A Distributed Architecture for Teleoperation over the Internet with Application to the Remote Control of an Inverted Pendulum

Denis Gillet, Christophe Salzmann, and Pierre Huguenin

Swiss Federal Institute of Technology, CH – 1015 Lausanne, Switzerland

Abstract. This paper discusses challenges in enabling teleoperation over the Internet for the specific case of real systems which exhibit fast dynamics. A distributed client-server architecture is proposed to provide the necessary level of interactivity for supervision and tuning, without compromising the essential control tasks. Enabling features include real-time simulation and augmented-reality visualization to enhance the user perception of the distant ongoing operations. The proposed approach is illustrated through the remote control of an inverted pendulum.

1 Introduction

To operate critical facilities efficiently, the process and manufacturing industries increasingly request global monitoring, as well as for sustainable support and immediate service from their suppliers. Among these services, telemaintenance is the most requested. As a consequence, new industrial network implementations are increasingly based on the Internet protocol to ensure a global and continuous access. However, these new capabilities are not yet exploited extensively due to the lack of predictability of the transmission delay inherent to the Internet and the lack of standardized deployment methodologies.

To contribute to the fulfilment of the market expectations related with emerging real-time services, an architecture is proposed to enable the remote supervision and remote tuning of distributed controlled systems, including their embedded or external control devices. These applications are among the most critical ones in industrial environments, since they handle the functions related with the dynamic behavior of real equipment.

The industrial equipment previously mentioned is just one example of physical systems that can be teleoperated. Didactic setups available in instructional laboratories can also be teleoperated in a flexible learning context to provide students with the necessary resources for practical experimentation, as described in [1], [2], and [3].

The architecture implemented for the teleoperation of both industrial equipment and didactic setups relies on a computer-based client-server scheme with Internet as a communication channel.

The remote operators are provided with client software to observe and to pilot the physical system via the Internet. The on-site server is the computer

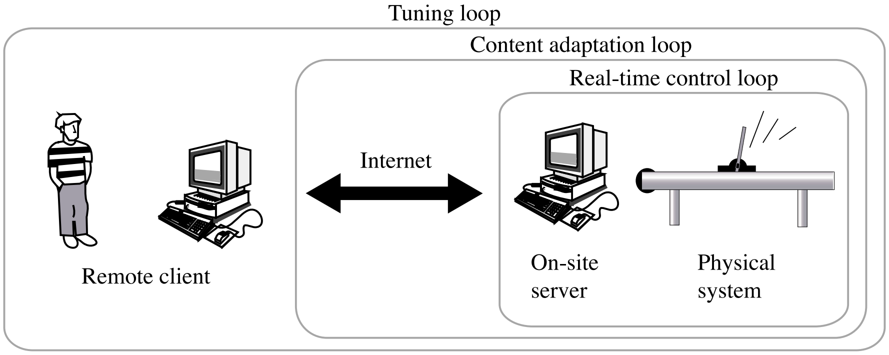


Fig. 1. Client-server Control Loops

that stands for the communication and operation interface with the real system and its instrumentation. To overcome the drawback of the operators not being present at the equipment location, additional devices have to be integrated, such as cameras to broadcast video views of the system and of its environment.

With such an architecture for teleoperation over the Internet, there are three imbricated control loops to implement (Fig. 1). The first one, dedicated to the real-time control of the real system, has to be implemented at the server side. A common and dangerous mistake is to close this loop across the Internet. Even if such an approach is challenging from a robust control point-of-view, it is unacceptable from an industrial point-of-view. Just think about the effect of the loss of a single or numerous measurement samples for stability and safety. The second one is the control loop designed to adapt the transmitted content to the available bandwidth in order to guarantee a sufficient quality of service. This is merely to ensure the reception of the information essential to carry out the desired operations according to their respective priority. The last loop is closed by the user for setting the conditions of operation or for tuning the distant real-time control loop. This is the loop the most sensitive to the transmission delay, because wrong actions can be performed by operators if the reactions of the system are not noticed fast enough for a valid interpretation.

In Sect. 2, the inverted pendulum and its local control are introduced. Various ways to handle the transmission delay that occurs when switching from local to remote control are described in Sect. 3. Then, details about the client-server architecture and the necessary features needed at the operator side to provide a sufficient quality of service are given in Sects. 4 and 5, respectively. Finally, concluding remarks and perspectives are expressed in Sect. 6.

2 The Inverted Pendulum and its Local Control

The pendulum described in this section is an inverted pendulum with two degrees of freedom. It emulates, in two dimensions, a juggler trying to keep a broomstick in equilibrium on his fingertip. The control objective is to simultaneously keep the pendulum stick in the upright position and the supporting cart at the center of the rail, using a single actuator.

The inverted pendulum shown in Fig. 2 is made up of a 4-meter horizontal rail mounted on top of a case that contains both the electronics and the power supply. A cart, moved by an electric motor by means of a metallic belt, can slide along this rail. The rotational axis of the pendulum stick is on this cart. No actuator exists for this axle. The only way to change the stick position is to accelerate the cart. Hardware security switches that turn the power off can be found at each end of the rail.

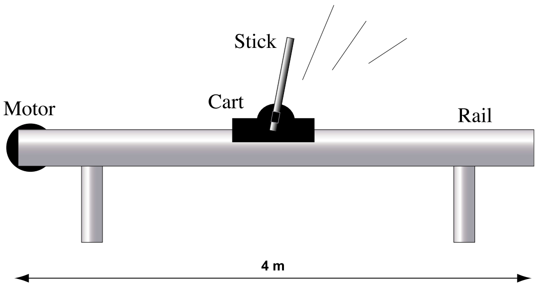


Fig. 2. The inverted pendulum

This laboratory-scale system can be controlled by any type of computer equipped with a National Instruments PCI-1200 family DAQ board. This portability feature turns the setup into an ideal support for educational demonstrations or trade shows.

The underlying state-space multivariable control principle aims at minimizing the energy spent by the controller to keep the stick raised and the cart at the center of the rail. The implementation of the corresponding linear quadratic regulator (LQR) relies on both angular and longitudinal position measurements, as well as the respective velocity estimates.

A representative physical model of the system is required for the controller design, as well as a good knowledge of its physical parameters. It is worth developing a good model because it leads to a much more accurate control than empirical methods. Moreover, the model can also be used for real-time simulation as needed for advanced teleoperation.

The swing-up algorithm implemented to move the stick from the lower stable vertical position to the upper unstable one is a very simple bang bang procedure: a constant voltage value is chosen to be applied to the motor and its sign is worked out adequately depending on the stick position in order to induce and amplify the stick oscillations.

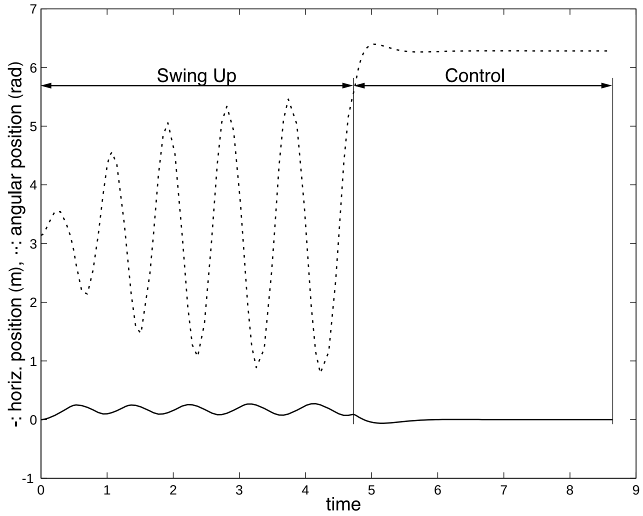


Fig. 3. Swing-up control signal

The swing-up procedure chosen is quasi-optimal from a time point-of-view. Its sequence and its intuitive justification are given as follows. First, the cart accelerates and the stick swings up due to its inertia. Then, the cart reaches its maximal velocity and the stick swings down by gravity. Once the stick crosses the lower point of its trajectory, it has accumulated the maximal possible amount of kinetic energy. Thus, it is the optimal time to inverse the voltage polarity (for a more accurate solution, the dynamics of the electrical drive has to be taken into account). This sequence has to be repeated until the stick reaches a pre-defined interval around the vertical position (Fig. 3). In this restricted control sector, the state-space controller can be activated to stabilize the pendulum.

The swing-up tuning parameters are the constant voltage value and the control sector size. The voltage level is a trade-off between the horizontal range of the cart displacement (the boundaries of the rail have not to be reached) and the angular velocity of the stick. The control sector size depends on the damping the controller can introduce and its robustness to the system nonlinearities, such as the ones due to the angular position of the stick.

The controller parameters that can be modified by local operators, in addition to the swing-up tuning ones, are the sampling period and the four state feedback gains. The same settings have to be available when remote operations are considered. Enabling fast remote prototyping of the controller requires that the effect of any change in these parameter values can be noticed as soon as possible by the operators, even in the case of a significant transmission delay.

3 Transmission Delay Handling

The varying transmission delay (approximately half the Round Trip Time) in Internet communication is the most important problem to handle when implementing teleoperation solutions. Depending on how the information packets transmitted are routed and how many routers are crossed, the delay can vary from a few milliseconds to hundreds (Fig. 4).

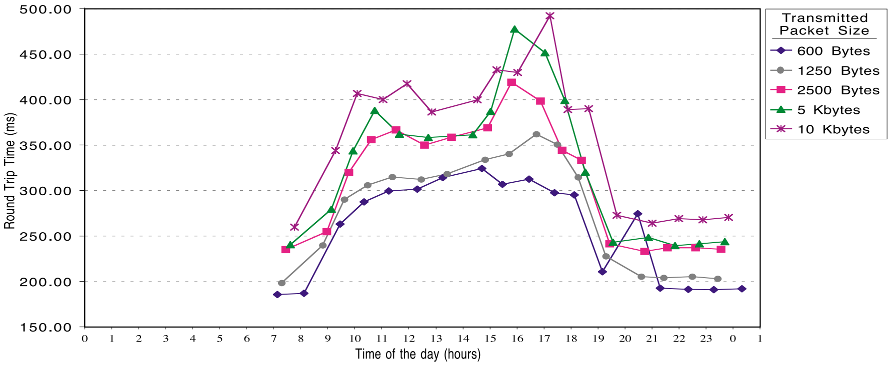


Fig. 4. Example of the transmission delay variation in Internet communication

The main difficulty for the operator occurs if the time constants of the remote system are of the same order of magnitude or faster than the time delay. Unfortunately, this condition appears to be true for most of the mechatronics systems.

The solution to partially alleviating this difficulty relies on an intelligent characterization and management of the data streams transmitted, such as the measurement stream, the video stream, the tuning stream and the coordination stream [4]. Such schemes aim to use optimally the available bandwidth. Since the transmission delay is incompressible, the additional solutions only aim to reduce its annoying effect on the sensorimotor behavior of the operators.

There are two classes of additional solutions, the client-side and the server-side ones. The client-side solutions mainly rely on predicting the remote ongoing operations to help the operator in anticipating the effect of its actions. The server-side solutions rely on intelligent context analysis carried out automatically by the server to feed back synthetic and composite results that can reduce the reaction time of the operators, once such a high level of information becomes available to them.

In automatic control, where the dynamic models are usually known and where the context of operation is always predefined, client-side solutions are more interesting. In mobile robotics, where the context of operation may evolve significantly and where more intelligence is embedded in the on-site system, server-side solutions can be more convenient. The teleoperation of an inverted pendulum clearly belongs to the first class.

4 Client–Server Architecture

The teleoperation server software is made of two distinctive parts, the user interface and the part handling the real-time operations, including the interface with the outside world. For local experimentation these two parts are located on the same computer.

The main concept in turning a locally-controlled setup into a remotely-controlled one consists of moving the user interface of the monitoring and control software away from the physical system. Two distinctive parts result: the remote client and the on-site server.

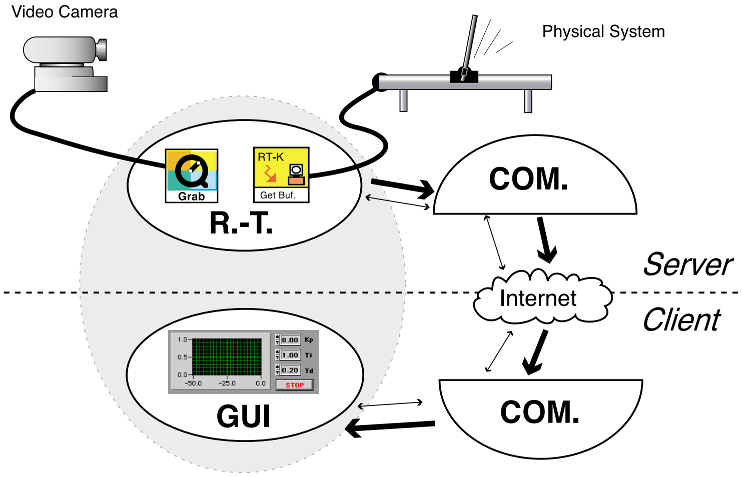


Fig. 5. Software components at the client and server sides

The remote client is a computer equipped with the functionalities necessary to observe and to act on the physical system. The client application is used to generate excitation signals and observe corresponding responses. The main objective of such an interface is to provide the user with a general view of the equipment, and to allow full control of the operations.

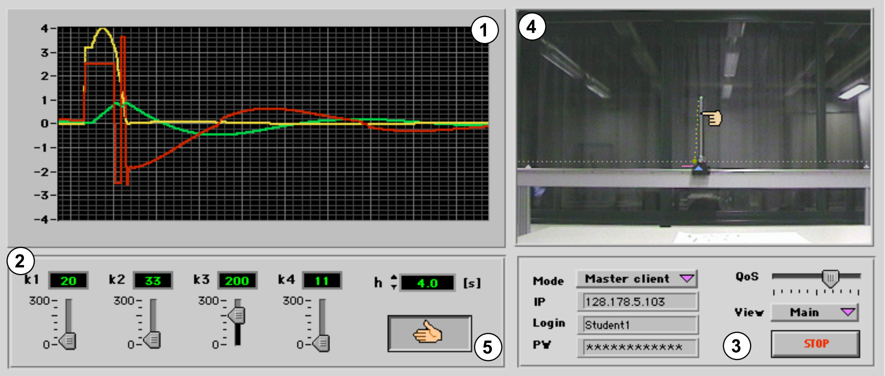
The on-site server is the computer located near the real system and equipped with the hardware interface to the sensors and actuators. The video camera and microphone can be seen as sensors. The server application receives the client commands and transmits them to the equipment. It also returns its physical and operational states to the client, including an image.

Three components (Fig. 5) are necessary to build the client and the server applications [4]. The client and the server application can be designed by adding a communication component to the two components used in local implementation: the real-time and the GUI ones. The client application is made up of the GUI and the communication components. The server application is made up of the real-time and the communication components. The server may require a basic user interface for supervision of the ongoing operations. The communication component allows the client and server applications to exchange information with other computers distributed in different geographical locations. This module also takes care of security issues regarding network management. For example, it prevents unauthorized access and schedules login to avoid conflicts.

By isolating carefully these modules in the development process, it is easy to port a local solution to a remote one, or port the remote solution to different physical systems.

5 Operator–Side Features

The user interface at the client side typically features an area showing a scope that displays all the relevant signals (see label 1 in Fig. 6), and also includes areas with dialog boxes that permit setting the controller parameters and specifying the access rights of the users (labels 2 and 3 in Fig. 6). In more advanced implementations the control algorithm may also be presented as a user option [5].



k1 20

k2 33

k3 200

k4 11

h 4.0 [s]

Mode Master client

IP 128.178.5.103

Login Student1

PW ************

View Main

Fig. 6. User Interface of the remote client software

To enhance the user perception, a virtual representation of key parts of the real system is superimposed on the video image (see the dotted lines in area 4 in Fig. 6). Such a composite view is called augmented reality. Useful information can be added, such as the reference signal for the controller (triangle drawn on the target track position), the force applied on the cart (length of the horizontal line drawn at the base of the cart), or even a virtual hand used to apply a disturbance on the pendulum (see button 5 and the hand displayed in area 4 which appears when the button is pressed). Different views of the distant equipment can be selected. When the bandwidth available is not large enough to ensure a sufficiently high video throughput, the display is limited to the virtual representation which in turn can be animated using real data or, in the worst case, using samples generated by a real-time simulator. In the pendulum example, the animation of the virtual image can be achieved from knowledge of only two variables, namely, the angular and longitudinal positions of the pendulum.

The implementation of a real-time simulator in the client software permits the user to carry out off-line simulations for pre-validation purposes. In addition, the simulator can be used to provide synthetic data that can be posted on the user interface when packet losses occur during transmission. This gives continuity to the display, and provides the user with a sense of real-time behavior. In these cases, it is important to post an indicator that announces that the data shown is provided by a model due to the absence of reliable network data.

6 Concluding Remarks and Perspectives

The teleoperation of controlled systems that exhibit fast dynamic behavior is a challenging application when it is carried out over the Internet. It constitutes an innovative real-time service that brings new opportunities for both industry and academia. In industry it enables the monitoring and the maintenance of critical facilities. In academia it allows access to and sharing of laboratory resources for collaborative research and flexible education.

The client-server architecture implemented that features three hierarchical control loops poses the basis for efficient operations that require a high degree of interaction. It constitutes a framework for the development of adaptive schemes that cope with varying bandwidth and transmission delay, as well as quality of service constraints.

The inverted pendulum is a convenient introductory example that permits underlining the principal requirements related with the teleoperation of controlled mecatronic systems. The most important ones are: the necessity to ensure a cadenced stream of information to reproduce the dynamic behavior of the on-site system, the need for hybrid representation to enhance the operator perception of the ongoing operations, and the possibility to remotely perturb the systems for validation purposes.

References

1. Gillet D., Salzmann Ch., Longchamp R., Bonvin D. (1997) Telepresence: An Opportunity to Develop Practical Experimentation in Automatic Control Education. In: European Control Conference, Brussels, Belgium
2. Bhandari A., Shor M.H. (1998) Access to an Instructional Control Laboratory Experiment through the World Wide Web. In: American Control Conference, Philadelphia, USA
3. Overstreet J.W., Tzes A. (1999) Internet-Based Client/Server Virtual Instruments Designs for Real-Time Remote-Access Control Engineering Laboratory. In: American Control Conference, San Diego, USA, 1472–1476
4. Salzmann Ch., Latchman H.A., Gillet D.; Crisalle O.D. (1998) Requirements for Real-Time Experimentation over the Internet. In: International Conference on Engineering Education, Rio de Janeiro, Brazil
5. Piguet Y.,Gillet D. (1999) Java-Based Remote Experimentation for Control Algorithms Prototyping. In: American Control Conference, San Diego, USA, 1465–1469