

2017-03-17

Teleoperation control of Baxter robot using Kalman filter-based sensor fusion

Li, Chunxu

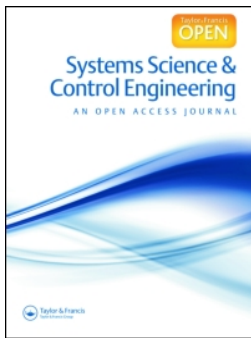
<http://hdl.handle.net/10026.1/9779>

10.1080/21642583.2017.1300109

Systems Science & Control Engineering

Informa UK Limited

All content in PEARL is protected by copyright law. Author manuscripts are made available in accordance with publisher policies. Please cite only the published version using the details provided on the item record or document. In the absence of an open licence (e.g. Creative Commons), permissions for further reuse of content should be sought from the publisher or author.



Systems Science & Control Engineering

An Open Access Journal

ISSN: (Print) 2164-2583 (Online) Journal homepage: <http://www.tandfonline.com/loi/tssc20>

Teleoperation control of Baxter robot using Kalman filter-based sensor fusion

Chunxu Li, Chenguang Yang, Jian Wan, Andy SK Annamalai & Angelo Cangelosi

To cite this article: Chunxu Li, Chenguang Yang, Jian Wan, Andy SK Annamalai & Angelo Cangelosi (2017) Teleoperation control of Baxter robot using Kalman filter-based sensor fusion, Systems Science & Control Engineering, 5:1, 156-167, DOI: [10.1080/21642583.2017.1300109](https://doi.org/10.1080/21642583.2017.1300109)

To link to this article: <http://dx.doi.org/10.1080/21642583.2017.1300109>



© 2017 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.



Published online: 17 Mar 2017.



Submit your article to this journal [↗](#)



Article views: 378




View related articles [↗](#)



View Crossmark data [↗](#)

Teleoperation control of Baxter robot using Kalman filter-based sensor fusion

Chunxu Li ^a, Chenguang Yang^{a,b}, Jian Wan^a, Andy SK Annamalai^c and Angelo Cangelosi^a

^aCenter for Robotics and Neural Systems, Plymouth University, Devon, UK; ^bZienkiewicz Centre for Computational Engineering, Swansea University, Swansea, UK; ^cMoray College, University of Highlands and Islands, Inverness, UK

ABSTRACT

Kalman filter has been successfully applied to fuse the motion capture data collected from Kinect sensor and a pair of MYO armbands to teleoperate a robot. A new strategy utilizing the vector approach has been developed to accomplish a specific motion capture task. The arm motion of the operator is captured by a Kinect sensor and programmed with Processing software. Two MYO armbands with the inertial measurement unit embedded are worn on the operator's arm, which is used to detect the upper arm motion of the human operator. This is utilized to recognize and to calculate the precise speed of the physical motion of the operator's arm. User Datagram Protocol is employed to send the human movement to a simulated Baxter robot arm for teleoperation. In order to obtain joint angles for human limb utilizing vector approach, RosPy and Python script programming has been utilized. A series of experiments have been conducted to test the performance of the proposed technique, which provides the basis for the teleoperation of simulated Baxter robot.

ARTICLE HISTORY

Received 4 February 2017
Accepted 23 February 2017

KEYWORDS

Kalman filter-based sensor fusion; vector approach; Baxter; MYO armband; Kinect

1. Introduction

With the rapid advancement of sensor and actuator techniques, the robot technology has been investigated to a great extent in wider areas, such as control system design, data sensor innovation, bionics and artificial intelligence. According to Garcia, Jimenez, De Santos, and Armada (2007), following the accomplishment of modern robots, industrial robots have now increasingly attracted considerable interest in the past decades, wherein, human–robot interaction (HRI) plays an increasingly important role in industrial robot application. The robot is thought to have the capacity to adjust to the modern needs. Human beings are able to satisfy the variation of the surrounding environment, hence, it would be effective if robot is operated by human depending on their actual abilities and this is defined as teleoperation (Tang, Wang, & Williams, 1998). The teleoperation innovation with the connection between human and robot has been widely investigated in Dautenhahn (2007). By using robotic teleoperation system, operator is able to control a remote robot conveniently through the internet.

However, there are some potential issues, such as genuine missing information, signal transmission delay due to the restrictions of transfer speed and system transmission protocol in Li, Ma, Yang, and Fu (2014). Despite these issues, it is still a critical and helpful instrument in the fields of HRI research. Hence, numerous teleoperation

applications have been reported in the literature. In Ijspeert (2008), a mutual control strategy for Baxter robot controller has been created. A strategy for emulating human written work aptitudes to a Baxter robot controller has been dealt by Yang, Chang, Liang, Li, and Su (2015a). Researchers are able to teleoperate a robot calligraphically by utilizing electromyography (EMG) signals and a haptic device. In Yang, Chen, Li, He, and Su (2015b), surface electromyography signals have been upgraded to perform teleoperation. The human operators can detect the situation in a haptic way and adjust muscle compression subliminally. Li et al. (2014) depict the improvement of a simulated robot teleoperation stage in light of hand signal acknowledgement utilizing visual data.

In addition, a direct approach to give robot a chance to imitate the human motion is developed, which is known as motion capture technology and it is a perfect strategy to transfer human abilities to robot side (Breazeal, 2000). To realize human motion capture, human body itself ought to be followed first. In the literature, there are various methods to achieve human motion capture. The most widely used method is to estimate the markers from the body of human operator, however, this may lead to a number of inconveniences to the user. Another method is utilizing image processing from typical cameras. However, this strategy is not reliable owing to the unstable body location capacities during the imaging process.

CONTACT Chenguang Yang  cyang@theiet.org

Other methods include stereo-vision cameras that have been applied to motion capture for depth data analysis. Unfortunately, its processing time is long, without effectiveness in real-time applications.

As one of the enabling techniques for teleoperation, motion capture primarily incorporates two interfaces, which are remotely wearable device input interface and detecting interface based on vision system. Moreover, a few sensors have been utilized for the visual system, such as Leap Motion and Kinect. In Reddivari et al. (2014), human motions are obtained by Kinect sensor, and by utilizing the vector approach, people can ascertain the joint points of Baxter robot. In Liu, Zhang, Fu, and Yang (2013), the welder-related work is caught by the Leap Motion sensor, which is estimated by a soldering robot with teleoperation networks. Moreover, the wearable devices, for example, exoskeleton (Jo, Park, & Bae, 2013) or joystick, or Omni haptic device (Ju, Yang, Li, Cheng, & Ma, 2014) are normally used. In this paper, we investigate the wearable device, MYO Armband together with the motion capture system using a Kinect sensor to teleoperate a Baxter robot optimized by Kalman filtering (KF) based sensor fusion. In Yang, Wang, and Hung (2002), KF method was used to overcome the shortcomings of Wiener filtering. KF is widely used as it can estimate the past, current and future state signal, even if the exact nature of the model is not known.

Essentially, filtering is a procedure of signal processing and transformation (removing or reducing undesirable components and enhancing the required components). People can implement the above steps either by software or hardware. Wherein, KF is a software filtering method minimizing mean square error, which is widely used as the best estimation criterion. Furthermore, KF uses the state space model of signal and noise to estimate the value of the previous time, after that the observed values of current time are updated. Based on the equations of established and observation system, the algorithm estimates the minimum mean square error of the signal to be processed.

Some researchers proposed the strategies using Kinect and MYO armbands to obtain the joints angles of human arms and then teleoperated with robots, for example, in Reddivari et al. (2014) and Yang, Chen, and Chen (2016). However, the accuracy of calculation of angular data in both the above strategies had issues owing to the existence of the noises. This paper develops an optimum strategy to eliminate the influence of the noises using KF-based sensor fusion. Precisely, the vector approach was used to calculate the five required joints angles, then the KF algorithm was applied to output a series of more accurate data with less errors. Additionally, we have effectively applied the control systems to teleoperate the Baxter robot. In order to do this, MYO armbands worn on the operator's lower arm and upper arm are utilized to identify and measure the angular velocity of human arm motion, and a Kinect-based body tracking system has also been utilized.

2. Preliminary

2.1. System configuration

To delineate the teleoperation of robot utilizing motion capture, an illustrative system was assembled. It comprises the body tracking system, Baxter robot and MYO armbands, as shown in Figure 1.

Motion capture is obtained by the Kinect sensor. In spite of the fact that there are diverse types of Kinect, it is utilized due to its low cost and it can provide the data required for this research. The Kinect device is associated with a remote computer, wherein, Processing programming software was introduced and used to get the position information from Kinect sensor.

This research utilizes Baxter, which is a semi-humanoid robot with arms of seven DOF joints and has impact avoidance abilities. Operators can control it for study through torque, speed and position mode, respectively. The overall experimental system was associated with and controlled by the development workstation, a remote

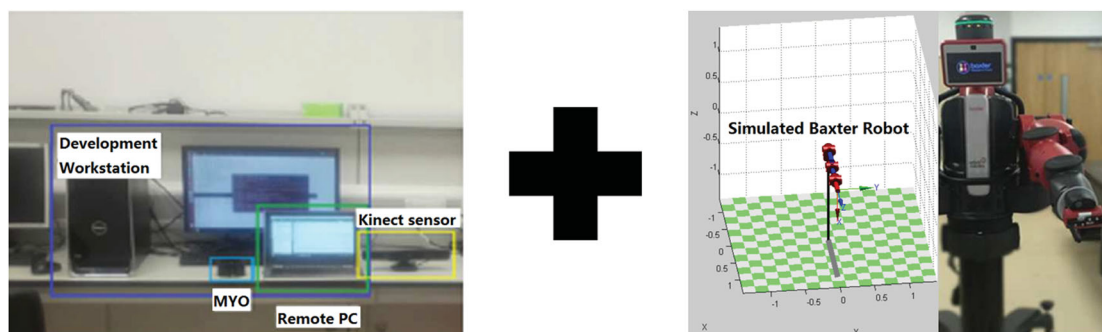


Figure 1. The experimental teleoperation system, left: development Workstation, MYO armband, Kinect sensor and Remote PC; right: MATLAB robotic toolbox based simulated Baxter robot.

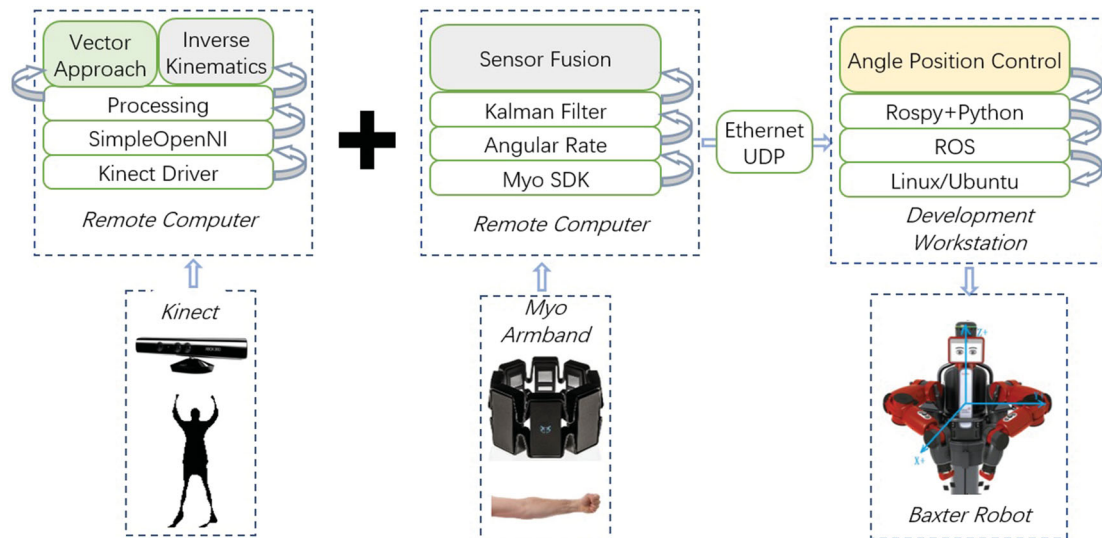


Figure 2. Diagram for the rule of research in total.

computer with Ethernet link, as well as a pair of MYO armbands. The principle of teleoperation system is represented as Figure 2.

2.2. Kinect sensor and development workstation

The Kinect sensor, utilized as a part of the proposed teleoperation system, is an arrangement of sensors created as a fringe device along with the Xbox video game device. Kinect is produced by Microsoft and generally utilized as a part of human motion 3D following, for example, non-verbal communication, motions and so on through its profundity camera in Cheok (2010). Besides, a RGB camera and a double infrared profundity sensor are located in front of the Kinect in Henry, Krainin, Herbst, Ren, and Fox (2012). Utilizing image and profundity sensors, Kinect can distinguish developments of a client. It does not require people to wear any additional interfaces. The image and profundity sensors are mounted on a base

with an engine, which permits it to change orientation for all the sensors, as in Figure 3.

From the Figure 3, the depth detector of the Kinect is contained within two units, which are the monochrome CMOS sensor and the infrared projector (label 1, Figure 3). They are working together, which is the basis of motion capture in Reddivari et al. (2014).

The image depth and RGB data collected from Kinect is illustrated in Figure 4. Human body can be rearranged by arranging several straight lines together to show human's positions and poses in 3D space. Kinect collects the statistic during the teleoperation process, such as human joint positions, speeds, and then sends them to the robot, by doing this, human-robot cooperation is achieved progressively. Compared to the traditional motion following device with complex programming, high cost and inconvenient setup, Kinect can be implanted in the control system working with open source (Sucipto, Harsoyo, & Rusmin, 2012).



Figure 3. Image of Kinect sensor: 1. Depth sensors, 2. RGB camera, 3. Motorized base. Modified from Reddivari et al. (2014).

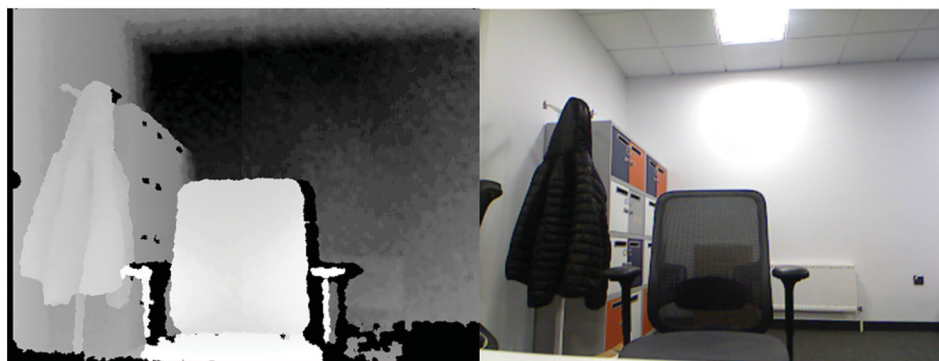


Figure 4. Image of depth and RGB sensor data collected from Kinect.

Numerous programming projects are accessible to be used with Kinect, such as OpenKinect, OpenNI, Microsoft Kinect for windows SDK (Boulos et al., 2011). OpenKinect is designed free for equipping the Kinect with computers and other devices in Villaroman, Rowe, and Swan (2011). OpenNI is able to support a large amount of different devices apart from Kinect in Villaroman et al. (2011), which utilizes NITE to get the skeleton information of the operator, according to Bolívar (2012). Microsoft Kinect for windows SDK is another commonly used platform produced by Microsoft in Bolívar (2012). SimpleOpenNI is an OpenNI and NITE wrapper for Processing in Ronchetti and Avancini (2011).

The utilization of programming software is imperative and ought to be relied on: (a) capacity of separating skeletal information; (b) similarity with different operating systems, for example, Windows and Linux; (c) clear documentation; and (d) straightforwardness for quick confirmation of calculations. After appropriate examination, Processing programming software which fulfils all the prerequisites is utilized. Operators can program via Kinect with SimpleOpenNI wrapper for OpenNI and NITE, according to Jean (2012), and skeleton information can be collected on both Windows and Linux platform in Zhou, Chang, and Li (2009). Processing is based on Java, hence fundamentally the same syntax can be used. All the functions utilized in this paper are given below:

PVector: A class to depict as a few dimensional vectors, particularly the Geometric vector. However, the statistic stores the parts of the vector (x , y for 2D, and x , y , z for 3D). The magnitude and direction can be obtained using the methods `mag()` and `heading()`, according to Ramos (2012).

pushMatrix() and *popMatrix()*: They can transfer the present transformation matrix into the matrix stack. The `pushMatrix()` can record the current coordinate system information to the stack and `popMatrix()` restores them. The `pushMatrix()` capacity and `popMatrix()` capacity are utilized in conjunction with other transformation

functions and might be used to control the extent of the changes (Ramos, 2012).

2.3. Robot Operating System and Rospy

Robot Operating System (ROS) is an adaptable operation system for the programming of robot. It is gathering the tools, conventions and libraries, which expects to rearrange the complicated task of robot activities to be more simple, according to Calo (2011). And the ROS can be setup under multiple platforms.

Rospy is a related Python customer library for ROS. The Rospy customer API empowers Python software engineers to rapidly interact with parameters, services and topics of ROS. The plan of Rospy requires usage speed (i.e. designer time) over runtime execution, hence calculations can be immediately prototyped and examined inside ROS. It is additionally perfect for the particular codes which have no critical path, for example, codes for initialization and configuration. A large amount of the ROS instruments are composed in Rospy script to develop the introspection abilities. A large number of the ROS devices, for example, Rostopic and Roservice, are based on top of Rospy (Cousins, 2010).

2.4. User Datagram Protocol

User Datagram Protocol (UDP) is one of the central individuals from the Internet convention suite (the arrangement of system conventions utilized for the Internet). Through this, PC software can send information, under this situation according to datagrams, to different PCs with an Internet Protocol (IP), by doing this, the unique transmission channels or information paths can be established without earlier interchanges. UDP is reasonable for purposes where the error is small enough and can be ignored or performed for the software, instead of the overhead of such handling at the system interface level. According to Tawfik et al. (2014), UDP was frequently

utilized for time sensitive applications owing to the fact that dropping packets are desirable to wait for delayed packets, where it is difficult to appear an alternative in a constant system.

2.5. MYO armband

MYO armband (shown in Figure 5) is a wearable device created by the Thalmic Labs. When the operator wears MYO armband on his/her arm, it can communicate with system through the function of Bluetooth. In addition, it has been built in eight EMG sensors and one IMU sensor with nine axis, hence the hand posture and arm motion can be detected. When people move their arm muscles, the eight EMG sensors can distinguish their hand gestures. Because every user has distinctive muscle size, type of skin and so on, the sensors can create information by electrical driving forces from arm muscles of operators. Critically, calibrating process is necessary for every user before using MYO armbands. Hence, the MYO armband can identify the motions and gestures of human limbs in a more accurate way.

2.6. Baxter research robot

In the experimental stage of the research, we utilized the simulated Baxter robot. The Baxter robot comprises of one torso, one 2-DOF head and two 7-DOF arms, which are shoulder joint: s_0, s_1 , elbow joint: e_0, e_1 and wrist joint: w_0, w_1, w_2 , respectively (Cousins, 2010). There are also coordinated cameras, torque sensors, encoders and sonar with the Baxter robot. Researchers can directly programme Baxter using open source, such as a standard ROS interface and so on. Seven Serial Elastic Actuators drive all joints of the Baxter robot arm, which gives passive consistence to minimize the constrain of any effect or contact (Cousins, 2010). Commonly, people teleoperate

and program the Baxter robot using ROS through Baxter RSDK running on the platform Ubuntu 12.04 LTS, wherein, ROS is an open source system with libraries, module devices and correspondences (Cousins, 2010). It improves the task of displaying and programming on a various types of automated platforms (Cousins, 2010).

The installation of the simulated Baxter robot in this research is demonstrated in Figure 1.

2.7. Kalman Filter

KF method is used to fuse the real-time dynamic low-level extra sensor data, which recursively decides statistical significance of the optimal fusion data combination by using the statistical characteristics of the measurement model. If the system has a linear dynamic model, the system noise and sensor noise can be represented by a white noise model with Gaussian distribution, KF would provide the unique statistically optimal estimate for the fusion data. The recursive nature of KF makes processed unnecessarily. The KF is divided into the continuous time KF and the discrete KF.

The actual physical system is usually continuous, as a consequence, the description of discrete systems often cannot completely replace the continuous time system. The mathematical model of the continuous system can be shown by the following formula from Davari, Gholami, and Shabani (2016),

$$\begin{aligned}\dot{x}(t) &= A(t)x(t) + B(t)u(t) + G(t)\omega(t), \\ y(t) &= H(t)x(t) + v(t),\end{aligned}\quad (1)$$

where x and u are n -dimensional state variables; y is dimension measurement vector; A is $n \times n$ -dimensional system matrix; G and B are $n \times r$ -dimensional system voice matrix; H is $m \times n$ -dimensional measurement



Figure 5. Image of MYO Armband.

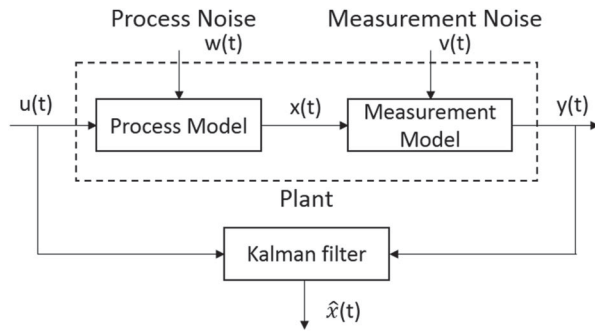


Figure 6. Working principle of continuous time KF.

matrix; ω is the zero-mean white noise vector of the r -dimensional continuous system; v is the continuous zero-mean value of the m -dimensional measurement of the white noise vector.

The continuous time KF equation, according to Särkkä and Solin (2012) is as follows:

$$\begin{aligned}\dot{\hat{x}}(t) &= A(t)\hat{x}(t) + B(t)u(t) + K(t)[y(t) - H(t)\hat{x}] \\ K(t) &= P(t)H^T(t)R^{-1}(t) \\ \dot{P}(t) &= P(t)H^T(t) + A(t)P(t) \\ &\quad - P(t)H^T(t)R^{-1}H(t)P(t) + G(t)s(t)G^T(t),\end{aligned}\quad (2)$$

where K is the filter gain matrix, \hat{x} is the estimated value of x and P is the estimated covariance matrix. Continuous time KF is based on the measured values of the continuous time process, and its method of solving the matrix differential equation is used to estimate the time continuous value of the system state variable, hence, this algorithm loses recursion. Its working principle is demonstrated in Figure 6.

3. Motion capture by Kinect

3.1. General calculation

Commonly, the motion capture calculations for upper limb depend on distances, locations and joint angles. The length between two specified points with two- and three-dimensional points can be obtained given by the following equations:

$$d_{2D} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \quad (3)$$

$$d_{3D} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}, \quad (4)$$

where (x_1, y_1) and (x_2, y_2) are points in 2D space, d_{2D} is the distance between these two points, (x_1, y_1, z_1) and (x_2, y_2, z_2) are points in 3D space, d_{3D} is the length between these two points.

The angles at all the joints are obtained by the law of cosines. The most extreme calculable angle is 180

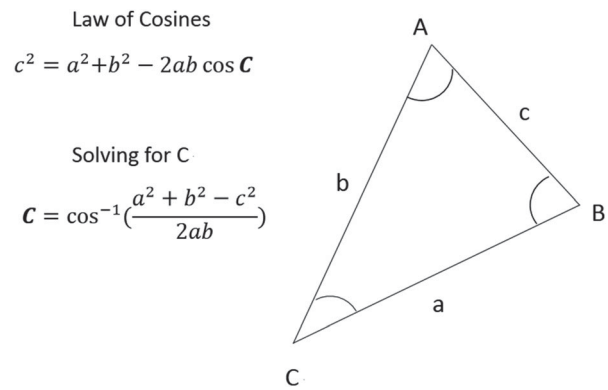


Figure 7. Mathematical principle description.

degrees. While computing the angles among the joints, an extra point is required to define at 180–360 degrees. After collecting the motion capture statistics, a triangle is drawn by utilizing any two joint points. From the other two points, the third point of the triangle can be obtained. Under this case, the coordinated statistics for every point of the triangle is known, we are able to find out the length of every side, instead of the value of each angle, which is still unknown. As shown in Figure 7, the magnitude of any coveted point can be calculated by applying the law of cosines.

Computations for the points of joint illustrate the length of sides a, b, c . Similarly, we can also calculate the angles of triangle using the law of cosines.

3.2. Vector approach

Kinect can identify every single joint with coordinate data of human body and supply with feedback about its statistics. All these directions are transformed into vectors and the particular angles of the joints can be obtained.

The coordinates of human body joints collected from Kinect in this strategy are under the Cartesian space, additionally, the particular angles from arms are computed. After mapping process by Kinect sensor, they are sent to teleoperate the Baxter as indicated by our requirement.

The five points Shoulder Pitch, Yaw and Roll as well as Elbow Pitch and Roll, shown as seen from Figure 8, are computed from the arm positions data that are extracted from the Kinect.

The computation of vectors is illustrated in Figure 9. According to Reddivari et al. (2014), the intense lines CO and CD illustrate left upper and ahead part of arm for human separately. Intense line BO represents the distance from left hip to left shoulder, and AO represents the length between right shoulder and left shoulder. The directions with coordinated data B_{X+} , B_{Y+} and B_{Z+} shows the axis system of Kinect in Cartesian space, where the point B is the origin.

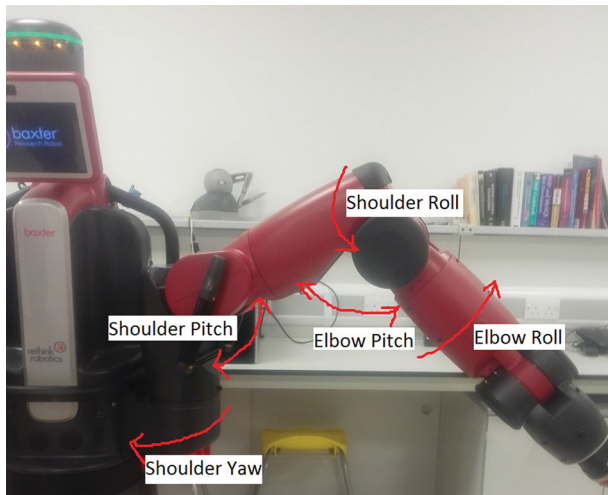


Figure 8. Demonstration of all related angles in Vector Approach: Shoulder Pitch, Yaw and Roll, Elbow Pitch and Roll.

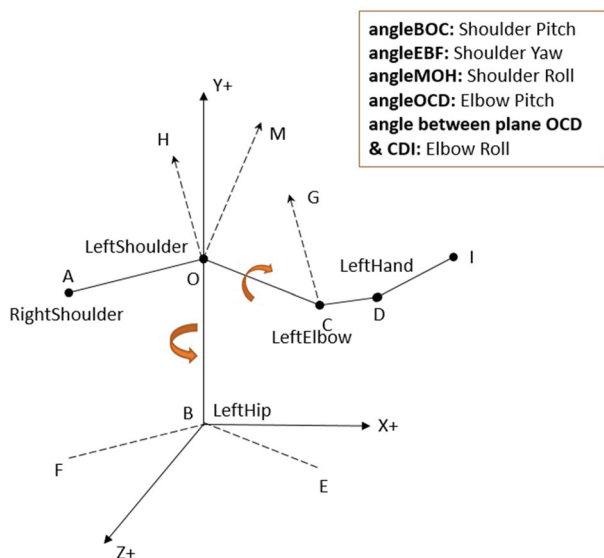


Figure 9. The principle of vector approach in mathematical computing (Reddivari et al., 2014).

Methodology for Computing Shoulder Pitch and Elbow Pitch: As in Figure 9, according to Reddivari et al. (2014), the angle $\angle BOC$ (Shoulder Pitch) is obtained by the distance of two points from vectors \overline{OB} to \overline{OC} . The computing methodology is then able to be defined by utilizing the three specified joints' position, which are shoulder (point O), elbow (point C) and hip (point B). Delivering these three points using the angle Of() function gives feedback of the value for angles, which are sent to Baxter directly. The $\angle OCD$ (Elbow Pitch), which is the angle among \overline{CD} and \overline{OC} , can be computed through sending hand, elbow and shoulder values into the angle Of() for working as well (Reddivari et al., 2014). In this methodology, we can use the angle Of() command in the Processing software to calculate any angles between two vectors.

Methodology for computing Shoulder Yaw: As we can see from Figure 9, according to Reddivari et al. (2014), the angle $\angle EBF$ (Shoulder Yaw) is obtained by a similar method by utilizing both shoulder point and elbow point, which are point A, O and C, respectively, where the vectors \overline{OC} and \overline{OA} are grouped together. However, the two above mentioned vectors \overline{OC} and \overline{OA} need to be anticipated into the plane XZ. By doing this, we are able to obtain the vectors \overline{BF} and \overline{BE} . Angle $\angle EBF$ (Shoulder Yaw) is the value of angle among \overline{BF} and \overline{BE} , which can be computed by utilizing angle Of() command in Processing.

Methodology for computing Elbow Roll and Shoulder Roll: The Elbow Roll is the angle between plane OCD and CDI, which can be calculated by angle Of() function. According to angular estimations, Shoulder Roll is most difficult to be calculated (Reddivari et al., 2014). As the computing is not straightforward and all the points are given in 3D plane, hence, the similar computing method utilized above cannot be accessible here.

In order to calculate the required angles, the point made by the vectors from elbow to hand, where its plane is opposite to the one from shoulder to elbow. It is going through the shoulder joint as well. The reference vector must be stably concerning the body. As a result, the reference vector can be computed by the intersecting point of vectors from shoulder to shoulder and shoulder to elbow.

For this situation, according to Reddivari et al. (2014), the normal line from that intersecting point of two vectors is used for continuous calculation. The vector \overline{OM} can be obtained by verifying the intersecting point between vectors \overline{OC} and \overline{OA} . The vector \overline{OM} is vertical to the plane obtained from the vectors \overline{OA} and \overline{OC} . Clearly, vector \overline{OM} and vector \overline{OC} are vertical from each other.

Along these lines, the normal line vector \overline{CG} can be decided via intersecting point of vectors \overline{CD} and \overline{OC} , which is additionally vertical to vector \overline{OC} . At this point, we can obtain the vector \overline{OH} by deciphering vector \overline{CG} along the vector \overline{CO} to point O. The angle $\angle MOH$ between vectors \overline{OH} and \overline{OM} is defined as Shoulder Roll (Reddivari et al., 2014).

The orientation angles sent by the Kinect can be separated by utilizing PMatrix3D with Processing software. The PMatrix3D outputs the required rotation matrix as well, where the current coordination framework is well given the backup into the stack. It is then delivered to the shoulder joint, additionally, the rotation matrix is utilized to transform into the coordination system data. Every single computation in this capacity will be decided within the obtained coordination framework.

After the computation of Shoulder Roll and Elbow Roll angles, the rotation matrix from the stack can be recovered to obtain the initial coordination framework. The right Shoulder Roll is additionally computed with the

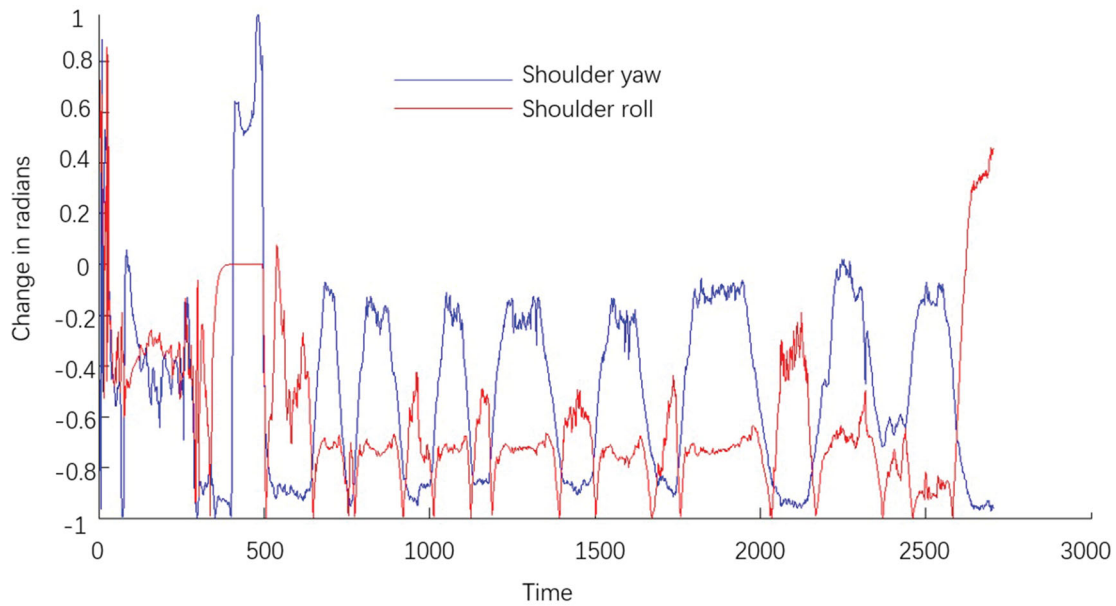


Figure 10. Error of vector approach.

similar method. Furthermore, a small change has been applied to the vectors coordination system.

As the function used to calculate roll angles is not accurate, the error needs to be corrected. Every value changes of Shoulder Roll is along with the value changes of Shoulder Yaw. After that the statistics are plotted in to the MATLAB, as seen in Figure 10. From several times' trials and the above method, the error is mostly revised by the equations demonstrated as follows:

$$\gamma_s = -\gamma_s - \beta_s/2 - 0.6, \quad (5)$$

where we define the angle of left Shoulder Roll is γ_s and the angle of left Shoulder Yaw is β_s . Presently these returned values of angles are sent to Baxter development platform for further advanced work utilizing UDP protocol. The data packets created by the server is sent through the function introduced above. So far, every single angular value is sent to teleoperate the Baxter robot with the Python script based on the KF sensor fusion.

4. Measurement of angular velocity by MYO armband

The joint angles are obtained by computing the integral of angle velocity. Any positions of human operator's arms can be used as the initial position, where the joint angles are assumed to be zero, according to Yang et al. (2016). When the operator moves his arm to a new pose P , the rotation angles(joint angles) are the pose P with respect point to the initial pose in Yang et al. (2016).

As shown in Figure 11, the frame (X_1, Y_1, Z_1) represents the orientation of MYO armband in the initial position.

The frame (X_2, Y_2, Z_2) represents the current orientation of the MYO. From the first MYO armband worn on the upper arm, we can obtain three angles' angular velocity $v1_x, v1_y, v1_z$, which represent Shoulder Roll, Pitch and Yaw, respectively.

From the second MYO armband worn on the forearm, we can get the angles' velocity $v2_x, v2_y$, which represent Elbow Roll and Pitch.

Existing in the joint angular velocity measured by the joint angle, there will be errors, however here the errors

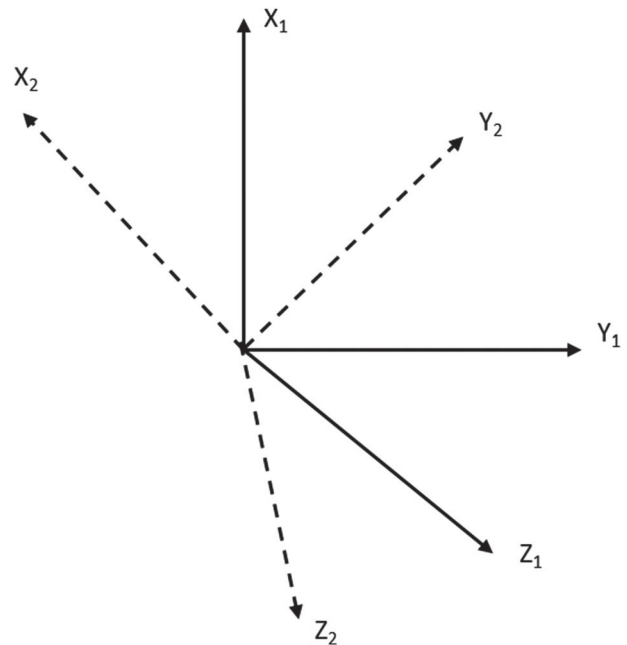


Figure 11. The orientation of the MYO in the initial pose and the current pose.

will be superimposed. The shoulder joint error will be superimposed on the elbow joint, resulting in a greater elbow error. In addition, the integration time problem also leads to the existence of errors. Although the sampling frequency of IMU in the MYO is 50 Hz, the resulting angle will have a large difference in value when the joint angle is calculated from the angular velocity integral in the program in Yang et al. (2016). Here, the method for the angular velocity was extended from the method of measurement for the angles using MYO armbands mentioned in previous research (Yang et al., 2016). In summary, in this paper, MYO armband is used to measure angular velocity of each joint, and Kinect is used to get the angles of each joint.

5. KF-based sensor fusion

The KF method has two basic assumptions: (1) a sufficiently accurate model of the information process is a linear (or time-varying) dynamic system excited by white noise; (2) the measurement signal contains additional white noise components for each time. When the above assumptions are satisfied, a KF method can be applied. In this paper, all the data collected from Kinect sensor and MYO armbands fulfilled the above requirement, hence the continuous time KF is used to fuse the data from different sensors.

The equation (mentioned in the previous section) is for the continuous time KF, wherein, x_0 , ω and v are not related to each other, according to Davari et al. (2016) and Särkkä and Solin (2012), which is demonstrated as follows:

$$\begin{aligned} E\{x(0)\} &= m_x(0); & E\{\omega(t)\} &= 0; & E\{v(t)\} &= 0 \\ E\{[x(0) - m_x(0)][x(0) - m_x(0)]^T\} &= P(0) \\ E\{x(0)\omega^T(t)\} &= 0; & E\{x(0)v^T(t)\} &= 0 \\ E\{\omega(t)v^T(t)\} &= 0; & E\{\omega(t)\omega^T(\tau)\} &= s(t)\delta(t - \tau) \\ E\{v(t)v^T(\tau)\} &= r(t)\delta(t - \tau), \end{aligned} \quad (6)$$

where s is the system noise variance intensity matrix of continuous system; r is the array of measured noise variance intensity; $M_x(0)$ and $P(0)$ are the initial mean value of x and the initial covariance matrix, respectively; $\delta(t - \tau)$ is the Dirac δ function.

We assume that every single joint of human arms is taken into account separately to research, which gives that all the KF factors are the first order, hence, here $A=0$, $B=1$, $G=1$ and $H=1$. Then the KF equations are simplified as below,

$$\begin{aligned} \dot{x}_i &= u_i + \omega \\ y_i &= x_i + v, \end{aligned} \quad (7)$$

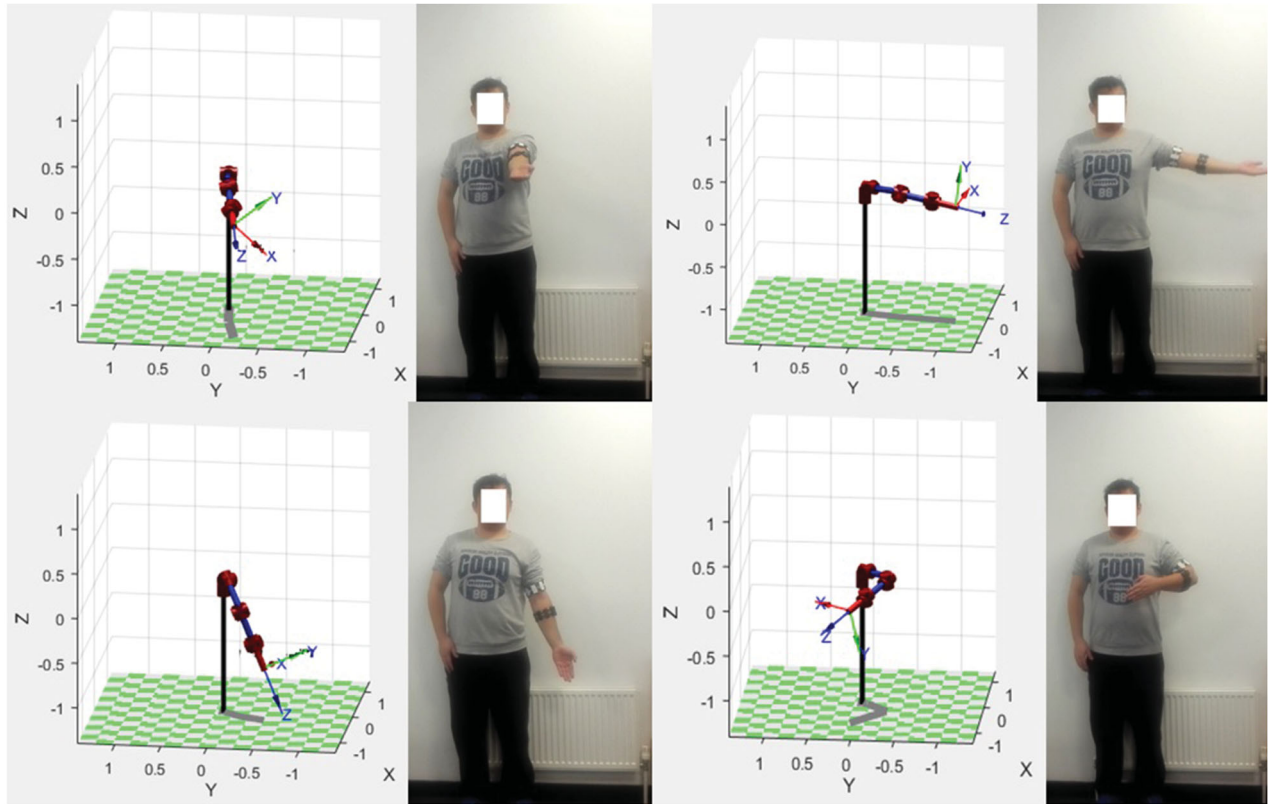


Figure 12. Demonstration of experiment at the different positions.

where in this special case, y_i is the angular position of the number of i joint collected from the Kinect sensor. And u_i is the angular velocity of the number of i joint of operator's arm motion.

$$\begin{aligned}\dot{\hat{x}}_i &= u_i + k(y_i - \hat{x}_i) \\ k &= pr^{-1} \\ \dot{p} &= p - pr^{-1}p + s,\end{aligned}\quad (8)$$

where k is the filter gain matrix, p is the estimated covariance matrix, \hat{x}_i is the required (satisfied) data obtained from KF-based sensor fusion, which is also the statistic that needs to be sent to development workstation via UDP.

6. Experimental analysis and results

The indoor experiment environment is of sufficient illumination. One operator stands in front of Kinect at the distance about 2 m. As we demonstrated in previous section, we choose the Shoulder Pitch, Shoulder Yaw, Shoulder Roll, Elbow Pitch and Elbow Roll to apply in this experiment. Following the experimental data collection, a simulated Baxter robot in MATLAB was used to teleoperate with the operators.

6.1. Experimental set-up

An operator worn a pair of MYO armbands faces to the Kinect sensor (as seen in Figure 12) with different arm movements. The operator wore one MYO armband near



Figure 13. Graphical result after the Kalman Filter-based sensor fusion (Shoulder Pitch, Shoulder Roll, Shoulder Yaw, Elbow Pitch, Elbow Roll).

the centre of the upper arm and wore the other near the centre of the forearm. The former measures the orientation and angular velocity of Shoulder Pitch, Yaw and Roll. The latter predicts the orientation and the angular velocity of Elbow Pitch and Roll. Before the experiments, it is significant to calibrate the MYO armband and warm up the EMG sensors, in order to recognize different hand postures better for the MYO armband. The operator should not move even any short distances, under this case, only both arms of operator can freely move with a small and stable speed.

6.2. Experimental results

Figure 13 illustrates the graphical results of the five selected DOFs with different trajectory after the KF-based sensor fusion between Baxter and the operator. The Kinect sensor gives us the position difference between robot's real trajectory and the base point and the MYO armbands gives us the angular velocity of those five angles accordingly. Then they were fused together via KF. Here, the experimental data of the operator's arm motion from Kinect and MYO, and the optimum output from Kalman filter-based sensor fusion and the angular statistics of the simulated Baxter robot were taken, respectively for the test. From the graph, it is concluded that the total performance of motion capture system becomes improved by applying the KF-based sensor fusion.

The method of using Kinect and MYO armband after KF-based sensor fusion to teleoperate a Baxter robot was developed and validated. The experimental results shown in Table 1 demonstrate a series of ratios, which are the different values of the five angles between those obtained by KF and those directly collected by Kinect. Because the values obtained via KF are optimum, and the noises during the teleoperation process have been removed. Hence, that ratio is approximately defined as the efficient improvement, which is denoted as r_e as defined in (8) below. The ratios shown in Table 1 averagely at 3.158%, 4.086%, 3.442%, 3.269% and 3.673% for the angular positions of Shoulder Pitch, Shoulder Roll, Shoulder Yaw, Elbow Pitch and Elbow Roll, respectively.

$$r_e = \frac{p_{KF} - p_{Kinect}}{p_{Kinect}}, \quad (9)$$

where p_{KF} and p_{Kinect} are the experimental data of different angular positions obtained from the KF-based

Table 1. Table for efficient improvement of different angular positions.

Data	Shoulder Pitch	Shoulder Roll	Shoulder Yaw	Elbow Pitch	Elbow Roll
Ratio	3.185%	4.086%	3.442%	3.269%	3.673%

sensor fusion and directly collected from the Kinect, respectively.

7. Conclusion

In this paper, a KF-based sensor fusion is applied to obtain an improved performance. In order to do this, a Kinect sensor is utilized to capture the motion of operator arm with vector approach. The vector approach can precisely calculate the angular data of human arm joints, by selecting five out of seven joints on each arm. Then, the angular velocity of human operator's arm can be measured by a pair of MYO armbands worn on the operator's arm. The continuous time KF method output the designed data with less error, after that, the data will be applied to the joints of the simulated Baxter robot, respectively for teleoperation.

The MYO armband is utilized in this research owing to its portability and its accurate computation of the values for angular velocity of shoulder and elbow motion. It worked together with Kinect, which aims to apply Kalman filter-based sensor fusion providing a way with users to enhance the accuracy of the teleoperation process. Several experimental works have illustrated the great accuracy and efficient improvement of the recommended designed techniques, averagely at 3.158%, 4.086%, 3.442%, 3.269% and 3.673% for the parameters of Shoulder Pitch, Shoulder Roll, Shoulder Yaw, Elbow Pitch and Elbow Roll, respectively. More research needs to be done with focus on the practical usage of different types of sensors for the future work, even different types of algorithms, for example the fuzzy logic (Luo, Wang, Liang, Wei, & Alsaadi, 2016) and (Lin, Chen, & Wang, 2016), to replace KF to carry out the sensor fusion in a promoted way.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Chunxu Li  <http://orcid.org/0000-0001-7851-0260>

References

- Bolíbar, J. (2012). *Kinect audio-runner: Audio feedback for improving performance in long-distance running* (Master of science thesis). KTH Royal Institute of Technology, Stockholm.
- Boulos, M. N. K., Blanchard, B. J., Walker, C., Montero, J., Tripathy, A., & Gutierrez-Osuna, R. (2011). Web GIS in practice x: A microsoft Kinect natural user interface for Google earth navigation. *International Journal of Health Geographics*, 10(1), 45.

- Breazeal, C. L. (2000). *Sociable machines: Expressive social exchange between humans and robots* (PhD thesis). Massachusetts Institute of Technology.
- Calo, R. (2011). Open robotics. *Maryland Law Review*, 70(3), 571–613.
- Cheok, A. D. (2010). Kawaii/cute interactive media. In *Art and technology of entertainment computing and communication* (pp. 223–254). Springer.
- Cousins, S. (2010). ROS on the pr2 [ros topics]. *IEEE Robotics & Automation Magazine*, 17(3), 23–25.
- Dautenhahn, K. (2007). Socially intelligent robots: Dimensions of human–robot interaction. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1480), 679–704.
- Davari, N., Gholami, A., & Shabani, M. (2016). Multirate adaptive Kalman filter for marine integrated navigation system. *The Journal of Navigation*, 1–20. doi:10.1017/S0373463316000801
- Garcia, E., Jimenez, M. A., De Santos, P. G., & Armada, M. (2007). The evolution of robotics research. *IEEE Robotics & Automation Magazine*, 14(1), 90–103.
- Henry, P., Krainin, M., Herbst, E., Ren, X., & Fox, D. (2012). RGB-D mapping: Using Kinect-style depth cameras for dense 3d modeling of indoor environments. *The International Journal of Robotics Research*, 31(5), 647–663.
- Ijspeert, A. J. (2008). Central pattern generators for locomotion control in animals and robots: A review. *Neural Networks*, 21(4), 642–653.
- Jean, J. S. (2012). *Kinect hacks: Tips & tools for motion and pattern detection*. O'Reilly Media.
- Jo, I., Park, Y., & Bae, J. (2013). A teleoperation system with an exoskeleton interface. Paper presented at 2013 IEEE/ASME international conference on advanced intelligent mechatronics (pp. 1649–1654), Wollongong, Australia.
- Ju, Z., Yang, C., Li, Z., Cheng, L., & Ma, H. (2014). *Teleoperation of humanoid Baxter robot using haptic feedback*. Paper presented at 2014 international conference on multisensor fusion and information integration for intelligent systems (MFI) (pp. 1–6). Beijing, China: IEEE.
- Li, C., Ma, H., Yang, C., & Fu, M. (2014). *Teleoperation of a virtual icub robot under framework of parallel system via hand gesture recognition*. Paper presented at 2014 IEEE international conference on fuzzy systems (FUZZ-IEEE) (pp. 1469–1474), Beijing, China.
- Lin, C., Chen, B., & Wang, Q.-G. (2016). Static output feedback stabilization for fractional-order systems in TS fuzzy models. *Neurocomputing*, 218, 354–358.
- Liu, Y., Zhang, Y., Fu, B., & Yang, R. (2013). *Predictive control for robot arm teleoperation*. Paper presented at IECON 2013–39th annual conference of the IEEE industrial electronics society, (pp. 3693–3698), Vienna, Austria.
- Luo, Y., Wang, Z., Liang, J., Wei, G., & Alsaadi, F. E. (2016). H control for 2-d fuzzy systems with interval time-varying delays and missing measurements. *IEEE Transactions on Cybernetics*, 47(2). doi:10.1109/TCYB.2016.2514846
- Ramos, E. (2012). Processing. In *Arduino and Kinect projects* (pp. 35–60). Springer.
- Reddivari, H., Yang, C., Ju, Z., Liang, P., Li, Z., & Xu, B. (2014). *Teleoperation control of Baxter robot using body motion tracking*. Paper presented at 2014 international conference on multisensor fusion and information integration for intelligent systems (MFI) (pp. 1–6), Beijing, China.
- Ronchetti, M., & Avancini, M. (2011). *Using Kinect to emulate an interactive whiteboard*. MS in Computer Science, University of Trento.
- Särkkä, S., & Solin, A. (2012). On continuous-discrete cubature Kalman filtering. *IFAC Proceedings Volumes*, 45(16), 1221–1226.
- Sucipto, A., Harsoyo, A., & Rusmin, P. H. (2012). *Implementation of gesture recognition on aquarium application*. Paper presented at 2012 IEEE international conference on system engineering and technology (ICSET) (pp. 1–4), Bandung, Indonesia.
- Tang, G., Wang, Z., & Williams, A. (1998). *On the construction of an optimal feedback control law for the shortest path problem for the dubins car-like robot*. Paper presented at Proceedings of the thirtieth southeastern symposium on system theory, 1998 (pp. 280–284). IEEE.
- Tawfik, M., Lowe, D., Salzmann, C., Gillet, D., Sancristobal, E., & Castro, M. (2014). *Critical factors in the architectural design of modern educational remote laboratories*. Paper presented IEEE frontiers in education conference (FIE) Proceedings (pp. 1–8), Madrid, Spain.
- Villaroman, N., Rowe, D., & Swan, B. (2011). *Teaching natural user interaction using OpenNI and the Microsoft Kinect sensor*. Paper presented at Proceedings of the 2011 conference on information technology education (pp. 227–232). *The Journal of the ACM*.
- Yang, C., Chang, S., Liang, P., Li, Z., & Su, C.-Y. (2015a). *Teleoperated robot writing using EMG signals*. Paper presented at 2015 IEEE international conference on information and automation (pp. 2264–2269), Lijiang, China.
- Yang, C., Chen, J., & Chen, F. (2016). *Neural learning enhanced teleoperation control of Baxter robot using IMU based motion capture*. Paper presented at 2016 22nd IEEE international conference on automation and computing (ICAC) (pp. 389–394), Colchester, UK.
- Yang, C., Chen, J., Li, Z., He, W., & Su, C.-Y. (2015b). *Development of a physiological signals enhanced teleoperation strategy*. Paper presented at 2015 IEEE international conference on information and automation (pp. 13–19), Lijiang, China.
- Yang, F., Wang, Z., & Hung, Y. (2002). Robust Kalman filtering for discrete time-varying uncertain systems with multiplicative noises. *IEEE Transactions on Automatic Control*, 47(7), 1179–1183.
- Zhou, X., Chang, N.-B., & Li, S. (2009). Applications of SAR interferometry in earth and environmental science research. *Sensors*, 9(3), 1876–1912.