An integrated system for supporting remote maintenance services

Paolo Sangregorio*, Alberto Luigi Cologni*, Franklin Caleb Owen† and Fabio Previdi*
*Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione
Università degli Studi di Bergamo 24044 Dalmine (BG)
Email: paolo.sangregorio@unibg.it
†Mechanical Engineering Department, California Polytechnic State University
San Luis Obispo, CA 93407 USA

***Abstract*—In today's marketplace machine producers are shipping their systems all around the world. Maintenance costs are becoming ever greater due to travel costs for sending service personnel to wherever the machines are installed to solve a particular problem. In the majority of cases, a technician solves the problem quickly, so that the travel overhead becomes a large part of the repair cost. This makes remote diagnosis, debugging and repair an obvious advantage. This paper describes a remote maintenance methodology that uses wireless and mobile technology to record and transmit video and machine operational parameters together for remote viewing and analysis. This information is sent to the machine manufacturer's headquarters, where a thorough analysis of it is done by experts there, and a solution is then transmitted to the remote site to correct the problem. The structure of this remote diagnostic system is described, and details about its implementation on a real machine used in manufacturing are given.**

I. INTRODUCTION

Never more than today, companies are competing fiercely to provide high quality at the lowest possible costs for their products and services. Maintenance plays a crucial role in this competition (see Lee et al. (2006), Lu and Sy (2009), Espindola et al. (2010)). In today's globalized marketplace a product can be shipped many miles away, making a maintenance intervention very expensive. It's common that vendors provide maintenance service contracts to their customers and thus it's important to optimize these services by making them quicker and less time consuming. Additionally, the increasing needs of production efficiency and flexibility require ever more complex machines with a widespread adoption of programmable electronics. Such machines require skilled technicians to diagnose and fix faults (see Mori et al. (2008), Ong and Lee (2004), Iung (2003) and Wang et al. (2007)). These skills are rarely available at the remote site, so it becomes necessary for the vendor to intervene and send a technician out to solve the problem. This results in a longer down-time. For expensive machines, often down-time is more expensive than the actual repair, in terms of lost production. Thus, advanced maintenance systems can play a significant role in solving remote problem, even given their strong limitations in the case of human-in-the-loop machines (see Ermidoro et al. (2014)). These machines involve a close cooperation with a human assembly worker. The machine's serves the role of a helper and works closely with the worker to perform a task on an assembly line that requires a human presence. With this type of machine it is often hard to determine remotely whether a problem is the result of a real failure of the machine or results from misuse by the operator. To make this determination, a remote maintenance system needs to be able to consider the operator behavior too. The most suitable solution in this case is to acquire a video of the operator interacting with the machine, while simultaneously recording machine operating data that permits the understanding of the internal machine status. With the explosion in mobile technology, portable computing power is now available to exploit for this application: smartphones and tablets have a wide set of communication interfaces, concentrated computational power and a well known HMI. The aim of this paper is to present an integrated remote maintenance solution, specialized for a semi-automated machine, based on smart devices. This system increases the efficiency of service technicians during remote maintenance operations. The integrated remote maintenance solution is presented, and applied to a real industrial manipulator as a demostration. The paper is organized as follows: in Section II the approach is presented with details about the single components. In Section III its application on a real manipulator is described in detail. Finally, Section IV is dedicated to final assessment and future developments.

978-1-4673-7929-8/15/$31.00 © 2015 IEEE

II. SYSTEM ARCHITECTURE

The system architecture for this remote maintenance system has been designed to be modular. This will allow its application on a number of different machines as well as its retrofit on machines already in service.

The overall architecture is presented in Figure 1 and it is composed by the following components:

- **Machine** The machine that needs to be monitored. This machine has its own control electronics and interfaces;
- **Gateway** Hardware that allows streaming of the machine's operating parameters over a wireless connection. A smart device connects to it for gathering this data;
- **Smart device** The smart device can be any smartphone or tablet which has enough computational power to download data and record video simultaneously. Its main tasks are:

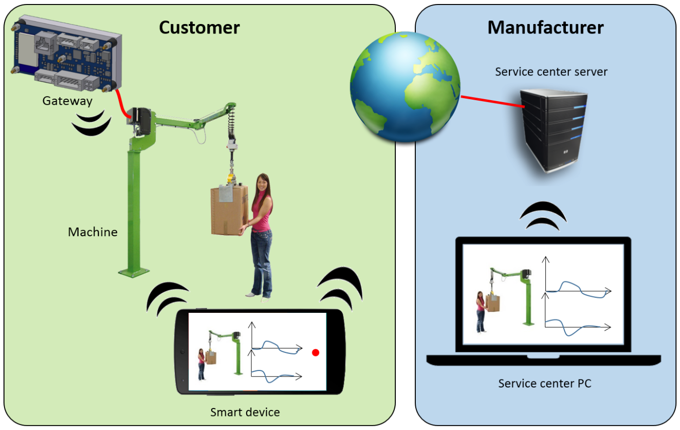

Fig. 1. The integrated remote maintenance service architecture

- ◦ Acquire information from the machine, through the wireless gateway;
- ◦ Record a video of the operator interacting with the machine, and store it together with all the acquired data;
- ◦ Package and route the synchronized video and operating data from the remote customer location to the machine manufacturer's headquarters;
- ◦ Serve as a smart HMI to reconfigure the machine and change its behavior, once a solution has been devised;

- **Smart device app** The application that runs on the smart device for acquiring and sending data;
- **Network infrastructure** A suitable network is required to send the acquired data to the manufacturer. In the most common case, the suggested method is the Internet;
- **Service center server** The server at the machine manufacturer's headquarters that accepts all the data from remote machines stores them in a structured way, accessible by diagnostic technicians;
- **Service center PC** A diagnostic technician will get the recorded data on his/her PC in the service center. This PC will display the data, allowing the technician to analyze and fix the problem;
- **Service center PC app** The application that allows the diagnostic technician to review the acquired data in a synchronized way;

The *machine* is connected to the *gateway* through an available connection on the machine; no specification is given about this connection because it is strictly related to the communication environment on the remote machine. The *gateway* needs to be developed specifically for allowing communication with the machine. It connects to the *smart device* using a wireless (IEEE 802.11b/g) connection. Once the data is acquired, the *device* connects to the manufacturer *service center server* using an Internet connection (through a local wireless access point). The *service center PC* connects to the server using a LAN network connection.

The following is a detailed description of the main components.

A. Machine

The approach presented in this paper is as general as possible, so that it will work on every machine. The only requirement is that the machine have some fieldbus where information and variables are available to be read by a gateway that connects to the same fieldbus.

B. Gateway

The gateway is required to route information coming from the machine fieldbus to a wireless network; therefore two interfaces are needed. The interface for the machine's fieldbus obviously depends on the bus that is available on the machine. The connection for the wireless network instead is based on the standard IEEE 802.11b/g protocol. Other communication protocols have been considered, such as Bluetooth, but that would limit the number of suitable smart devices that can be used for communication. iOS devices in fact have limitations imposed by the Apple on Bluetooth connections that would require a certification of the gateway (for more details on these limitations see Apple MFi Program). Using Wi-Fi, the gateway is compatible with a wide range of available smartphones, guaranteeing a high standard of performance in the communication.

To allow a better development and debugging of the system, the gateway has been provided with additional interfaces. The following wired connections are available:

- Analogical input (0-3.3V)
- Digital input (24V)
- Digital output (24V)
- Serial communication interface (RS-232)

For the wireless connection, a Wi-Fi Roving Networks RN-131G/C module connected via UART has been selected. This module has an integrated CPU which already implements the whole TCP/IP stack. It can work in the classic Wi-Fi configurations that are: *(i) infrastructure*, *(ii) Ad-Hoc* and *(iii) Soft-Ap*. The (i) mode allows the module to connect to an existing network, while the (ii) mode allows to create a point to point network between the module and another device. In the last mode (iii) the module behaves like it was an access point, with a limitation of seven connected clients.

The configuration used in this work is Soft-AP: with this configuration, each gateway exposes a network where the smart device can connect to debug the machine.

C. Smart device app

The device used for this application is an Android device, thus the application is written in Java. The interaction diagram is depicted in Figure 2.

The main application sections are:

1) **Main menu** The main screen of the application where the user can select the operation to run;

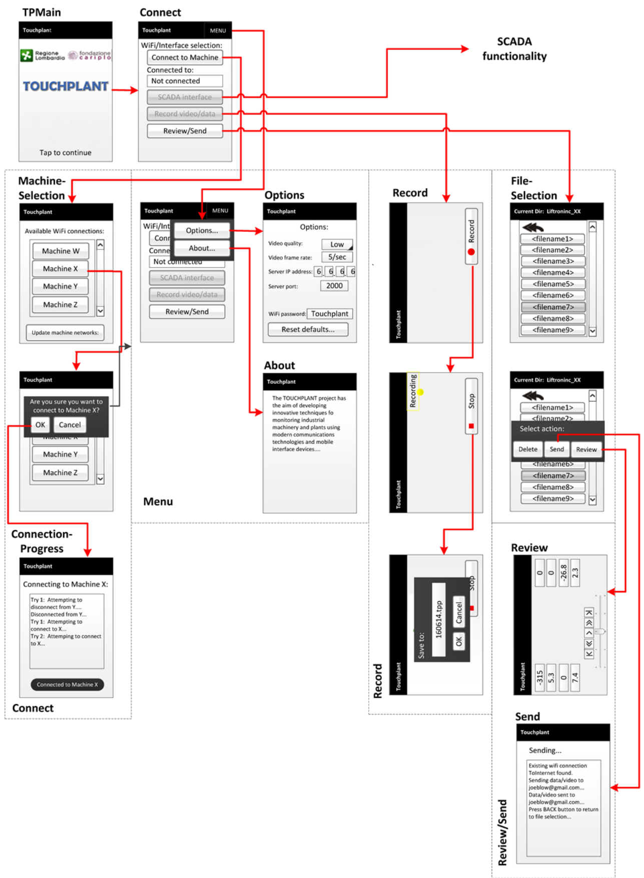

Fig. 2. The application structure

2) **Machine selection** From this screen, the user can select which machine he/she wants to connect to. The list of machines is generated by scanning wireless Soft-AP gateways and listing the devices found. Each wireless SSID is called with the serial number of the machine. Once the user selects the device, the connection is made and maintenance operations are enabled;

3) **Record** This section allows the video capture of the operator using the machine together with the synchronized capture of data coming from the machine. The user records the operator using the machine. When the operation is finished, the user stops the recording and stores it under a given name;

4) **Review and Send** This section allows the user to review the recorded video and data. The video is shown fullscreen on the smart device with a data overlay that shows the values of the recorded variables synchronized with the video. The user can change which variables to display and can upload the package to the service center server to allow a more detailed review by a diagnostic technician;

5) **Options** The options section allow the user to configure the quality of the video and its framerate. This is useful on older devices that do not have enough computational power to record high-quality video while receiving data or when the network connectivity is not reliable for sending large files. Other options that can be configured in this view are the server address for the upload and the wireless password for the gateway;

Figure 3 is an information-flow diagram for the system.

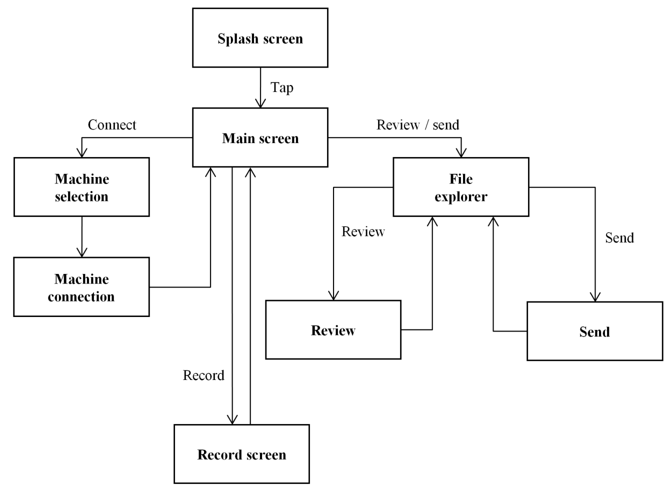

Fig. 3. Information flow for remote maintenance system

D. Service center server

The server that resides in the service center collects all the recordings coming from smart devices into a centralized database. This server has a web interface so that the operator can browse all the installed machines for every customer, review information related to the software version and components for a machine and have a list of the recordings available for a specific machine. The server has been developed in PHP, and the communication with the smart device uses the JSON format.

E. Service center PC App

In order to diagnose, debug and maintain a remote machine, a PC application was developed to review and analyze the transmitted information. Maintenance services usually work with standard PCs, so this program was written for use in the Windows environment.

The application consists of a simple one-form software (see Figure 4): starting from the menu it is possible to open the received files (using a local path or a network URL. Once the file is open it is possible to watch the acquired video while the acquired signals are plotted vs. time.

The main features of the diagnostic program are:

- manage the video stream as a typical video player (play, pause, forward, rewind, stop, etc.);
- define and change which signals are plotted on the four available axes, with a maximum of 6 signals per plot;

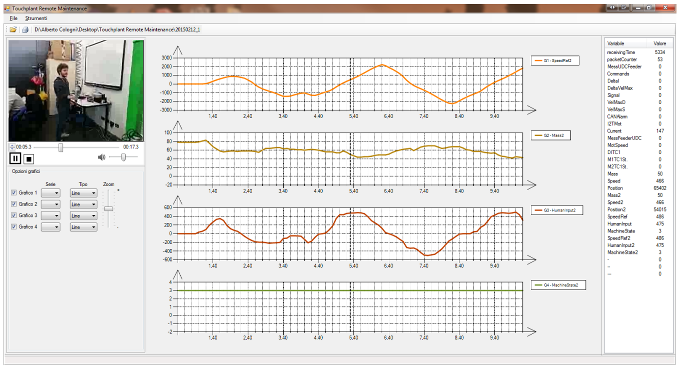

Fig. 4. Screen-shot of the desktop application

- move the video display and use drag and drop to change the signals plotted;
- verify all the acquired signals.

In order to customize the acquired signals for each machine, every time the user loads a file, a JSON configuration file is loaded from the service center server. With this arrangement, it is also possible to change the number of acquired signals and assign, for each one, a meaningful name.

III. Application to an Industrial Manipulator

This system has been implemented on a small overhead crane used in the assembly of automobiles and airplanes. The crane automatically compensates for the weight of the object to be mounted. Thus a worker can use it to install engines in automobiles single-handedly. This manipulator has a vertical mast and a horizontal, hinged arm. At the top of the mast a brushed motor is mounted. It has a wire rope that leads down to an end-effector, which is a specialized jig used to grasp the part to be mounted. (See see Previdi et al. (2010) and Previdi et al. (2012)). This manipulator works closely with a human assembly worker, and, for this reason, debugging of assembly problems requires recording the worker/machine interaction during the assembly process. Thus this system is perfect for validating the application of this methodology.

The architecture of the machine is composed of three electronic boards that communicate via a CAN backbone (the power supply board, the control board and the motor control board).

The gateway for reading information from this machine has been equipped with a CAN interface for connection to this backbone. With a CAN connection, the gateway can gather all the data streaming from the machine and intercept all the messages running between the boards.

The relevant operating data for remote maintenance logged for the manipulator are:

- **Motor speed** to detect problems related to the motion system, such as the grasping system or the lift cables;
- **Motor current and voltage** to determine if the electrical components are working properly. These signals in combination with the motor speed can be used to diagnose the motor's condition;
- **Alarms and messages** to understand when an error is triggered. This then can be correlated to the other signals and to user intervention;
- **Operator commands** to have information about what the worker is doing and when he/she is interacting with the machine. This allows the diagnosing technician to understand if a problem is related to some specific combination of the inputs or if it is connected to a specific usage by the operator;

IV. CONCLUSIONS AND FUTURE DEVLEOPMENTS

In this paper an integrated system for supporting remote maintenance services is presented. The main advantage of this methodology is the the fusion of different information streams: it allows a remote diagnostic technician to observe the operator using the machine while reviewing data from the machine. This innovative solution extends the current typical remote maintenance services in the field of semi-automatic. It has been tested on a real industrial assembly machine.

In the near future the Windows-based diagnostic will be installed on the server of the machine producer, enabling us to test the entire system.

Acknowledgment

This research activity has been carried out in the context of the *Touchplant project*, a project funded with the joint support of *Regione Lombardia* and *Fondazione Cariplo*.

References

Ermidoro, M., Formentin, S., Cologni, A.L., Previdi, F., and Savaresi, S.M. (2014). On time-optimal anti-sway controller design for bridge cranes. In *American Control Conference (ACC), 2014*, 2809–2814. IEEE.

Espindola, Pereira, D.B., Henriques, C.E., Botelho, R.V., and S., S. (2010). Using mixed reality in the visualization of maintenance processes. *Proceedings of the 1st IFAC Workshop on Advanced Maintenance Engineering, Services and Technology*, 30–35.

Iung, B. (2003). From remote maintenance to mas-based e-maintenance of an industrial process. *Journal of Intelligent Manufacturing*, 14(1), 59–82.

Lee, J., Ni, J., Djurdjanovic, D., Qiu, H., and Liao, H. (2006). Intelligent prognostics tools and e-maintenance. *Computers in industry*, 57(6), 476–489.

Lu, K.Y. and Sy, C.C. (2009). A real-time decision-making of maintenance using fuzzy agent. *Expert Systems with Applications*, 36(2), 2691–2698.

Mori, M., Fujishima, M., Komatsu, M., Zhao, B., and Liu, Y. (2008). Development of remote monitoring and maintenance system for machine tools. *CIRP Annals-Manufacturing Technology*, 57(1), 433–436.

Ong, M.H. and Lee, S.M. (2004). Evaluating the use of multimedia tool in remote maintenance of production machinery in the automotive sector. In *Robotics, Automation and Mechatronics, 2004 IEEE Conference on*, volume 2, 724–728. IEEE.

Previdi, F., Fico, F., Belloli, D., Savaresi, S., Pesenti, I., and Spelta, C. (2010). Virtual reference feedback tuning (vrft) of velocity controller in self-balancing industrial manual manipulators. In *American Control Conference (ACC), 2010*, 1956–1961. IEEE.

Previdi, F., Fico, F., Savaresi, S.M., Belloli, D., and Pesenti, I. (2012). Direct design of a velocity controller and load disturbance estimation for a self-balancing industrial manual manipulator. *Mechatronics*, 22(8), 1177–1186.

Wang, W., Peter, W.T., and Lee, J. (2007). Remote machine maintenance system through internet and mobile communication. *The International Journal of Advanced Manufacturing Technology*, 31(7-8), 783–789.