# Telematic Control and Communication with Industrial Robot over Ethernet Network

M.W. Abdullah\*, H. Roth, J. Wahrburg Institute of Automatic Control Engineering University of Siegen Siegen, Germany \*abdullah@zess.uni-siegen.de

*Abstract* - This research presents and evaluates an approach to control and communicate with industrial robot control system over Ethernet network in order to implement online guiding and live data exchange. One of the main benefits of this fullytelematic control is to give the ability to integrate external sensors so that a guarded-motion can be implemented or to use sensorguided motion by generating online trajectories. A comparison in terms of execution time, sensor integration, and security is made between the proposed fully-telematic control approach on one side, and the semi-telematic approach and the classical offline control of industrial robot on the other side.

# Keywords - telematic control; indutrial robot; Ethernet network; sensors integration.

#### I. INTRODUCTION

The usage of industrial robots is increasing rapidly worldwide and especially after the advanced developments in the industrial robot technologies, such as in the built in controlling software and the interface options with different external devices. However, despite of variety of sectors where industrial robots are being used, the International Federation of Robotics reported that the global density of industrial robots was around 58 per 10,000 employees in 2014 [1]. This number can be seen as a low density if the advantages of using industrial robots in the plants are considered. For example, industrial robots increase the production rate and reduce the operational cost when long term view is taken. In addition, the robots give better product quality since they are more precise and have low repeatability error compared to the human operators. When looking into the applications where the implantations of industrial robots are limited, as an example a simple assembling application of electronic parts in automotive industry, it can be noticed that one of the main reasons in this low usage is because of the limited ability of industrial robot to deal with unforeseen events, i.e. accidents or changes in workspace. Another reason that can be found more in small and medium enterprises is due to the variety of the given tasks and their different configurations in the processes and the environments that requires flexible handling and adaption from the operator [2]. Therefore, fully automated processes could not be achieved unless the industrial robots can meet these dynamic requirements by providing the same

M. Weyrich Institute of Automation and Software Engineering University of Stuttgart Stuttgart, Germany michael.weyrich@ias.uni-stuttgart.de

flexibility as the human operator does [3]. Thus, in order to have automated production lines with high efficiency and performance in the processes, it is required to have intelligent industrial robots that can sense the surrounding area by reacting to unforeseen events that might occur and also by interacting with human operators. This can be achieved by sensor guided industrial robot with sensor based control. The keyword and first challenging phase in the implementation of sensor guided robot is the integration approach of external sensors with the control system of the industrial robots [4]. Such an approach should always consider less hardware modification and simplicity in integration with the existing commercial control systems that are already in service with more than 1.5 millions industrial robots [1].

#### II. BACKGROUND

Most, if not all, the industrial robots come with their own control systems provided by the robot manufacturer. The traditional way of using the industrial robot is to program offline once and run it forever concept. In some cases if the robot needs to work on conveyer on which the products come to robot station by, the trigger of robot program or the movement is done by the digital or analog input/output interface which is usually a default feature. Moreover, the commercial control systems do not give usually an access to the internal low-level control loops of robots. On the other hand, some recent advanced industrial robots give external sensor interface feature, however, they are usually more expensive, and not all types of sensors can be used directly.

Therefore, the first and most common approach the researchers tend to use in order to integrate external sensors with industrial robots is to replace the commercial control systems with their own developed ones [2]. This approach has two crucial issues; firstly the user will waive the warranty the robot manufacturer gives with the commercial control system, and secondly, an accurate dynamic model should be built to control the robot. The dynamic models for the manipulator in general can be driven by Euler-Lagrange method or Newton-Euler method in form of motion equation. Assuming that all the correct parameters are provided by the robot manufacturer and a solid control system is built, it is unlikely to find this approach used outside research labs and it is rarely found in the industry.

More recent approaches to realize sensor integration with commercial control systems are by using the alternative interface options that either exist inside the control system or require a hardware modification. Modern KUKA robots provide an explicit Robot Sensor Interface (RSI) where sensor data can be included with the defined trajectory for the internal control algorithm [5, 6]. EtherCAT is one of these interfaces where some of industrial robot control systems provide and many researchers have used it to realize real-time motion control [7, 8].

Other types of interface also have been used to integrates sensors with robot control system such as CAN bus and RS232 that usually require hardware modification either on the commercial control system or on the network of the work floor, such as the approaches were used in [9, 10, 11]. EtherNet/IP interface that provide real-time data transmission is used with VxWorks operating system to integrate external Force/Torque sensor with KUKA robot in [12]. Open source control systems, e.g. ROS [13], also have been contributing in a solution to standardize one platform for different types of industrial robots and support external sensors integration. On the other hand, according to IMS Research results, as shown Ethernet TCP/IP interface is the most common in Fig. 1, interface type among others Industrial Ethernet in the industry [14]. In fact, many devices such as HMIs, PLCs, machines, sensors, and even robots control system have made it as standard interface option. In the experiment presented in this research it is also the cases that Ethernet TCP/IP is a standard interface feature in the robot model used.



Fig. 1. World market for industrial Ethernet 2015 forecast.

To summarize, external sensor integration is still not a solved solution for many existing control systems of industrial robots. Only advanced types have their own dedicated interface for sensors such as in KUKA models. The other alterative requires hardware and software modification which is not always feasible due the closed architecture of control systems. Open source systems such as ROS have couple of limitation in the integration of sensors beside their complicated software that requires experts and high programming knowledge to modify the system in order to implement new sensors.

#### **III. PROPOSED TELEMATIC APPROACHES**

#### A. Objectives

In order to have the ability to integrate external sensors and to communicate with the commercial control system without replacing it or making a hardware modification, this research will present two approaches for such requirements. The interface link between the telematic machine, i.e. a PC, and robot control system will be carried through an Ethernet interface which is as mentioned previously the most common interface type. By doing so, not only external sensors can be integrated, but also live data such as; Tool Center Point (TCP) position and orientation, torques on joints, input/output pins status, and etc. can be transmitted from the robot control system to the telematic machine. This data can be useful for different applications, for instance, in this research it will be used to trigger commands by including it as feedback for telematic control system. To summarize, the objectives of this work are:

- Provide a simple and fast solution for external sensors integration with robot control systems that have limited sensor interfaces.
- Telematic control of industrial robot over Ethernet network. This opens the doors to use complicated guiding algorithms that is hard to be implemented directly on software of the robot control system.
- Standardize remotely command to be used with different types of control systems.
- Monitor robot's live data remotely.
- Study the performance and feasibility of the proposed approach in comparison with the traditional method of writing the programs inside the robot control system.

#### B. Hardware Setup

The industrial robot used here is a UR5 Robot manufactured by the Danish company Universal Robot. It is a low cost 6-DOF manipulator with 5 kg payload. UR5 control system provides couple of standard interface options including Ethernet TCP/IP socket. Using this interface real-time measurements, ex. joints'; positions, velocity, torques, etc, can be read with frequency of 125 Hz. It also gives the possibility to write command either directly from the socket or to be included inside the program loop as will be explained in the next subsection. In addition, a Force/Torque sensor is used as an external sensor to be integrated with the industrial robot. The F/T sensor, type ATI Gamma F/T Transducer, provides Ethernet interface through its NetBox using Row Data Transfer (RDT) with high-speed streaming data up to 7000 Hz by UDP protocol. Therefore, it cannot be integrated directly to the robot control system.

The only additional hardware used is a standard switch to connect the robot control system and external F/T sensor NetBox with the telematic machine as shown below in Fig. 2.



Fig. 2. System architecture.

## C. Approch' Concepts

To control a robot, a program should be written inside its control system using one of the available user interfaces. This program, that is written with the manufacturer provide own language, guides the robot for the required action as the control system understands commands format. The concept of purposed approach, which will be called *fully-telematic* control, is to write a small program with default provided robot language, inside the robot control system using the user interface device, PolyScop. The purpose of this program is to receive the messages from the telematic machine, which is in different format than the robot default command format. through the Ethernet interface and process them to instruct the robot for the required action. In this case, the program written with PolyScop and saved inside the control system will act as handler for the commands by processing them and converting them to the format that robot understand. As shown in Fig. 3, this program consists of five parts:

**Before Main Program:** this part initiates the communication with telematic machine by defining the IP address, the port number, and set the I/O. The PolyScop will act a client in this scenario and the telematic machine is the server.

**Main Manipulator Program:** handles the movement of robot by the received poses that were interpreted in the other part. Two type of movement are defined here, the movement in joint space and in tool space. The movement can be always interrupted if a request is sent from the telematic machine.

**Thread 1:** current pose of the robot TCP in terms of position and orientation (x, y, z,  $\theta_x$ ,  $\theta_y$ ,  $\theta_z$ ) is sent continuously to the telematic machine with frequency of 0.5 second.

**Thread 2**: is the part that checks and changes the status of I/O based on the received command.

**Thread 3:** works as security measure by continuously checking for a signal to interrupt the movement inside the main program.

These Threads are executed in parallel with the main program, therefore; it is possible to handle different commands in the same time and even when the robot is in the motion state.



Fig. 3. Communication between telemaitc machinne and the internal program written inside PolyScope.

Three types of messages are being sent to the control system and handled by the Threads parts. The pose message as shown in Fig. 4 consists of ten variables. Table 1 explains each variable usage. The format and length of the message can be changed or extended by the user based on the control system specifications and features provided. The purposed message has the essential variables to define the path for the robot.

Pose Message (10 Variables)												
Моv. Тур.	Ref. Cor.	Position			Orientation			Opt.				
		×	У	z	θ×	θγ	θz	C1	C2			

Fig. 4. Pose message format sent from the telematic machine.

TABLE I.					
Pose Message Variables	Definition				
Movment Type	1: Joint Space 2: Linear 3: Circular Blend				
Reference Coordinate	<ol> <li>Base coordinate</li> <li>Tool coordinate (TCP)</li> <li>User specified coordinate</li> </ol>				
Position	Targeted position of TCP				
Orinetation	Targeted orientation of TCP				
Optional variables	Two variables that can be used to set the velocity and acceleration or blend radius. If set to 0 then the default settings is used.				

The second type of messages is designated for the digital I/O. It contains three variables, first one defines whether to read or write from I/O and second variable specifies the number of port. In case of the read request for the state of the input port, third variable will be ignored and the Thread that handle this type of message will send to the telematic machine the state of the requested input number. For setting the digital output, variable three in this case will define the state of this output. The last message type is to interrupt the movement of the robot or the execution of the program based for example on the sensor data or user request. When an interrupt signal is sent, the robot will wait for a new pose to move to. The main advantage of this approach is a safe execution of the commands since the internal software of the control system is handling them. In addition, error handling scenario can be defined inside the control system and thus, no need to stop the robot and reset the process. The downside of this approach is the dual processing of information inside the telematic machine and the control system which will lead to a delay or longer execution time of the commands. This approach is called a fully-telematic because the control of the robot is always under the supervision of the telematic machine, and it is always possible to interrupt a movement or the execution of a specific command.

For the the semi-telematic control approach, that is used for example by ROS and suggested by robot manufacturer, the commands are executed directly inside the control system. This is done by sending each command line individually and waiting for the control system to execute it, and only after that the next command line is sent. In this case, there will be no handler for commands, i.e. PolyScop, which will lead to a faster execution time. On other hand, to retrieve a specific data from the robot, a request should be sent each time, and not like the pervious approach where a parallel program is sending the data continuously. However, retrieving the data directly is much faster as will be shown later in the verification tests. Another pitfall with this approach is the error handling, if the command has a wrong or missing value then there are three possible scenarios. First, there will be no indication sensed, i.e. the robot will ignore the wrong command. Second possibility is the robot will execute the wrong command which will lead to a possible collision with itself or the surrounding objects. And last scenario is when the robot enters in the *Emergency State* where a human operator should manually restart the robot. The reason of calling this approach as semitelematic is because the telematic machine will lose the control during excitation of the command and can listen or perform the coming command only after finishing the current one.

The communications methods of two the approaches are shown in Fig. 5. The commands in the semi-telematic approach should be formatted according to the script language provided by robot manufacturer, thus, it cannot be universal as in the fully-telematic approach. For example, to move the robot TCP linearly to the targeted position from user defined coordinate, the first approach needs to send [2, 3, x, y, z,  $\theta_x$ ,  $\theta_y$ ,  $\theta_z$ , 0, 0]. While in the second approach, the user coordinate should be defined first in individual request, then the type of the movement should be defined each time the move function is used. For the example with the robot mode used in this research, above the request will be *movel(x, y, z, \theta\_x, \theta\_y, \theta\_z, \alpha, v,) where \alpha and v represent the acceleration and velocity respectively.* 



Fig. 5. Communication channels between URControl and Telematic Machine in the two approaches.

### D. Telematic Machine

The telematic machine used in this research is a laptop running with Windows 8 operating system. It has an Intel iCore 3 processor with 1.7 GHz and 4 GB RAM. The controlling program was implemented in LabVIEW software, see Fig. 6 below, where the visualization for the F/T sensor and robot status is created for the user. Also manual controls based on two approaches were created so the user can select between the messages and their format that will be sent to the robot control system.



Fig. 6. Telematic Machine Control Software GUI.

### E. Experiment and Preminiary Results

First experiment is conducted to check the execution time for a specific request sent by the the telematic machine. In this experiment, the telematic machine is requesting the current external force at TCP which is measured internally by the control system. The execution time is calculated as the delta time between request when data package leaves the telematic machine and answer data package arrives to telematic machine. The data was captured by Wireshark network analyzer software. The average time as shown in Fig. 7 is 13.8 ms for the first fully-telematic approach. A fourth *Thread* was defined inside the internal program written with PolyScope to response to the request for TCP force. This thread consist of five commands lines starting with setting a variable to received message over Ethernet, interpreting the message, calling the function get tcp force(), sending the results to telematic machine, and finally a loop to continuously reading coming messages.



Fig. 7. Execuation time for specific requests in fully (blue) and semi (red) telematic control approaches.

On the other hand, the average execution time for the semitelematic approach was 1.71 ms. These preliminary results show that the second approach has as expected a faster execution time, since the communication is directly between the telematic machine and the robot control system without a middle man. The second experiment aimed to compare the process time of same assembling application in the two proposed approaches and the classical offline control. In this peg-in-hole assembling application, the robot has to move to five poses as can be seen in Fig. 8. First pose represents the home pose from which the robot should start and end the process. The peg is gripped through two poses, 1 and 2, and this because the end-effector needs to approach the peg vertically to grip correctly. Same steps are used also in the assembling phase, i.e. 3 and 4 poses.



Fig. 8. Peg-in-Hole assembling application five poses.

In the offline programming approach, the average process of time of 10 runs to the complete assembling application was 7.38 seconds. There were two waits steps of 0.5 second during the gripping and releasing of the peg immediately after reaching pose 2 and 4, and this is necessary to give the time to the gripper to close and open before the robot start the movement again. The same wait steps were also used in the semi-telematic control. But since the robot should receive the next move-command only after reaching the previous pose, a check loop was needed to compare the incoming message. Each message is sent from the robot with frequency of 125 Hz and that contain the actual position of robot. Thus, the trigger to send the next move-command is by comparing the sent pose with the actual pose sent received. The average process time in this approach was 10.21 seconds. For the fully-telematic control approach, again the wait steps were as in previous approaches. As for triggering the next move-command, a reached signal is sent from the program written inside the PolyScope to indicate that the robot has reached the required pose. The average process time in this approach was 12.45 seconds. The robot speed and acceleration were the same in all the approaches.

Sensor based automated peg-in-hole assembling using vision sensor and F/T sensor was carried with the purposed fully-telematic approach and presented in [15]. This intuitive search algorithm to detect the hole uses different the models that interrupted the contact forces shown in Fig. 9. The assembling task was carried in seven steps compared and the execution time, including image processing to primarily detect the location of the hole, was around 27 to 32 seconds.



Fig. 9. F/T sesnor reading during peg-in-hole assembling steps.

#### IV. CONCLUSION

This paper presented an approach to guide industrial robot with a telematic machine over Ethernet interface. With this approach external sensors can be integrated with existing robot control systems that have limited sensor interfaces, and thus, more intelligent guiding algorithms can be implemented. This will give the opportunity for the industrial robots to be used in applications that require sophisticated processes such as assembling of electronic parts. In Table 2, the results of the comparison between the proposed approach and the existing other two approaches for controlling of industrial robots are given.

TABLE II.

Comparison between control	Approach				
methods	Fully- Telematic	Semi- Telematic	Offline (Classic)		
Real-Time Guiding		•	•		
External Sensors Integration	•	•			
Guard-Motion	•				
Online Trajcotries Generation	•	•			
Live Data Exchange	•	•			
Telematic Control	•	•			
Peg-in-hole assembling time (s)	12.45	10.21	7.38		

#### REFERENCES

- [1] Executive Summery World Robotics 2013; Industrial Robots, International Federation of Robotics-IFR, 2014.
- [2] A. Blomdell, G. Bolmsjö, T. Brogårdh, and others, in: Extending an Industrial Robot Controller, Implementation and Applications of a Fast Open Sensor Interface. IEEE Robotics and Automation Magazine, September 2005.
- [3] E. Kus, R. Grüninger, and R. Hüppi, in: Integration of Intelligent Sensors for Sensor guided Motions in Industrial Robot Applications. IEEE International Conference on Automation and Logistics Qingdao, China, September 2008.
- [4] T. Kreoger and F. M. Wahl in: Multi-Sensor Integration and Sensor Fusion in Industrial Manipulation: Hybrid Switched Control, Trajectory Generation, and Software Development. IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems, 2008.
- [5] R. Winkler and J. Suchy in: Position Feedback in Force Control of Industrial Manipulators An Experimental Comparison with Basic Algorithms. IEEE Conference on Robotic Sensor Environments, 2012.
- [6] KUKA Robot GmbH in: KUKA Robot Sensor Interface, 2009.
- [7] Il-Kyun Jung and Sun Lim in: An EtherCAT based Real-time Centralized Soft Robot Motion Controller. International Conference on Instruments and Measurements, Sensor Network and Automation 2012.
- [8] T. Choi, H. Do, J.Kyung, D. Park and C. Park in: Control of 6DOF Articulated Robot with the Direct-teaching Function using EtherCAT. IEEE International Symposium on Robot and Human Interactive Communication (2013).
- [9] S. Luo and S. Zhu in: Open Architecture Multi-Axis Motion Control System for Industrial Robot Based on Can Bus. International Conference on Automatic Control and Artificial Intelligence (2012).
- [10] [10] P. Dzitac A. M. Mazid in: A Depth Sensor to Control Pick-and-Place Robots for Fruit Packaging. International Conference on Control, Automation, Robotics & Vision, 2012.
- [11] M. Lotz, H. Bruhm and A. Czinki in: A New Force Control Strategy Improving the Force Control Capabilities of Standard Industrial Robot. Journal of Mechanics Engineering and Automation 4 276-283, 2014.
- [12] J. Loske and R. Biesenbach in: Force-torque sensor integration in industrial robot control. 15th International Workshop on Research and Education in Mechatronics, 2014.
- [13] Steve Cousins in: Welcome to ROS. IEEE Robotics & Automation Magazine, page 13-14, 2010.
- [14] The World Market for Industrial Ethernet: Industrial Ethernet book Issue 69/42, 2012.
- [15] M. W. Abdullah, H. Roth, M. Weyrich, and J. Wahrburg in: An Approach for Peg-in-Hole Assembling using Intuitive Search Algorthim based on Human Behavior and Carried by Sensors Guided Industrial Robot. The 15<sup>th</sup> IFAC/IEEE/IFIP/IFORS Symposium on Information Control Problems in Manufacturing, Ottawa-Canada, May 2015.