

Motion Control in VR — Real-time Upper Limb Tracking via IMU and Flex Sensor

Kaiyuan Zhu, Liang Shi
Stanford University

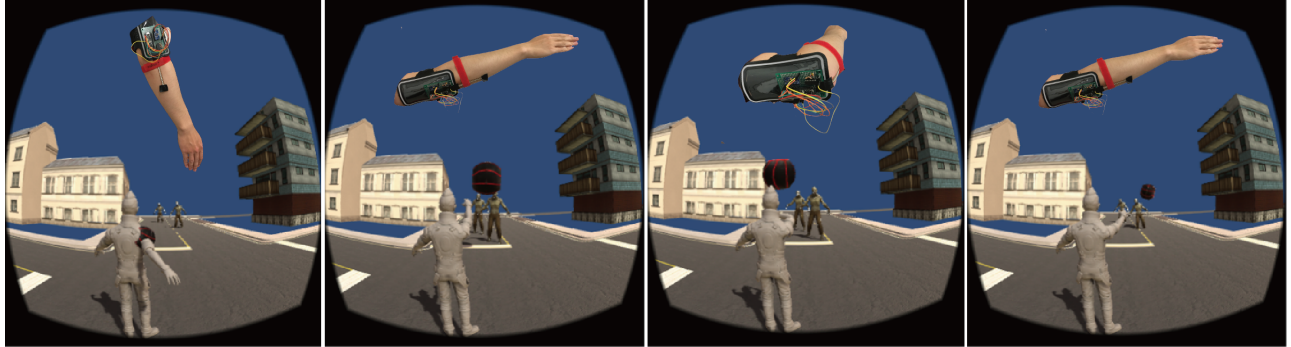


Figure 1: Upper limb gestures for controlling the avator in virtual environment. The side-view of player’s gestures are overlaid on top of each image. (Left) The upper limb points straight down to ground. The player enters the idle mode where an energy ball attaches to his shoulder. (Mid-left) The upper limb is raised up parallel to the ground. The player enters the loading mode where energy ball floats up to the air. (Mid-right) The upper limb is flexed to 90 degree. The player enters the spinning mode where energy ball starts to fast spin and ready for launch. (Right) The upper limb is unflexed back to gesture in loading mode. The player enters the launch mode where the energy ball is shot out.

Abstract

Real-time limb motion tracking offers a natural and intuitive way to interact with virtual environment in VR head-mounted displays. In this report, we present an affordable while accurate and low-latency tracker design using a single IMU plus a flex sensor. A kinematically-constrained model is developed to fuse the 7-degree measurement and reconstruct the state of upper limb. We build a prototype together with a third person shooting game to demonstrate the effectiveness of our design.

Keywords: upper limb tracking, inertial measurement units, flex sensor, real-time

1 Introduction

The bloom of virtual reality(VR) has presaged a revolution on the existing ways of doing things — from playing video games, watching live events, to shopping apparel online or interacting with a doctor. Instead of offering an immersive viewing experience alone, devices like HTC Vive allows the user to physically participate in the virtual world with two wireless controllers. To further strengthen such interaction, a full upper limb tracking would presumably create a more realistic feeling, in particular for games like third-person shooter where the player will synchronize with an avatar.

A number of sensing technologies have been developed for real-time motion tracking. These technologies consist of mechanical, inertial, acoustic, magnetic, optical, and radio frequency sensing. Recent advances in low-cost micro-electro-mechanical systems (MEMS) sensors with self-contained inertial measurement units(IMU) has made it possible to build wrist-watch-sized tracker[Bachmann et al. 2001; Bachmann et al. 2003]. An IMU typically consists of a tri-axial accelerometer, a tri-axial gyroscope and a tri-axial magnetometers. They measure acceleration, angular velocity and magnetic field independently in real-time without the assist of artificially generated source, thus will not suffer from range limitation and interference.

IMU-based upper limb tracking has been proposed using multiple IMUs design, typically with one on each independent segment, i.e arm, forearm. The relative orientation of each segment comparing to an Earth-fixed frame is reconstructed from corresponding IMU measurement. They together resemble the full state of upper limb.

In this report, we show by enforcing kinematically-constrained model, a single IMU plus a flex sensor is enough to obtain an accurate reconstruction. The flex sensor is a 1D sensor that alters the resistance according to the extent of flex. We built a wearable poker-size sensor prototype using off-the-shelf devices. To demonstrate the effective our design, we created a virtual environment in which the player is surrounded by zombies. To kill the zombies, the player has to make different upper limb gestures to launch energy balls. By designing a collection of gestures, we thoroughly evaluate the sensitivity and robustness of our sensor. Through our work, we wish to stimulate a follow-on research about how to simplify the sensor design by capitalizing on the intrinsic constraints of human body.

2 Related Work

Upper limb motion tracking has drawn a widespread research interest for its broad applications in neurological disorders diagnosis, medical rehabilitation, human computer interaction and gaming. A number of technologies have been developed for human motion capture including mechanical trackers, active magnetic trackers, optical tracking systems, acoustic, and inertial tracking systems. In this section, we offer a detailed review of inertial sensor based systems and briefly introduce the others.

Mechanical trackers can be generalized into two categories. Body-based systems attach exoskeleton to the articulated structure and goniometers within the skeletal linkages measure joint angles[gyp]. Ground-based systems attach one end of a boom or shaft to a tracked object and typically have six degrees of freedom (DOFs)[pha]. Such systems normally track a single rigid body, but can provide haptic feedback.

Optical trackers can also be generalized into two categories. Pattern recognition systems sense structured illumination to determine the position and/or orientation [Welch et al. 1999]. Image-based systems use multiple cameras to track actively or passively marked points on moving objects for 3D reconstruction [MCS ; QMC].

Active magnetic trackers use sets of orthogonally mounted mini coils to sense a set of sequentially generated magnetic fields. The induced current in each sensor coil helps determine the orientation. Variation of current intensity across the sensor coils is proportional to the distance from the field transmitter and it helps determine the position [Raab et al. 1979].

Ultrasonic trackers use time-of-flight and triangulation or phase-coherence to locate the position. In either design, the time interval or phase difference of the emitted reference signal and the bounced back signal is measured for calculating the distance between sensor and target.

Inertial sensors were first used in the detection of human movements in the 1950s [Inman et al. 1953]. However, these sensors were not commercially available until, in recent years, their performance had been dramatically improved while their size had been pushed down to fit in small wearable devices.

In particular, a number of multiple-IMU design have be proposed for upper limb tracking. The underlying idea of these methods is based on integration/double integration of partial or full IMU measurements. In [Zhou and Hu 2005], the upper limb is modeled as a kinematic chain using six joint variables: three for the shoulder(arm) and three for the elbow(forearm). An extended Kalman filter is applied to fuse the data from accelerometer and gyroscope in order to reduce errors and noise. In [Yun and Bachmann 2006], the aforeheadmetioned model is further extended by a quaternion-based Kalman filter which avoids the singularity introduced by Euler representation. The magnetometer measurements are also incorporated to improve the accuracy. Xsens [Roetenberg et al. 2009] and InterSense [Foxlin and Harrington 2000] employed similar ideas by using hybrid inertial sensors and magnetic or ultrasonic sensors. In [El-Gohary et al. 2011], El-Gohary et al. combines kinematic models designed for control of robotic arms with the unscented Kalman filter (UKF) to estimate the angles of human arm and forearm using two Opal sensors. The above methods can be considered as orientation-based approaches which require only one integration.

The other approaches use double integration to directly track the position with the help of additional constraints, such as geometrical model [Tao and Hu 2008; Kim et al. 2011; Zhang et al. 2009] and zero-velocity update (ZUPT) [Chen and Hu 2012]. In [Tao and Hu 2008], Tao et al. presented a 3D hand trajectory tracking system by fusing IMU and webcam data. A physical arm constraint is applied to improve the systems performance. In [Kim et al. 2011], Kim et al. proposed a wearable tracking system with two IMU mounted on wrist and elbow. The upper limb trajectory is calculated by tracking the position of forearm and upper arm. Additional geometrical models are enforced to mitigate the double integration drift errors.

Recent advances in depth-sensing camera also spawned a serious of work combining IMU with devices like Microsoft Kinect. In [Bo et al. 2011], Lanari et al. developed a knee angle estimation system by fusing data from both IMU and Kinect. For an accurate orientation and position tracking, different geometrical models are taken into account in sensor fusion [Mefoued 2014; Jeong 2014; Tian et al. 2015].

In [Hyde et al. 2008], Hyde et al. proposed a joint rotation model for upper limb and a tracking approach focusing on minimizing the number of sensors. Inspired by this work, we present a single IMU

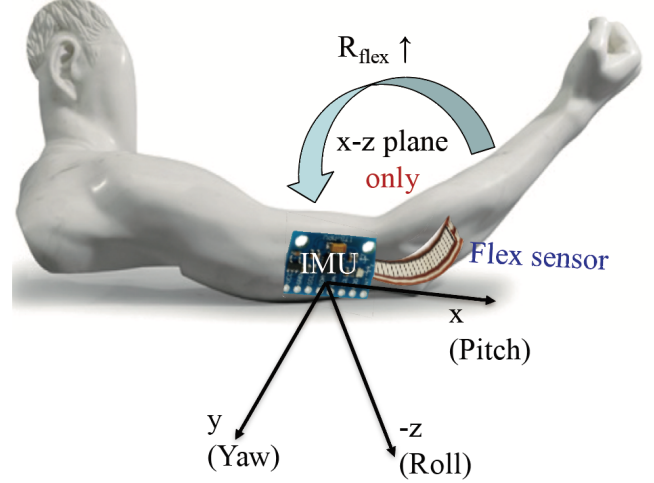


Figure 2: Kinematic model of upper limb

plus flex sensor design that achieves the orientation and positional tracking of upper limb. We note our design is not sensitive to rotation of forearm, nevertheless, it uniquely determines the position of hand.

3 Kinematic Model of Upper Limb

We first introduce our kinematic model of upper limb and show how we can reduce the necessity of using a second IMU. Fig. 2 illustrates the schematic design of our sensor and the kinematic model. We mount the only IMU on the arm at the location close to elbow. The flex sensor spans the entire elbow and reaches to the end of arm and the start of forearm. This arrangement allows IMU and flex sensor to be integrated in one circuit board.

We establish the IMU coordinates based on the right-hand rule and aircraft principle axes. The x -axis (Pitch) heads to the direction where the arm points out, the y -axis (Yaw) lies in the IMU plane and heads perpendicular to x -axis, the z -axis (Roll) stands orthogonal to the xy plane so that the $-z$ direction points out from the IMU and away from the arm.

From IMU, the three-axial gyroscope measurement $\omega_x, \omega_y, \omega_z$ and three-axial accelerometer measurement a_x, a_y, a_z are read out and used to track the orientation of arm. Through this report, we use quaternion-based representation to avoid the gimbal lock. Define the initial position of upper limb to be pointing straight down to the ground, where the gravity direction aligns with the x -axis. In that state, the accelerometer vector $\mathbf{v}_{up} = (-1, 0, 0)$ with magnitude $9.81m/s^2$. Let rotation quaternion $\mathbf{q}^{(t)}$ represent the rotation from sensor frame to inertial frame at time t and $\mathbf{q}^{(0)} = (1, 0, 0, 0)$ be the initial quaternion. At time t , the accelerometer measurements can be represented as vector quaternion $\mathbf{q}_a^{(sensor)} = (0, a_x, a_y, a_z)$; the gyroscope measurements can be represented as rotation quaternion $\mathbf{q}_\Delta = (\cos(\frac{\Delta t l}{2}), \frac{\omega_x}{l} \sin(\frac{\Delta t l}{2}), \frac{\omega_y}{l} \sin(\frac{\Delta t l}{2}), \frac{\omega_z}{l} \sin(\frac{\Delta t l}{2}))$ with $l = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}$. We employ the complementary filter to update $\mathbf{q}^{(t+\Delta t)}$:

1. compute accelerometer measurement in inertial coordinates

$$\mathbf{q}_a^{(inertial)} = \mathbf{q}^{(t)} \mathbf{q}_a^{(sensor)} \mathbf{q}^{(t)-1}$$

2. compute normalized vector part of $\mathbf{q}_a^{(inertial)}$,

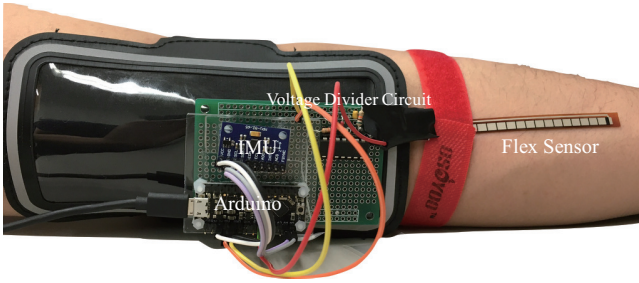


Figure 3: Sensor prototype

$$\mathbf{v}_{\text{inertial}} = \left(\frac{q_{a1}^{(\text{inertial})}}{\|q_a^{(\text{inertial})}\|}, \frac{q_{a2}^{(\text{inertial})}}{\|q_a^{(\text{inertial})}\|}, \frac{q_{a3}^{(\text{inertial})}}{\|q_a^{(\text{inertial})}\|} \right)$$

3. compute angle between $\mathbf{v}_{\text{inertial}}$ and \mathbf{v}_{up} ,

$$\phi = \arccos\left(-\frac{q_{a1}^{(\text{inertial})}}{\|q_a^{(\text{inertial})}\|}\right)$$
4. compute the normalized rotation axis between $\mathbf{v}_{\text{inertial}}$ and \mathbf{v}_{up} ,

$$\mathbf{v}_{\text{tilt}} = \mathbf{v}_{\text{inertial}} \times \mathbf{v}_{\text{up}}$$
5. compute $\mathbf{q}^{(t+\Delta t)}$ using complimentary filter,

$$\mathbf{q}^{(t+\Delta t)} = \mathbf{q}((1 - \alpha)\phi, \mathbf{v}_{\text{tilt}}) \mathbf{q}^{(t)} \mathbf{q}_{\Delta}$$

By rotating the initial position with $\mathbf{q}^{(t+\Delta t)}$, the orientation of arm can be determined in real-time.

To simplify the second IMU to flex sensor, we made key observation that once the orientation of arm is determined, the forearm can *only* rotate along y-axis, or equivalently flex within x-z plane. This is a consequence of the limited degree of freedom enforced by the joints connecting arm and forearm. To measure the extent of forearm flexing, we design a circuit using operational amplifier to convert the resistance change to voltage variation. A linear model is then fitted to map every voltage measurement to the corresponding flex angle. A detailed implementation will be illustrated in the next section.

In conclusion, we no longer treat upper limb tracking as a set of independent problem for each segment. Instead, it is solved in a two-stage fashion, where the orientation of arm is first estimated from IMU measurements and the rotation of forearm along *y*-axis is calculated from flex sensor measurements.

4 Implementation

We integrate the microprocessor, IMU, flex sensor and signal conditioning circuit onto a single circuit board. Fig. 3 shows the sensor prototype attached on the arm. We use Arduino Metro-mini as the microprocessor for IMU and flex sensor data streaming and processing. We choose MPU-9250, an IMU module that contains gyroscope, accelerometer and magnetometer, though only gyroscope and accelerometer measurements are used for calculation. The Arduino reads the IMU data through Inter-Integrated Circuit (I²C).

Fig. 4 shows the signal conditioning circuit for the flex sensor. It consists of an operational amplifier (Op-Amp) chip with the non-inverting (+) terminal connecting to reference voltage $R_{\text{Ref}} = 2.5\text{V}$ and the inverting (-) terminal connecting to a reference resistor and the power supply. The flex sensor bridges the inverting terminal and the output terminal. Based on the Golden Rules of Op-Amp, the output voltage of the circuit is determined by

$$V_{\text{out}} = V_{\text{Ref}} - IR_{\text{flex}}, \quad (1)$$

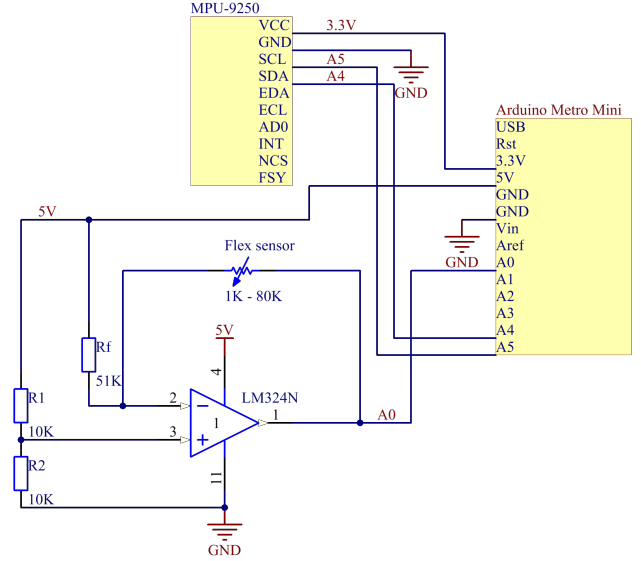


Figure 4: Schematics of the signal conditioning circuit

where I is a constant current given by

$$I = \frac{V_{CC} - V_{\text{Ref}}}{R_f}, \quad (2)$$

supply voltage $V_{cc} = 5.0\text{V}$ and reference resistance $R_f = 51\text{k}\Omega$. The advantage of our design over naive voltage divider is the output voltage becomes a linear function of the resistance of flex sensor. Note our design assumes the resistance of flex sensor is a linear response of the flex angle. In reality, the physical transfer function is not necessarily linear but also depends on other variables such as the pivot of flexing. In our test, the resistance of flex sensor R_{flex} reaches the minimum of $12\text{k}\Omega$ with no flex and climbs up to the maximum $50\text{k}\Omega$ when the elbow is fully flexed. Thus, we chose the reference resistor of $51\text{k}\Omega$, resulting a predicted output between 1.9V with no flex to 0V with fully flexed elbow. The circuit output is connected to the Analog-Digital-Converter(ADC) pin on the microprocessor. In the Arduino microprocessor, the digitized voltage reading is mapped to $0 - 90$ degree flexing angle using linear model

$$\theta_{\text{flex}} = -(V_{\text{out}} - 390)/3. \quad (3)$$

The rotation quaternion of the arm and the flexing angle of the elbow are feed to the PC via a USB cable at 9600 baud rate.

Fig. 5[left] shows a third person controller we download from Unity Asset Store. His entire body is assembled following the hierarchy demonstrated in Fig. 5[Right]. For instance, the arm object can be found by tracing down the following path
 Skeleton→Hips→Spine→...→Neck→Shoulder→Arm

In this model, the shoulder object is the parent of arm object, and the arm object is the child of shoulder object. Upon obtaining the arm's rotation quaternion from the Arduino microprocessor, we attach a C# script under the arm object and use the following line of code to control the motion of the arm

```
transform.rotation = initRotation*
transform.parent.rotation*inertialRotation;
```

where three quaternions are multiplied sequentially. The "inertial-Rotation" is the rotation quaternion passed from the Arduino microprocessor. The "transform.parent.rotation" is the rotation state of

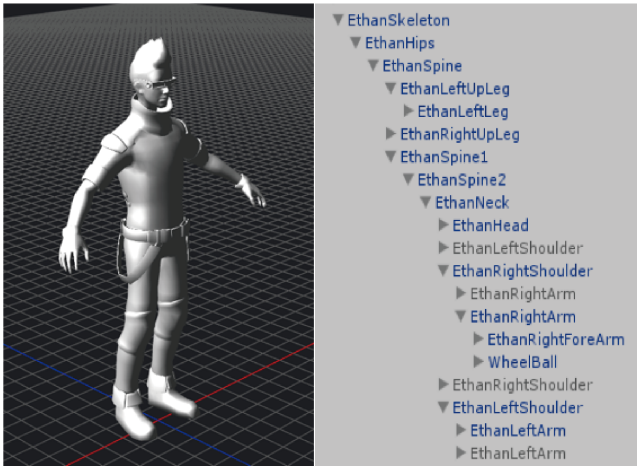


Figure 5: (Left) Body model of the third person controller. (Right) Model hierarchy of the third person controller.

the parent object, i.e., the shoulder’s rotation in quaternion. Since “transform.rotation” in Unity represents the rotation state in world space, we need to pre-multiply microprocessor output “inertialRotation” by its parent rotation state “transform.parent.rotation” such that the arm will rotate with respect to the shoulder. The “initRotation” is a quaternion that rotates the initial state of avatar’s arm to a kinematically nature orientation. Here, the “initRotation” is chosen such that avatar’s arm synchronizes with the player’s for the given initial posture. For different initial postures, “initRotation” can be different. Before the start of the game, we ask the player to hold his or her right arm straight down until the actual game starts. This ensures the orientations of avatar’s arm can be correctly synchronized with the player’s throughout the game.

To reconstruct the flexing state of forearm, we attach another script under the forearm object.

```
transform.rotation = transform.parent.rotation
*Quaternion.Euler(flex, -flex, 0.0f);
```

The “flex” is the flexing angle outputted from Arduino microprocessor. Note that we not only rotate in y -axis because when the sensor is attached to the arm, there is usually a constant 45 degree offset which is counteracted by the above formulation. If the axis is strictly aligned, the rotation should only be performed on one axis. Again, we pre-multiply “transform.parent.rotation”, i.e., the rotation state of arm object, such that the forearm will rotate with respect to the arm based on the flex data measurement.

5 Evaluation

To demonstrate the effectiveness of our sensor, We designed a third-person shooting game in the virtual environment using Unity, which is shown in Fig. 6. We implement the gesture control with a state machine. An avatar of the player arrives in an abandoned town with zombies surrounded. When the player points his arm straight down, he keeps staying in the idle mode, i.e Fig. 1 [Left]. As long as he raises and points out his arm, he enters the loading mode where an energy ball rises up from the ground Fig. 1[Mid-left]. Once he flexes his forearm, he steps into the spinning mode Fig. 1[mid-right], where the ball starts to spin fast and ready to be launched. Finally, when he wave the forearm back to straight, the ball is shot out to the direction he targeted Fig. 1[Right]. He can restore the position of the ball by flexing his arm again. The zombie will be blown to the sky if it is hit by the flying ball.

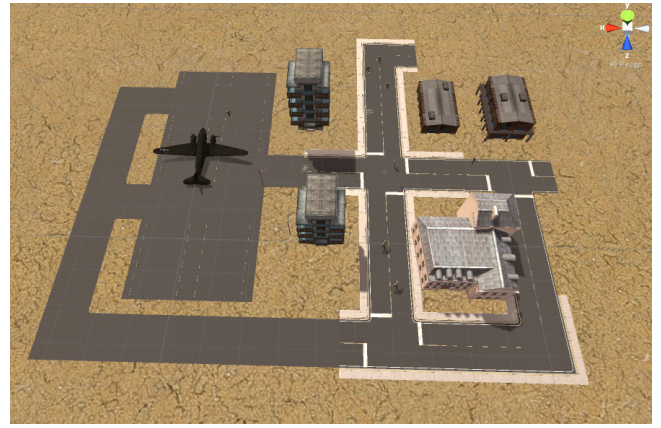


Figure 6: Full virtual environment

In experiment, our sensor correctly detects different upper limb gestures and display the corresponding motions in the game.

6 Discussion

We choose a cost effective approach that allows synchronization of physical arm gestures to the motion of virtual character. However, there are some limitations of our current design. The circuit board needs to be mounted on the arm with a correct location and orientation, and one end of the flex sensor needs to be strapped to the forearm. If the player does not wear the device correctly, the IMU will give an offset to the orientation of the avatar’s arm. The flex sensor response also differs from people to people due to the variation of arm thickness. If the flex sensor does not respond to the elbow motion due to improper wearing, and the forearm in the virtual environment will rotate in a wrong manner. The flex sensor may experience fatigue over time and gives inconsistent output. In addition, the complimentary filter does not eliminate the drift in yaw direction. As a result, the orientation of the virtual arm will drift over time. To improve our model, we may consider implement more robust filter algorithm such as the Kalman filter. We may also search for replacements of the flex sensor that is more suitable to wear around human elbow.

Acknowledgements

We would like to thank Prof. Wetzstein and Robert for the fruitful discussion.

References

- BACHMANN, E. R., MCGHEE, R. B., YUN, X., AND ZYDA, M. J. 2001. Inertial and magnetic posture tracking for inserting humans into networked virtual environments. In *Proceedings of the ACM symposium on Virtual reality software and technology*, ACM, 9–16.
- BACHMANN, E. R., YUN, X., MCKINNEY, D., MCGHEE, R. B., AND ZYDA, M. J. 2003. Design and implementation of marg sensors for 3-dof orientation measurement of rigid bodies. In *Robotics and Automation, 2003. Proceedings. ICRA’03. IEEE International Conference on*, vol. 1, IEEE, 1171–1178.
- BO, A., HAYASHIBE, M., AND POIGNET, P. 2011. Joint angle estimation in rehabilitation with inertial sensors and its integration with kinect. In *EMBC’11: 33rd Annual International Confer-*

- ence of the IEEE Engineering in Medicine and Biology Society, IEEE, 3479–3483.
- CHEN, L., AND HU, H. 2012. Imu/gps based pedestrian localization. In *Computer Science and Electronic Engineering Conference (CEEC), 2012 4th*, IEEE, 23–28.
- EL-GOHARY, M., HOLMSTROM, L., HUISINGA, J., KING, E., MCNAMES, J., AND HORAK, F. 2011. Upper limb joint angle tracking with inertial sensors. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, IEEE, 5629–5632.
- FOXLIN, E., AND HARRINGTON, M. 2000. Weartrack: A self-referenced head and hand tracker for wearable computers and portable vr. In *Wearable Computers, The Fourth International Symposium on*, IEEE, 155–162.
- Gypsy4 mechanical motion capture 2004 [online]. available: <http://www.animazoo.com/products/gypsy4.htm>.
- HYDE, R. A., KETTERINGHAM, L. P., NEILD, S. A., AND JONES, R. J. 2008. Estimation of upper-limb orientation based on accelerometer and gyroscope measurements. *Biomedical Engineering, IEEE Transactions on* 55, 2, 746–754.
- INMAN, V. T., EBERHART, H. D., ET AL. 1953. The major determinants in normal and pathological gait. *J Bone Joint Surg Am* 35, 3, 543–558.
- JEONG, J. 2014. Normalized estimate of adaptive filter for non-minimum phase transfer function component: An illustrative case study. *Neurocomputing* 133, 295–308.
- KIM, J., YANG, S., AND GERLA, M. 2011. Stroketrack: wireless inertial motion tracking of human arms for stroke telerehabilitation. In *Proceedings of the First ACM Workshop on Mobile Systems, Applications, and Services for Healthcare*, ACM, 4.
- Motion capture systems from vicon peak 2005 [online]. available: <http://www.vicon.com/>.
- MEFOUED, S. 2014. A robust adaptive neural control scheme to drive an actuated orthosis for assistance of knee movements. *Neurocomputing* 140, 27–40.
- Products: Phantom 1.5/6dof and 3.0/6dof 2005 [online]. available: http://www.sensable.com/products/phantom_ghost/premium6dof.asp.
- Qualisys motion capture analysis system of kinematics data 2005 [online]. available: <http://www.qualisys.com/>.
- RAAB, F. H., BLOOD, E. B., STEINER, T. O., AND JONES, H. R. 1979. Magnetic position and orientation tracking system. *Aerospace and Electronic Systems, IEEE Transactions on*, 5, 709–718.
- ROETENBERG, D., LUINGE, H., AND SLYCKE, P. 2009. Xsens mvn: full 6dof human motion tracking using miniature inertial sensors. *Xsens Motion Technologies BV, Tech. Rep.*
- TAO, Y., AND HU, H. 2008. A novel sensing and data fusion system for 3-d arm motion tracking in telerehabilitation. *Instrumentation and Measurement, IEEE Transactions on* 57, 5, 1029–1040.
- TIAN, Y., MENG, X., TAO, D., LIU, D., AND FENG, C. 2015. Upper limb motion tracking with the integration of imu and kinect. *Neurocomputing* 159, 207–218.
- WELCH, G., BISHOP, G., VICCI, L., BRUMBACK, S., KELLER, K., ET AL. 1999. The hiball tracker: High-performance wide-area tracking for virtual and augmented environments. In *Proceedings of the ACM symposium on Virtual reality software and technology*, ACM, 1–ff.
- YUN, X., AND BACHMANN, E. R. 2006. Design, implementation, and experimental results of a quaternion-based kalman filter for human body motion tracking. *Robotics, IEEE Transactions on* 22, 6, 1216–1227.
- ZHANG, Z., HUANG, Z., AND WU, J. 2009. Hierarchical information fusion for human upper limb motion capture. In *Information Fusion, 2009. FUSION'09. 12th International Conference on*, IEEE, 1704–1711.
- ZHOU, H., AND HU, H. 2005. Kinematic model aided inertial motion tracking of human upper limb. In *Information Acquisition, 2005 IEEE International Conference on*, IEEE, 6–pp.