AUTONOMOUS CONTROL ISSUES IN A TELEROBOT

Vincent Hayward

McGill University McGill Research Center for Intelligent Machines
3480 University Street Montreal, Quebec Canada H3A 2A7

ABSTRACT

We discuss the structure of a possible autonomous control structure for a telerobot, and then establish corespondance with concepts familiar in the field of teleoperation: Traded, shared and supervisory control. Systems such as telerobots can be described from various perspectives, leading to various system decompositions which each imposes constraints on the overall design.

1. INTRODUCTION

The stated goal of "telerobots" is to integrate both the autonomous and teleoperated control of robotic tasks. It is expected that such a system will combine the capabilities of an autonomous robot control system and of a teleoperated one. An increased robustness of the system and a reduced need for human intervention are some of the principal advantages expected from that approach.

On the other hand, combining autonomous and teleoperated control of manipulators creates specific problems that are discussed in this paper from the point of view of the autonomous control system design.

Bejczy distinguishes two basic approaches to the effect of combining autonomous and teleoperated control.[2,3] In traded mode, the control may switch from fully autonomous control to full teleoperation. In shared mode, certain degrees of freedom are allocated to the autonomous controller and the others to the teleoperated control. The concept of supervisory control, as exemplified by the work of Sheridan,[12] also poses challenges in terms of the design of a cooperating autonomous control system. Supervisory control entails having a human operator specifying general instructions to a partially autonomous control system that is capable of producing an integrated summary display of the results from which the choice of the next instructions can be based.

First, we will discuss the structure of a possible autonomous control structure, and then establish corespondance with concepts familiar in the field of teleoperation: Traded, shared and supervisory control.

Telerobotic systems may be described from three distinct viewpoints which are (1) a collection of models which account for the physics of the system, (2) levels of abstraction, and (3) a configuration, or control structure. While a one-to-one correspondence between these entities would be convenient, its implementation is unclear due to their dynamic nature. Furthermore, modern control structures tend to blur levels of organization. For example, force interaction during task execution may be carried out using hybrid control or manipulator controlled compliance. Using such schemes, joints are no longer controlled individually, making it difficult to isolate "joint level" control.

In this paper, the term "layer" will refer to the actual (physical) organization of the system, and the term "level" will refer to the abstract (conceptual) organization of the task. We wish to emphasize the important difference between these two notions.

Control can be adequately partitioned into hierarchical control layers if the underlying physics of the system are themselves hierarchical. Large scale technological systems attempt to enforce such a structure. For example, power distribution networks are designed in such a way that local failures do not disturb the function of the whole: normally a short-circuit in one house does not affect the neighbouring one.

Robot control systems can be molded into the hierarchical framework if the variability of tasks to be performed is small. Unfortunately, in applications such as space automation, the considered class of tasks is very broad and the systems to be controlled no longer consist solely of manipulators. The task itself must be what is controlled, leading to a great deal of variability. For each task, we obtain a different physical system, thus potentially different levels to describe it.

As the dividing line between autonomous control and teleoperation is shifting, the organization of a telerobot must be flexible enough to accomodate incremental development. This imposes additional constraints on the design.

2. A POSSIBLE AUTONOMOUS CONTROL STRUCTURE

2.1 Sensors

Sensors are fundamental components of flexible automation devices such as telerobots because they alleviate the need to provide *exact and complete* knowledge about the task. Sensors are used to acquire information at run-time which is not available or would be impractical to gather beforehand, or which is too cumbersome, In short, sensors are used to deal with uncertainty.

There are two ways, possibly combined, for dealing with uncertainty. The task can be designed such that its *natural constraints* cause the task to converge toward a goal by naturally descending its potential function.[4]

If the conditions cannot allow such a design, sensors have to be used to actively overcome the lack of convergence (for example track a surface) and the lack of modeling. In that case, sensors are used to establish the discrepancy between a *desired state* and the actual state, in view of computing correcting commands (control theory). Sensor thresholds can be used to trigger finite state changes (automata). Finally, sensors can also be used to assert the effective completion of a task. It should be noted that this latter use is conceptually identical to the other uses, but is occurring at a higher level of abstraction. In all cases, an error analysis must be performed beforehand in order to determine all permissible thresholds. This later analysis is potentially computationally intractable and therefore must be carefully considered at the planning stages of a

CH2556-9/88/0000-0122 $1.00 1988 IEEE

task.

For sensors, there is clearly a physical layer. In the case of sensors it can be called the instrumentation layer. In some cases, sensor data can be utilized almost directly, for example in a tracking task. Most often, raw data will require some kind of filtering, but still will represent the sensed variables. Filtering only requires knowledge about the signal itself. One level up, data needs to be aggregated. in order to construct models. Aggregation requires knowledge about the properties of the sensor. Finally, aggregated data will require interpretation, that is, one will have to account for the nature of what is being sensed in order to derive the required information. Interpretation also requires knowledge about properties of what is being sensed.

In short, there are apparently four layers of sensing hierarchy, which will not necessarily match layers of the others hierarchies. This sensor organization must be apparent in the database that specifies available sensing capabilities (Figure 1).

Interpretation Layer (Knowledge about the world)
Aggregation Layer (Knowlededge about sensors)
Filtering Layer (Knowledge about the signal)
Instrumentation Layer

Fig. 1 Sensing Layers

2.2 Commands

The identification of levels of commands relies on the assumption that any command can be expressed in terms of other commands, hopefully simpler and concerning lower levels of abstraction.

At the highest levels of abstraction, tasks may be described in terms of mathematical logic. The task description consists of plans which describe the states through which the system should go for the task to be accomplished. Although in theory, mathematical logic is the only tool required to describe all details of a plan, in practice this approach quickly leads to untractable difficulties. Plans must rely on very high levels of abtraction to be successfully coded.

In a telerobotic system the next layer may correspond to robot programs which encapsulate basic operations in terms of strings of gross motions, and sensor-based fine motions (guarded and compliant) together with the operation of end-effector and peripheral equipment. In the case of the coordination of multiple manipulators, processes described by finite state automata were found to be a convenient abstraction (Nilakantan and Hayward 1988). The next layer will be concerned with trajectories, which allows us to abstract the mechanical system in terms of a point in velocity/force space. Finally, the lowest layer will consist of dynamic control algorithms applied to explicit setpoints. whether they concern the motion of manipulators, end-effectors. or peripheral equipment

Levels of abstraction are depicted in Figure 2

AI Planner (Predicates and Logic)
Strategy (Skeletons)
Actions (State change relevant to the plan)
Motions (End point mouvements. Homing, tracking)
Control (Velocity, force)
Joints (Angles and torque)

Fig. 2 Command Levels

The commands have to execute on a physical and computational structure, consequently, a mapping is established between the command structure and the functional one.

Figure 3 depicts this binding process, analogous to compilation in a computer. If the functional hierarchy can directly execute the abstract task description, this binding step may be omitted and everything is interpreted. If the binding is performed at run-time within levels, we obtain a more complex but more versatile system. Finally if the entire binding is done at run time, we obtain a very complex system, but immeasurably more versatile (probably what biological systems do).[7]

$$\| \text{Levels} \| \rightarrow \textit{Binding} \rightarrow \| \text{Layers} \|$$
$$\Uparrow$$
$$\textbf{Context}$$

Fig. 3 Binding

Once generated, commands must be applied to a run-time system which will have all the attributes of a concurrent, hierarchical distributed real-time system. Thus, software engineering plays a central role in the implementation of such a run-time structure which must gracefully handle exceptions and deal with context changes. We expected exception handling to be one of the major software challenge in the design of a telerobot. In many ways, this command decomposition reflects the control architecture selected at JPL.[1,10]

2.3 Site and Computational Hierarchy

The computational architecture of a telerobot is distributed over several sites linked to each other via communication channels. The local site is normally situated in a laboratory environment with powerful computer facilities. The operator site is located near an operational site within physical reach from the human operator, as close as safety allows. The actual manipulator control system must be installed very close to the operational site. In general, the commands produced from model-based planning, are carried out via efferent channels and convey back relevant information about the task execution through afferent channels.

Each of the sites must perform, at various rates, with various capabilities and levels of abstractions, the functions pertaining to a robotic task execution. Since this organization is naturally distributed, one must find how the task execution decomposition can be mapped onto the architecture while satisfying all constraints imposed by computational limitations and channels bandwidth and delay.

2.4 Motion Planning

Motion planning can be viewed as a process occurring between the task planning process and the servo-control process. The role of motion planning is to satisfy a set of constraints dictated by the manipulator itself (its work envelope, kinematic and dynamic properties, and possibly other considerations such as deflection), the task (nominal trajectories must converge toward a goal state under model uncertainty), the environment (motions must only generate wanted collisions with controlled force), and design parameters such as energy or joint travel minimization.

Many of the motion planning techniques require extensive computations. In consequence, the very first stages of task planning consists of deciding how much motion planning must be done off-site, at task preparation time, and how much can be done on-site. Of course, in the latter case, a larger amount of flexibility and adaptation can be expected from the system. In the "programming by showing" systems, all trajectories are stored in a fixed manner. In sensor-based motions, reference coordinates and target positions can be determined at run time. The dynamics of the manipulator must be utilized on-line to set bounds on accelerations. However, the task preparation phase must insure that the resulting trajectories are collision-free. It is a desirable goal to include that capability at run-time as well as it would increase the adaptivity of the system to new situations.

There also exits a classification in the nature of motions. Gross motions are utilized to move manipulator and loads over large distances. In this case, the principal constraint imposed onto the motion is avoidance of collisions. In the case of docking, for example, there exits additional constraints such as following a well defined path in Cartesian space. Fine motions will be used to reduce the discrepancy between expected model-based trajectories and actual trajectories constrained either by physical contact or proximity sensing. The collision avoidance problem then takes a different nature and must be more concerned with unwanted collisions in the vicinity of the work site.

2.5 Granularity of Description

There also exists a hierarchy in the abstractions and models used to describe a telerobotic system. At the highest level, the system and its task can be described in terms of formal logic, once hardware details are abstracted. The implementor will have to choose the level of abstraction or *grain size* at which such a description is appropriate. Although in theory, mathematical logic is the only tool required to describe any kind of system, in practice, one must decide when this approach becomes appropriate or ceases to be so. At a less high level of abstraction, it might be useful to describe the system in terms of automata. in other words, the system is described in terms of *state changes*. Some formalisms such as Petri Nets have been proposed to express concurrency. At an even lower level, the system can be described in terms of *processes*. Processes have a finite life time and explicitly deal the notion of time, hence the importance of synchronization mechanisms. At a lower level, descriptions are made in terms of continuous functions (i.e. kinematics) and continuous feed-back control.

3. TELEOPERATION VS. AUTONOMY

Shared, traded and supervisory control can in fact be viewed under the same angle. From the autonomous control point of view, the intervention of the human operator can be viewed as the replacement of of the system's functions by a human operator acting through a bi-directional man-machine interface. For the sake of clarification, the system should be designed in such a way that the intervention should not be viewed as a *mode change* but as the replacement of a functional unit by another one, all things remaining equal. Distinction between shared control and traded control then becomes a difference in levels, but not in nature. In effect, a re-mapping of information channels can be performed (Coiffet and Gravez).

There are two cases: The human operator decides to interrupt an autonomous task execution. Presumably this execution will resume at a later moment, the autonomous system decides to stop the task because, for example, sensors indicate large discrepancies with expected model-based values. At interruption time, the autonomous system must be capable to express in a concise manner which state has been reached. It will also have to express which allowable states are expected, if dynamic re-planning is to be avoided.

This problem can be solved if all levels of a control hierarchy are capable to display their current context of execution, thus allowing the human operator to make an informed decision that will not disturbed the current plan. Futhermore, at resumption time, the human operator must be capable to specify to the system, the state information that has been changed while the system was under human operator control. This problem has received a lot of attention in artificial intelligence research under the headline of the "frame problem".[5]

Shared control is likely to be easier to implement since the human operator is scheduled to intervene at a predefined section of the task execution. From the autonomous controller, shared degrees of freedom are conceptually analogous to compliant motions. The net result is a separation of the operational space into subspaces in which what is not commanded (velocities, respectively forces) is derived from the robot behavior and reincorpotated in the task model.

Other shared control options can be put in terms which are familiar to robot programming techniques. For example, position specification can be provided by a human operator asynchronously with the robot's actual motions. There exist in fact a continuum of possibilities based of the notion of sensory preview. At the other end of the spectrum, we have kinesthetic coupling. In that later case, performance degrades very quickly as the amount of asynchrony and delays augments.[13]

Thus, transmission delays will also affect directly the design of the autonomous control system as more robust strategies will have to be used as the delays increase, reducing the possibility of human intervention.

We would like to point out, that delays are also a notion attached to any dynamical system. In consequence, specific problems arise in attempting to accurately synchronize manipulators and/or machinery,[8,9] and consequently in controlling them form human operator commands (no matter how dexterous, a manipulator will delay execution of a command causing a velocity change).

4. CONCLUSION

The handling of these problems in a structure manner requires the identification of the relationships between the various models that contribute to the description of a robotic task. The relative importance of these relationships can vary from task to task. In other words, robotic commands that we hold for "primitives" are in fact extremely context sensitive. For traded control, from the autonomous control point of view, task planning and robot programming need to be highly structured to allow for explanation capabilities. Collision prevention plays a central role and need to be higly automated. For shared control, high quality compliant motions are essential and manipulator design plays a central role.

5. AKNOWLEDGEMENTS

The author wishes to thank, L. Daneshmend, A. Nilakantan, A. Topper, A. Foisy L-P. Demers, S. Aubry for many great discussions.

Support from NSERC (Natural Sciences and Engineering Research Council of Canada), JPL (Jet Propulsion Lab) and Spar Aerospace is gratefully acknowledged.

6. REFERENCES

1. Balaram J. Loksin A., Kreutz K., and Beahan J. 1987. A run-time control architecture for the JPK Telerobot. *Proceedings of the Workshop on Space Telerobotics*. G. Rodriguez, Editor, NASA/JPL Publication 87-13, Pasadena, Ca.

2. Bejczy, A. K., Vušković, M. 1980. An interactive manipulator control system. *Mini and Micro-computers*, Vol. 5, No. 1.

3. Bejczy, A. K., Szakali, Z. 1987. A synchronized computational architecture for generalized bilateral control of robot arms. *Proc. of SPIE, Space Station Automation III*, Vo. 851. Cambridge, MA.

4. Erdmann M., Mason T. M. 1986. An exploration of sensorless manipulation. *IEEE Int. Conference on Robotics and Automation*, San Francisco, CA.

5. Genesereth, M. R., Nilsson, N. J. 1987. *Logical foundations of artificial intelligence*. Morgan Kaufman Publishers Inc.

6. Hayward, V., Aubry, S. 1987 (November). On describing a robotic scene, *Intelligent Robots and Computer Vision: Sixth in a Series, Advances in Intelligent Robotics Systems, SPIE*, Cambridge, MA.

7. Hayward, V., 1987 (July). An integrating survey of robot manipulator control and connections with their biological counterparts. *McGill Research Center for Intelligent Machines Technical Report*, CIM-87-13, McGill University, Montréal, Canada.

8. Hayward, V., Hayati, S. 1988 (June). Kali: An environment for the programming and control of cooperative manipulators. 1988 American Control Conference. Atlanta, GA.

9. Nilakantan, A., Hayward V. 1988 (November). Synchronizing Multiple Manipulators. *Second International Symposium on Robotics and Manufacturing, Research, Education and Applications*. Albuquerque, New Mexico.

10. Schenker P. S. 1987. Program objectives and technology outreach. *Proceedings of the Workshop on Space Telerobotics*, G. Rodriguez, Editor, NASA/JPL Publication 87-13, Pasadena Ca.

12. Sheridan, T. 1987 Telerobotics. *10th IFAC World Congress on Automatic Control*, Munich, FGR.

13. Stark, L., Tendick, F., Kim, W. S. 1988 Cooperative control in telerobotics. *IEEE 1988 Int. Conf. on Robotics and Automation*, Philadelphia. PA.