Hindawi Publishing Corporation
Advances in Human-Computer Interaction
Volume 2016, Article ID 7921295, 10 pages
http://dx.doi.org/10.1155/2016/7921295

Research Article

Kinect-Based Sliding Mode Control for Lynxmotion Robotic Arm

Ismail Ben Abdallah,[1,2] Yassine Bouteraa,[1,2] and Chokri Rekik[1]

[1]*Control and Energy Management Laboratory (CEM-Lab), National School of Engineers of Sfax, Sfax, Tunisia*
[2]*Digital Research Center of Sfax, Technopark of Sfax, BP 275, Sakiet Ezzit, 3021 Sfax, Tunisia*

Correspondence should be addressed to Yassine Bouteraa; yassinebouteraa@gmail.com

Received 30 November 2015; Revised 24 May 2016; Accepted 7 June 2016

Academic Editor: Alessandra Agostini

Recently, the technological development of manipulator robot increases very quickly and provides a positive impact to human life. The implementation of the manipulator robot technology offers more efficiency and high performance for several human's tasks. In reality, efforts published in this context are focused on implementing control algorithms with already preprogrammed desired trajectories (passive robots case) or trajectory generation based on feedback sensors (active robots case). However, gesture based control robot can be considered as another channel of system control which is not widely discussed. This paper focuses on a Kinect-based real-time interactive control system implementation. Based on LabVIEW integrated development environment (IDE), a developed human-machine-interface (HMI) allows user to control in real time a Lynxmotion robotic arm. The Kinect software development kit (SDK) provides a tool to keep track of human body skeleton and abstract it into 3-dimensional coordinates. Therefore, the Kinect sensor is integrated into our control system to detect the different user joints coordinates. The Lynxmotion dynamic has been implemented in a real-time sliding mode control algorithm. The experimental results are carried out to test the effectiveness of the system, and the results verify the tracking ability, stability, and robustness.

1. Introduction

The use of manipulator robot controlled by a natural human-computer interaction offers new possibilities in the execution of complex tasks in dynamic workspaces. Human-robot interaction (HRI) is a key feature which differentiates the new generation of robots from conventional robots [1, 2].

Many manipulator robots have similar behavior with human arm. Thanks to the gesture recognition using Kinect sensor, the motion planning is easier when the robotic arm has the same degree of freedom of the human arm. Gesture recognition using Kinect sensor is based on the skeleton detection, although current development packages and toolkits provide base to identify the joints position of the detected skeleton.

Kinect is a motion detection and recognition smart sensor which allows human/computer interaction without the need of any physical controllers [3]. Indeed, Kinect sensor is a motion sensor that provides a natural user interface available for several applications in different fields including game-based learning systems [4, 5], stroke rehabilitation [6, 7], helping visually impaired people [8, 9], navigation systems [10–12], and other fields [13, 14]. Based on its internal processor, Kinect sensor can recognize human movement patterns to generate a corresponding skeleton coordinates that can be provided in a computer environment such as Matlab [15, 16] and LabVIEW [17, 18]. Therefore, the dynamic gesture recognition technology has gained increased attention.

Several approaches have been developed, in order to improve recognition or take advantage of the existing algorithms. Reference [19] proposes easy gesture recognition approach in order to reduce the effort involved in implementing gesture recognizers with Kinect; the practical results with these developed packages are acceptable. Furthermore, an eigenspace-based method was developed in [20] for identifying and recognizing human gestures by using Kinect

3D data. A user adaptation scheme for this method was added to enhance Eigen3Dgesture and the performance of the constructed eigenspace of Kinect 3D data. Based on Microsoft's "Kinect for Windows SDK," [21] uses the API for measuring movement of people with Parkinson's disease.

Gesture control is mainly used for telemanipulation in several modes. In [22], a client/server structured robot teleoperation application system is developed in networked robot mode. Reference [23] describes gesture based telemanipulation of an industrial robotic arm in master-slave mode for unstructured and hazardous environments, a maximum velocity drift control approach is applied allowing the amateur user to control the robot by simple gestures, and force control was combined with the gesture based control to perform safe manipulation. However, this solution is costly and depends on reliability of the force-torque sensor. In addition, a method of human-robot interaction using markerless Kinect-based tracking of the human hand for teleoperation of a dual robot manipulator is presented in [24]. Based on hand motion, the user can control the robot manipulators to perform tasks of picking up and placing; the work was applied in virtual environment.

For robot teleoperation, many researches have been done providing the possibility of controlling the robotic systems dedicated for complex tasks [25]. Several techniques are used to control robot motion; a human-robot interface based on hand-gesture recognition is developed in [26–30]. Others have based on hand-arm tracking; [31] presents a method of real-time robot manipulator teleoperation using markerless image-based hand-arm tracking.

Many problems occurred during the human-robot interaction when using the Kinect sensor. After the acquisition of 3D data Kinect, these data will be processed to control the robot. Several approaches have been used to check the control. A Cartesian impedance control is used to control the dual robot arm [32]; this approach allows the robot to follow the human arm motion without solving inverse kinematic problems and avoiding self-collision problems. Likewise, a proportional derivative control is implemented with inverse kinematic algorithm in [33]. In [34], a PID controller is also used to control a 2-degree of freedom Lego Mind storm NXT robotic arm.

Based on online HIL (Hardware-in-the-Loop) experimental data, the acquired input data from the Kinect sensor are processed in a closed loop PID controller with feedback from motors encoders. Although these results are satisfactory side position control, they remain imperfect from velocity control stand point due to the use of simple controller used. Moreover, compared to the conventional computed torque control used in [35], the Lynxmotion dynamic has been implemented in a real-time sliding mode control algorithm.

To combine the real and virtual objects and the implemented algorithms, we need to design a user interface that manages the overall system. Indeed, the human-machine interface (HMI) is highly recommended in robotics fields and many human-machine interfaces have been developed to control the robotic systems [36, 37]. In [38], a human-machine interface is designed to control the SCARA robot using Kinect sensor. This interface is implemented in C# using Kinect library OpenNI.

In this paper, we have used the package provided by Virtual Instrument Package Manager for LabVIEW. The recognition algorithm calculates the angle of each human arm joint dedicated for control. This algorithm is detailed in Section 1.

Compared to Shirwalkar et al. [23], firstly, the proposed control strategy takes into account the dynamics system. Also our algorithm detects all user joints and not only the hand, while for the security side we propose an algorithm security which avoids any type of parasite that may occur in the Kinect sensing field. Moreover, the proposed algorithm avoids any singularity position.

Likewise, a proportional derivative control is implemented with inverse kinematic algorithm [33]. However, on the one hand, this classic controller ignores the robot dynamics. On the other hand, this controller is limited to position control and does not consider the motion control.

Regarding the work in [38], we develop a generic interface that encompasses the different components of the global system: virtual 3D manipulator, Kinect sensor feedback values, skeleton image, implemented dynamic model, implemented sliding mode control, implemented recognition algorithm for calculating the joints position, and implemented security algorithm. This LabVIEW-based user interface is described in Section 5.

The present paper is organized as follows: in Section 2, we presents the recognition method. The dynamic model of the Lynxmotion robotic arm is detailed in Section 3. In Section 4, the control design based on the sliding mode control approach is described. Human-machine interface is presented as well as experiment results in Section 5. Finally, we conclude with a work summary.

2. Recognition Method

Several methods are used to detect the human arm movements. Kinect sensor is the most popular thanks to skeleton detect and depth computing. The Kinect is a motion sensor that can measure three-dimensional motion of a person. In fact, Kinect sensor generates a depth and image (x, y, and z) of human body. In addition, the Kinect sensor can provide space coordinates for each joint.

To enjoy these benefits, we need an Application Programmer's Interface (API) to the Kinect hardware and its skeletal tracking software. Microsoft's "Kinect for Windows SDK" was used to provide an API to the Kinect hardware. This API provides an estimate for the position of 20 anatomical landmarks in 3D at a frequency of 30 Hz and spatial and depth resolution of 640×480 pixels as shown in Figure 1. The default smoothing parameters are as follows:

(i) Correction factor = 0.5.

(ii) Smoothing factor = 0.5.

(iii) Jitter radius = 0.05 m.

(iv) Maximum deviation radius = 0.04 m.

(v) Future prediction = 0 frames.

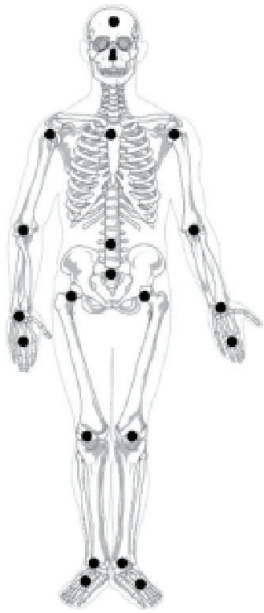

Figure 1: Kinect for Windows SDK detected joints.

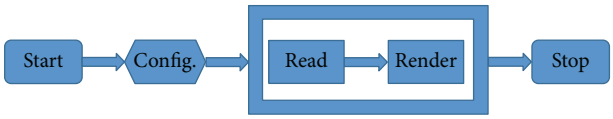

Figure 2: Operational Layout for the Kinect LabVIEW toolkit.

Based on Microsoft Kinect SDK, the Kinesthesia Toolkit for Microsoft Kinect is developed at University of Leeds initially for the medical rehabilitation and surgical tools. In addition, the toolkit helps the NI LabVIEW programmers to access and use the popular functions of the Microsoft Kinect camera such as RGB video, depth camera, and skeletal tracking. The toolkit comes fully packaged using JKI's VI Package Manager.

In addition, the toolkit allows the user to initialize and close the Kinect's different components through polymorphic VIs (RGB camera, depth camera, and skeletal tracking) dependent on the functionalities they require. The Operational Layout for the Kinect LabVIEW toolkit is described in Figure 2.

The first step is to initialize the Kinect; this act opens the device and returns a reference. The second step is to configure the Kinect. In this step, we can choose the RGB stream format, enable the skeleton smoothing, and select the appropriate event (video feedback from the RGB camera, 3D skeleton information, or depth information). In the next step, we can get skeleton, depth, and video information from the Kinect in the main loop such as while loop. The additional implementations can be put in the same loop. Finally, we stop the Kinect and close the communication.

In this paper, a user interface containing a skeleton display screen has been developed as shown in Figure 3. The screen

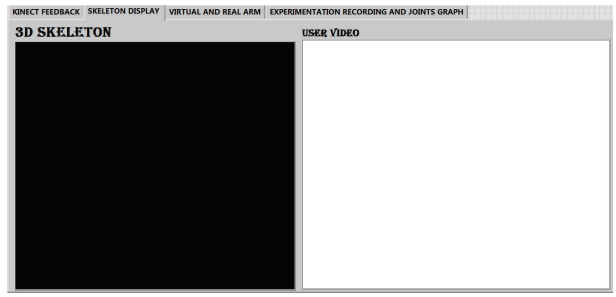

Figure 3: Skeleton and user display screen.

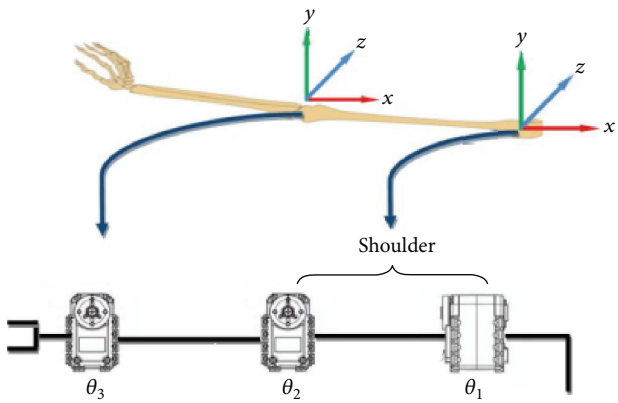

Figure 4: Human arm and manipulator robot model.

has been reserved to the skeleton display and the user video recording.

In this work, we focus on the human arm portion of the skeleton data. In fact, the human arm can be considered as being a 4DOF manipulator arm similar to Lynxmotion robotic arm. In our case, a robot manipulator model has been developed with a 3DOF. Therefore, we assume that the user has the opportunity to control the robot based on his right arm gestures. The three considered movements are as follows:

(i) Flexion-extension for the right shoulder joint.

(ii) Abduction-adduction for the right shoulder joint.

(iii) Flexion-extension for the right elbow joint.

The similarity of the human arm and the robot manipulator model is shown in Figure 4.

After the acquisition of each elementary joint position, an additional calculation is performed to determine the desired speed and acceleration. All these parameters will be considered as the desired trajectories for the manipulator robot control algorithm.

3. Dynamic Model of Lynxmotion Robotic Arm

The dynamic model deals with the torque and force causing the motion of the mechanical structure. In this section, robotic arm dynamic model has been computed based on

Euler-Lagrange formulation. The potential and kinetic energies of each link have been computed by the two following equations:

$$\begin{aligned} u_i &= -m_i g^T {}_i^i P, \\ K_i &= \frac{1}{2} m_i v_{c_i}^T v_{c_i} + \frac{1}{2} {}_i^i w^T {}_i^i I {}_i^i w, \end{aligned} \tag{1}$$

where m_i is the ith link mass, v_{c_i} is the ith link linear velocity, ${}_i^i w$ is the ith link angular velocity, ${}_i^i I$ is the ith inertia tensor, and ${}_i^i P$ is the ith link position.

Further calculation, the dynamic model of a full actuated robot manipulator is expressed as follows:

$$M(q)\ddot{q} + N(q, \dot{q}) = \tau, \tag{2}$$

where τ is $n \times 1$ vector consisting on applied generalized torque, $M(q)$ is $n \times n$ symmetric and positive definite inertia matrix, $N(q, \dot{q})$ is the vector of nonlinearity term, and $q \in R^{n \times n}$.

The vector of nonlinearity term hails from centrifugal, Coriolis, and gravity terms described in the following equations:

$$\begin{aligned} N(q, \dot{q}) &= V(q, \dot{q}) + G(q), \\ V(q, \dot{q}) &= B(q)[\dot{q}, \dot{q}] + C(q)[\dot{q}]^2. \end{aligned} \tag{3}$$

Here, robot manipulator dynamic equation can be written as

$$\tau = M(q)\ddot{q} + B(q)[\dot{q}, \dot{q}] + C(q)[\dot{q}]^2 + G(q), \tag{4}$$

where $B(q)$ is matrix of Coriolis torque, $C(q)$ is matrix of centrifugal torque, $[\dot{q}, \dot{q}]$ is vector of joint velocity obtainable by $[\dot{q}_1 \cdot \dot{q}_2, \dot{q}_1 \cdot \dot{q}_3, \ldots, \dot{q}_1 \cdot \dot{q}_n, \dot{q}_2 \cdot \dot{q}_3, \ldots]^T$, and $[\dot{q}]^2$ is vector witch can be obtained by $[\dot{q}_1{}^2, \dot{q}_2{}^2, \dot{q}_3{}^2, \ldots]^T$.

In our work, a 3 DOF manipulator model has been developed using the Lagrange formulation. The goal was to determine the Coriolis, centrifugal, and gravitational matrices for real implementation reasons.

4. Control Design

Robot manipulator has highly nonlinear dynamic parameters. For this reason, the design of a robust controller is required. In this section, we study the motion controller "sliding-mode control."

The dynamic model of the robot manipulator is described as follows:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau, \tag{5}$$

where $q = [q_1, q_2, q_3]^T$ and $\tau = [\tau_1, \tau_2, \tau_3]^T$.

Let $q_d(t)$ denote the desired trajectory. The tracking error is defined as

$$e = q_d(t) - q(t). \tag{6}$$

Define $\dot{q}_r = \dot{q}_d + s(q_d - q)$, where s is a definite positive matrix.

According to the linear characteristic of robotic, we obtain

$$\widehat{M}(q)\ddot{q}_r + \widehat{C}(q, \dot{q})\dot{q}_r + \widehat{g}(q) = \gamma(q, \dot{q}, \dot{q}_r, \ddot{q}_r)\widehat{P}, \tag{7}$$

where $\widehat{P}$ is the robot estimated parameters vector and $\gamma(q, \dot{q}, q_r, \dot{q}_r)$ is a repressor matrix.

Define

$$\begin{aligned} \widetilde{H}(q) &= H(q) - \widehat{H}(q), \\ \widetilde{C}(q) &= C(q, \dot{q}) - \widehat{C}(q, \dot{q}), \\ \widetilde{g}(q) &= g(q) - \widehat{g}(q). \end{aligned} \tag{8}$$

The sliding vector is selected as

$$s = \dot{e} + \mu * e. \tag{9}$$

We propose the sliding mode control law as follows:

$$\tau = \widehat{M}(q)\ddot{q}_r + \widehat{C}(q, \dot{q})\dot{q}_r + \widehat{g}(q) + s + k * \operatorname{sgn}(s), \tag{10}$$

where $k_i = \sum_j \overline{\gamma}_{ij} \overline{\overline{P}}_j$.

Define

$$\begin{aligned} \overline{\overline{P}}_i \text{ Such } \quad \forall i \ \left|\widetilde{P}_i\right| &\le \overline{\overline{q}}_i, \\ \overline{\gamma}_{ij} \text{ Such } \quad \forall i \ \left|\gamma_{ij}\right| &\le \overline{\gamma}_{ij}. \end{aligned} \tag{11}$$

Theorem 1. *The proposed controller guarantees the asymptotic stability of the system.*

Proof. Select the LEC as

$$\begin{aligned} V(t) &= \frac{1}{2} s^T M(q) s \Longrightarrow \\ \dot{V}(t) &= s^T M(q)\dot{s} + \frac{1}{2} s^T \dot{M}(q) s \\ &= s^T M(q)\dot{s} + s^T C(q) s \\ &= s^T \left[M(q)\ddot{q}_r + C(q, \dot{q})\dot{q}_r + g(q) - \tau\right]. \end{aligned} \tag{12}$$

Replacing τ by its expression (10) yields

$$\begin{aligned} &\dot{V}(t) \\ &= s^T \left[\widetilde{M}(q)\ddot{q}_r + \widetilde{C}(q, \dot{q})\dot{q}_r + \widetilde{g}(q) - k * \operatorname{sgn}(s) - s\right] \\ &= s^T \left[\gamma(q, \dot{q}, \dot{q}_r, \ddot{q}_r)\widetilde{P} - k \cdot \operatorname{sgn}(s) - s\right], \end{aligned} \tag{13}$$

where $\gamma(q, \dot{q}, \dot{q}_r, \ddot{q}_r) = [\gamma_{ij}]$ such as $|\gamma_{ij}| \le \overline{\gamma}_{ij}$.

And $\widetilde{P} = [\widetilde{P}_i]$ such as $|\widetilde{P}_i| \le \overline{\overline{P}}_{ij}$.

Yield

$$\begin{aligned} \dot{V}(t) &= \sum_i \sum_j s_i \gamma_{ij} \widetilde{P}_j - \sum s_i K_i \operatorname{sgn}(s_i) - \sum_i s_i^2 \\ &= \sum_i \sum_j s_i \gamma_{ij} \widetilde{P}_j - \sum_i \sum_j \left|s_i\right| \overline{\gamma}_i \overline{\overline{P}}_j - \sum_i s_i^2 \le -\sum_i s_i^2 \\ &\le 0. \end{aligned} \tag{14}$$

This proves the stability. □

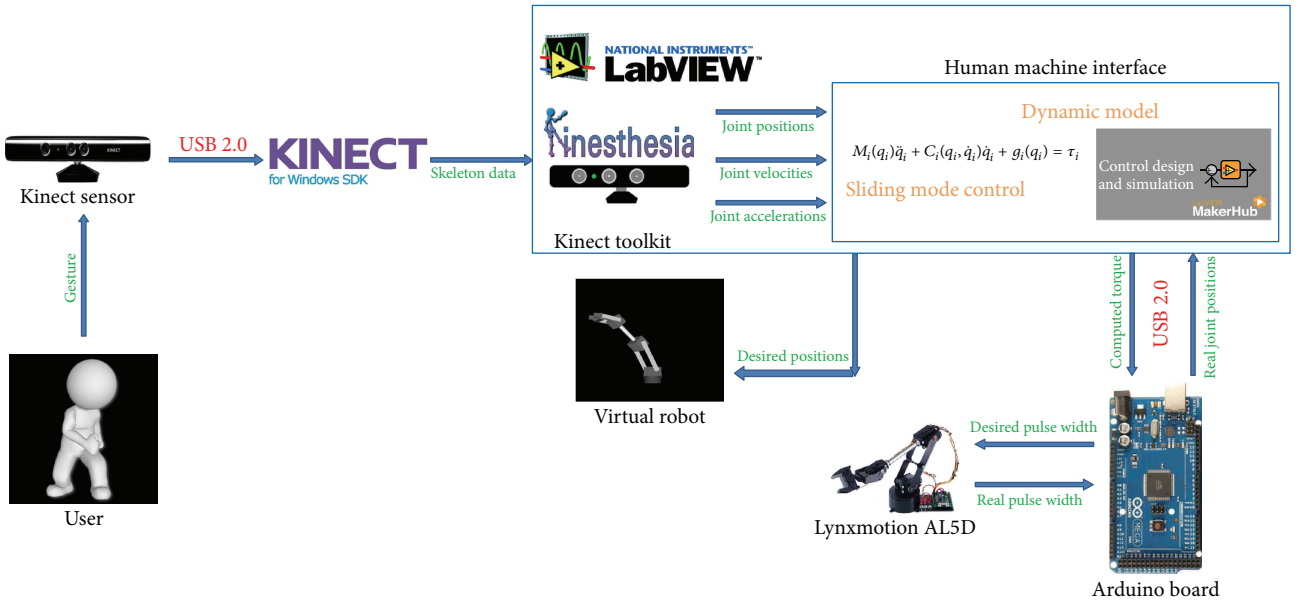

Figure 5: System overview.

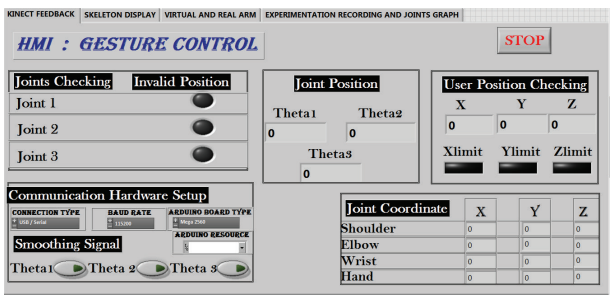

Figure 6: First screen of the user interface.

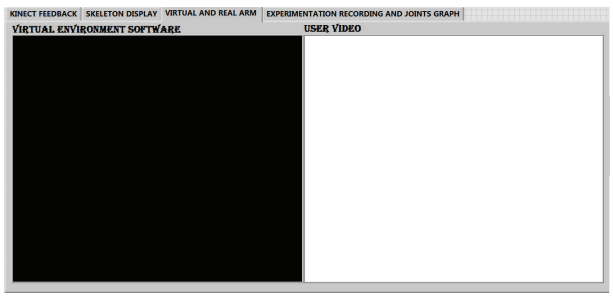

Figure 7: Third screen of the user interface.

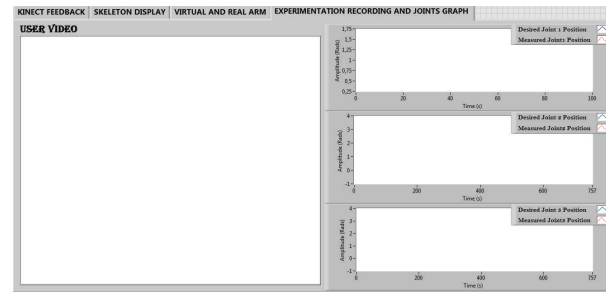

Figure 8: Fourth screen of the user interface.

5. Experiment Results

In the beginning, we had some packages to install. Indeed, the JKI's VI Package Manager such as Kinesthesia Toolkit for Microsoft Kinect, LabVIEW Interface for Arduino (LIFA), and control design and simulation module are strictly required for the implementation and then for the proper system functioning.

In this section, we present a LabVIEW-based human-machine interface. The virtual robot has been designed in 3D LabVIEW virtual environment. The virtual robot has the same kinematics parameters of the real robot such as joints axes and links distances. A user interface screen has been reserved to display the virtual robot movement versus the real robot.

The control law based on the sliding mode control has been successfully implemented with the parameters discussed in the previous section.

Going from gesture applied by the human arm, Kinect sensor captures the human arm motion and abstracts it in 3D coordinates. Using Kinect for Windows SDK, the skeleton data are provided and transmitted to the recognition algorithm. Based on Kinesthesia Toolkit for Microsoft Kinect, the human arm joint positions, velocities, and accelerations are computed and considered as desired trajectories of the robotic arm. By helping of LabVIEW control design and simulation, the developed dynamic model of 3DOF Lynxmotion robotic arm is implemented. Also, a sliding mode control algorithm is implemented ensuring the high performance in the trajectories tracking. Furthermore, the computed torque

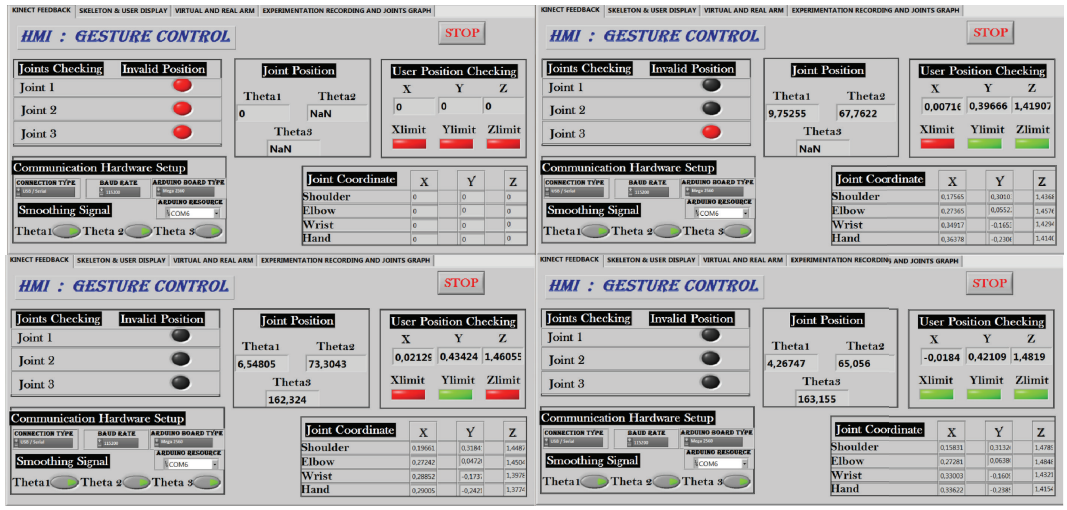

Figure 9: User position checking process.

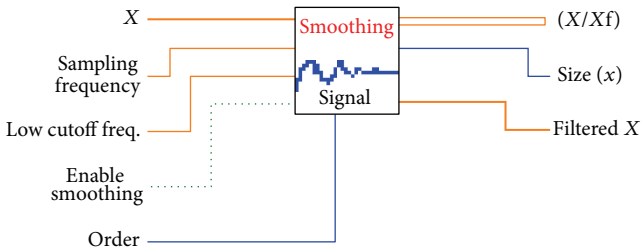

Figure 10: Butterworth-based designed filter.

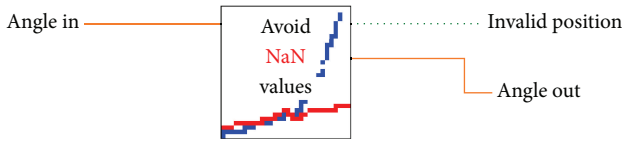

Figure 11: Avoid invalid position sub-VI.

is transmitted to the servomotors moving the mechanical structure of the robotic arm. Moreover, the desired joint positions are applied in the virtual environment to move the virtual robot in order to prove the compatibility of the desired motion in both real and virtual environment. Hence, an augmented reality approach is proposed to validate the control system.

The system overview can be presented by Figure 5.

In our experiment, we only use the first three joints and the remaining joints are not considered. Note that the user interface contains four screens.

The first screen is designed to the setting and hardware configuration. The screen displays the tracking process state such us invalid positions, joint positions, and 3D coordinates and user position checking. A security algorithm has been developed to secure the tracking process. Note that where the security program indicates a skeleton detection error, for one reason or another, the last calculated angle (already validated) of each joint will be sent to the robot until the operator solves the problem. Based on the feedback node which returns the last valid result with a nonnumeric input, the security algorithm is developed and abstracted in sub-VI for importing in the user interface. Figure 11 illustrates the imported sub-VI. In addition, the user can also set the Arduino board communication such us connection type and baud rate and Arduino board type and select the Arduino resource.

In the tracking process, the operator's arm can twitch, leading to unwanted random movement. In order to avoid this undesired movement, we propose a Butterworth low pass filter-based smoothing signal algorithm. In the same screen, the user can enable the smoothing signal algorithm by pressing on the bouton control. The designed filter is abstracted in a sub-VI as shown in Figure 10. The filtered trajectories, presented as the desired trajectories in Figures 15, 16, and 17, demonstrate the effectiveness of the filtering algorithm.

Finally, the stop button is set in the developed interface to stop the tracking process and close all connected hardware. This screen is developed as shown in Figure 6.

The second screen, presented in Figure 3, is designed to record the user's video and skeleton display. Figure 7 shows the third screen to prove the augmented reality. In this page, we show the virtual and real environment. The user and the virtual robot will be displayed in the user video portion whereas the virtual robot will be presented in the virtual environment software.

The remaining screen describes the joints graphs and operator's video recording during the experimentation as shown in Figure 8.

After running, the operator must select the first screen, called Kinect Feedback, to adjust its position. A user position checking is triggered to secure the operator's position in the Kinect sensing field. A green color of the signaling LED indicates the good state of the recognition process. Otherwise, the LED color changes to red to indicate the

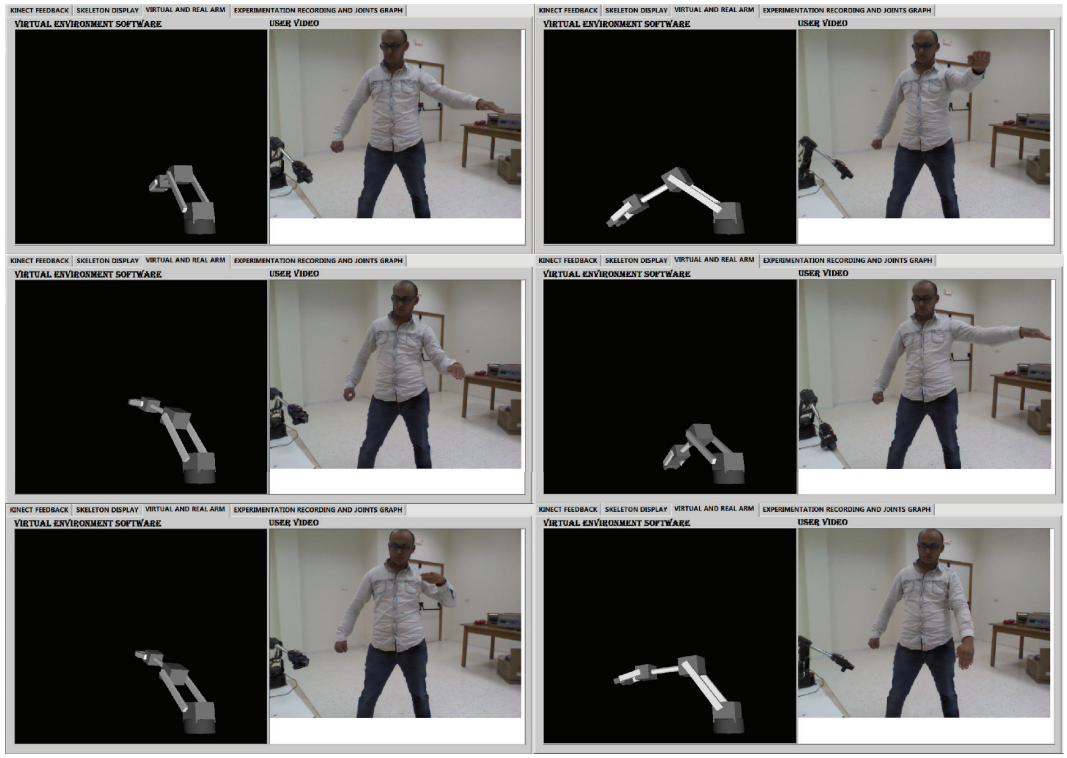

Figure 12: Augmented reality process.

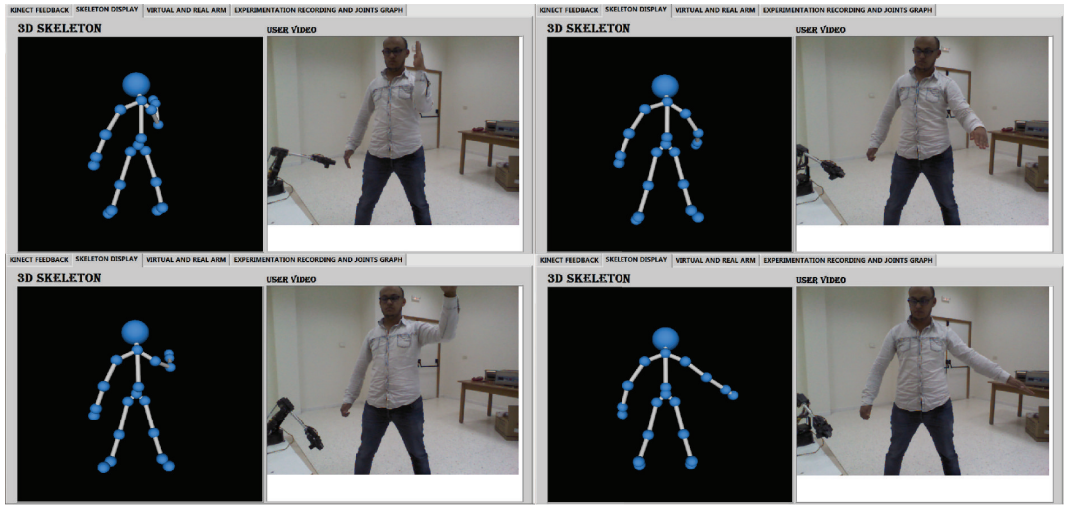

Figure 13: User and skeleton display.

problem of tracking. Figure 9 describes the executed screen in the user position checking process.

Based on Kinect toolkit for LabVIEW, the operator's skeleton data will be read by the LabVIEW interface. Then a gesture recognition method is executed to determine each joint angle. The calculated joint positions are considered as desired trajectories for the sliding mode control implemented in the developed software. Ultimately, the computed torque will be sent to the Atmel-based Arduino card via USB protocol. Therefore, the servomotors of the three first joints will be actuated to move the mechanical structure of the Lynxmotion robot.

A three-degree of freedom virtual robot is used to perform this experiment. Essentially, the augmented reality evaluated both the ability of the robot manipulator to copy human hand, arm motions, and the ability of the user to use the human-robot manipulator interface. The proposed controller is validated by the augmented reality. In fact, by building a virtual robot by using LabVIEW other than the real robot, as shown in Figure 12, virtual robot movements

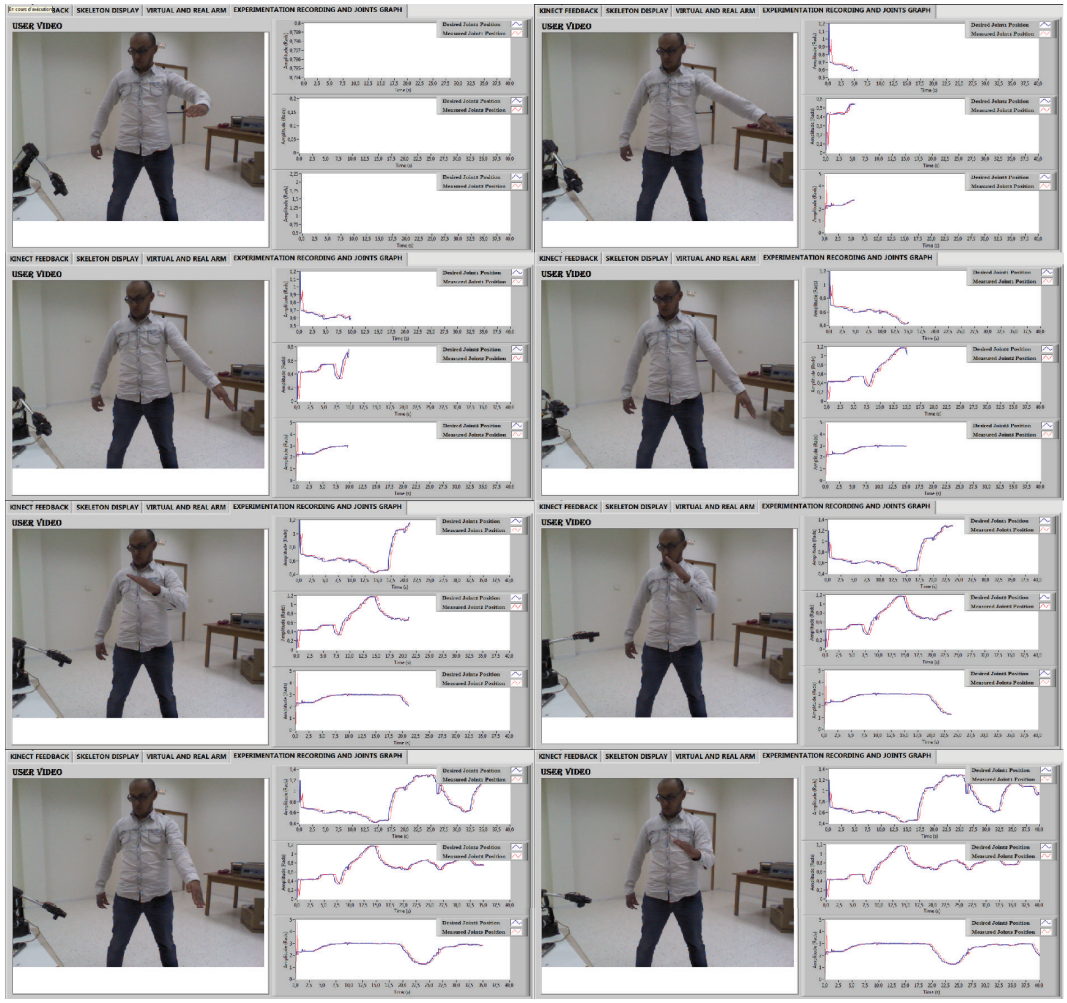

Figure 14: Gesture control process and real-time graphs.

are similar to the real robot. Therefore, the proposed control method is validated.

Figure 13 proves how the user can control the manipulator robot based in his detected skeleton.

Photographs series depicting gesture control process execution, each five seconds, with real-time graphs have been presented in Figure 14.

In the teleoperation process, the tracking error convergence is very important and the performance of our system is based on the tracking error optimization. Indeed, in the case that the elementary joint cannot track his similar human arm joint, the robotic arm will not be able to perform the desired task because of the accumulation of each joint position error. In this case, the end-effector localization is different regarding the desired placement. Therefore, the user must adapt the gesture with the current end-effector position to reduce the end-effector placement error. Hence, the adaptive gestures estimated by the user require a particular intelligence and such expertise. In another case, when the user applies the desired gesture with a variable velocity, in this situation, the control algorithm must respect the applied velocities and accelerations to conserve the convergence of the tracking error. In order to overcome this constraint, the use of high performance sliding mode controller is highly required.

Figures 15, 16, and 17 show trajectories of each joint and the desired trajectory for each. Both trajectories are as a function of time. The illustrated results show the effectiveness of the proposed control approach. The illustrations have only one second as response time. In experimental condition, this time can be regarded as excellent. This control strategy should be improved if we hope to exploit this system in telemedicine applications. Moreover, the developed teleoperation approach can be used for industrial applications, in particular, for pic and place tasks in hazardous or unstructured environments. The real-time teleoperation system offers a benefit platform to manage the unexpected scenarios.

6. Conclusion

This paper applies the gesture control to the robot manipulators. Indeed, in this work we use the Kinect technology to establish a human-machine interaction (HMI). The developed control approach can be used in all applications,

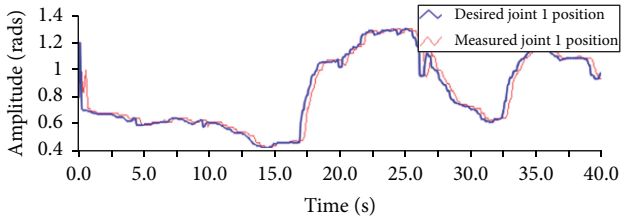

Figure 15: Desired and measured joint 1 position.

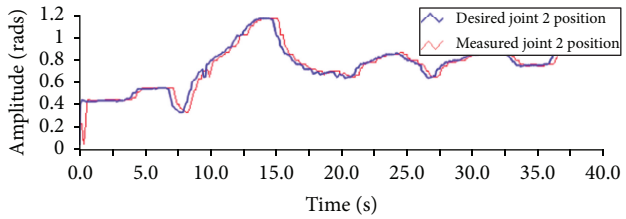

Figure 16: Desired and measured joint 2 position.

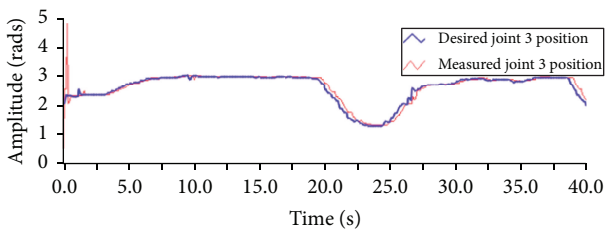

Figure 17: Desired and measured joint 3 position.

which require real-time human-robot cooperation. Trajectory tracking and system stability have been ensured by sliding mode control (SMC). Control and supervision both have been provided by a high-friendly human-machine interface developed in LabVIEW. Experiments have shown the high efficiency of the proposed approach. Furthermore, the Kinect-based teleoperation system is validated by augmented reality. In conclusion, the proposed control scheme provides a feasible teleoperation system for many applications such as manipulating in hazardous and unstructured environment. However, there are some constraints when applying the proposed method, such as the sensibility of the desired trajectory generated by the human arm even in case of random and unwanted movements. Indeed, the work space and the regional constraints must be respected in the control schemes by filtering the unaccepted trajectories. Moreover, for pic and place applications, grasping task is not considered in the proposed telemanipulation system. In perspective, this drawback can be overcome by using the electromyography biosignal for grasping task control.

Competing Interests

The authors declare that they have no competing interests.

References

[1] Y. Bouteraa, J. Ghommam, N. Derbel, and G. Poisson, "Nonlinear control and synchronization with time delays of multiagent robotic systems," *Journal of Control Science and Engineering*, vol. 2011, Article ID 632374, 9 pages, 2011.

[2] Y. Bouteraa, J. Ghommam, G. Poisson, and N. Derbel, "Distributed synchronization control to trajectory tracking of multiple robot manipulators," *Journal of Robotics*, vol. 2011, Article ID 652785, 10 pages, 2011.

[3] G. A. M. Vasiljevic, L. C. de Miranda, and E. E. C. de Miranda, "A case study of mastermind chess: comparing mouse/keyboard interaction with kinect-based gestural interface," *Advances in Human-Computer Interaction*, vol. 2016, Article ID 4602471, 10 pages, 2016.

[4] C.-H. Tsai, Y.-H. Kuo, K.-C. Chu, and J.-C. Yen, "Development and evaluation of game-based learning system using the microsoft kinect sensor," *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 498560, 10 pages, 2015.

[5] C.-H. Chuang, Y.-N. Chen, L.-W. Tsai, C.-C. Lee, and H.-C. Tsai, "Improving learning performance with happiness by interactive scenarios," *The Scientific World Journal*, vol. 2014, Article ID 807347, 12 pages, 2014.

[6] K. J. Bower, J. Louie, Y. Landesrocha, P. Seedy, A. Gorelik, and J. Bernhardt, "Clinical feasibility of interactive motion-controlled games for stroke rehabilitation," *Journal of NeuroEngineering and Rehabilitation*, vol. 12, no. 1, article 63, 2015.

[7] M. Štrbac, S. Kočović, M. Marković, and D. B. Popović, "Microsoft kinect-based artificial perception system for control of functional electrical stimulation assisted grasping," *BioMed Research International*, vol. 2014, Article ID 740469, 12 pages, 2014.

[8] N. Kanwal, E. Bostanci, K. Currie, and A. F. Clark, "A navigation system for the visually impaired: a fusion of vision and depth sensor," *Applied Bionics and Biomechanics*, vol. 2015, Article ID 479857, 16 pages, 2015.

[9] H.-H. Pham, T.-L. Le, and N. Vuillerme, "Real-time obstacle detection system in indoor environment for the visually impaired using microsoft kinect sensor," *Journal of Sensors*, vol. 2016, Article ID 3754918, 13 pages, 2016.

[10] D. Tuvshinjargal, B. Dorj, and D. J. Lee, "Hybrid motion planning method for autonomous robots using kinect based sensor fusion and virtual plane approach in dynamic environments," *Journal of Sensors*, vol. 2015, Article ID 471052, 13 pages, 2015.

[11] T. Xu, S. Jia, Z. Dong, and X. Li, "Obstacles regions 3D-perception method for mobile robots based on visual saliency," *Journal of Robotics*, vol. 2015, Article ID 720174, 10 pages, 2015.

[12] W. Shang, X. Cao, H. Ma, H. Zang, and P. Wei, "Kinect-based vision system of mine rescue robot for low illuminous environment," *Journal of Sensors*, vol. 2016, Article ID 8252015, 9 pages, 2016.

[13] H. Kim and I. Kim, "Dynamic arm gesture recognition using spherical angle features and hidden markov models," *Advances in Human-Computer Interaction*, vol. 2015, Article ID 785349, 7 pages, 2015.

[14] A. Chávez-Aragón, R. Macknojia, P. Payeur, and R. Laganière, "Rapid 3D modeling and parts recognition on automotive vehicles using a network of RGB-D sensors for robot guidance," *Journal of Sensors*, vol. 2013, Article ID 832963, 16 pages, 2013.

[15] A. Procházka, O. Vyšata, M. Valis, and R. Yadollahi, "The MS kinect use for 3d modelling and gait analysis in the Matlab

environment," in *Proceedings of the 21th Annual Conference: Technical Computing 2013*, Prague, Czech Republic, 2013.

[16] B. Y. L. Li, A. S. Mian, W. Liu, and A. Krishna, "Using Kinect for face recognition under varying poses, expressions, illumination and disguise," in *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV '13)*, pp. 186–192, Tampa, Fla, USA, January 2013.

[17] C. Muhiddin, D. B. Phillips, M. J. Miles, L. Picco, and D. M. Carberry, "Kinect 4 ... holographic optical tweezers," *Journal of Optics*, vol. 15, no. 7, Article ID 075302, 2013.

[18] P. Qiu, C. Ni, J. Zhang, and H. Cao, "Research of virtual simulation of traditional Chinese bone setting techniques based on kinect skeletal information," *Journal of Shandong University of Traditional Chinese Medicine*, no. 3, pp. 202–204, 2014.

[19] R. Ibañez, Á. Soria, A. Teyseyre, and M. Campo, "Easy gesture recognition for Kinect," *Advances in Engineering Software*, vol. 76, pp. 171–180, 2014.

[20] I.-J. Ding and C.-W. Chang, "An eigenspace-based method with a user adaptation scheme for human gesture recognition by using Kinect 3D data," *Applied Mathematical Modelling*, vol. 39, no. 19, pp. 5769–5777, 2015.

[21] B. Galna, G. Barry, D. Jackson, D. Mhiripiri, P. Olivier, and L. Rochester, "Accuracy of the Microsoft Kinect sensor for measuring movement in people with Parkinson's disease," *Gait & Posture*, vol. 39, no. 4, pp. 1062–1068, 2014.

[22] K. Qian, H. Yang, and J. Niu, "Developing a gesture based remote human-robot interaction system using kinect," *International Journal of Smart Home*, vol. 7, no. 4, pp. 203–208, 2013.

[23] S. Shirwalkar, A. Singh, K. Sharma, and N. Singh, "Telemanipulation of an industrial robotic arm using gesture recognition with Kinect," in *Proceedings of the IEEE International Conference on Control, Automation, Robotics and Embedded Systems (CARE '13)*, Jabalpur, India, December 2013.

[24] G. Du and P. Zhang, "Markerless human–robot interface for dual robot manipulators using Kinect sensor," *Robotics and Computer-Integrated Manufacturing*, vol. 30, no. 2, pp. 150–159, 2014.

[25] A. Hernansanz, A. Casals, and J. Amat, "A multi-robot cooperation strategy for dexterous task oriented teleoperation," *Robotics and Autonomous Systems*, vol. 68, pp. 156–172, 2015.

[26] F. Terrence, F. Conti, G. Sébastien, and B. Charles, "novel interfaces for remote driving: gesture, haptic and PDA," in *SPIE Telemanipulator and Telepresence Technologies VII*, vol. 4195, pp. 300–311, 2000.

[27] E. Ueda, Y. Matsumoto, M. Imai, and T. Ogasawara, "A hand-pose estimation for vision-based human interfaces," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 4, pp. 676–684, 2003.

[28] J. Zhang and A. Knoll, "A two-arm situated artificial communicator for human–robot cooperative assembly," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 4, pp. 651–658, 2003.

[29] B. Ionescu, D. Coquin, P. Lambert, and V. Buzuloiu, "Dynamic hand gesture recognition using the skeleton of the hand," *EURASIP Journal on Applied Signal Processing*, vol. 13, pp. 2101–2109, 2005.

[30] Y. Kim, S. Leonard, A. Shademan, A. Krieger, and P. C. W. Kim, "Kinect technology for hand tracking control of surgical robots: technical and surgical skill comparison to current robotic masters," *Surgical Endoscopy*, vol. 28, no. 6, pp. 1993–2000, 2014.

[31] J. Kofman, S. Verma, and X. Wu, "Robot-manipulator teleoperation by markerless vision-based hand-arm tracking," *International Journal of Optomechatronics*, vol. 1, no. 3, pp. 331–357, 2007.

[32] R. C. Luo, B.-H. Shih, and T.-W. Lin, "Real time human motion imitation of anthropomorphic dual arm robot based on Cartesian impedance control," in *Proceedings of the 11th IEEE International Symposium on Robotic and Sensors Environments (ROSE '13)*, pp. 25–30, Washington, DC, USA, October 2013.

[33] R. Afthoni, A. Rizal, and E. Susanto, "Proportional derivative control based robot arm system using microsoft kinect," in *Proceedings of the International Conference on Robotics, Biomimetics, Intelligent Computational Systems (ROBIONETICS '13)*, pp. 24–29, IEEE, Jogjakarta, Indonesia, November 2013.

[34] M. Al-Shabi, "Simulation and implementation of real-time vision-based control system for 2-DoF robotic arm using PID with hardware-in-the-loop," *Intelligent Control and Automation*, vol. 6, no. 2, pp. 147–157, 2015.

[35] I. Benabdallah, Y. Bouteraa, R. Boucetta, and C. Rekik, "Kinect-based Computed Torque Control for lynxmotion robotic arm," in *Proceedings of the 7th International Conference on Modelling, Identification and Control (ICMIC '15)*, pp. 1–6, Sousse, Tunisia, December 2015.

[36] H. Mehdi and O. Boubaker, "Robot-assisted therapy: design, control and optimization," *International Journal on Smart Sensing and Intelligent Systems*, vol. 5, no. 4, pp. 1044–1062, 2012.

[37] I. Ben Aabdallah, Y. Bouteraa, and C. Rekik, "Design of smart robot for wrist rehabilitation," *International journal of smart sensing and intelligent systems*, vol. 9, no. 2, 2016.

[38] C. Pillajo and J. E. Sierra, "Human machine interface HMI using Kinect sensor to control a SCARA robot," in *Proceedings of the IEEE Colombian Conference on Communications and Computing (COLCOM '13)*, pp. 1–5, Medellin, Colo, USA, May 2013.