 2 Analog Devices ADXL311JE); and a 3 axis gyroscope (consisting of 3 Analog Devices ADXRS300ABG). Various passive components and a voltage regulator are also present. Each sensor is capable of resolving global orientation relative to the gravity and Earth's magnetic field vector. The sensors communicate with the capture subject radio as SPI slaves. Two minidin connectors allow multiple sensors to be chained together. The bare sensors have a 1.5x1.5.0.75 inch footprint (Figure 5).

Sensors can be added or removed as needed prior to the capture session. The limiting factor in our application is the bandwidth of the Linx radios (33,600 bps). Future plans include migrating to 802.11b, where the limiting factor will become the speed at which the SPI is able to operate. This is governed by the length of the physical connections as well as the Mega32 16Mhz operating speed.

The capture subject radio, operating as SPI master, polls the SPI bus for the presence of a particular sensor ID. Should a sensor with that ID be attached to the system, that sensor will respond to the SPI master and the orientation information will be transferred. If the ID is not found, the SPI master tries two additional times in case the sensor is in a critical section, then moves on to the next sensor ID. The maximum ID in our application is 15, although this could be changed in other applications. Therefore, the current system is scalable from 1 to 15 sensors.

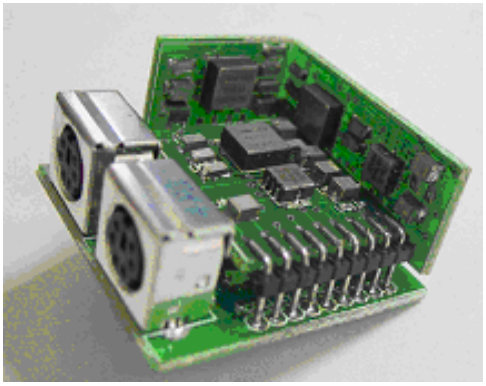

Fig. 5. Hardware for an assembled IMU sensor unit.

5. Robonaut Teleoperation from the Motion Suit

In the the following paragraphs, we address the three main issues for mapping global sensor orientation from a human performer onto the kinematics of a humanoid robot, such as Robonaut[1]:

(1) the parent-child order placement of the sensors;
(2) the decomposition of each sensed orientation into the correct joint angles;

(3) the offset adjustment for obtaining (approximately) equal postures.

Robotic kinematics structures require each joint orientation to be defined in its local frame. In this way, local orientations can be easily converted to the angular values to be achieved by each joint actuator. Individual sensors of the motion suit provide orientation with respect to a global frame. Consequently, orientation information from each sensor needs to be converted to a local frame, according to parent-child relationship of the sensors placement.

In our experiments, we used two sensors to capture arm motions. Sensor 1 was placed on the upper arm of the teleoperator, and sensor 2 was placed in the lower arm (Figure 7). Clearly, sensor 2 needs to be considered a child of sensor 1 in this setup.

Let q_1 and q_2 be the global orientations received by sensors 1 and 2, respectively. As q_1 is considered to be the root of our simple hierarchy of sensors, it does not need any corrections. However q_2 needs to be transformed to its local frame:

$$q_2^{local} = {q_1}^{-1} q_2 \tag{16}$$

Rotations q_1 and q_2^{local} are correctly expressed in their local frames, however they still need to be decomposed in the correct Euler angles sequence in order to obtain the correct angular rotations for each of the controlled Robonaut's arm joints. The trigonometric equations involved in the decomposition of a rotation into Euler angles can be found in an introductory Robotics textbook[38].

Further adjustments might be required, depending on the specific kinematics present in the target robotics structure. For instance, in Robonaut's case, the three axis of rotations in Robonaut's shoulder do not intersect at a single point. Such alignment exists in the sensed rotation q_1, and therefore a correction through Inverse Kinematics would be required for a perfect match. As the misalignment in Robonaut's shoulder is small, we did not include such correction step in the experiments presented here.

The final joint angles obtained are relative to the decided initial reference orientation (q_{ref} in Section 3) of the motion suit. Offsets must be added to these values in order to match the reference posture of the target structure. The offsets are defined by the joint values of the target structure (Robonaut's arm) required to achieve the same posture as the initialization posture defined for the motion suit. Some angular offsets requires negation in order to conform with the direction of orientation of the corresponding actuated joint.

Let α_1, α_2, α_3 be the three final Euler angles obtained from q_1, after all the corrections previoulsy explained (including the offset adjustment). Similarly, let α_4, α_5, α_6 be the final angles obtained from q_2^{local}.

Similar to a human arm, Robonaut's arm has a total of 7 DOFs (Figure 7). In the experiments presented in this paper, only 5 of them are being teleoperated. Angles α_1, α_2 and α_3 are mapped to the 3 joints affecting the rotation of Robonaut's

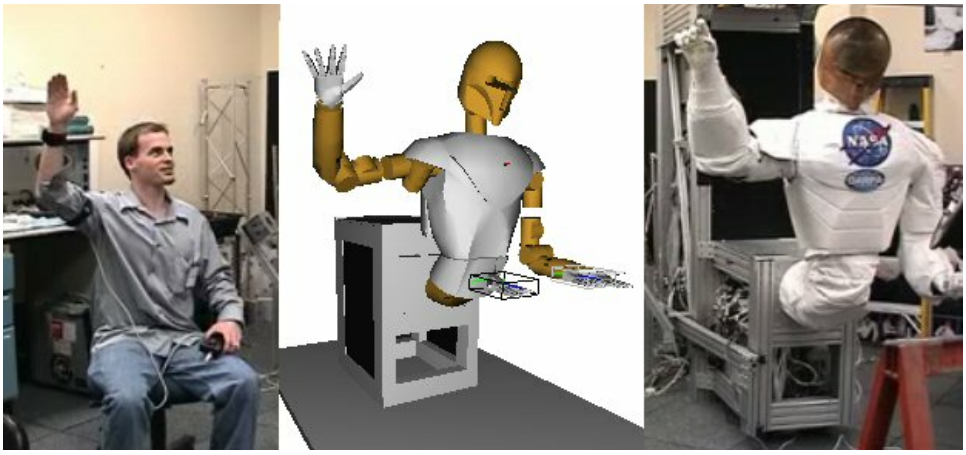

Fig. 6. Two sensors are placed in the right arm of the user; the radio unit can be seen in the left hand of the user (left image). Joint angles are derived from the sensors and mapped to the arm of a simulated model of Robonaut (center image). The obtained end-effector position and orientation in our simulator are sent via network to the Robonaut control interface for actuation (right image).

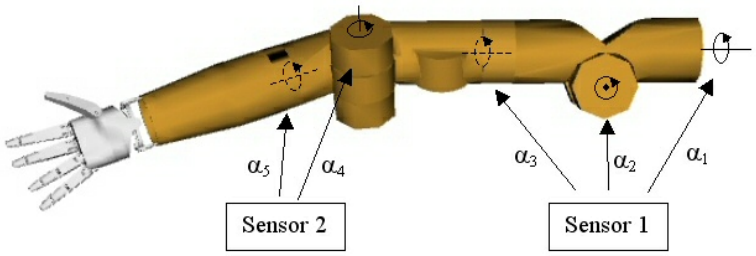

Fig. 7. Our model of Robonaut's right arm. In accordance with Robonaut, this arm has 7 DOF: 2 in the shoulder, 1 in the upper arm, 1 in the elbow, 1 in the lower arm, and 2 in the wrist.

shoulder. Angles α_4 and α_5 are mapped to the elbow flexion and lower arm twist. Angle α_6 is not required to be used. A third sensor placed in the user's hand would be required in order to teleoperate the remaining 2 joints of Robonaut's wrist.

Due to safety concerns, the maintainers of Robonaut only permit external control of the robot's arms through its built-in inverse kinematics (IK) mechanism. The IK mechanism requires commands be sent as the 6 DOF position and orientation of the endeffector in Robonaut's coordinates (in relation to a frame located within Robonaut's chest). Robonaut's built-in Jacobian-based Inverse Kinematics is then responsible to reconstruct the posture. This circumstance requires us to explicitly model Robonaut kinematically in order to translate motion capture data into IK

Fig. 8. Snapshot sequences from Robonaut teleoperation using our IMU-based motion capture suit for (top two rows) circle and (bottom two rows) box hand trajectories.

commands for Robonaut. Figure 6 illustrates such mapping from the sensors to the intermediate model and finally to Robonaut.

As illustrated in Figures 8 and 9, we evaluated our motion suit through six teleoperation trials. For each trial, data published by the intermediate model and Robonaut were saved to files. A comparison of the endeffector trajectories for these six trials are shown in Figure 9. As shown in the plots, teleoperation with our motion suit performed well with observable decrease in performance as the motion becomes more complicated. We attribute this performance decrease to human factors issues and latency in Robonaut actuation.

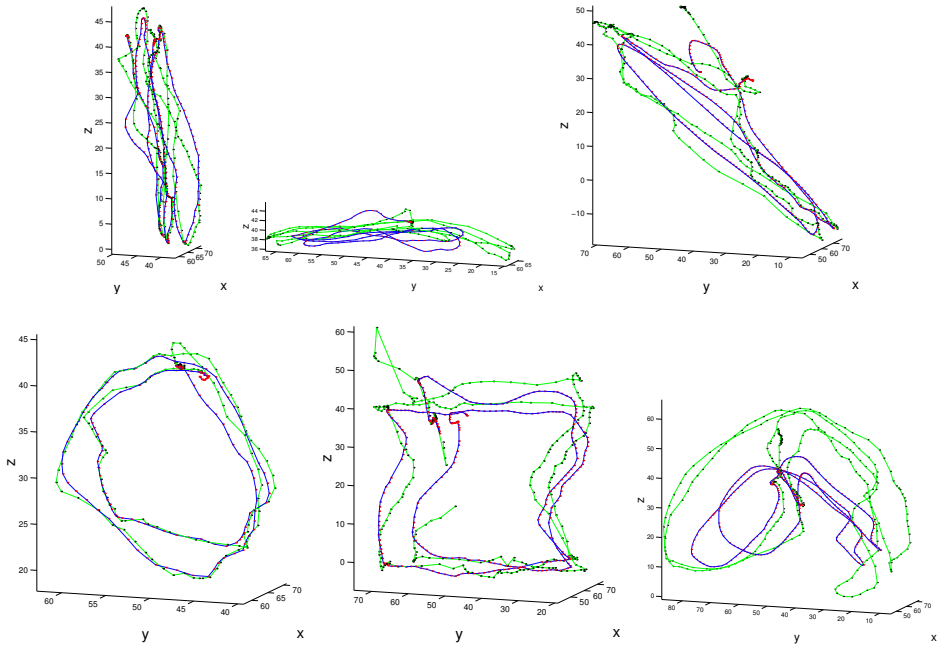

Fig. 9. Comparisons of endeffector trajectories for teleoperation of Robonaut. Two curves are shown in each plot for the trajectory achieved by Robonaut (in blue) and the trajectories of command positions sent from our intermediate model (in green) in Robonaut's coordinates. The top row contains plots for (left) vertical, (center) horizontal, and (right) diagonal endeffector movements. The bottom row contains plots for (left) circle, (center) box, and (right) infinity symbol endeffector movements.

Although not backed by recorded data, each recorded motion in the intermediate model was executed autonomously and without mishap on Robonaut after the teleoperation trials. Each recorded motion was deliberately teleoperated to begin and end near the same configuration. This property allowed the autonomous playback to continue looping through the motion repeatedly. Such experiments further demonstrate the applicability of dynamic trajectory formation[9,10].

6. Discussion

Our aim in developing an IMU-based motion capture system is to provide practical systems for utilizing human motion in the field as well as a lab. While many applications exist, we primarily want to use such systems for "on-the-fly" humanoid robot teleoperation and collecting corpuses of human motion for learning from demonstration. In its current state, our inertial capture system acheives this practicality. As we address in the following paragraphs, however, the system is subject to several caveats that can be addressed with realizable extensions.

Realizing practical untethered motion capture. There are four primary factors in realizing our aims for practical untethered motion capture: cost, equipment size, communications, and reliability. Considering the cost and size of existing capture systems, our motion suit is relatively inexpensive and compact. The equipment cost for each sensor is less than $300 USD and radio equipment is less than $100 USD. The sensors, radios, and batteries are were constructed from standard components, yet the form factor of the sensors and subject interface can easily be worn by a human. With more components more specific to IMU-based capture and wireless communications, this form factor could potentially be reduced such that worn sensors can be hidden under clothing.

For communications and reliability, our motion suit has demonstrated reliable accuracy for Robonaut teleoperation over 916MHz radio commuications. Informally, our suit has operated succesfully for periods up to 1 hour at distances up to 100 feet. Our choice of IMUs and radios was driven more by cost than accuracy. As IMU technology improves, so will the reliable performace of inertial-based motion capture. Additionally, the need for off-board communications can be removed by processing or collecting sensor data on the subject interface using increasingly smaller on-board computing devices.

Scalability, position estimation, and global human orientation. A benefit of our motion capture design is the ability for each sensor to estimate orientation independently. This feature provides an inherent scalability in that the sensors are continually publishing orientation information regardless of changes to the number of sensors or their configuration. The scalability issue for our system arises for corresponding sensor information to a kinematic topology. Our system currently relies upon associations between sensor identifiers and kinematic joints. Such correspondences could be done manually or through establishing wired communication between sensor pairs with a parent-child relationship, as in our current system.

A truly scalable system, however, would estimate sensor-kinematic correspondences by estimating kinematic topology. One approach to estimating kinematic topology could use a variant of work by O'Brien et al.[39] that partitions sensors into kinematic subgroups based on rotation about an individual joint. Another approach could treat the problem as a sensor network. This approach would rely on another sensing modality, such as sonar or wireless signal strength, to localize the sensors for position.

One limitation of our current system is that three sensors are required for completely untethered operation. More specifically, one additional sensor is required to account for the global rotation of the human with respect to their initial orientation. Without this additional sensor, the user is restricted to forward/backward global movement. Global turning or twisting results in a skewing of the capture data. We have tested unrestricted motion capture with three sensors; however, only two of these sensors were working during our tests with Robonaut.

Sensor initialization. Our application requires the sensors to be placed in a particular orientation, with connectors pointed up, in order to be initialized prop-

erly. This requirement is a side effect of the limited 32k flash programming space in the Mega32 microcontroller. Each sensor attempts to resolve its initial orientation upon power-up using trigonometric functions based on g_{sensed} and m_{sensed} to calculate initial yaw, pitch, and roll angles to form q_{ref} at $t = 0$. This process is similar to the error angle calculation, which has three cases, but due to memory limitations only one of these cases could be implemented. The chosen orientation is natural because the sensors are oriented with connectors pointed upwards when the body is in a neutral standing pose. Future improvement would include a memory expansion by at least 32k, which would allow this initialization to occur in any orientation. The ability to determine this initial orientation on-the-fly by sending a software reset rather than cycling the system power is currently being explored.

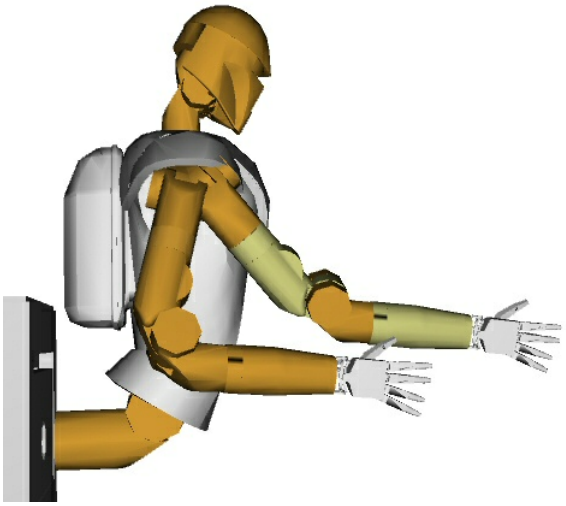

Fig. 10. An illustration of reindexing motion suit data for relative robot movement. This illustration shows two models of Robonaut, one indicative of values from the motion suit (the darker contracted arm) and one indicative of Robonaut's actual configuration (the lighter extended arm).

Human-robot factors and synchronization. During our teleoperation trials, it became apparent that human factors issues were more significant than we anticipated. These issues include initial human-robot synchronization, reindexing, and actuation delay. Due to safety concerns, Robonaut will not accept control commands unless its actual arm endeffector position is matched, within some close threshold, by the desired endeffector position sent by the motion suit. This initial synchronization had to be executed by the teleoperator through blind search (i.e., without feedback). Additionally, situations may arise during teleoperation where it is advantageous for the user's and the robot's configuration to be significantly different. These situations could be due to differences in kinematics or performing

relative movement in more less physically challenging coordinates.

As suggested by the Robonaut team, this problem was fixed through reindexing (Figure 10). Reindexing saves an offset transformation between the teleoperator and the robot. This allows the user's motion to be relative to the initial Robonaut position rather than control in absolute coordinates. Furthermore, relative control allows the teleoperator to chose a confortable working position independent of Robonaut's posture. Relative control can be reindexed at anytime during operation, allowing to address a larger reachable space.

Reindexing, however, does not account for the latency between teleoperator motion and robot actuation, result are compensation errors by the teleoperator. These errors occur when the human moves too fast for the robot to actuate and then attempts to compenstate for the robot's latency. The resulting motion is not necessarily faithful to either the human's motion or the robot's capabilities. Incorporation of feedback, through haptics or another sensation, to indicate deviation between the human and robot configurations could aleviate this issue.

7. Conclusion

In this paper, we have presented the design for an untethered motion capture system based on inertial sensing. The functionality of our system was demonstrated through experiments in teleoperating the NASA Robonaut. Our motion capture system has been presented as a cost-effective and unencumbering means for extending motion collection beyond structured environments. In the future, similar systems will play an important role in research areas such as human-robot interaction and learning from demonstration.

References

1. R. O. Ambrose, H. Aldridge, R. S. Askew, R. R. Burridge, W. Bluethmann, M. Diftler, C. Lovchik, D. Magruder, and F. Rehnmark. Robonaut: Nasa's space humanoid. *IEEE Intelligent Systems*, 15(4):57–63, Jul-Aug. 2000.
2. N. Pollard, J. K. Hodgins, M. Riley, and C. G. Atkeson. Adapting human motion for the control of a humanoid robot. In *IEEE International Conference on Robotics and Automation (ICRA 2002)*, pages 1390–1397, May 2002.
3. S. Jacobsen, M. Olivier, F. Smith, D. Knutti, R. Johnson, G. Colvin, and W. Scroggin. Research robots for applications in artificial intelligence, teleoperation and entertainment. *The International Journal of Robotics Research*, 23(4):319–330, Apr 2004.
4. G. Martinez, I. A. Kakadiaris, and D. Magruder. Teleoperating robonaut: A case study. In *The British Machine Vision Conference (BVMC 2002)*, pages 757–766. British Machine Vision Association, 2002.
5. J. Amat, M. Frigola, and A. Casals. Virtual exoskeleton for telemanipulation. In *International Symposium on Experimental Robotics (ISER 2000)*, pages 21–30, Honolulu, HI, USA, 2000.
6. A. Dasgupta and Y. Nakamura. Making feasible walking motion of humanoid robots from human motion capture data. In *IEEE International Conference on Robotics and Automation*, pages 1044–1049, 1999.

7. J. Kuffner. Goal-directed navigation for animated characters using real-time path planning and control. In *Workshop on Modelling and Motion Capture Techniques for Virtual Environments (CAPTECH 1998)*, pages 171–186, Geneva, Switzerland, 1998. Springer-Verlag.
8. M. J. Matarić, V. B. Zordan, and M. Williamson. Making complex articulated agents dance: An analysis of control methods drawn from robotics, animation, and biology. *Autonomous Agents and Multi-Agent Systems*, 2(1):23–44, Mar 1999.
9. O. C. Jenkins and M. J. Matarić. Performance-derived behavior vocabularies: Data-driven acqusition of skills from motion. *To appear in the International Journal of Humanoid Robotics*, 2004.
10. A. J. Ijspeert, J. Nakanishi, and S. Schaal. Trajectory formation for imitation with nonlinear dynamical systems. In *IEEE Intelligent Robots and Systems (IROS 2001)*, pages 752–757, Maui, Hawaii, USA, 2001.
11. R. Peters, C. Campbell, W. Bluethmann, and E. Huber. Robonaut task learning through teleoperation. In *IEEE International Conference on Robotics and Automation (ICRA 2003)*, pages 2806–2811, 2003.
12. W. Ilg, G. H. Bakir, J. Mezger, and M. A. Giese. On the representation, learning and transfer of spatio-temporal movement characteristics. In *IEEE International Conference on Humanoid Robotics (Humanoids 2003)*, Karlsruhe, Germany, Oct 2003.
13. M. Okada and Y. Nakamura. Design of the continuous symbol space for the intelligent robots using the dynamics-based information processing. In *IEEE International Conference on Robotics and Automation (ICRA 2004)*, pages 3201–3206, 2004.
14. C. Breazeal, A. Brooks, J. Gray, G. Hoffman, C. Kidd, H. Lee, J. Lieberman, A. Lockerd, , and D. Mulanda. Humanoid robots as cooperative partners for people. *Submitted to the International Journal of Humanoid Robotics*, 2004.
15. M. N. Nicolescu and M. J. Matarić. Natural methods for robot task learning: Instructive demonstration, generalization and practice. In *Autonomous Agents and Multi-Agent Systems (AAMAS 2003)*, pages 241–248, Melbourne, AUSTRALIA, July 2003.
16. C. Rose, M. F. Cohen, and B. Bodenheimer. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics & Applications*, 18(5):32–40, Sep-Oct 1998. ISSN 0272-1716.
17. L. Kovar, M. Gleicher, and F. Pighin. Motion graphs. *ACM Transactions on Graphics*, 21(3):473–482, 2002.
18. J. Lee, J. Chai, P. S. A. Reitsma, J. K. Hodgins, and N. S. Pollard. Interactive control of avatars animated with human motion data. *ACM Transactions on Graphics*, 21(3):491–500, 2002.
19. Y. Li, T. Wang, and H.-Y. Shum. Motion texture: a two-level statistical model for character motion synthesis. *ACM Transactions on Graphics*, 21(3):465–472, 2002.
20. M. G. Choi, J. Lee, and S. Y. Shin. Planning biped locomotion using motion capture data and probabilistic roadmaps. *ACM Transactions on Graphics*, 22(2):182–203, 2003.
21. L. Goncalves, E. Di Bernardo, and P. Perona. Reach out and touch space (motion learning). In *International Conference on Face and Gesture Recognition*, pages 234–241, 1998.
22. H. Sidenbladh, M. J. Black, and L. Sigal. Implicit probabilistic models of human motion for synthesis and tracking. In *European Conference on Computer Vision*, volume 1, pages 784–800, Copenhagen, Denmark, 2002.
23. M. Motion. Gypsy 3.0. http://www.metamotion.com.
24. P. Inc. Fastrak. http://www.polhemus.com.
25. V. M. Systems. http://www.vicon.com.

26. N. D. Inc. Optotrak. http://www.ndigital.com.
27. C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.
28. D. Gavrila and L. Davis. 3d model-based tracking of humans in action: A multi-view approach. In *IEEE Conference on Computer Vision and Pattern Recoginition*, pages 73–80, San Francisco, CA, USA, 1996.
29. J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 126–133, Hilton Head, SC, USA, 2000.
30. C.-W. Chu, O. C. Jenkins, and M. J. Matarić. Markerless kinematic model and motion capture from volume sequences. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR 2003)*, pages II: 475–482, Madison, Wisconsin, USA, June 2003.
31. M. Inc. Shapetape. http://www.measurand.com.
32. A. Mazzoldi, D. D. Rossi, F. Lorussi, E. P. Scilingo, and R. Paradiso. Smart textiles for wearable motion capture systems. *Autex Research Journal*, 2(4):199–203, 2002.
33. V. Group. Alert. http://www.verhaert.com.
34. E. R. Bachmann, R. B. McGhee, X. Yun, and M. J. Zyda. Inertial and magnetic posture tracking for inserting humans into networked virtual environments. In *ACM Symposium on Virtual Reality Software and Technology (VRST)*, pages 9–16, Banff, Canada, 2001.
35. D. Fontaine, D. David, and Y. Caritu. Sourceless human body motion capture. In *Smart Objects Conference (SOC 2003)*, Grenoble, France, 2003.
36. E. R. Bachmann, I. Duman, U. Usta, R. B. McGhee, X. Yun, and M. J. Zyda. Orientation tracking for humans and robots using inertial sensors. In *International Symposium on Computational Intelligence in Robotics and Automation (CIRA 1999)*, pages 187–194, Monterey, CA, Nov 1999.
37. K. Shoemake. Animating rotation with quaternion curves. In *Computer Graphics*, volume 19, pages 245–254, 1985.
38. J. J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley, Reading, MA, USA, 1989.
39. J. O'Brien, R. Bodenheimer, G. Brostow, and J. Hodgins. Automatic joint parameter estimation from magnetic motion capture data. In *Graphics Interface*, pages 53–60, May 2000.