Design Simulation of a Web-Based Supervisory Control System

Yeuk F. Ho and Lawrence W. Stark
Telerobotics and Neurology Unit,
University of California, Berkeley,
Berkeley, CA 94720
Email: et@rov.berkeley.edu & stark@pupil.berkeley.edu

Keywords: robust; model-based image processing; redundancy; JAVA; distributed system.

ABSTRACT

A Web-based tele-operating system has been simulated in our lab with model-based supervisory control and model-based, top-down image processing for robot pose recovery. These modules have also been integrated so as to achieve robust performance with reduced human attendance.

Web-based network connection design and 3D display are based on currently available web technologies. The most difficult design aspect was to achieve smooth integration between web technologies and the hardware physical entities. Constraints from the remote environment were carefully considered, since the system will be used to build a lava-diverting dam at the base of an active volcano in Japan.

INTRODUCTION

Background. In recent years, since the pioneering paper of Ferrell and Sheridan (1967) tele-operation has been a subject of intense research due to advances in the field of computer science (hardware and software platforms), affordable sensory technology (GPS tracking systems, machine vision) and availability of uniform and robust communication platforms; the Internet is the outstanding example. Applications of such technology include exploration and inspection of hard-to-reach or hostile environments such as other planets, ocean depths and disabled nuclear power plants.

Motivation. Our lab is involved in the design of a tele-operation system. The purpose of the system is to prepare a foundation at the foot of a volcanic mountain in Japan for construction of a dam for diverting further lava flows safely into the sea (Figure 1, lower). Due to the presence of poisonous gas in the environment, tele-operation was the first choice re methodology for the task. Currently, the system is tele-operated, but with direct manual control and video camera visual displays. Daily tasks include removal of volcanic dirt by bulldozers and dump trucks, preparation of dam foundation by rollers and compaction vibrators and actual construction of the dam. The system under design has to fulfill several requirements. The first is high level of autonomy to reduce the day-to-day human attendance involved. A second requirement is a high level of robustness in the remote feedback control functions. The third requirement is flexibility in deployment of the distributed system; a fourth requirement is flexibility while continually upgrading the system.

Direct vs Model-Based Tele-Operation

Most current tele-operating systems are basic extensions of direct manual control (Beczy, 1980). The obvious drawbacks include reduced stability of the control loops due to long delays in the loops as well as impedance mismatches in the master-slave environments. Proper operator training and reconditioning of signals between the remote and local sites has been shown to reduce these shortcomings (Kim et al. 1987; Buttolo and Hannaford 1995)

Another approach is model-based or supervisory control tele-operation in which the two environments are de-coupled (Ferrell and Sheridan 1967; Stark et al. 1987; Yoerger and Slotline 1987; Moray et al. 1989; Sheridan 1992; Blackmon and Stark
Local control station and remote work site environments interact not with each other, but only to computer models functioning at their respective sites (Figure 1, upper). Synchronization, with low communication costs, may be thus achieved by rapid exchange of only model parameters between the local and remote models. This method avoids problems of direct manual feedback control by shifting moment-to-moment sensing and feedback control responsibility to remote environments. Local control operators, acting in supervisory manner, need only interfere infrequently and indirectly with the remote operation and with only a finite set of high level commands.

A simple example, to highlight the differences between the two modes of control is to consider how to move a remote vehicle from point A to B. Direct control requires the operator to send steering and acceleration signals to the remote vehicle and so to drive the vehicle to its destination. In model-based control operator specifies the destination and the remote closed-loop feedback system will operate independently to drive the vehicle to that point. This later mode provides a high degree of autonomy and therefore was chosen as our primary design feature to meet our requirements.

**Sensing Means**

To provide for the long operation hours demanded of the remote control system, robustness in control law and feedback sensors was essential. We addressed this need of robust feedback with a model-based visual feedback system. Cameras are 2D sensors and therefore provide a rich raw data input with sufficient redundancy for robust analysis. The usual problems in visual sensors are complexity of algorithms and the time and computational power needed to analyze the raw data. By using model-based top-down image processing, IP, (Stark et al. 1988; Zelnio 1991; Wong and Li 1992; Stark 1993; Beveridge and Riseman 1994; Yong and Stark 1995; Ho and Stark 1997), we create a domain of prior calibrated models to define the set of recognizable objects. The models further restrict the methods and sequences for analyzing the images and for verifying the existence of a recognizable calibrated model in the images. In essence, model-based top-down IP differs from the traditional bottom-up methods by shifting from search and analysis for any objects possibly existent in an image to checking for and confirming the presence of a limited set of objects in known loci. With this restriction to top-down processing, it clearly becomes the method of choice as a real-time feedback sensing mechanism (Figure 1, upper).

In addition, the global positioning system, GPS, for the on-board tracking system provides redundant feedback signals for robust feedback control. This enrichment, together with internal sensors on the vehicle joints and engine gauges and short range emergency "soft bumpers" introduced the advantages and problems of sensor fusion (beyond the scope of the present paper).

**JAVA / C++ Mixed Environment**

To achieve high flexibility of system deployment and incremental upgrade, we choose to use JAVA as our preferred platform for software development. JAVA has the obvious advantage of cross-platform compatibility; tight integration with the Internet makes it a good candidate for the network framework we are designing. Another advantage of JAVA is its remote method invocation technology, RMI, which transcends the traditional client/server based framework. Client/Server based network frameworks provide the software designer with a set of low level communication tools for byte-level data exchanges. It however burdens the designer with the need to create her communication protocol over the channel, to handle execution of algorithms on the local machine on behalf of a remote connection request, and to deal with all possible errors that may occur. RMI unifies the network communication into object levels so that functionalities of remote objects (JAVA objects created in another process or machine) can be invoked locally with data exchanges, encapsulated as ‘serializable’ JAVA objects. That is, with the exception of an additional network-related error, local objects and remote objects behave very similarly, allowing designers to focus on system design. The only software components that employ the C++ language serve low-level feedback control laws and the IP algorithm units running on the remote vehicle embedded system. This choice is based on ability of C++ to provide both the necessary hardware interfacing as well as the computation efficiency of native coding. We are now in the process of evaluating DCOM/ActiveX (beyond the scope of the present paper).

**System Components**

The major system components are the i) remote vehicles, ii) IP procedures, iii) control laws, iv) supervisory interface, v) world model, vi) path planner, and vii) data logger (Figure 2); following are detailed description of each.

**Remote Vehicles.** The hardware under remote control includes modified construction vehicles such as bulldozers, dump trucks and rollers. They have been equipped with external sensors such as GPS tracking systems and internal sensors such as encoders at manipulator joints, engine status gauges and fuel level sensors. Visual feedback is provided through stereo cameras mounted at the drivers’ seats for drivers’ points-of-view of the remote environment. Additional supporting vehicles included non-vibrating camera vehicles for third person points-of-view for visual feedback, and communication relay vehicles. Camera signals are digitized with CMOS-based capturing devices. Remote control signals to the hardware are
provided through a Programmable Logic Array, PLA, interfacing with the controlling software through serial ports.

**Image Processing and Control Law Components.** The Window CE, WCE, embedded system was chosen as the computation platform on the remote vehicles for system support of network communication and for ease of software development. As mentioned, the low-level control laws for vehicle operation and for the IP units will operate on WCE as C++ modules. The control laws provide path-tracking functionality for model-based supervisory control, while model-based IP, coupled with the GPS system provides redundant positional feedback for control. Communication between the remote vehicle and the outside world will be a socket-based communication system. More flexibility may be obtained by using a full x86 system and again this is being evaluated by our group (beyond the scope of the present paper).

**Supervisory Control Interface Component.** A JAVA-based operator interface is provided for supervisory control center for remote operations (Figure 3). The General Supervisory Interface includes special input hardware such as joysticks, with or without force-feedback, or spaceballs (Our simulation has only keyboard and mouse interfaces for cross-platform compliance). The interface is used to enter operating parameters, for example, boundaries of operation areas, for the path planner component to enable detailed execution sequence planning (Figure 4). Direct manual control input remains for backup operations.

The Graphical Display is the main information feedback for the operator. It includes simple graphics provided by the Abstract Window Toolkit, AWT, for system status, hardware conditions, and progress status. The kinematic model of the remote environment uses 3D computer graphics. Under JAVA, this is provided either by Liquid Reality, programmatic interface to Virtual Reality Markup Language engine (VRML) or by JAVA3D.

**World Model Component.** This component serves as the JAVA object for task coordination. It maintains the states of all components and serves as the information center for all the other components in coordinating their execution of diverse functions. It also relays planned tasks submitted by operators and commands the remote vehicles’ motion sequences. This is the only object connecting to the remote vehicles’ C++ objects through sockets. For stability purposes, it is possibility to have redundant world model objects, since it is so essential a component for total system operation (Figure 1). For example, one residing at the control center enables fairly typical control between vehicle sensing and vehicle control; another also residing in the control center provides for display and further distance communication. Thus communication between these two world models can be limited to parametric data flow. Note that the short distance between the control center and the vehicle (200 meters) permits us to keep the “so called remote world model in the control center”. Of course with the Martian-Rover, the remote model would reside on the vehicle.

**Path Planner Component.** This component serves as the motion planning algorithm for task execution. The operator submits task parameters from the Operator Interface and the resultant planned execution sequence is graphically displayed for operator approval, before being sent to the World Model for execution.

**Data Logging Component.** This component utilizes JAVA database connectivity, JDBC connections, to the sequential query language, SQL, database for logging operation progress and error events. It mainly interacts with the World Model for data archiving.

**SIMULATION**

**Purpose.** In parallel with the system design stage, new studies were carried out or old ones reviewed to verify the feasibility of the components. The main concepts to be verified included:---

a. *Efficiency of the HMI*, Human Machine Interface using Model-Based Supervisory Control. Two main sub-topics were user adaptation to intermittent control and command strategies and the effectiveness of a synthetic graphical display of the remote environment for operator visual feedback (Figure 3).

b. *Robustness of the Low Level Control Law* in tracking operations under uncertainty. Two main types of uncertainty are unknowns in the mechanics of the system (terrain variation, unmodelled vehicle dynamics, plant noise and noise in feedback control) and in measurement processes (GPS measurement uncertainty, visual feedback noise such as image noise, unmodelled objects in scenery, and lighting variation (Figure 5).

c. *Robustness in communication mechanism.* The TCP/IP protocol, at best, does not guarantee timely delivery and may even drop packets when traffic is heavy. In addition, our choice of using standard HTTP protocol over TCP/IP makes the communication between the world model and the remote vehicle C++ components discrete and intermittent in nature. Therefore, we need to investigate further the impact of our various choices re communication system design and performance.
RESULTS

Feasibility of Supervisory Control. Previous studies examined human performance in tele-operation with direct manual control or supervisory control and with or without virtual display (Stark et al 1987; Stark et al 1988; Kim, Takeda and Stark 1988b; Bejczy, Kim and Venema 1990). They confirmed the feasibility of supervisory control scheme with a virtual display as the main visual feedback mechanism. The main conclusion was that while the initial performance under supervisory control was inferior to that of direct manual control, training eventually brings the performances to comparable levels. In addition, the virtual display helped the operators to perceive the remote environment by providing flexible, non-feasible synthetic viewpoints of the remote environment, as compare to fixed and realizable viewpoints displayed with direct camera feedback (Figure 3).

Robust Remote Feedback Control Loop. To verify the robustness of the remote feedback control loop, a simulation environment (Figure 4) using OpenGL to generate the simulated video inputs to the IP algorithm, was created (Ho and Stark 1997). Gaussian white noise was injected both in the plant dynamics (multiplicative noise) and in the simulated images (accumulative gray-level noise) to model unknown approximations, perturbations, and errors in both control plant model and in robot pose tracking. Gaussian white noise was used in the simulation because the diversity and complexity of all the kinds of errors involved made more precise error modeling unnecessary complicated. Different measures were derived to estimate the robustness of the image processing algorithm and the stability of the path-tracking control law. Our results (Figure 5, in part), studied and verified stability of the IP algorithm under moderate noises level. When the IP system did fail, the GPS tracking system and the internal encoders could be used to provide secondary partial feedback to sustain the feedback loop and, when necessary, to re-initialize the model on which the IP algorithm relies. Alternatively, a reconstruction of the 3D models from 2D images could be employed (Yong and Stark 1995). The IP result was used as the primary sensor output despite the complexity of IP involved because unlike GPS, it can be used to track remote objects (for example, other humans in the working environment) and because its failure may also indicates events not predicted by the model that should alert the operator (for example, video camera failure).

Robustness in Communication. We have two modes of communication: i) the RMI method between general JAVA components, and ii) the HTTP protocol joining the world model, the remote low-level control law, and the IP algorithm; both written in C++. The RMI method was chosen at this stage, but further evaluation was required re the HTTP protocol. A simple web client for control interface and a web server (Microsoft IIS) was connected to a simulation of the remote environment as described in the above through an ISAPI web server extension routine. Direct manual control was first implemented and pose of the robot was numerical displayed on the web client interface. An important stage in the development was to design and construct a browser based client to connect with the web server and to implement the path planner JAVA component. Performance of this reduced web-based supervisory control system was then evaluated against network delay to verify the flexibility and upgradability of our system design.

SUMMARY

This web-based tele-operating system was motivated to design a system to control the construction of a lava-diverting dam at the base of an active volcano in Japan. Our laboratory approach rests upon model-based schemes, both for supervisory control and for IP. Because of the desire for practical application, the IP algorithms were chosen to provide robust operational platforms, while at the same time, reducing demands on the human operator for routine tasks so as let her focus on high level path planning and circumventing emergency situations. Simulation studies were performed to document the feasibility and robustness of the model-based supervisory control scheme and the model-based IP algorithms. Additional studies of system integration within web technologies are being performed at the present time.

ACKNOWLEDGEMENT

We wish to acknowledge cooperation with Fujita Research Inc., Encino, California with Drs. Ken Kawamura and Hideyoshi, Masuda, their technical monitors, who are working with us on an extended project. We thank laboratory colleagues Drs. Ted Blackmon, Frank Tendick, Yong Yu, An Nguyen and Hiroshi Oda for discussions; also Dr. Kamran Siminou, Neuro-Optics, and Drs. Stephen Ellis and Michael Sims, scientific monitors for research grants with NASA-Ames Research Center, for additional support.

REFERENCES

JOURNAL


CONFERENCE PROCEEDINGS

CONFERENCE PROCEEDINGS


JOURNAL


JOURNAL


CONFERENCE PROCEEDINGS


JOURNAL


CONFERENCE PROCEEDINGS


CONFERENCE PROCEEDINGS


BOOK


JOURNAL


BOOK


JOURNAL


JOURNAL


CONFERENCE PROCEEDINGS


CONFERENCE PROCEEDINGS


CONFERENCE PROCEEDINGS


JOURNAL

Figure 1: **Video Input & World Model**

Image of bulldozers (lower) and the graphical representation of them in the world model (upper) with output of the Path Planner displayed as wire frame image.
Figure 2: Functional Object Connection.

World Model Object is the central location wherein system states and resource availabilities are updated. GPS sensors and lower level IP Objects (operating on video signals, V, from onboard cameras) on camera cars and working vehicles are used to update status of working vehicles. Sensor Fusion updates the World Model. Human Operator at Supervisory Control views Display and approves the Path Planner; then Control Object issues commands to vehicles. Database stores history. Double line represents WEB connections.

Figure 3: Supervisory Control Interface
Visualization of a planned motion sequence from two virtual viewpoints (upper panels). The tele-operator Graphical User Interface (GUI) (lower panel) is an original design using mouse and joystick as its main inputs (Blackmon and Stark 1997). As described in text the present design implements a JAVA applet running inside a web browser with mouse and keyboard.

Figure 4: Diagram of Supervisor Control Scheme.

Operator uses Supervisory Control Interface and Path Planner to plan out actual sequences of movements, each then sent to the Low Level feedback controller for controlling robot plant output. Redundant Feedback is provided both by GPS sensors and by Image Processing Algorithm through sensor fusion. Feedback also updates World Model for system consistency. Only in simulation do the added noises, N1 & N2, and Image Synthesis exist.
Figure 5: Typical measure used to evaluate robustness of the remote control loop.

Average number of visual enhancements, VEs, tracked under different combination of noises (4, 3, 2, 1). Horizontal axis is standard deviation of zero-mean gaussian plant noise as a fraction of instantaneous command velocity. Vertical axis is standard deviation of zero-mean gaussian pixel noise. A minimum of two VEs are required for pose estimation. As shown, the system has a high margin of safety over a large range of noise. In addition, the two noises have independent effects on the system at moderate noise levels (heavy horizontal and vertical dashed lines). As noise is increased, trade-off between tolerance to image noise and to control noise appears (light oblique dash line).