Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 63 (2017) 184 – 188

www.elsevier.com/locate/procedia

The 50th CIRP Conference on Manufacturing Systems

Gesture control of cyber physical systems

Gergely Horváth[a,b,*], Gábor Erdős[a,b]

[a] *Institute for Computer Science and Control, Hungarian Academy of Sciences, Budapest 1111, Hungary*
[b] *Department of Manufacturing Science and Engineering, Budapest University of Technology and Economics, Budapest 1111, Hungary*

[*] Corresponding author. Tel.: +36 1 279 6181; *E-mail address:* gergely.horvath@sztaki.mta.hu

Abstract

The next generation of robots should be able to work well in the presence of human operators safely without posing potential life hazard. To realize the collaborative and symbiotic work of humans and robots, it is essential to use a convenient interface that is natural for humans. The design of the interface must consider ergonomic aspects and must not hinder the work of the operator. The interface have to be unambiguous for both the robot and human operator. In this article, we present a control for cyber physical systems. In order to properly interpret and execute commands in a human-robot shared workspace, it is essential to have a virtual model, where commanded actions can be justified from feasibility and safety point of view. This is established by implementing the digital twin of the robot cell. This virtual model contains both the model of the robot and the human operator. The controller allows simultaneous control of robots, both the actual machines and the models in cyber-space, as well as updating the posture and gestures of the human operator.

The implemented gesture control language is modular. Modularization creates the opportunity to reconfigure the cell with minimal energy expenditure and still use the same gestures, in order to realize different production use-cases.

Peer-review under responsibility of the scientific committee of The 50th CIRP Conference on Manufacturing Systems
Keywords: gesture control; cyber physical system; human robot collaboration

1. Introduction

For a long time the sole purpose of autonomous robots was to redeem human workers from labor. Robots are already used in many different fields such as assembly robots for production lines, transport robots in industrial facilities, or domestic areas like household maintenance as cleaning robots. The next generation of robots, however, has to be able to work in a shared workplace with human operators complementing and not substituting their work. The implementation of such shared workplaces raises new problems, such as ensuring safe operation of human and robot, defining simple and unambiguous communication interface between the human operator and the robot, to name a few.

Robots working with humans in a shared workplace require a certain degree of autonomy. It is necessary that the robot knows its state and its surrounding and capable of selecting from a list of feasible actions. The actual state of the workplace (e.g. position of the human worker, command of the worker, etc.) could be obtained only by processing raw sensor data (e.g. camera, depth sensor, etc.) with dedicated algorithms. If the sensor inputs and the processing algorithms are available, then the robot has the opportunity to make adequate decisions that takes into account the actual state of the workplace. In order to endow the robot with this autonomy, first the actualized digital representation of the workplace should be derived. This digital representation should provide the accurate position of the objects in the surrounding, the position of the human operators and the interaction channel with the operator. This virtual model is called the digital twin [1].

The human robot interface has to be such that it considers ergonomic aspects and does not hinder the work of the human operator. An emerging technology in human robot interaction is, robot control through human gestures. Faudzi et al. [2] used images to recognize different hand gestures to control a robot through a specially made control circuit. Raheja et al. [3] report also a work using computer vision—with no depth data—that recognizes the direction of pointing. Many research use the cheap Kinect sensor for depth data based gesture recognition, like Bernier et al. [4] who works out a proper way to segment the data into gestures and pre-, and post-strokes, applying machine learning algorithms. An elderly care robot was programmed to recognize calling gestures in [5] by Zhao et al. Gerard et al. [6] uses gesture recognition and demonstrates it through an item pick-up scenario.

These papers addresses and solves many problems for single robots, but they fail to address collaborative workplaces with multiple acting agents. In this paper we present a method to control a cyber-physical system with novel gesture control realizing safe operation through applying the principle of digital

Peer-review under responsibility of the scientific committee of The 50th CIRP Conference on Manufacturing Systems
doi:10.1016/j.procir.2017.03.312

twin.

Topic of the research was motivated by problems taken from assembly processes. In certain assembly operations precise fitting and the lifting and positioning of heavy objects are present. There are situations where certain assembly operations take place in hidden, or hard to reach places. In these scenarios the cooperation of human and machine is necessary as the human operator can easily adapt to reach any kind of hidden place, while robots can lift even extremely heavy weight objects. In the presented scenario the usage of the digital twin is twofold:

- collision detection of the human operator and the robot, realizing safety measures.
- gesture control to signal the robot to move on to the next preprogrammed task (e.g. moving the part currently in production into the next assembly position).

In Section 2 the concept of digital twin is reviewed. In Section 3 our gesture recognition algorithm is laid out. In Section 4 our use-case is detailed, while in Section 5 we summarize our results.

2. Digital twin

The digital twin of the workcell, shared by human and robots, is modeled as a *linkage*, which is capable of capturing the geometric and kinematic relations of the static and moving objects. The linkage concept is developed in LinkageDesigner [7], which is an add-in application package to Wolfram Mathematica™. A linkage is basically a graph whose nodes denote links, while edges are constraints between the links (see in Figure 1). Four kinds of constraints are considered: revolute, prismatic and universal joints, as well as fixed transformation. Two links and a constraint between them define together what is usually referred to as a *kinematic pair*. The linkage of the workcell defines an open kinematic chain. The linkage captures the following basic properties of the workcell

- Description of the links
  - link volume (via 3D triangle mesh)
  - local link reference frame
- Kinematic constraints between the links
  - fixed transformations
  - parameterized variable transformations representing the moving kinematic pairs

The linkage structure can capture more information than required directly for one mechanism, in fact, it can model a complete workcell including robot, human operator, workpiece, fixture, stationary equipment, feeder devices (e.g. turntable), and other technologically relevant volumes. Also, it is capable of modeling spatial relations (e.g. relative location within the workcell), as well as spatial and kinematic constraints. Since the spatial relation of moving objects of the workcell are modelled with kinematic transformations it is possible to update the actual transformation based on measured sensor data. This way the virtual twin can be actualized and provide an accurate digital representation for various planning and decision making processes. Due to its versatility, the linkage can serve as the pivotal data structure for modeling the digital twin of the workcell.

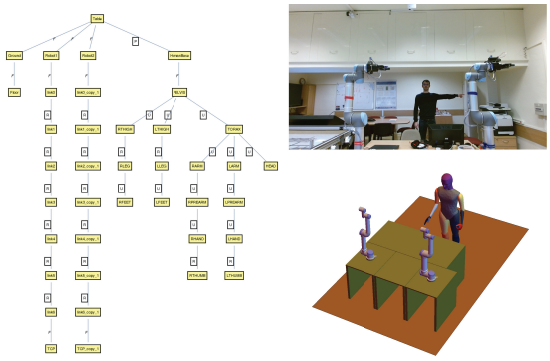

Fig. 1. Linkage graph (left), photo (top right) and virtual model (bottom right) of the workcell

3. Gesture control

We realized control for the cyber-physical system through the usage of hand-gestures. The system is able to distinguish between various hand gestures like an outstretched arm forward, upward or sideways vertically, or in different angles upwards and downwards.

The recognition system is built up of a Microsoft Kinect™ version 2 sensor, and a computer running the gesture recognition algorithm. The gesture recognition process has three distinct levels:

- first the signal of the sensor has to be acquired and some preprocessing executed.
- a coordinate transformation has to be carried out in order to transform the joint coordinates from camera space, into model space.
- finally, the transformed joint data is used to differentiate between the pre-defined gestures.

3.1. Signal acquisition

The Kinect for Xbox One has multiple built in sensors. These include a time-of-flight sensor[8] for sophisticated depth vision. The resolution of the depth sensor is 512×424, with a 30Hz sampling rate.[9] The sensor also features a color camera, equipped with an active infrared capable sensor and a microphone. The color camera is full HD, meaning it captures video in 1080p. The frequency of the video stream is 30Hz, however, in poor lighting conditions it falls to 15Hz. The active IR sensor creates the opportunity to use the Kinect in low light environment.

These sensor data can be obtained using multiple software APIs, including the official Kinect for Windows SDK or the free and open source OpenKinect library project. We used the official Kinect SDK in our work, which presents the possibility to query the state of the sensors, or get notified when the sensor data gets updated. We utilized these features when implementing a C# class that stores the actual state of the sensors. After acquisition and data storage is done, on query the class

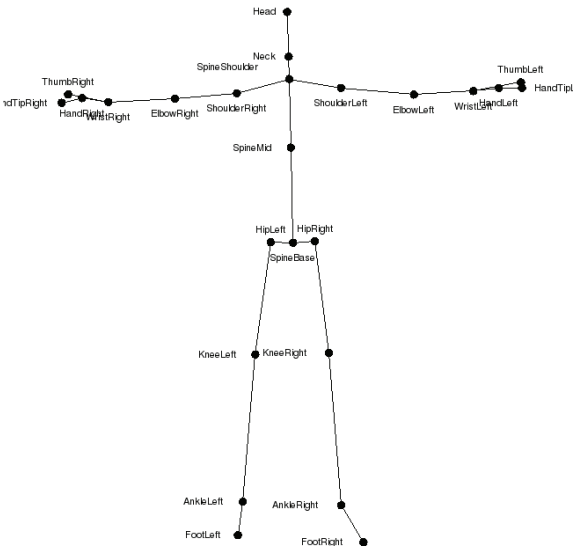

Fig. 2. The returned joints (left and right switched because image is made from the point of view of the sensor)

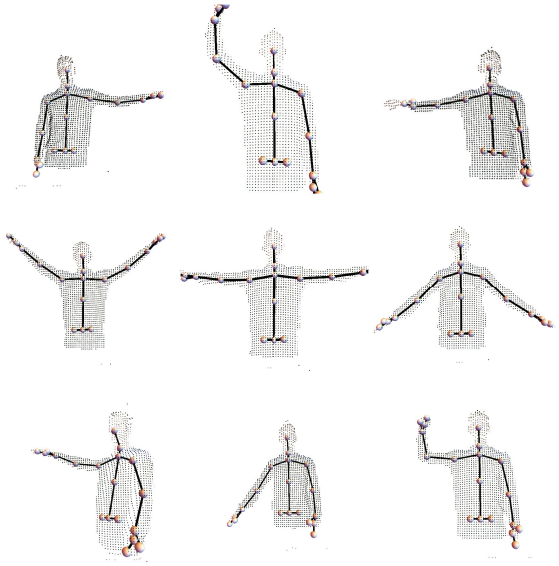

Fig. 3. The recognizable gestures. From left to right, from top to bottom: `LEFT`, `UP`, `RIGHT`, `DUP`, `LEFTRIGHT`, `DDOWN`, `FORWARD`, `DOWN`, `BACKWARD`.

Table 1. Coordinates of the human object on Figure 2

Joint Name	`SpineBase`	`Neck`	`Head`	
Coordinate	$\begin{pmatrix}-0.11\\-0.12\\2.39\end{pmatrix}$	$\begin{pmatrix}-0.09\\0.48\\2.43\end{pmatrix}$	$\begin{pmatrix}-0.09\\0.64\\2.45\end{pmatrix}$	$\cdots$

can return the last accumulated image, be it color-, depth-, or IR image.

The SDK also contains methods for rigging the point cloud of human operators—or players in the terms of the SDK—and returning the position of the joints. This functionality has been wrapped in a way so it does not only returns the joints, but the hand states as well. Figure 2 displays a rigged player while Table 1 the name of the joints returned. The names are self-assigned.

3.2. Coordinate transformation

The Kinect SDK returns the joint coordinates of the human player. The virtual model has to be adjusted to the real world. This means, the human model needs to have the same joint positions as the real operator, which have to be calculated. This calculation is much easier if the two models are in the same local reference frame. The reference frame of the Kinect sensor differs from the virtual reality's in two aspects:

- a unit in the Kinect sensors coordinate system is $1m$, while $1mm$ in our virtual model;
- the directions of the three main axis does not overlap (upwards is Y in Kinect space and Z in virtual model).

To arrange this a homogeneous transformation matrix has been created, which is used to transform points from one reference frame to another. It is a 4×4 matrix, where the upper left 3×3 sub-matrix is an orientation matrix (with possible scaling), while the first 3 elements of the last column are responsible for the translation. The last row is $0, 0, 0, 1$. In order to use a homogeneous transformation matrix, the vector also has to be extended with a forth coordinate, which is usually 1. The result of the transformation will be a 4 element long vector, where the last coordinate can be stripped to obtain the coordinates of the transformed vector.

We created a homogeneous transformation matrix that converts between the `SpineMid` joint in the reference frame of the Kinect and the virtual model. The created matrix looks like the following:

$$\begin{pmatrix}-1000 & 0 & 0 & 1000sm_x\\ 0 & 0 & 1000 & -1000sm_z\\ 0 & 1000 & 0 & 1000-1000sm_y\\ 0 & 0 & 0 & 1\end{pmatrix}$$

where sm_x, sm_y, sm_z are the coordinates of the `SpineMid` joint in the frame of the Kinect.

3.3. Gesture recognition

Gesture recognition is built using Wolfram Mathematica™, which provides an interface to utilize the class mentioned in Section 3.1. From the global coordinates of the joints the bones constructing the human operator is calculated, as the posture of the bones is the basis of the recognition. These include relative angular position of bones—compared to other bones—and global angular positions of bones, which are compared to the global axis. Figure 3 shows the recognized gestures.

The recognition process is a clustering. The objects are the set of various angles between the limbs of the human operator, and the groups are the different gestures. Our algorithm uses

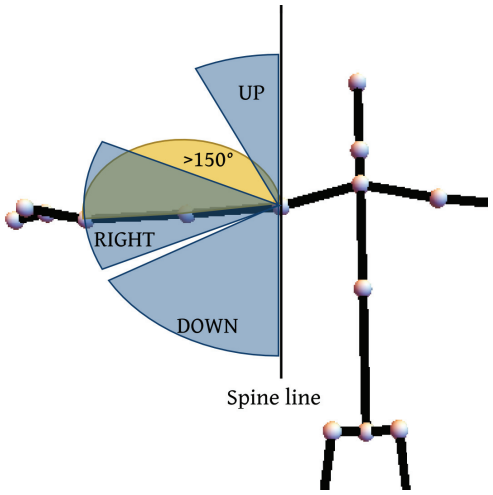

Fig. 4. Example of UP, RIGHT and DOWN gestures (blue) and the clustering definition of a straight right arm (yellow)

a rule-based method, where the rules are applied based on previous observations. These rules consider three aspects of the joints. These are:

- whether the gesture uses the left or the right arm (or both),
- if the arm is extended in the elbow or bent,
- in which direction the hand is pointing compared to the global directions.

Such an example, considering only the right hand, which is extended can be seen on Figure 4. The example shows the distinction between three major gestures: UP, RIGHT and DOWN. If predefined gestures cannot be recognized, its interpreted as a no-action.

3.4. Gesture control

Now the control process is the following. An instance of the implemented algorithm is running on a computer connected to the Kinect sensor. The algorithm executes a loop and queries the actual bone positions of the human operators in the view field of the Kinect. From the available data, it determines the operator to be the base of gesture recognition. The exact method for operator determination is shown in Section 4.

After the operator is chosen, recognition of its gesture is attempted. The posture of the human operator is classified according to the predefined rules. If no gesture rule can be fit on it, no command is sent out.

The commands are sent to a preprocessor which checks if the command is feasible and if it is, sends it to the controlled instances be it physical or virtual. In our experiment we controlled one robot in the real world and one virtual instance.

The activity diagram of the whole gesture recognition and control process can be seen on Figure 5. The figure demonstrates a single iteration of the recognition process. As it can be seen most of the involved units of the system are executing continuously, while the algorithm itself triggers when called and makes the 3D sensor capture the image. Later the system can be extended such a way, that the algorithm does not have to be called externally but it monitors the data stream of the 3D sensor.

4. Use case

The cell consists of two UR5 robots, a table those sit on, and multiple human operators. For the digital model of the workcell, the kinematically valid mechanism models of the machines and the human operators are created. During the construction of the mechanisms, proper joints are specified, guaranteeing the mobility of the models. The model of the two UR5 robots are identical. They contain 6 rotational joints which connect the successive links in a serial fashion. The model of the human operators are composed of 4 rotational joints and 12 universal joints. There is an additional planar joint (2 more rotational joints and a translational joint), which accommodates for the positioning of a human operator relative to the robots. Figure 1 shows the graph of the assembled workcell, with—for the sake of simplicity—one human operator.

As mentioned in Section 3.4 the operator to control the robots has to be chosen. In our case the closest human operator is chosen. To decide the right person, the distance of every human operator from both robots have to be measured. It would be a challenging task to make these measurements in the real world, as the human operators or the robots can move, which burdens the positioning of the distance sensors heavily. Because of the mentioned reasons virtual sensors are applied that perform the task at hand. Distance calculation is based on the PQP (Proximity Query Package) [10]. Figure 6 shows the calculated distances for one operator, considering multiple human joints. With the aid of the virtual sensors we are able to make the distance measurements in the virtual model, find the closest operator for every robot and find their ID, which can be sent back to the processing algorithm.

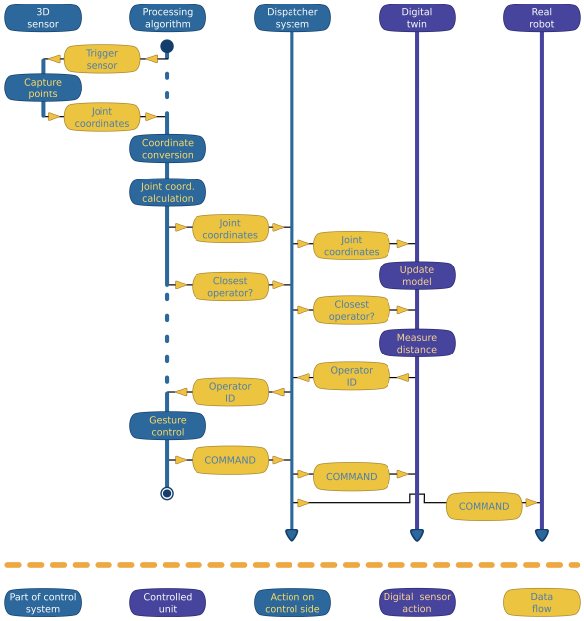

Fig. 5. Activity diagram of the whole system

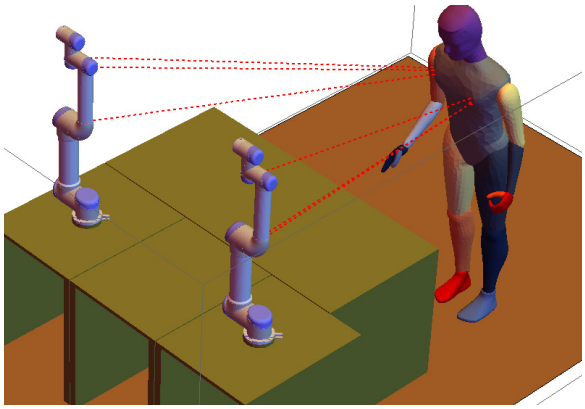

Fig. 6. Distance calculation of human operator and robot links

Table 2. Distance between robot and human operator. Closest points of robot link and human torax is given.

Link name	Distance	Point pair	
link2	1390.57	$\begin{pmatrix}-103.398\\312.323\\1425.9\end{pmatrix}$	$\begin{pmatrix}-125.673\\-1072.02\\1296.\end{pmatrix}$
link3	1390.34	$\begin{pmatrix}-103.398\\312.323\\1425.9\end{pmatrix}$	$\begin{pmatrix}-106.385\\-1072.29\\1299.9\end{pmatrix}$
link4	1417.5	$\begin{pmatrix}-109.166\\311.296\\1450.8\end{pmatrix}$	$\begin{pmatrix}-157.043\\-1092.72\\1639.9\end{pmatrix}$
link2_copy_1	1707.42	$\begin{pmatrix}-364.576\\327.468\\1451.3\end{pmatrix}$	$\begin{pmatrix}-1330.37\\-1072.61\\1302.\end{pmatrix}$
link3_copy_1	1677.49	$\begin{pmatrix}-381.163\\338.989\\1463.\end{pmatrix}$	$\begin{pmatrix}-1224.61\\-1094.9\\1678.7\end{pmatrix}$
link4_copy_1	1727.39	$\begin{pmatrix}-390.832\\345.705\\1469.9\end{pmatrix}$	$\begin{pmatrix}-1323.37\\-1092.67\\1682.8\end{pmatrix}$

With the proper operator IDs the processing algorithm can find the joints for the corresponding person, and analyze their gesture. If a known gesture is identified, a command for the robots are generated. This command is sent—through a dispatcher system—to both the virtual and real robots, to be carried out.

5. Conclusion and future work

The use-case explained in Section 4 demonstrates the possibility of integrating virtual sensors in a workcell. While distance measurement was used to find authorized personnel by its position to operate the robots, virtual sensors have the potential for diverse utilization. To name a few examples the virtual model can be extended with color data, making it possible to recognize QR-code stamps or even faces of the operators, or it is possible to use virtual sensors to measure the velocity or acceleration of the robots, without actually encumbering potentially low capacity equipment. The concept of digital twin can also be exploited in quality control (e.g. realtime checking if the operator strictly followed a predefined sequence of tasks, or failed/skipped some), along with advanced control implementing even more elaborate gesture languages or using feedback control on a robot, based on current state.

In our paper we showed how we kept the digital copy of a workcell up-to-date, exploiting data of available sensor. We also demonstrated how to extract information out of the digital twin, and how to implement digital sensor in a cyber-physical system. The obtained data have been successfully used in a gesture control scenario.

6. Acknowledgement

This research has been supported by the Hungarian Scientific Research Fund (OTKA), Grant No. 113038 and the GINOP-2.3.2-15-2016-00002 grant on an ”Industry 4.0 research and innovation center of excellence”.

References

[1] Rosen, R., von Wichert, G., Lo, G., Bettenhausen, K.D.. About the importance of autonomy and digital twins for the future of manufacturing. IFAC Papers Online 2015;48(3):567–572.

[2] Faudzi, A.A.M., Ali, M.H.K., Azman, M.A., Ismail, Z.H.. Real-time hand gestures system for mobile robots control. Procedia Engineering 2012;41:798–804.

[3] Raheja, J., Chaudhary, A., Maheshwari, S.. Hand gesture pointing location detection. Optik - International Journal for Light and Electron Optics 2014;125(3):993 – 996. doi:http://dx.doi.org/10.1016/j.ijleo.2013.07.167.

[4] Bernier, E., Chellali, R., Thouvenin, I.M.. Human gesture segmentation based on change point model for efficient gesture interface. In: 2013 IEEE RO-MAN. 2013, p. 258–263. doi:10.1109/ROMAN.2013.6628456.

[5] Zhao, X., Naguib, A.M., Lee, S.. Kinect based calling gesture recognition for taking order service of elderly care robot. In: The 23rd IEEE International Symposium on Robot and Human Interactive Communication. 2014, p. 525–530. doi:10.1109/ROMAN.2014.6926306.

[6] Canal, G., Escalera, S., Angulo, C.. A real-time human-robot interaction system based on gestures for assistive scenarios. Computer Vision and Image Understanding 2016;149:65–77.

[7] Erdős, G.. Linkagedesigner, the mechanism protyping system website. `http://www.linkagedesigner.com`; 2005. Accessed: 2016-12-30.

[8] Bamji, C.S., O'Connor, P., Elkhatib, T., Mehta, S., Thompson, B., Prather, L.A., et al. A 0.13μm cmos system-on-chip for a 512×424 time-of-flight image sensor with multi-frequency photo-demodulation up to 130 mhz and 2 gs/s adc. IEEE Journal of Solid-State Circuits 2015;50(1):303–319. doi:10.1109/JSSC.2014.2364270.

[9] website, M.C.. Kinect hardware. 2014. URL: `developer.microsoft.com/en-us/windows/kinect/hardware`.

[10] Larsen, E., Gottschalk, S., Lin, M.C., Manocha, D.. Fast proximity queries with swept sphere volumes. In: Proc. IEEE Int. Conf. Robot. Autom. 2000, p. 3719–3726.