VISION-BASED CONTROL IN AUTONOMOUS MANIPULATION FOR INTERVENTION MISSIONS AUV

Giacomo Marani * **Junku Yuh** **

* *Autonomous Systems Laboratory, University of Hawaii, USA*
** *National Science Foundation, USA*

Abstract: Autonomous manipulation refers to the capability of a robot system that performs intervention tasks requiring physical contacts with unstructured environments without continuous human supervision. Today, few AUVs are equipped with autonomous manipulators. SAUVIM (Semi Autonomous Underwater Vehicle for Intervention Mission, University of Hawaii) is one of the first AUVs capable of autonomous manipulation.
This paper presents a solution adopted within the manipulator control system of SAUVIM in order to address the problems inherent in autonomous underwater manipulation. The main approach is layered into different levels, where different behaviors take place, and the environment sensing is mainly based upon optical vision.

Keywords: Autonomous underwater manipulation, computer vision

1. INTRODUCTION

Most of today's underwater intervention tasks are performed by manned submersibles or remotely operated vehicles (ROVs). While autonomous underwater vehicles with no physical link to the human operator could permit operating in the areas where humans cannot go, such as under ice, in militarily denied areas, or in missions to retrieve hazardous objects, the low bandwidth and significant time delay inherent in acoustic underwater communications are considerable obstacles in remotely operating a manipulation system and reacting to unexpected situations in a timely manner.

The key element in improving underwater intervention capability in AUVs is autonomous manipulation. It is a challenging technology milestone, which refers to the capability of a robot system that performs intervention tasks requiring physical contacts with unstructured environments without continuous human supervision. Today, few AUVs are equipped with autonomous manipulators. SAUVIM (Semi Autonomous Underwater Vehicle for Intervention Mission, University of Hawaii, fig. 1) is one of the first AUVs capable of autonomous manipulation.

This paper presents a solution adopted within the manipulator control system in order to address the problems inherent in autonomous underwater manipulation. The main approach is layered into different levels, where different behaviors take place: a low-level layer, which interacts with the robot hardware, a medium-level layer where controls algorithms are executed, and a high-level layer where the task description is performed. Here, the environment sensing is based upon optical vision.

As an application of the above approach, this paper will present also initial experimental results of

Fig. 1. The SAUVIM autonomous underwater vehicle (Autonomous Systems Laboratory, University of Hawaii)

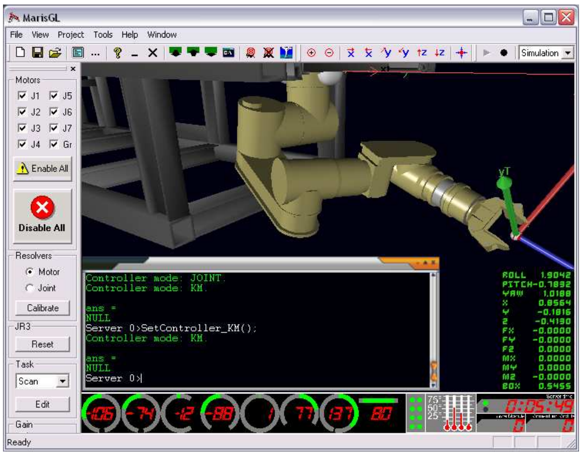

Fig. 2. The virtual reconstruction of SAUVIM manipulator and his workspace

autonomous intervention performed by SAUVIM in the oceanic environment. The operation included locating and cutting a cable, using a special tool integrated with the gripper and a camera mounted on the arm in order to locate the target.

2. AUTONOMOUS OPERATIONS

The principal obstacle in exchanging data with a generic AUV during a mission is the low bandwidth and a significant time delay inherent in acoustic subsea communication. This aspect makes very difficult operating the manipulator remotely.

Robot teleprogramming was proposed as an intermediate solution between supervised control systems and direct teleoperation when a significant time delay appears in the communication (Funda *et al.*, 1992; Paul *et al.*, 1993; Sayers *et al.*, 1998). The main idea is to make the operator feeling the system as a common teleoperation where the communication delay has disappeared. The concept consists in decoupling the local and remote zones by limiting the data exchange to a few symbolic information, in opposition to the common robot teleoperation system that use low level information (e.g. the joint position, motor velocity reference, etc.).

Usually, in teleprogramming systems, a partial copy of the remote information is used to create a virtual reality model of the remote environment. In this way, the main component of the virtual model is a predictive simulator. The interaction user-system is performed over a simulated environment and the stability of the local zone is not affected by the delay of the communication channel. The remote system, if not able to cope with an error, can transmit this information to the local system that may assume a safety state. It is matter of the remote system to avoid as much as possible these critical errors.

However this solution was introduced in order to teleoperate the manipulator by the user. In order to increase the level of autonomy, we introduce a different approach where the user provides, in place that a direct drive of the manipulator, a more high-level information as for example "unplug the connector" during a particular mission. In our implementation we use this concept in order to perform a generic intervention task. It is a different alternative between the still challenging and difficult problem of a fully autonomous control system and the classic teleoperation mode. In this approach, the function of the operator is to decide, after an analysis of the data, which particular task the vehicle is ready to execute and successively send the decision command. At this point, the low-level operation commands are provided by a remote pre-programmed subsystem, while the virtual reality model in the local zone uses only the few symbolic information received through the low bandwidth channel in order to reproduce the actual behavior of the system.

Figure 2 shows the actual virtual scene of SAUVIM reproduced in the local zone using only the sensor data available.

In this approach, the level of autonomy is related to the level of information needed by the system in performing the particular intervention.

Within the above scenario, the manipulator control system in the medium-level layer must ensure a reliable behavior within the workspace, avoiding collisions, system instabilities and unwanted motions while completing the required task, when it is theoretically executable. The control system must also address other general manipulation issues, such as task-space oriented, task priority assignment, and dynamic priority changes.

3. THE SAUVIM MANIPULATION SUBSYSTEM

SAUVIM (Semiautonomous Underwater Vehicle for Intervention Missions) (Yuh and Choi, 1999) involves the design, fabrication, and building of a semi-autonomous underwater vehicle for intervention missions such as underwater sampling, repair of underwater structures, and wreckage recovery.

To achieve these result the vehicle is equipped with a 7 degrees of freedom electro-mechanical robotic manipulator (MARIS 7080, fig. 3). The arm, designed for underwater applications at high

Fig. 3. MARIS 7080 Underwater Manipulator

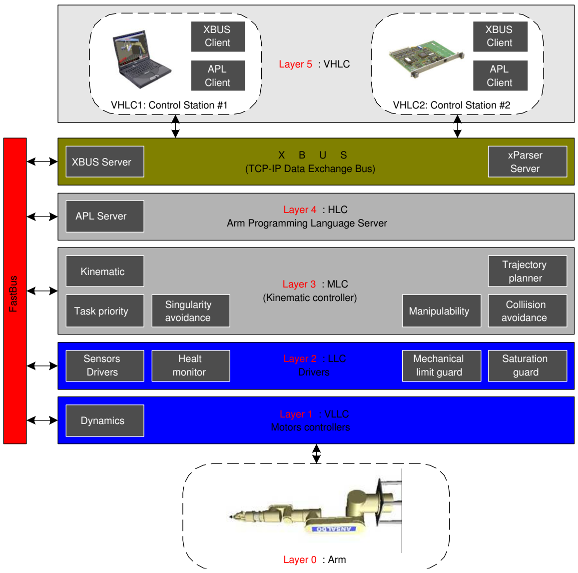

Fig. 4. MARIS Control System

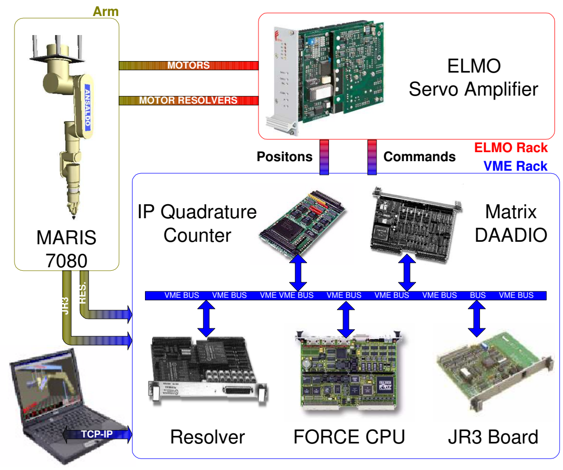

Fig. 5. Hardware architecture

depths, is internally compensated with appropriate oil. Each degree of freedom is actuated by a brushless motor with a reduction unit (harmonic drive). The accuracy of the angular measurement is assured by two resolvers (respectively before and after the reduction unit). A force/torque sensor set between the last degree of freedom of the wrist and the gripper, allows the acquisition of force and torque data acting on the gripper.

The VME-based computing architecture is shown in fig. 5. It uses a FORCE 68060 board to implement the control scheme, which in turn provides the references to the Elmo motor drives through the AD/DA board, a MATRIX MD-DAADIO. Two four-channel IP-Quadrature piggyback modules, mounted on a VME carrier board, acquire the encoder signal from the motor drivers. Two more boards are installed on the VME Bus, one for exciting and acquiring the signals from the resolvers on the output axis, and one for the JR3 force sensor. All the modules can be accessed by the FORCE board though memory locations mapped on the VME bus.

On the top of the above architecture, there is a well structured layered control system (fig. 4). The first layer, the Very Low Level Controller, basically is the hardware level. The second one hosts all the software drivers necessary for acquiring the sensor data and sending references to the actuator drives. The third layer, the Medium Level Controller, contains the coded algorithms for solving the direct and inverse kinematic problems: Resolved Motion Rate Controller, task prioritization with position prior orientation (Kim *et al.*, 2002), singularities avoidance (Marani *et al.*, 2002; Marani *et al.*, 2003; Marani *et al.*, 2006), collision avoidance and manipulability optimization. All the input and output of above algorithm, among with the sensor data from the layer 2, are directly accessible through the FastBus from the layer 4, the High Level Controller, that contains our Arm Programming Language server. The APL server layer is in turn connected with the communication layer that allows to exchange its data with the Arm Programming Language clients (e.g. loading executables).

The FastBus is responsible for sharing of information between all the layers. All input and output quantities are accessible form all the modules at all levels and, through the xBus, by the user. This facilitates the interaction with the user and allows debugging of the system.

The Arm Programming Language, a key component within the high level controller layer, has been described in (Marani *et al.*, 2005). It is suitable for real-time embedded control systems and offers at the same time flexibility, good performance and simplicity in describing a generic complex task. Its layer abstraction approach allows an easy adaptation to the hardware-specific requirements of different platforms. The language is completely math-oriented and capable of symbolic manipulation of mathematical expressions. The last is an important distinctiveness from most of the currently available robot programming languages. The procedural approach has been chosen in order to enhance the performance while maintaining the flexibility required for executing complex tasks. It is particularly suitable for real-time embedded systems, where the interaction of a generic algorithm with the time is often critical. Finally, the language system is organized in a client-server architecture, in order to separate the human interface from the execution layer, as obviously required in autonomous underwater manipulation systems.

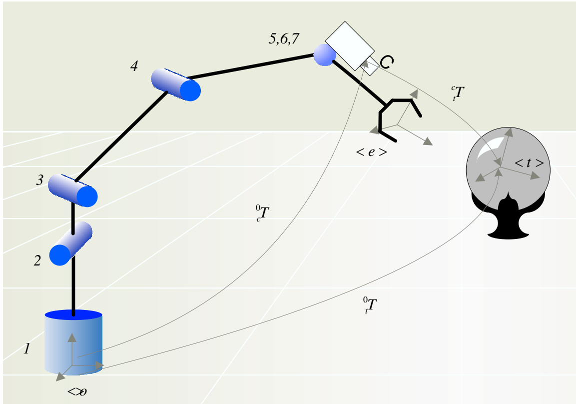

Fig. 6. Schematic representation of the arm workcell with camera and target

4. TARGET LOCALIZATION: THE VIDEO PROCESSING SYSTEM

A key feature of the autonomous manipulation system is the capability to locate the position of the target with respect to the base frame of the manipulator (which often may coincide with the vehicle frame) with a number of degrees of freedom sufficient to perform the required task. This goal is achieved using a video camera located on the wrist of the manipulator and a dedicated video processing system.

In our task, the target is a sphere of known diameter, represented in figure with the associated frame $< t >$. The transformation matrix T_t^0 of the target frame $< t >$ with respect to the base frame $< 0 >$ is given by:

$$T_t^0 = T_c^0 T_t^c \tag{1}$$

where T_c^0 is the transformation matrix of the camera frame $< c >$ with respect to the base frame $< 0 >$, while T_t^c is the transformation matrix of the target frame $< t >$ with respect to the camera.

The placement of the camera T_c^0 is easily computed using the joint position information of the arm and the relative position of the camera with respect to the joint on which it is mechanically coupled. This relative position may be precisely computed using a set of predefined movement of the joint hosting the camera, along the main axes.

The placement of the target T_t^c with respect to the camera is obtained, instead, using video processing of the acquired image. As a matter of fact, the problem may be seen as a 2d localization of a circle within the acquired image. Upon camera calibration, the cartesian position of the center of the sphere in the space may be easily computed from the center and the diameter of the 2D circle. The camera calibration is performed directly in the water, with the aid of the robotic arm that

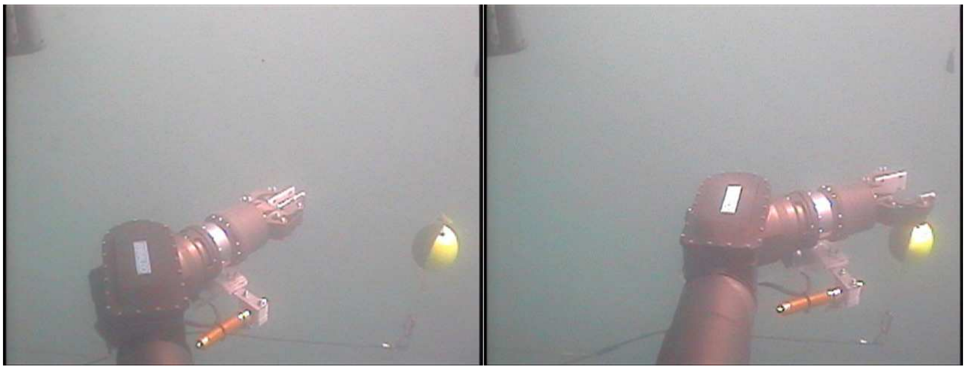

Fig. 7. Underwater scenes of the cutting operation

Fig. 8. Result of video processing: the circle extraction

makes possible the acquisition of a fixed optical pattern from different angles and distances.

The localization of the circle within the image is done using the following sequence of steps:

- Image filtering
- Edge extraction using Canny filter applied to the color image and using the color contrast gradient
- Fast Circle extraction using the line segments found in digital images (E. Kim and Kitajima, 2005)

This combination of algorithms gave us the best performance and robustness with respect to false or missed detections in an underwater environment. Figure 8 shows the result of the above sequence of steps applied to a single frame. In our actual implementation, the system is capable of processing about 10 frames per second, which is sufficiently high in order to lock and follow the target, in case of a relative movement of the target with respect to the vehicle.

5. THE CABLE-CUTTING DEMO

In order to test and validate the implementation of the presented solution, we prepared a SAUVIM mission with the goal of locating and cutting a cable, using a special tool integrated with the gripper. The cable passes through a sphere of 10 cm in diameter, which simplifies its localization in the space, and has one end attached to a floating device.

Figure 7 shows few snapshots of the operation. In details, the sequence of tasks performed to achieve the final goal is the following:

- Navigate to the target site.
- Extract the arm and perform a visual scan of the surrounding space, using the attached camera.
- During the scan, try to locate the target (in our case the sphere of fig. 8)
- Once the target has been detected, the arm enters in a tracking state, in order to place the gripper to a constant relative position with respect to the sphere. If the target moves, the arm follows it while maintaining the same relative position.
- Once the tracking system detects no movements of the target for a sufficient amount of time, the arm proceeds with the cutting task (fig. 7), which is performed in 2 seconds with the aid of a special cutting tool integrated within the gripper. During the time frame of the cutting operation, a small movement of the target with respect to the arm may still be corrected using the video feedback.

The only required human intervention is the decision on when starting the visual scan. Once the vehicle has reached the target site, it is matter of the supervisor to decide if the target is the correct one and, successively, to start the autonomous sequence of operation. From now on, the arm performs autonomously all the subsequent operations in order to reach the goal of cutting the cable. If any errors arises during any of the above autonomous steps (for example no target has been detected during the scan), the arm returns in its initial parking position and the mission is aborted. A mission log allows the operator to verify later the cause of the failure.

6. CONCLUSION

Today's underwater intervention tasks are performed with extensive human supervision, requiring high-bandwidth communication links, which results in limited applications. Autonomous underwater manipulation systems will make it possible to sense and perform mechanical work in areas that where humans cannot go, such as deep-ocean or under-ice explorations.

Autonomous manipulation for underwater intervention is a challenging technology milestone, and today there are still very few examples worldwide of applications using such technology.

In this paper we presented one of the first approach, implemented within the SAUVIM manipulation system. The solution has been tested with an experimental demo where the vehicle-manipulation system was used to cut a submerged cable. This operation could be seen as an autonomous rescue intervention task and results show that the presented approach is promising for autonomous underwater manipulation .

7. ACKNOWLEDGEMENTS

This work was sponsored in part by ONR under grant N00014-97-1-0961, N00014-00-1-0629, N00014-02-1-0840, N00014-03-1-0969, and N00014-04-1-0751. Any opinions, findings and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

E. Kim, M. Haseyame and H. Kitajima (2005). The extraction of circles from arcs represented by extended digital lines. *IEICE Transactions on Information and Systems*.

Funda, J., T. S. Lindsay and R. P. Paul (1992). Teleprogramming: Toward delay invariant remote manipulation. *Presence* (1), 29–44.

Kim, J., G. Marani, W. K. Chung, J. Yuh and S. R. Oh (2002). Dynamic task priority approach to avoid kinematic singularity for autonomous manipulation. In: *Proc., IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. pp. 1942–1947.

Marani, G., I. Medrano, S. K. Choi and J. Yuh (2005). A client-server oriented programming language for autonomous underwater manipulation. In: *The Proceedings of The Fifteenth (2005) International Offshore and Polar Engineering Conference*. Seoul, Korea.

Marani, G., J. Kim, J. Yuh and W. K. Chung (2002). A real-time approach for singularity avoidance in resolved motion rate control of robotic manipulators. In: *Proc., IEEE Int. Conf. on Robotics and Automation*. pp. 1973–1978.

Marani, G., J. Kim, W. K. Chung and J. Yuh (2003). Algorithmic singularities avoidance in task-priority based controller for redundant manipulators. In: *Proc., IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. pp. 1942–1947.

Marani, G., J. Yuh and S. K. Choi (2006). Autonomous manipulation for an intervention auv. In: *Guidance and Control of Unmanned Marine Vehicles* (G. Roberts and B. Sutton, Eds.). pp. 217–237. IEE's Control Engineering Series.

Paul, R. P., C. P. Sayers and M. R. Stein (1993). The theory of teleprogramming. *Journal of the Robotics Society of Japan* **11**(6), 14–19.

Sayers, C. P., R. P. Paul, J. Catipovic, L. Whitcomb and D. Yoerger (1998). Teleprogramming for subsea teleoperation using acoustic communication. *IEEE Journal of Oceanic Engineering* **23**(1), 60–71.

Yuh, J. and S. K. Choi (1999). Semi-autonomous underwater vehicle for intervention mission: An auv that does more than just swim. *Sea Technology* **40**(10), 37–42.