An agent based Human Robot Interaction in a supervised autonomous system.

Alon Levy and Reuven Granot [(*)]
Technion Israel Institute of Technology
[(*)] Center for Computational Mathematics and Scientific Computation Center for Computational Mathematics and Scientific Computation at University of Haifa

ABSTRACT

In human supervised autonomously controlled systems the human operator should concentrate attention on tasks, which need high skills and leave to the computer yet complex processes, which can be done autonomously. It is crucial to let the operator have a complete control over his tool by enabling him maximum freedom to make changes in the robot behavior as response to specific environments or events. A model based on a mixture of hybrid and behavior based systems enables human operator intervention at all levels of the control hierarchy. This paper deals about how this human intervention should be applied.

For human intervention at the low layers of the control hierarchy of RCS, where the control bandwidth is extremely high, a software entity, known as agent, which acts in the purpose of the human operator, will take care of the fluent transition between the state in which the robot operates and the one imposed by the human. This agent will be a task oriented control agent. For layers, with lower control bandwidth, the agent serves as an intelligent interface, which may restrict the less experienced operator from catastrophic intervention with the system activity. In such a case, the human operator should have the choice to completely switch off this feature.

INTRODUCTION

Supervised autonomy is beneficial because: (1) it removes humans from DDD (Dirty, Distant, Dumb) activities eliminating casualties from enemy action or accidents; (2) it increases efficiency by allowing one human controller to supervise the operation of many platforms; (3) it allows the platforms to function in synergistic and new ways with other platforms and sensors by taking the human out of the loop most of the time; the vehicles can then directly share sensory information, databases, experiences, and collective decision-making.

For most combat operations (as well as in some other examples), in order to leave the soldier operator aware of the battle development and react quickly to improve his personal safety, it is mandatory to shorten a very long delay needed to the human operator to adapt him from the remotely monitored environment back to the local real world. For remote operation in such environments, it is mandatory to upgrade from a continuously remote controlled system into a supervised autonomous one.

Remote work is far more costly to execute than what workers could accomplish directly with conventional tools and practices because task operations are slow and tedious due to difficulties of remote manipulation and viewing [(1)]. Decades of experience within the remote operations community (i.e. nuclear) shows that remote tasks may take hundreds of times longer than hands-on work; even with state of the art force-reflecting manipulators and television viewing, remote task performance execution is five to ten times slower than equivalent direct contact work. Modest improvements in the work efficiency of remote systems can have high payoffs by reducing the completion time for projects [(3)]. Additional benefits will accure from improved work quality, enhanced safety and supervision of many platforms by the same human operator.

Under supervisory control [(2,3)], an operator divides a problem into a sequence of tasks, which a system can achieve on its own [(3)]. In multi-operator teleoperation, humans share or trade control. Cooperative teleoperation tries to improve teleoperation by supplying expert assistance [4]. Several robot control architectures have addressed the problem of mixing humans with robots [6,7].

Human supervised autonomously controlled systems, also known as telerobots [3], are capable of focused perception and intelligent action. In a telerobot, a human operator generates tasks, and a computer autonomously closes some of the controlled loops. The human should concentrate attention on tasks, which need high skills and leave to the computer yet complex processes, which can be done autonomously, like moving a multi segmented robot arm.

Current real time control can guarantee that hard deadlines will be met, but cannot autonomously react to unpredictable events. The major problem of robotics (automation) is how to deal with unexpected contingent events. At one extreme there are those events, which can well be treated by the autonomous system. At the other extreme are those events, which need the intervention of a human operator. This intervention is based on human capabilities like intuition, experience or training (i.e. for maintenance). We assume that the current technology is far less capable to deal autonomously with these types of events and a human operator is required to be at least capable to communicate with the autonomous subsystem. Instead of a fully autonomous design, the automation will be introduced as an improved Man Machine Interface (MMI) by developing independent features, which enable the operator to better supervise a larger number of systems. So, the telerobot is developed as an improved tool [8] to serve the human, rather then a terrifying competitor for his job related glory. In order to leave the human operator with high motivation in completing his job, it may be crucial to let him have complete control over his tool by enabling him maximum freedom to make changes in the robot behavior as response to specific environments or events. Most of these will not be anticipated even by the best design.

A harmonic collaboration of man and machine is usually assumed to be hard to achieve. A mixed architecture, in which human and machine interact, each making use of its best characteristic features is now a technological imperative for many practical applications. Examples of such applications include operation of equipment in combat environment, mining, earthmoving and construction, as well as safe driving or security and area protection. A model based on a mixture of hybrid and behavior based systems, incorporated into RCS [9] enables human operator intervention at all levels of the control hierarchy and at every element of its intelligent nodes. However, this model does not define explicitly how the human intervention will be applied, which is the subject of this paper.

THE HUMAN SUPERVISOR AND SYTEM INTERACTION.

Supervision and intervention by a human would provide the advantages of on-line fault correction and debugging, and would relax the amount of structure needed in the environment, since a human supervisor could anticipate and account for many unexpected situations.

The problem of supervisory control is also one of decision making, i.e., given the observation of the state of the process, to decide from encoded knowledge what action to take. From this perspective, the human supervisory control is beter acomplished as a task oriented control [10]. Knowledge based systems and, in particular, rule-based approaches, is ideally suited for such a decision making task. The fuzzy controller is one such simple rule-based system. The knowledge for it does not derive only from an expert operator; it can also come from the designer of the system. Fuzzy logic gives the most efficient rule-based representation that deals with continuous variables and fuzzy control at the higher layers of the architecture hierarchy can naturally enable the human machine interaction [11] and so be used as an addition to conventional control mostly implemented at the lower layers of the hierarchy [10]. For example, humans drive a car with no measurements and no computation receaving decisions at low control bandwidth, while high control bandwidth conventionally controlled units of the car system autonomously perform activites. When they fail to perform, the human driver is forced to stop or reduce speed. The “drive by wire” car system, as been an improved machine upon the manually drived car, will perform autonomously under strict suprvision of a local or remote human driver. This driver, if is remote, may drive a few such systems; or if he is situated on the vehicle, may perform simultaneosly other tasks, probably of a higher complexity.

The human supervisor must have the ability to interfere and improve the system performance on the fly, using his past experience and wills. Otherwise, the problem reduces to be one, which may have been solved by apriori-improved design.

The human operator initiates an autonomous subtask at one of the higher layers of the control hierarchy and yet is able to interfere at all levels and at any time, limited only by his abilities. In such a model, the human operator can change system behaviors and practically interfere with actions performed in real time.

An autonomously performed task senses its environment, frequently processes the relevant data to achieve a conclusion about its state (make decision) and control action. Sometimes this activity is missing the expected performance due to imperfect sensors or their environment, inadequate data reduction or decision algorithms or just occurrence of some unexpected events. In this and similar situations the human operator should interfere and change the system's behavior by adapting some of control parameters or completely replace the controller with a previously prepared one.

The problem relates to the time delay introduced by a human operator, while monitoring the system's activities. While, in the remote control mode of operation, which is a continuous mode, the operator being in full attention, his control bandwidth determines the rate of response. In those situations, in which it is not possible to break the robot's operation, once the operator has to make a decision to interfere and only then take action, the human-robot system acts at two distinct time scales and control bandwidth. Inputs received from the operator (at low control bandwidth) being imposed on a running control task (high control bandwidth) may introduce instability.

THE CONTROLLING AGENT

A software entity, which acts in the purpose of the human operator, known as agent, will take care of the fluent transition between the state in which the robot operates and the one imposed by the human (12). This transition should take care about the imperfections, which are responsible for the improper operation of the robot, by disconnecting or adapting them to the new situation. The robot is assumed to fluently transfer from the original state to the newly defined one as expected by the experienced human operator.

For human intervention at the low layers of the control hierarchy of RCS, or other JAUGS [(13)] compliant architectures, where the control bandwidth is extremely high, this agent will be a task oriented control agent. The control agent will be a control subassembly, and it will be capable to replace on the fly the running controller or make necessary adaptation of the control parameters, without bringing the system to a temporary stop. In the same time the higher layers are informed about the expected disturbances. For layers, with lower control bandwidth, the agent serves as an intelligent interface, which may restrict the less experienced operator from catastrophic intervention with the system activity. In such a case, the human operator should have the choice to completely switch off this feature.

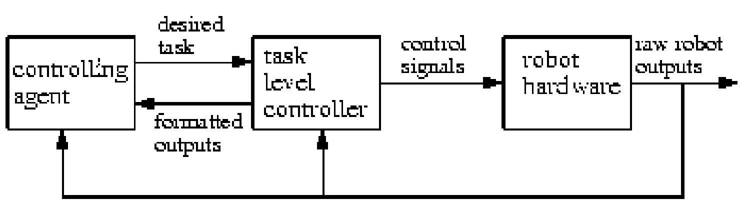

Figure 1. Task-level supervisory control system block diagram.

The task level controller in figure 1 stands for a simplified presentation of the controlling agent's role. Once integrated into the RCS reference model architecture, the Behavior Generator (BG) hierarchy takes care of the preplanned tasks. BG modules in higher layer nodes serve as controlling agents and are part of the hierarchy. The problem is to introduce an agent (the one representing the human operator), somehow outsider to the layered structure of RCS. This agent exchanges messages with the BG of the node responsible for the needed change or adaptation and may use all available resources of the relevant BG modules. It may also receive information from sensors at different levels of analyzed data.

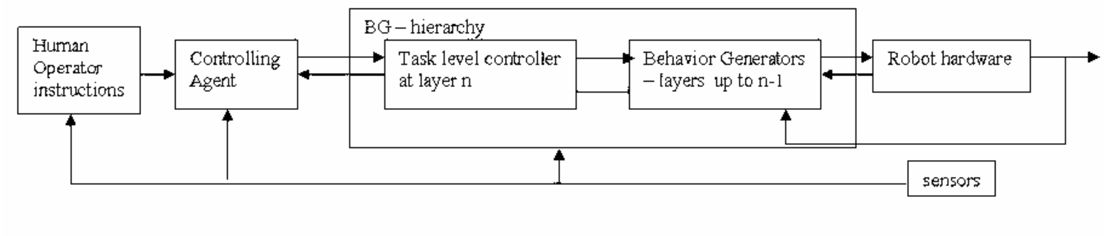

Figure 2. A model for human operator interaction, represented by a controlling agent, in a supervised autonomous RCS architecture based system.

CONTROL AGENT INTERACTION WITHIN RCS

According to the RCA architecture guides, the human operator will be represented by a software agent, which has to negotiate some improvement in the control process. That is why it will be called a control agent (CA).

The CA's goal is to implement the human intentions as a general guidance by replacing some running controllers smoothly with other alternatives. The alternative controllers, or behaviors, may be of completely different type or just adaptation of the already running behaviors. The objective to implement smoothly the human operator's wills means that the transaction from the running state to the new one must be done without making a system stop and taking care of system security (preventing it from reaching catastrophic event situations, or rich unstable behavioral situations).

There are two types of transfer actions the CA has to deal with:

- The new and old behaviors are competing on the same resources, in which case the transfer action may be done by a function, which activates continuously the new behavior in parallel to the continuous deactivation of the old one.

- The new and old behaviors are not competing on resources, so the transaction may be done instantaneously, once the new behaviors run in parallel with the old ones.

The system safety is another responsibility of the CA. Before implementing the human operator intentions (if not explicitly instructed otherwise), the CA has to check all the relevant data to ensure that no catastrophic event may occur from the prompt change in local goals.

The CA is external to RCS model, but can communicate with all the existing nodes. Once instructed to make a change at some explicitly pointed node, it will send a message to all supervisory nodes about the human operator wish to make/ check for a possible change in local goals. Any change regarding the Global Goal, should bring the system to a temporary stop, until the human operator will solve this contradiction, by adjusting the newly updated Global Goal. In this way the allowed adjustments made by the human operator at some node should be correlated with the local goals of the supervisory nodes, or the operator should be asked to connect the CA to a higher layer situated node. Once this is agreed, the CA tasks are now restricted to some heuristic improvements of the autonomous procedures. This may even be later observed as reducing the system performance, but been done because of the human instruction. The reason for such a non-optimal activity is the wish to leave the human in charge for responsibilities beyond the design characteristics of the system.

EXAMPLES OF CONTROL AGENTS ACTIVITY

- **Avoid obstacle behavior**:
  - The obstacle may be avoided using a right detour path, as well as a left detour one.
  - Based on autonomous decision making procedures, the system starts a left turn.
  - The human operator based on personal experience and his preferences, decides for a right detour and instruct so the CA.
  - The agent after sending messages to the node responsible for path control calculates the latest time at which the turn may still be changed from a left detour path to a right detour path.

    - The agent instructs the path control node to evaluate a new plan for a right detour and generates time constraints as well as other design requests.
    - If the new plan fits the design criteria, the CA waits to approve it by launching a competing behavior, as the new and the existing behaviors compete on the same resources: steering and acceleration of the wheels.
    - The new behavior strength is increased at the same time as the old behavior strength is decreased.
    - If the right detour path plan cannot be accepted in time, the CA informs the human operator, which may decide on an emergency stop.
- **A bulldozer with "blade in the ground", while moving earth in a specified direction is instructed to speed up**.
    - The CA after sending messages to the relevant supervisor node, has to decide between several alternatives:
        - "Raise the blade" and so reduce earth resistance force but perform a less deep excavation. The CA from the node responsible for job scheduling and excavation task completion will request value for different parameters. This is equivalent to the design/ choose of a new behavior
        - "Increase the tilt of the blade" and so also reduce earth resistance force. This action may need an increase in power and may be used only when the bulldozer is assumed to return to the same path, or the excavation depth is far from final.
        - "Increase motor power" and so reduce accuracy in direction and width (if acceptable). In this case the CA should ask for new performance calculations based on the increased motor power and decide on its proper value.
    - Independent of the CA choice, the new behavior is replacing the running one without any competition on resources. The transition may be done instantaneously, since the new behavior is only activating new sub-behaviors, without replacing those already running, except making some continuous variation in their parameters.

TYPES OF FUZZY SUPERVISORY CONTROL AGENTS [14]

The human intervention for the supervision tasks is best acomplished using a natural language for the human operator. Using fuzzy variables in order to instruct the robot about the updated tasks introduces a fuzzy control set of rules as the input for the Controllling Agent. As such, the fuzzy controller and the task level controller form a cascaded control in which the set of output set values of the fuzzy controller serve as input values for the underlying task level controller (i.e. a set of PID controllers).

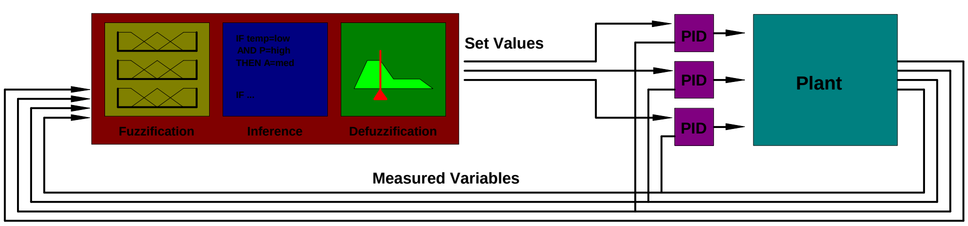

Figure 3: Fuzzy Logic Controller Outputs Set Values for Underlying PID Controllers

In some cases supervisory multi-variable controllers are designed from operator experience and experimental results rather than from mathematical models. Each single process variable is kept constant by a PID controller, while the set values for the PID controller come from the fuzzy logic system. In other cases, it could be reasonable to develop the complete closed loop control solution for the human supervisor instruction in a fuzzy system.

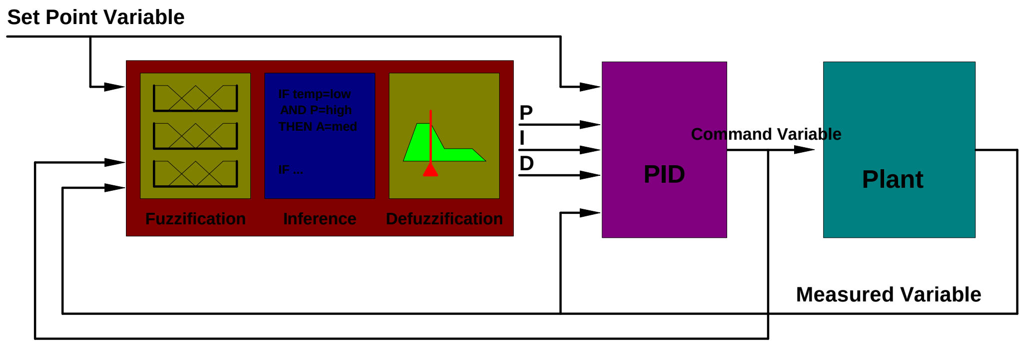

Figure 4: A fuzzy logic controller adapts the parameters of a conventional PID task level controller.

When the supervisory intervention causes large disturbances in the task level controller a fuzzy controller in parallel to the conventional controller may be a preferred solution.

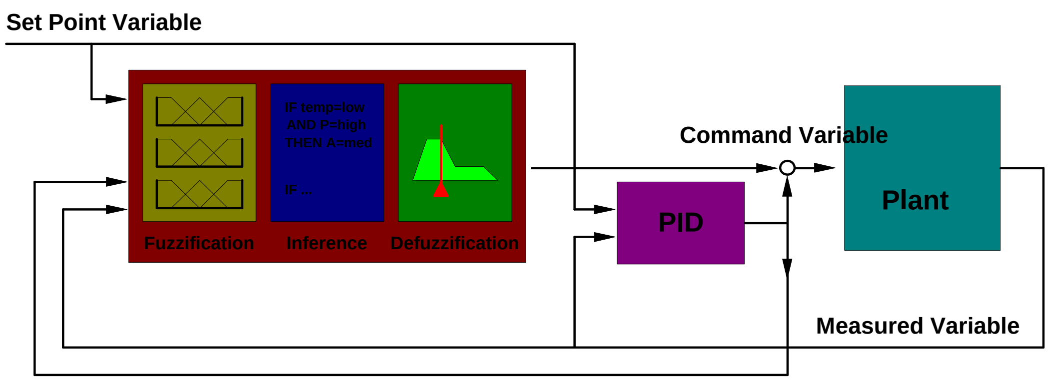

Figure 5: A fuzzy controller in parallel with the PID task level controller for large supervisory disturbances.

THE ROBUST STABILITY OF THE CASCADED CONTROL BUILT FROM A CONTROL AGENT CONNECTED TO THE TASK LEVEL CONTROLLER.

We assume that the task level controller is well behaved. It is a controller at the lower layers of the architecture and as such it has a high bandwidth with asymptotic stability: all outputs asymptotically progress in time to a constant value from any initial condition and oscillations are within the allowed decay.

For the Controlling Agent, in order to have robust stability it is required that the supervisory stability will denote bounded outputs in time and no limit cycling (limit cycling is a phenomenon in nonlinear systems in which a system is stable but not convergent).

REFERENCES.

1. W. P. Hamel and J. Osborn, "Robot Task pace Analyzer" U.S. Department of Energy, contract # DE-AR26-97FT34314, www.netl.doe.gov/publications/proceedings/97/97em/em_pdf/EMPI-11.PDF
2. Cannon, D. and Thomas, G. “Virtual Tools for Supervisory and Collaborative Control of Robots”, *Presence* 6(1), 1997.

3. Sheridan, T.B. (1992) "Telerobotics, Automation, and Human Supervisory Control", MIT Press.
4. Murphy, R. and Rogers, E. "Cooperative Assistance for Remote Robot Supervision", Presence 5(2), 1996.
5. McGovern, D., "Human Interfaces in Remote Driving", Technical Report SAND88-0562, Sandia National Laboratory, Albuquerque, New Mexico, 1988
6. Albus, J., et. al., "NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)", Technical Note 1235, NIST, Gaithersburg, Maryland, 1987.
7. Krotkov, E., et. al., "Safeguarded Teleoperation for Lunar Rovers", SAE 26th ICES, Monterey, California, 1996.
8. S. Murray and D. Murphy, "Underwater Telerobotics and Virtual Reality: A New Technology Partnership", http://www.spawar.navy.mil/robots/pubs/pacon96.pdf
9. Albus, J. S. and Meystel, A. M. (2001) "The Engineering of Mind", John Wiley&Sons. Inc
10. Ha, Q.P., Santos, M., Nguyen, Q.H., Rye, D.C. and Durrant-Whyte, H.F. "Robotic excavation in construction automation," IEEE Robotics and Automation Magazine, Vol. 9, No. 1, pp. 20-28, 2002. http://www.eng.uts.edu.au/~quangha/Pdf_papers/IEEE_RAM.pdf
11. Ha, Q.P., Santos, M., Rye, D.C. and Durrant-Whyte, H.F., (2001) "A task-oriented control architecture for robotic excavation" Proc. of the 5th World Multi-Conference on Systemics, Cybernetics and Informicatics (SCI'2001), Orlando USA, vol. 2, pp. 65-70. http://www.eng.uts.edu.au/~quangha/PhDProjects/Robust.html
12. R. A. Reyes and S. Tosunoglu, "Fuzzy Logic Control Strategies for Teleoperation Systems", 1999 Florida Conference on Recent Advances in Robotics
13. http://cimar.mae.ufl.edu/FLA99/session2/ray_fiuppr.pdf
14. Edward Tunstel, Marco A. A. de Oliveira, Sigal Berman: "Fuzzy behavior hierarchies for multi-robot control", International Journal of Intelligent Systems 17(5): 449-470 (2002)
15. Nilsson, N. "Toward Agent Programs with Circuit Semantics", Technical Report STAN-CS-92-1412, Stanford University Computer Science Department, 1992.
16. Nilsson, N. "Teleo-Reactive Programs for Agent Control", Journal of Artificial Intelligence Research, 1:139-158, 1994.
17. http://www.jauswg.org
18. M. W. Torrie, D.L. Cripps, J.P. Swensen (2002) "Joint Architecture for Unmanned Ground Systems (JAUGS) Applied to Autonomous Agricultural Vehicles, ASAE Conference on Automation Technology for Off-road Equipment, Hyatt Regency, Chicago, Il, July 27-28, 2002.
19. http://www.fuzytech.com