IMUSim: A Simulation Environment for Inertial Sensing Algorithm Design and Evaluation

A. D. Young ayoung9@inf.ed.ac.uk M. J. Ling m.j.ling@ed.ac.uk D. K. Arvind dka@inf.ed.ac.uk

Centre for Speckled Computing, School of Informatics, University of Edinburgh 10 Crichton Street, Edinburgh EH8 9AB, United Kingdom

ABSTRACT

The use of wireless devices with accelerometers and gyroscopes to measure the movements of humans and objects is a growing area of interest. Applications range from simple activity detection to detailed full-body motion capture using networks of sensors worn on the body. A variety of algorithms have been proposed for these applications, but opportunities for accurate evaluation and comparison have been limited due to the many difficulties with performing rigorous experiments. We present a simulation environment, specifically for inertial sensing applications, designed to tackle this problem. We simulate sensor readings based on continuous trajectory models, and show how suitable models can be generated from existing motion capture or other sampled data. We show a good match between our simulated data and real sensor data for human movements. We also model a wide range of real-world issues such as non-ideal sensors, magnetic field distortions, timing factors and radio packet losses. To demonstrate the capabilities of our simulator, we present new results comparing four existing orientation estimation algorithms for human motion capture.

Categories and Subject Descriptors

I.6.4 [Simulation and Modelling]: Model Validation and Analysis

General Terms

Theory, Algorithms, Experimentation

1. INTRODUCTION

Accelerometers and gyroscopes-collectively known as inertial sensors-measure linear accelerations, the effects of gravity, and angular velocities. These parameters can be used in various ways to monitor many aspects of movement. In particular, they can be used to estimate and track orientation. Devices which combine multi-axis inertial sensors to do this are known as Inertial Measurement Units (IMUs), and often

Copyright 2011 ACM 978-1-4503-0512-9/11/04 ...\$10.00.

also include magnetometers to provide a heading reference from the Earth's magnetic field. With 3-axis measurements from all three types of sensor, the complete 3D spatial rotation of an IMU relative to the Earth can be estimated, albeit subject to a number of ambiguities dependent on the environment, sensors and nature of the motion.

Historically these sensors were primarily used for aircraft and missile guidance, but reductions in size, cost and power have made them usable in other applications. Many consumer devices now include accelerometers for simple motion and orientation detection, and some recent smartphones include a full IMU sensor suite. Inertial sensors have also been integrated into wireless sensor devices, for various purposes but particularly for detecting, classifying and tracking human movements with devices worn on the body, in fields including healthcare, sports and human-computer interaction. The sensing and analysis techniques used range from simple activity detection, through movement classification, to detailed full-body motion capture using networks of wireless IMUs worn on different parts of the body, driving a model of the subject's body structure. The examples given in this paper relate to the latter application, but the simulator presented can be used to simulate IMUs for any purpose.

Full-body motion capture using IMUs has many potential advantages over other capture technologies, but many realworld issues make system design difficult, and ambiguities in the sensed data present a challenge to algorithm design. Numerous algorithms have been proposed [1–8], yet evaluation of the accuracy of these has been quite limited, due to the cost and difficulty of full system development, and the many problems with performing rigorous experiments on realistic motions.

We believe that although real-world validation will always ultimately be essential, in this particular field there is a great deal still to be gained through better simulations at the design stage. Comparisons between algorithms and other aspects of system design have been hindered to date by the lack of any common framework within which they can be meaningfully tested. Existing wireless sensor network (WSN) simulators do not address the specific needs of IMU algorithm design, and no previous work has addressed these needs. We present a simulation environment targeted specifically at IMUs. Our contributions in this paper are:

- 1. Description of the IMU simulation environment, including: IMU trajectory requirements and generation, magnetic field modelling, and sensor models.
- 2. Validation of the simulation against real IMU measurements for human motion.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IPSN'11, April 12-14, 2011, Chicago, Illinois.

- 3. Empirical modelling of a real world motion capture environment, and validation of simulated magnetic field model.
- 4. New results comparing four orientation estimation algorithms using the simulation environment.

2. RELATED WORKS

2.1 WSN Simulators

There have been a large number of WSN simulators [9] developed, however none addresses the task of accurately simulating the sensed data of an IMU undergoing complex movements such as those of the human body.

WSN simulators have generally concentrated on the behaviour of large scale networks, and the communications between nodes. Application level simulation of WSNs is provided by simulators such as TOSSIM [10], and Castalia [11], while COOJA [12] and Avrora [13] provide more detailed simulation of node hardware. While these simulators support node mobility and environment models, the provided implementations are generally simplistic since detailed models are not required. In contrast, as will be shown in this paper, the motion and environment models are crucial in IMU simulation, and the requirements for physical accuracy and consistency are stringent.

As our simulation framework concentrates mainly on the accurate physical simulation of IMU data, it should be seen as complementary, rather than competitive, to the existing simulators. Indeed, nothing prevents the models proposed in this paper from being added to existing simulators should this be desired, or existing models of e.g. node hardware or radio channels being used in our simulator. Our code is specifically designed to facilitate this.

2.2 IMU Accuracy Testing

The use of multiple IMU systems for motion capture of human subjects has often been demonstrated qualitatively by presenting images of subject posture and reconstructed posture [2,4,5,8,14]. These qualitative demonstrations provide no scientific basis for algorithm comparison.

In order to assess the accuracy of IMU algorithms quantitatively it is necessary to have access to a true reference of the device orientation and position. It is possible to perform experiments with both IMUs and optical motion capture [3, 6, 15], however given the costs of the respective systems, and the expertise required to operate them, this is not always a viable option. Furthermore, accounting for the sources of error between the two systems can be challenging, requiring careful data pre-processing, and giving potentially ambiguous results.

As combined optical/inertial capture is so challenging, studies often use computer controlled systems, such as pan and tilt units, to generate device motion [4, 5, 16, 17]. This approach allows the static and dynamic accuracy of IMUs to be assessed over long periods, but it is difficult to generate realistic motions such as might be experienced during human motion capture. The movements of mechanical systems are subject to many more constraints than those of a human body, with fewer degrees of freedom and limits on speeds, accelerations and variation of motions. Building a mechanical system able to accurately reproduce human movement would be prohibitively expensive, as can be seen from the cost of present humanoid robots which still fall far short of human freedom of movement.

For these reasons, the authors began work on generating simulated IMU data based on optical capture data of human movements. Some evaluations of IMU algorithm performance based on early simulation work were published by Young [7, 18]. These papers did not give full details of the simulation methods, which have since been revised, and the simulated data was not validated against real measurements.

3. IMU MOTION CAPTURE THEORY

As the examples in this paper focus on IMU-based human motion capture, we provide here an introduction to this area.

An IMU-based motion capture system requires the subject to wear one IMU on each segment of the body [2]. A segment in this sense is any part of the body which surrounds a rigid section of bone structure and must therefore move predominantly as a unit, e.g. the forearm, upper leg, or head. The orientation of each IMU, and hence the segment to which it is attached, is estimated from its sensor data. When the orientation of all segments of the body is known at a given moment, the complete posture can be reconstructed. This whole process is repeated at high frequency to achieve motion capture.

On its own, this method only estimates relative positions between parts of the body. If the absolute movements of the subject are required, these can be estimated by deadreckoning approaches, or measured separately with the aid of external infrastructure such as GPS, computer vision systems or ultra-wideband (UWB) radio range finding.

There are therefore several stages to the processing required: orientation estimation of each IMU, which is in turn often divided into vector observation, gyroscope integration and data fusion steps; reconstruction of the posture; and, optionally, translation estimation. Some of these stages may be combined in some methods, and some may be distributed in various ways into a wireless network of IMUs.

3.1 Vector Observation

The orientation of an IMU equipped with 3-axis accelerometers and magnetometers can be estimated from the locally measured directions of two global vectors: magnetic north, and acceleration due to gravity. The measurement of both is subject to various sources of noise and interference. The task of estimating orientation from imperfect measurements of reference vectors is called vector observation, and requires two or more non-zero, non-collinear, vectors in the rotated co-ordinate frame that have known directions in a reference co-ordinate frame. Vector observation algorithms attempt to find a rotation matrix \boldsymbol{R} that minimises Wahba's loss function [19]:

$$L(\boldsymbol{R}) = \frac{1}{2} \sum_{i=1}^{n} a_i \|\boldsymbol{b}_i - \boldsymbol{R}\boldsymbol{w}_i\|^2, \qquad (1)$$

where b_i are the observed vectors in the rotated co-ordinate frame (acceleration and magnetic field vectors), w_i are the reference vectors in the reference co-ordinate frame (gravity and magnetic north), and a_i are a set of non-negative weights applied to each vector to account for individual accuracy. Numerous solutions to Wahba's loss function have been proposed [20]. Some are shown to find the optimal \mathbf{R} , others sacrifice optimality for speed. Vector observation allows for absolute estimation of IMU orientation but is sensitive to noise in the vector observations, particularly the non-gravitational linear accelerations of the IMU which can be thought of as noise superimposed on the observation of gravity.

3.2 Gyroscope Integration

Using a 3-axis gyroscope the complete angular rate vector of an IMU can be measured. Since angular rate is the derivative of orientation it can be integrated to find an IMU's current orientation, working from a known initial orientation at a previous time. To do this accurately requires a sampling rate at least in the region of 100-200Hz. In practice, this integrated estimate will still drift from the truth with each step due to noise, bias, and numerical integration errors.

3.3 Data Fusion

By combining gyroscope integration and vector observation, a more accurate orientation estimate can be obtained which exploits both the angular rate accuracy obtainable from gyroscope data and the absolute, but noisy, orientation estimates provided by vector observation. This requires a method to fuse the two sources into a single state estimate.

A common tool for this form of data fusion task is the Kalman filter model, which provides the optimal recursive estimator for linear systems with well specified state dynamics. Extensions to the standard Kalman filter, such as the Extended and Unscented Kalman filters, allow estimation of non-linear systems. A number of Kalman filters have been proposed specifically for tracking IMU orientation [1,3,4,21]. Due to the matrix operations required to propagate system state and estimated covariance, the complexity of these filters is $O(n^3)$ where n is the size of the state vector.

Complementary filters provide an alternative and simpler method, which has the advantage of being more practical to implement *in situ* on a low-power wireless IMU [15]. Complementary filtering exploits the complementary properties of multiple observations of a signal corrupted by noise with different frequency properties. Gyroscope integration suffers from drift, i.e. low frequency noise, whilst vector observation suffers from acceleration noise at higher frequencies. By selecting suitable frequency cut-offs a composite filter can be constructed that passes the signal of interest with unity gain and zero latency while blocking the majority of the noise. The major disadvantage of complementary filters is that they do not provide any estimate of uncertainty in their results.

3.4 Posture Reconstruction

Reconstruction of subject motion from IMU data is generally performed by combining estimated IMU orientation with a forward kinematic rigid body model of the subject [2]. Estimated orientations of IMUs attached to the subject limb segments can directly drive the body model, or a further level of data fusion can be applied to model subject dynamics and bio-mechanical kinematic constraints [14, 22].

To obtain the orientation of a body segment from that of an IMU attached to it, it is necessary to first know the orientation of the IMU relative to that segment. This can be achieved by deliberate alignment of the IMUs as they are attached, or by observing the orientations of all IMUs while the subject is in a known calibration posture such as a T-stance.

3.5 Translation Estimation

Estimation of IMU translation through integration of estimated IMU linear acceleration is prone to extreme drift with time. Double integration of acceleration is required, leading to quadratic growth in bias errors. As gravitational acceleration must be removed from accelerometer data prior to integration the orientation estimation accuracy is of paramount importance. An error as low as 1 mrad, significantly less than typical static accuracy measurements of existing IMUs, results in an acceleration of approximately 0.01 m/s^2 which, left uncorrected, would result in an error of 4.5 m after 30 s [23].

As direct integration of IMU acceleration is so prone to error it is common to apply additional processing, such as using Zero Velocity Update, to reset velocity estimates to zero when an IMU is expected to be stationary, for example during the stance phase of a walking gait [24, 25]. Alternatively, the subject posture can be used to estimate position based on tracking expected ground contact [7].

3.6 Summary of Problem

The full reconstruction of subject motion from IMU data is a challenging problem, requiring fusion of data from many different sources and at different levels of abstraction.

A range of methods have been proposed for each aspect of the task, however it is hard to meaningfully compare their performance. Results can be highly dependent on sensor hardware and on the nature of individual motions. Realworld experiments are difficult and time-consuming to conduct rigorously for realistic motions, and results can be hard to analyse meaningfully.

For these reasons, we believe a good common simulation environment is the key to improving work in this field. To be relevant however, a simulation must be capable of generating truly representative sensor data for realistic human movements and environments, modelling IMU-specific problems such as sensor noise, bias, misalignment, cross-axis effects and field distortions as well as supporting models of more general sensor network issues such as time synchronisation and packet losses.

4. SIMULATION ENVIRONMENT

Our simulation environment is designed to support modelling all aspects of IMU operation in an extensible fashion. The simulator is implemented in the Python scripting language, making use of the strong set of existing libraries including NumPy, SciPy, SimPy, Cython and matplotlib. These tools provide an environment that will also be familiar to users of Matlab. The decision to implement in Python was driven by the strong high level support for complex mathematical systems that these libraries provide, and the ease of interfacing with other code in future.

An overview of the simulation environment is shown in Figure 1. The key functionality of the simulator is the generation of realistic inertial sensor data from models of the environment, sensor hardware, and the position and rotation trajectories that simulated IMUs take through the course of the simulation. IMUs can have independent clocks, allowing for the simulation of timing skew between samples.

The gravitational and magnetic fields in the environment are modelled as continuous vector fields. Ideal simulated sensor values at a given instant are obtained by evaluating



Figure 1: Block diagram of simulation framework showing main components involved in generating simulated IMU data from existing motion capture data.

trajectory derivatives at that moment along with the environmental vector fields at the sensors's position, rotated into its local co-ordinate frame. These ideal values are processed by a sensor subsystem which models the imperfections of the sensor and analogue-to-digital convertor (ADC).

Algorithms to run on the IMUs are written out directly in normal Python code, with access through a platform object to the simulated IMU hardware, including the sensor subsystems and a radio model to communicate with other IMUs or a simulated host system. The wireless channel model used for the radio can also make use of the position and orientation information from the IMU trajectories at the time a transmission attempt is made.

The simulator is primarily designed for offline simulation, with correct behaviour and clarity of code overriding concerns of performance. However, the performance is sufficient to simulate a single IMU in real time on a typical 2GHz PC.

4.1 IMU Trajectory Functions

In order for the simulator to generate the sensor measurements of an IMU, its trajectory through space must be known. Two functions must be defined for each IMU giving its position:

$$\boldsymbol{p}: T \to \mathbb{R}^3, \ T \in [0, t_{max}], \ \boldsymbol{p} \in C^2,$$
 (2)

and its rotation, represented as a quaternion:

$$\theta: T \to SO(3), \ T \in [0, t_{max}], \ \theta \in C^2.$$
 (3)

The velocity, acceleration, rotational rate and rotational acceleration vectors in the IMU local frame may be evaluated as:

$$\boldsymbol{v}(t) = \theta^*(t) \otimes \boldsymbol{p}'(t) \otimes \theta(t), \qquad (4)$$

$$\boldsymbol{a}(t) = \boldsymbol{\theta}^*(t) \otimes \boldsymbol{p}^{\prime\prime}(t) \otimes \boldsymbol{\theta}(t), \tag{5}$$

$$\boldsymbol{\omega}(t) = 2\theta^*(t) \otimes \theta'(t), \tag{6}$$

$$\boldsymbol{\alpha}(t) = 2\theta^*(t) \otimes \theta^{\prime\prime}(t), \tag{7}$$

respectively, where \otimes is the quaternion multiplication operator, and * indicates the quaternion conjugate.

The position and rotation functions must be continuous, so that readings can be taken at any point in time, and twice differentiable, so that linear and angular accelerations can be obtained. Although any suitable functions can be used, the simulator includes tools to generate interpolating trajectory functions from discrete time trajectory data.

For sampled position data, p is defined by fitting three independent spline functions to the x, y, z components of the data. Cubic splines are used to give the required C^2 continuity. Defining θ from sampled rotations is more complicated as the most common forms of quaternion interpolation, the SLERP [26] and SQUAD [27] algorithms, are only C^0 and C^1 continuous respectively. The quaternion B-spline algorithm of Kim et al. [28] provides the necessary continuity.

In order to perform realistic experiments, IMU simulations should be based on motions representative of the system's intended usage. For the most accurate simulations possible for a given application, these motions should be obtained through capture of target motions using other methods such as optical capture. While potentially costly and time consuming, the benefits of an accurate simulation in reducing the risk of IMU system design may make this effort worthwhile. Alternatively, for simple or well-modelled motions the trajectory functions could be defined directly.

Generation of IMU data from pre-existing motion data offers a useful compromise solution. Use of existing data ensures that test motions are realistic without incurring the costs of performing dedicated experiments, and the use of freely available data allows for simplified comparisons between competing algorithms. For human motion the Carnegie Mellon University motion capture corpus¹ is a suitable and valuable resource.

4.1.1 Trajectories from rigid body models

In human motion the trajectories of individual points on the body are not independent, since they are constrained by the structure of the skeleton. For this reason human motion is normally modelled as movements of a jointed rigid body model. An example of a rigid body model is shown in Figure 2. Most human motion capture data is processed to fit this type of model, and interchange formats such as BioVision Hierarchy (BVH) files include the model structure along with sampled positions of the root joint and rotations of all joints. To simulate an IMU attached at some offset relative to a segment in this model, we require trajectory functions which maintain the physical constraints of the skeleton. The functions must take into account the kinematic chain of joints connecting the IMU position to the root of the model.

Given a body model, its root position and joint rotations as C^2 continuous functions of time, and Equations 2–7, then the positions, linear velocities, and linear accelerations of any point attached to the model can be calculated in a recursive manner. The position of a point, p, offset from a joint, J_k , by an offset vector in joint local co-ordinates, o,

¹Data available from http://mocap.cs.cmu.edu. The database was created with funding from NSF EIA-0196217.



Figure 2: Rigid body model example, showing joint tree structure and sensors attached at offsets.

at time, t, can be calculated as:

$$\boldsymbol{p}(\boldsymbol{o}, J_k, t) = \boldsymbol{p}(\boldsymbol{0}, J_k, t) + \boldsymbol{R}_{J_k}(t) \cdot \boldsymbol{o}, \qquad (8)$$

where $\mathbf{R}_{J_k}(t)$ is the rotation of joint J_k at time t.

The positions of the joints, $p(0, J_k, t)$ can in turn be calculated by applying Equation 8 recursively:

$$\boldsymbol{p}(\boldsymbol{0}, J_k, t) = \begin{cases} \boldsymbol{p}_{J_0}(t), \ k = 0\\ \boldsymbol{p}(\boldsymbol{o}_{J_k}, J_{k-1}, t), \text{ otherwise,} \end{cases}$$
(9)

where: o_{J_k} is the offset of the of the joint J_k , in its parent's local co-ordinate frame; joint J_{k-1} is the parent of joint J_k in the body model tree; and joint J_0 is the root.

The velocity of a point in the model can be calculated as:

$$\boldsymbol{v}(\boldsymbol{o}, J_k, t) = \boldsymbol{v}(\boldsymbol{0}, J_k, t) + \boldsymbol{R}_{J_k}(t) \cdot (\boldsymbol{\omega} \times \boldsymbol{o}), \qquad (10a)$$

$$\boldsymbol{v}(\mathbf{0}, J_k, t) = \begin{cases} \boldsymbol{p}'_{J_0}(t), \ k = 0\\ \boldsymbol{v}(\boldsymbol{o}_{J_k}, J_{k-1}, t), \text{ otherwise,} \end{cases}$$
(10b)

the acceleration of a joint as:

$$\boldsymbol{a}(\mathbf{0}, J_k, t) = \begin{cases} \boldsymbol{p}_{J_0}'(t), \ k = 0\\ \boldsymbol{a}(\boldsymbol{o}_{J_k}, J_{k-1}, t), \text{ otherwise.} \end{cases}$$
(11)

and the acceleration of an offset point as [6]:

$$a(o, J_k, t) = a(0, J_k, t) + R_{J_k}(t) \cdot (\alpha \times o + (o \cdot \omega) \omega - o ||\omega||^2)$$
(12)

where $\boldsymbol{\omega}$ and $\boldsymbol{\alpha}$ are the rotational rate and acceleration of joint J_k at time t.

Figure 3 shows the visualisation of an example motion taken from the CMU motion capture corpus. The trajectory generation functions allow the source motion data to be arbitrarily resampled, allowing for the effects of timing factors such as IMU synchronisation error and delay between ADC samples to be accurately modelled. This would not be possible with a discrete time approach.

4.1.2 Trajectory Filtering

Source motion data captured by optical motion capture systems, while highly accurate, can often suffer from high frequency noise. This noise, if left unfiltered, results in substantial noise in derivative values that completely masks the underlying form of the functions. In order to perform useful simulations it is necessary to filter the raw source data to remove this noise.



Figure 3: Visualisation of motion trajectories for a walking subject. Source data at 120 Hz converted to continuous time, model posture displayed at 5 Hz and the velocity vector of the right foot displayed at 50 Hz.

For position data the simulation environment makes use of smoothing splines rather than pure interpolating splines to mitigate the effects of random residual errors in the source motion capture. The smoothing spline has the property that it minimises the second derivative, the acceleration, while still maintaing the overall form of the observations. The default standard deviation is 1 mm, based on the typical residual errors reported for the popular Vicon and Qualisys optical capture systems.

Rotational data is more complex to process as the current quaternion B-spline implementation does not support smoothing directly. We therefore acheive smoothing by applying a smoothing spline to each quaternion component individually before the quaternion spline is generated. As quaternions capture three degrees of freedom in four component values, simple filtering of individual components introduces distortions when sampled at low frequencies. However, with source motion capture gathered at high sample rates, this effect is minimal.

4.2 Vector Field Modelling

The gravitational and magnetic fields of the Earth form vector fields and can therefore be modelled respectively by the functions:

$$g: \mathbb{R}^3 \to \mathbb{R}^3, \tag{13}$$

$$b: \mathbb{R}^3 \to \mathbb{R}^3. \tag{14}$$

For a simulation confined to an area which is miniscule compared to the curvature of the Earth's surface, the gravitational and magnetic fields can be considered to be effectively parallel vector fields, and thus:

$$g(x, y, z) = (g_x, g_y, g_z)^T = (0, 0, g_0)^T,$$
(15)

$$b(x, y, z) = (b_x, b_y, b_z)^T = b_0 \begin{pmatrix} \cos(\phi) \cos(\theta) \\ \cos(\phi) \sin(\theta) \\ \sin(\phi) \end{pmatrix}, \quad (16)$$

where $g_0 \approx 9.81 \text{m/s}^2$ is the local gravitational acceleration; and $b_0 \approx 50 \mu \text{T}, \theta, \phi$, the local magnetic field strength, declination and inclination respectively.

While ideally the required vector fields could be modelled by Equations 15 and 16, both fields can be distorted by nearby objects. Therefore we consider the causes and effects of distortions to each of the modelled fields.

4.2.1 Gravitational Field Distortions

The Earth's gravitational field varies with latitude, height, the tidal effect of the moon and the sun, and variations in the local distribution of mass [29]. The effect of variation with latitude can be estimated by the International Gravity Formula while the effects of other variations on experienced acceleration can be estimated by applying Newton's laws of gravitation and motion:

$$\boldsymbol{F} = \frac{Gm_1m_2}{\|\boldsymbol{r}\|^2} \frac{\boldsymbol{r}}{\|\boldsymbol{r}\|},\tag{17}$$

$$\boldsymbol{F} = m\boldsymbol{a}.\tag{18}$$

The global variation experienced due to latitude and height, about 0.5% and 0.3% of the nominal field strength respectively, can be ignored for simulations within a small area. Furthermore, these parameters affect only the magnitude of the experienced acceleration and not its direction. The tidal pull of the moon and sun results in a acceleration of up to approximately 0.003% of the Earth field. This amount of acceleration results in a maximum possible error of less than 0.02° [17].

The effect of variation in distribution of mass in the area around an IMU is more complex to analyse. The gravitational acceleration measured is determined by the vector sum of all forces generated by surrounding masses:

$$a = \sum_{i=0}^{N} \frac{Gm_i}{\|\boldsymbol{r}_i\|^2} \frac{\boldsymbol{r}_i}{\|\boldsymbol{r}_i\|}.$$
 (19)

While this could be evaluated for each point in an IMU trajectory, it would present a significant computational overhead, dependent on the granularity of the environment model. As the mass of surrounding objects is so small relative to the mass of the Earth, these effects can generally be ignored and the simple model of Equation 15 used. As illustration, the point mass required to introduce a significant error, on the order of 0.1° at a distance of one metre, is equivalent to a sphere of lead thirty-five meters in diameter [17].

4.2.2 Magnetic Field Distortions

The Earth's magnetic field is commonly approximated as a magnetic dipole with a magnetic south pole near the Earth's north pole, however this model is not accurate on a local level. Variations in the surrounding materials, such as iron rich mineral deposits in natural settings, or steel used in building construction, cause distortions in the observed field. These distortions can be easily observed using a normal magnetic compass and can result in heading errors of up to 180° .

Accurate modelling of magnetic fields using Maxwell's equations is extremely complicated, relying on exact knowledge of geometry and electro-magnetic properties. Therefore, for the purposes of simulation magnetic field modelling is approximated in two ways: firstly, through simplified models of ideal solenoids; and secondly, through empirical measurements of real world environments.

Modelling of ideal solenoids is possible using the Biot-Savart law and the theory of superposition. The magnetic field of each solenoid is estimated using the model proposed by Derby and Olbert [30]. This model allows for the efficient calculation of the magnetic field at a point due to a single solenoid. By simulating multiple parameterised solenoids, with various translations and rotations, and combining their fields through superposition complex fields can be created.

While simulation using ideal solenoid models allows for generation of physically accurate fields, these are not necessarily realistic, as they exclude many complex interactions with surrounding materials. Use of empirically measured fields provides greater realism, at the expense of an initial investment in time.

Generating a vector field model from observations of the field is a complex process. The accuracy of the field model depends on the resolution of the observations, with more observations leading to increased fidelity. Taking many field observations in a structured manner, for example a regular three dimensional grid, is a laborious process. Taking a large number of unstructured readings, using an optical motion capture system for position and orientation of a threedimensional magnetometer, is a far faster proposition. In either case, once empirical data has been gathered, an interpolation algorithm must be used to generate the complete field model. Field interpolation is performed using the natural neighbour interpolation algorithm by Hemsley [31].

4.3 Sensor Subsystem Modelling

Modelling of IMU sensor hardware is important as the physical properties that the IMU senses must pass through various transformations before being rendered as digital values within a micro-processor. The current framework models three main properties of the IMU hardware: the sensor transfer function, relating physical properties to voltages; injection of sensor noise; and ADC quantisation.

Transfer functions for individual sensors are based on manufacturer information provided in device datasheets. These transfer functions are generally specified as linear functions from physical quantities to voltages. Three-dimensional measurements are provided by sensor triads, either in monolithic integrated packages or composed of multiple devices. To model cross-axis sensitivity and non-orthogonal axis alignment a transform matrix is applied. Additional realism may be added through the use of temperature dependency and sensitivity to other physical parameters, for instance the effects of linear acceleration on rotational rate gyroscopes. Further empirical characterisation, such as nonlinearity measurements or time varying bias conditions, could be used to increase simulation accuracy for specific sensors.

As an example, Equation 20 shows the transfer function of the Analog Devices ADXRS300 rotational rate gyroscope, configured for a sensitivity of $1200^{\circ}/s$ [32]. The transfer function relates the rotational rate in radians per second, ω , to the analogue voltage at the ADC input, V_{gyro} , including the contributions of: linear acceleration sensitivity, A; offset voltage, O; rotational rate sensitivity, S; and nonorthogonality modelling matrix, T.

$$V_{\text{gyro}} = ST \left(\omega + Aa \right) + O$$
$$\approx \frac{1.25 \times 10^{-3} \pi}{180} T \left(\omega + \frac{0.2}{g_0} a \right) + 2.5 \qquad (20)$$

Practical sensors all suffer from some level of noise in output signals. As observed sensor noise, n, is caused by many independent sources it can generally be modelled as an additive zero mean white Gaussian process with standard deviation σ :

$$n = \mathcal{N}(0, \sigma). \tag{21}$$

This can be confirmed, and a suitable value of σ determined experimentally, by measurement of the sensor output while static. For example, Figure 4 shows a normal distribution quantile-quantile plot for the noise from a Freescale



Figure 4: Quantile-quantile plot of accelerometer noise (ADC points) for Freescale MMA7260Q at 1.5 g sensitivity.

MMA7260Q tri-axial accelerometer. The high r^2 correlation value indicates a good fit to the normal distribution. Autocorrelation or power spectral density plots can be used to confirm frequency properties.

Filtered sensor values, derived from IMU trajectories, are calculated as double precision floating point values to provide maximal consistency with analogue signals. To emulate the effects of analogue to digital conversion these values are quantised to discrete values. Voltages are converted to signed ADC values by the parametric ADC transfer function:

$$ADC(V, b, V_{\text{ref}}) = \begin{cases} -2^{b-1}, & V < -V_{\text{ref}}, \\ \left\lfloor \frac{V}{V_{\text{ref}}} 2^{b-1} + \frac{1}{2} \right\rfloor - 2^{b-1}, \\ 2^{b-1} - 1, & V \ge V_{\text{ref}}, \end{cases}$$
(22)

where, b is the number of bits in the ADC output and V_{ref} is the ADC reference voltage, generally taken to be $V_{cc}/2$.

As the sensor sub-system is modelled as a composite function of individual transfer functions, increased fidelity can be added by substituting empirical characterisations of components. This allows the simulator to be adapted in the future to support more complex models including details such as sensor non-linearities.

4.4 Wireless Communications Modelling

Because our simulations include the trajectories taken by each IMU, and in the case of human movements also the motion of the body, the opportunity exists to model in detail the effects on wireless communications between IMU devices. This could be of particular value in the design of worn or carried wireless devices, which experience time varying obstruction by the user's own body, leading to markedly different channel properties compared to traditional WSN applications [33].

At present however, we are not aware of channel models which could be easily plugged in to fulfil this function. Traditional radio channel models, such as the model used in the Castalia WSN simulator [11], model packet reception probability as a stochastic function of transmission distance and transmitter power. These models generally do not take into account the specifics of local environmental obstructions. More recent models have begun to consider the effects of environmental obstructions such as walls [34]. However, all of these models are based on static deployments of nodes within static environments. Development of body sensor network specific radio models, dealing explicitly with the effects of the human body on radio propagation [35, 36], are still in their infancy. Such models can be integrated easily into the simulator when ready.

For the results in Section 6, a simple stochastic packet reception model was used. A constant bit error rate probability, Pr(e), was assumed for all transmissions, and the presence of one or more bit errors results in packet loss. The probability of a packet of length l bits being lost, Pr(l) is therefore simply:

$$Pr(l) = 1 - (1 - Pr(e))^{l}.$$
(23)

This model has the advantage that, having only a single parameter, it is easy to tune for varying degrees of packet throughput.

5. VALIDATION OF SIMULATOR

In order to test the accuracy of our simulations, we have conducted some experiments to directly compare our simulated sensor values to those measured by real IMUs. To achieve this, we used an optical motion capture system to capture the movements of a subject who was also wearing wireless IMUs. In addition to the normal markers on the subject, the positions and rotations of the IMUs themselves were tracked using three markers attached to each IMU. The overall methodology was similar to that in [6], although unlike that experiment we did not adjust sensor calibration based on the optical capture.

We sampled the magnetic field in the capture area, using the magnetometer of an IMU swept around the capture volume whilst being tracked by the optical system. These measurements were later used to generate an interpolated field model of the capture area. A subset of the samples are illustrated in Figure 5a, for the region near the floor where there was strongest distortion from steel structures below. The median of the field vectors has been subtracted to illustrate the distortions alone.

We collected capture data from a subject with three IMUs on the left upper leg (femur), lower leg (tibia) and foot of the subject. The subject was first captured in a standing pose. The positions and rotations of the body segments at a single moment in this pose were used to obtain the constant offset parameters of the rigid body model described in Section 4.1.1. The subject was then captured performing various movements. For each movement, the positions of the root joint (hip) marker and the rotations of each segment were then used as source data for the position and rotation splines of the model. We then ran simulations of IMUs attached to this model. The constant position and rotation offsets of these simulated IMUs relative to their joints were taken again from the single standing reference frame; the IMU marker data in the movements themselves were not used to obtain our results.

Figure 6 displays complete sensor data comparisons for a test motion in which the subject swings their leg forward and back, thus generating accelerations along the chain of joints. In general there is very strong correlation between the simulated and real values, indicating correct behaviour of the trajectory generation and rigid body modelling. Similar results were obtained for other motions such as walking.

Note that no noise has been added to the simulated signals in this example; these are the idealised values to which



(a) Magnetic field distortion samples near the floor of the capture area. The volume shown is approx. $2 \times 2 \times 0.2$ m.



(b) Effect of using interpolated field. Faded lines from real data, dotted lines from simulation with constant field, solid lines from simulation with interpolated field.

Figure 5: Magnetic field distortions and their effects on IMU data.

random noise can be added according to the parameters of individual sensors. The generation of noise to given statistical parameters is a well understood task; it is the low frequency information from the trajectory model which is challenging to obtain and hence of interest here.

The importance of magnetic field modelling is illustrated in Figure 5b, showing real and simulated values, with and without distortion, for the magnetometer at the foot. As the foot passes close to the floor at the bottom of the swing there is a substantial change in the measured field vector. If the magnetic field is modelled as a constant, there is a large error between the simulated and real values. By using an interpolated model of the field, a much closer match is achieved.

6. DEMONSTRATION OF SIMULATION EN-VIRONMENT

In order to demonstrate the application of IMUSim to the development and testing of wireless inertial motion capture algorithms we have implemented four algorithms:

- **Complementary Filter (CF)** an implementation of the original complementary orientation filter by Bachmann [2], with the original Gauss-Newton iterative vector observation replaced by the FQA algorithm [37] as in later work.
- Gated Complementary Filter (GCF) a modification of Bachmann's original filter by Young [15], to use Gram-Schmidt ortho-normalisation for vector observation, and to reject clearly erroneous gravitational field measurements.
- **Extended Kalman Filter (EKF)** an implementation of an Extended Kalman filter design by Yun et al. [4], using FQA for vector observation.
- Linear Acceleration Estimation Filter (LAEF) an implementation of the distributed linear acceleration estimation filter proposed by Young et al. [6].

The source motion capture data for simulations was subject 16 trial 15 from the CMU motion capture corpus. No gravitational or magnetic field distortions were included in the simulation. All algorithms were simulated at a sampling rate of 100 Hz; realistic sensor models including bias, cross-axis sensitivity and noise; and a 12 bit ADC. For the Linear Acceleration algorithm a lossy wireless channel with a bit error rate probability of 10^{-3} was used for inter-IMU communication.

6.1 Effects of Linear Acceleration

The implemented algorithms all have a similar structure with rotational rate gyroscope data being fused with vector observations of the gravitational and magnetic field. For a moving subject the IMU accelerometers measure the vector sum of the gravitational field and the subject's linear acceleration. The effect of this is highly significant to the observations of the gravitational field as illustrated by Figure 9. For clarity we have selected a single IMU, attached to the top of the right tibia, for analysis, however, the same effects are seen for all IMUs in the model.

The large variation in observed gravitational vector measurements has a great effect on the accuracy of vector observation, and hence on filtered orientation accuracy, as illustrated in Figures 7 and 8.

In the CF and EKF implementations, whose designs assume that the gravitational acceleration vector is measured with Gaussian noise, the corruption of the vector observation leads to substantial error. The particularly poor performance of the EKF implementation is due to the poor tuning of the measurement noise covariance matrix. Yun et al. [4] estimate the covariance of the vector observation process from measurements of a static IMU. This leads to far greater confidence in the output of the vector observation than is reasonable for an IMU undergoing substantial linear accelerations, as in the case of a walking subject.

The GCF implementation, that detects linear acceleration based on the magnitude of the measured acceleration, has little opportunity to perform drift correction, and the vector observations that are made are frequently inaccurate,



Figure 6: Comparison of measured and simulated data for swinging leg motion. Solid lines in time series plots are simulated data, faded lines are real IMU measurements.



Figure 9: Comparison of gravitational acceleration (black line) and total acceleration (grey line) for a realistic IMU model attached to the top of the right tibia. The total acceleration shows much greater variation due to the effects of linear acceleration caused by subject motion.



Figure 7: Vector observation quaternion components (solid lines) and true orientations (dotted lines) for realistic IMU model attached to the top of the right tibia.

Figure 8: Estimated IMU orientation quaternion components (solid lines) and true orientations (dotted lines) for realistic IMU model attached to the top of the right tibia.

Table 1: Packet delivery rate and estimated orientation accuracy for the LAEF algorithm with varying BER.

	$BER = 10^{-3}$		$BER = 10^{-4}$	
Joint	DR(%)	$\mathrm{RMSE}(^{\circ})$	DR(%)	$\mathrm{RMSE}(^{\circ})$
root		4.1 ± 0.6		4.1 ± 0.6
rfemur	88.3 ± 1.4	3.4 ± 1.8	98.4 ± 0.5	3.5 ± 1.8
rtibia	78.1 ± 2.3	3.3 ± 1.7	97.1 ± 0.8	3.7 ± 2.1
rfoot	69.3 ± 2.6	3.6 ± 1.5	96.0 ± 1.1	4.4 ± 1.7
lfemur	87.7 ± 1.5	3.4 ± 2.0	98.1 ± 0.5	3.6 ± 2.1
ltibia	77.6 ± 2.0	3.4 ± 1.8	97.0 ± 0.6	3.7 ± 2.1
lfoot	68.7 ± 2.1	3.9 ± 1.7	95.8 ± 0.7	5.0 ± 2.1

leading to corruption of estimated state.

The LAEF implementation, that is specifically designed to estimate linear accelerations due to rigid body motion, shows much less variation in the accuracy of vector observation, directly leading to its markedly better performance.

6.2 Effect of Packet Loss on Linear Acceleration Estimation Filter

The CF, GCF, and EKF, algorithms operate solely on locally sensed data to estimate orientation, In contrast, the distributed LAEF algorithm requires estimated linear acceleration data to be transferred between IMUs. The original paper on the LAEF [6] acknowledges that packets will be lost, but does not present any results regarding what happens in this case.

To test the sensitivity of the LAEF algorithm to packet loss simulations were performed with Bit Error Rates (BER) of 10^{-3} and 10^{-4} . Fifty independent simulations were performed for each BER, the results of which are shown in Table 1.

Two points are evident from the simulation results: orientation accuracy is a function of kinematic chain length, and increased BER does not adversely affect algorithm accuracy.

Investigation of the simulation results yielded the likely cause for the increase of error with kinematic chain length. Due to the numerical differentiation required by the LAEF algorithm, to estimate angular acceleration from rate gyroscope data, minor numerical errors result even when using floating point numbers and perfect sensors. These minor errors are amplified as the length of the kinematic chain increases. When using realistic sensor models with noise, as in the presented results, the accuracy of linear acceleration estimation is substantially reduced compared to the behaviour with ideal sensor models.

Increased BER reduces the possibility to perform gyroscope drift correction. However, provided the time between drift corrections is sufficiently small, no large error can accumulate. With the simple BER channel model used in the current simulations packet loss is not temporally correlated so typically there are no long periods without reception. Further work is required to investigate the effects of temporal packet loss correlation on orientation accuracy.

7. CONCLUSIONS

A new simulation environment targeted specifically at inertial sensing applications has been developed, that can simulate IMU measurements from trajectory definitions for any application. It allows continuous trajectories to be synthesised from existing discrete time trajectory data or rigid body motion capture, reducing the barriers to realistic experimentation, and allowing comparisons to be performed using commonly available data sets.

The generated IMU sensor data has been validated against real IMU measurements for simulations based on captured human motions and an empirical environmental model. The use of the simulator in obtaining new results in orientation algorithm comparison has also been demonstrated. The simulations have revealed problems with proposed algorithms that were not recognised in the original publications due to the lack of realistic test scenarios.

The simulator is available at http://www.imusim.org/ under an open source license. It is designed to be easy to extend with new models and algorithms, allowing it to continuously evolve as a common tool for research and development in a wide range of inertial sensing applications.

Acknowledgements

This work was supported by the UK Engineering and Physical Sciences Research Council under the Basic Technology Research Programme, Grant C523881 entitled "Research Consortium in Speckled Computing".

8. REFERENCES

- E. Foxlin, "Inertial head-tracker sensor fusion by a complementary separate-bias Kalman filter," in *Proc.* of the IEEE Virtual Reality Annual Int. Symp., 1996, pp. 185–194.
- [2] E. R. Bachmann, "Inertial and magnetic tracking of limb segment orientation for inserting humans into synthetic environments," Ph.D. dissertation, Naval Postgraduate School, California, 2000.
- [3] H. Luinge and P. Veltink, "Measuring orientation of human body segments using miniature gyroscopes and accelerometers," *Medical and Biological Engineering* and Computing, vol. 43, no. 2, pp. 273–282, 2005.
- [4] X. Yun and E. R. Bachmann, "Design, implementation, and experimental results of a quaternion-based Kalman filter for human body motion tracking," *IEEE Trans. on Robotics*, vol. 22, no. 6, pp. 1216–1227, 2006.
- [5] A. Young, M. Ling, and D. K. Arvind, "Orient-2: A realtime wireless posture tracking system using local orientation estimation," in *Proc. of the 4th workshop* on *Embedded networked sensors*. ACM, 2007, pp. 53–57.
- [6] A. D. Young, M. J. Ling, and D. K. Arvind, "Distributed estimation of linear acceleration for improved accuracy in wireless inertial motion capture," in *Proc. of the 9th ACM/IEEE Int. Conf. on Information Processing in Sensor Networks*, 2010, pp. 256–267.
- [7] A. D. Young, "From posture to motion: The challenge for real time wireless inertial motion capture," in *Proc.* of the 5th Int. Conf. on Body Area Networks. ICST, September 2010.
- [8] S. Sun, X. Meng, L. Ji, J.-K. Wu, and W. Wong, "Adaptive sensor data fusion in motion capture," in *Proc. of the 13th Int. Conf. on Information Fusion*, July 2010.

- [9] E. Egea-Lopez, J. Vales-Alonso, A. Martinez-Sala, P. Pavon-Marino, and J. García-Haro, "Simulation tools for wireless sensor networks," in *Proc. of the Int.* Symp. on Performance Evaluation of Computer and Telecommunication Systems, 2005, pp. 559–566.
- [10] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and scalable simulation of entire TinyOS applications," in *Proc. of the 1st Int. Conf. on Embedded Networked Sensor Systems.* ACM, 2003, pp. 126 – 137.
- [11] H. N. Pham, D. Pediaditakis, and A. Boulis, "From simulation to real deployments in WSN and back," in Proc. of the IEEE Int. Symp. on a World of Wireless, Mobile and Multimedia Networks, 2007, pp. 1–6.
- [12] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with COOJA," in Proc. of the 31st IEEE Conf. on Local Computer Networks, 2006, pp. 641–648.
- [13] B. L. Titzer, D. K. Lee, and J. Palsberg, "Avrora: scalable sensor network simulation with precise timing," in *Proc. of the 4th Int. symposium on Information Processing in Sensor Networks*. ACM, 2005.
- [14] Z. Zhang, L. W. Wong, and J.-K. Wu, "3D upper limb motion modeling and estimation using wearable micro-sensors," in *Proc. of the Int. Conf. on Wearable* and *Implantable Body Sensor Networks*. IEEE Computer Society, 2010, pp. 117–123.
- [15] A. Young, "Comparison of orientation filter algorithms for realtime wireless inertial posture tracking," in *Proc. of the 6th Int. Workshop on Wearable and Implantable Body Sensor Networks.* IEEE Computer Society, June 2009, pp. 59–64.
- [16] E. Foxlin, M. Harrington, and Y. Altshuler,
 "Miniature 6-DOF inertial system for tracking HMDs," in *Proc. of SPIE Helmet and Head-Mounted Displays III*, 1998, pp. 214 – 228.
- [17] A. D. Young, "Wireless realtime motion tracking system using localised orientation estimation," Ph.D. dissertation, University of Edinburgh, 2010.
- [18] A. D. Young, "Use of body model constraints to improve accuracy of inertial motion capture," in Proc. of the Int. Conf. on Wearable and Implantable Body Sensor Networks. IEEE Computer Society, 2010, pp. 180–186.
- [19] G. Wahba, "A least squares estimate of satellite attitude," SIAM Review, vol. 7, no. 3, pp. 384–386, 1965.
- [20] F. Markley and D. Mortari, "Quaternion attitude estimation using vector observations," *Journal of the Astronautical Sciences*, vol. 48, no. 2, pp. 359–380, 2000.
- [21] T. Harada, H. Uchino, T. Mori, and T. Sato, "Portable orientation estimation device based on accelerometers, magnetometers and gyroscope sensors for sensor network," in *Proc. of IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, 2003, pp. 191–196.
- [22] D. Roetenberg, H. Luinge, and P. Slycke, "Xsens MVN: Full 6DOF human motion tracking using miniature inertial sensors," Xsens Motion Technologies B.V., Tech. Rep., 2009.

- [23] G. Welch and E. Foxlin, "Motion tracking: no silver bullet, but a respectable arsenal," *IEEE Computer Graphics and Applications*, vol. 22, no. 6, pp. 24–38, 2002.
- [24] L. Ojeda and J. Borenstein, "Personal dead-reckoning system for GPS-denied environments," in *Proc. of the* 5th IEEE Int. Workshop on Safety, Security and Rescue Robotics. IEEE, 2007, pp. 1–6.
- [25] X. Yun, E. R. Bachmann, H. Moore, and J. Calusdian, "Self-contained position tracking of human movement using small inertial/magnetic sensor modules," *Proc.* of the IEEE Int. Conf. on Robotics and Automation, pp. 2526–2533, 2007.
- [26] K. Shoemake, "Animating rotation with quaternion curves," ACM SIGGRAPH computer graphics, vol. 19, no. 3, pp. 245–254, 1985.
- [27] K. Shoemake, "Quaternion calculus for animation," Math for SIGGRAPH (ACM SIGGRAPH'91 Course Notes #2), 1991.
- [28] M.-J. Kim, M.-S. Kim, and S. Y. Shin, "A general construction scheme for unit quaternion curves with simple high order derivatives," in *Proc. of the 22nd Annual Conf. on Computer Graphics and Interactive Techniques.* ACM, 1995, pp. 369–376.
- [29] R. Boynton, "Precise measurement of mass," in Proc. of the 60th Annual Conf. of the Society of Allied Weight Engineers. SAWE Inc., May 2001.
- [30] N. Derby and S. Olbert, "Cylindrical magnets and ideal solenoids," *American Journal of Physics*, vol. 78, no. 3, pp. 229 – 235, March 2010.
- [31] R. Hemsley, "Interpolation on a magnetic field." Bristol University, Tech. Rep., 2009.
- [32] H. Weinberg, Application Note AN-625: Modifying the Range of the ADXRS150 and ADXRS300 Rate Gyros, Analog Devices, 2003.
- [33] L. Hanlen, D. Miniutti, D. Rodda, and B. Gilbert, "Interference in body area networks: Distance does not dominate," in *IEEE 20th Int. Symp. on Personal*, *Indoor and Mobile Radio Communications*, 2009, pp. 281–285.
- [34] O. Chipara, G. Hackmann, C. Lu, W. Smart, and G. Roman, "Practical modeling and prediction of radio coverage of indoor sensor networks," in *Proc. of the* 9th ACM/IEEE Int. Conf. on Information Processing in Sensor Networks. ACM, 2010, pp. 339–349.
- [35] S. Lee, M. Lerotic, A. Sani, Y. Zhao, J. Keegan, Y. Hao, and G. Yang, "Articulated postures for subject-specific RF simulation," in *Proc. of the Int. Conf. on Wearable and Implantable Body Sensor Networks.* IEEE Computer Society, 2010, pp. 96–101.
- [36] D. Smith, D. Miniutti, L. Hanlen, D. Rodda, and B. Gilbert, "Dynamic narrowband body area communications: Link-margin based performance analysis and second-order temporal statistics," in *Proc. of the IEEE Conf. on Wireless Communications* and Networking, 2010, pp. 1–6.
- [37] Y. Xiaoping, E. Bachmann, and R. McGhee, "A simplified quaternion-based algorithm for orientation estimation from Earth gravity and magnetic field measurements," *IEEE Trans. on Instrumentation and Measurement*, vol. 57, no. 3, pp. 638–650, March 2008.