

Realtime Motion Compensation for ROV-based Tele-operated Underwater Manipulators

Marc Hildebrandt
Underwater Robotics Department
DFKI Bremen
28359 Bremen, Germany
Email: marc.hildebrandt@dfki.de

Leif Christensen

Jochen Kerdels

Jan Albiez

Frank Kirchner

Abstract—This paper presents a novel underwater movement compensation algorithm for stabilization of manipulator position utilizing not ROV movements for disturbance compensation, but overlaid manipulator movements. A model based estimator is used to predict vehicle movement and provide the manipulation system with the necessary time to compensate for the estimated motion. It describes the conceptual benefits of this approach compared with common station-keeping algorithms, and shows how previous methods can be combined with the new approach in order to further improve manipulator position accuracy. The method is validated in a number of experiments, which show its feasibility and outstanding performance.

I. INTRODUCTION

ROV-based manipulators are one of the few possibilities to manipulate objects in the deep sea environment. With increasing demand for facilities and structures in depths beyond the access of human divers and the increasing complexity of such structures, the challenges and applications for manipulator systems are steadily increasing. In order to match these requirements either new manipulation paradigms or novel control systems are imperative. This paper focuses on the possibility to assist human operators in complex tasks executed with ROV-based tele-operated manipulators by introducing a motion-compensation layer between the operator and the manipulator, greatly improving the performance of such systems.

II. PROBLEM ANALYSIS

A ROV-based manipulator system suffers the problem of two coupled high-DOF systems: the 6 DOF ROV platform plus the (usual) 4-7 DOF manipulator. In a Heavy-Workclass ROV these two systems typically are operated by two individual operators, one steering the ROV and the second performing the tele-operated manipulation. One of the main tasks of the ROV-operator is keeping the ROV as steady as possible to avoid additional movement of the manipulator, especially the end-effector of the manipulator, since this is the interaction point with the environment. From a manipulation perspective these added 6 DOF severely aggravate the manipulation problem, and need to be kept as small as possible.

This problem may be addressed in two different ways: either by keeping the ROV's position as stable as possible using station keeping algorithms or else by recognizing and compensating for the ROV movement by correction of the

manipulator end-effector position. The former has been topic of intense research, and a number of well-performing methods exist (e.g. [1], [2], [3]). ROV motion compensation using the manipulator is scarcely seen in underwater systems, since manipulator systems are usually directly tele-operated, with little to no computer control, making automated compensation algorithms difficult to implement. In principle such an algorithm should outperform a station-keeping algorithm, since a manipulator generally has more dynamic movement properties than a ROV. This should result in higher precision of the movement compensation. The implementation and evaluation of a manipulator based compensation algorithm is the topic of this paper.

The general problem with underwater manipulators is their inadequacy in respect to computer control. Designed as tele-operated master-slave systems for harsh environments and rough manipulation tasks they lack the precision and sensory feedback of industrial manipulators, making the transfer of algorithms from traditional manipulation systems to underwater manipulators very difficult. The direct consequence is that widespread hydraulic underwater manipulators are hardly used for research on manipulation topics. Either customized manipulation systems are utilized (e.g. the 500m rated dual-arm workcell of the AMADEUS phase II project [4] or the electric MARIS 7080 in the SAUVIM project [5], [6], [7]), or during experimental validation phases industrial robots are used (e.g. [8] describes a visual servoing algorithm for uninstrumented underwater manipulators which is tested on an industrial manipulator). Both workarounds suffer from severe drawbacks: Using custom manipulator systems prevents application of the developed algorithms on commercial underwater vehicles, since the capabilities of the manipulator systems differ. Designing algorithms for existing underwater manipulators but testing them only on industrial manipulators merely achieves a 'proof of concept' validation, saying little about the transferability to the original target system even if the different manipulator capabilities are tried to account for. These considerations are the reason that the authors design and test their algorithms on a real underwater manipulator, the hydraulic Schilling Robotics Orion7P with all its limitations and drawbacks.

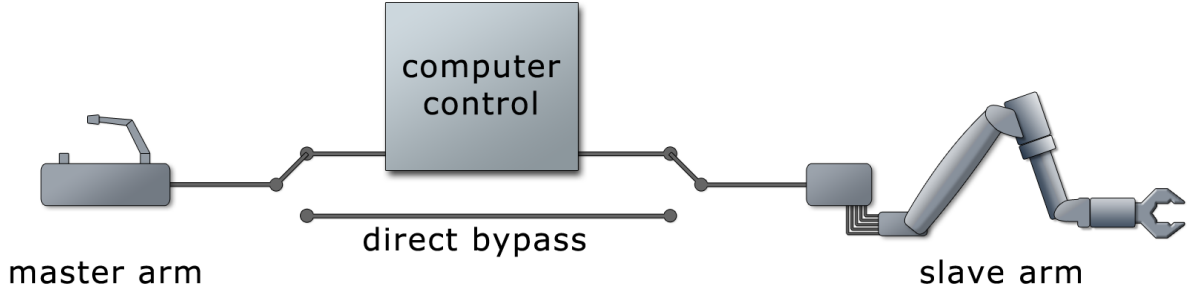


Fig. 1. Control flow in the manipulator system: the Orion7P may be directly controlled (using the bypass) or with a computer controlled middle layer included.

III. PROPOSED SOLUTION

The proposed method described in this paper addresses the problem by actively compensating for ROV movement with the manipulation system. The method is realtime capable and works without major changes on existing systems, making the integration into existing systems straightforward and easy. Its performance was verified using an Orion7P from Schilling Robotics [9] integrated into the ROV simulator at the underwater testbed of the German Research Center for Artificial Intelligence (DFKI) (for further details see [10]). It is based on the computer control system of the used Orion7P manipulator ([11]), and is designed as intermediate layer between user input and manipulator movement. The compensation algorithm can be divided into three discrete parts: prediction of vehicle movement, calculation of manipulator compensation movement and execution of the compensation movement with the manipulator. Since the control/sampling frequencies of the involved devices do not need to match these three steps run continuously but asynchronously.

The prediction of vehicle movement is based on movement information from the ROV, which can originate from a number of sources: acoustic speed measurement systems like DVL (doppler velocity log), acoustic positioning systems (LBL, SBL or USBL (long- short- or ultra-short-baseline systems)), IMU systems (inertial measurement units) or optical positioning systems utilizing camera information and computer-vision algorithms to name just some popular possibilities. The precision of the retrieved data varies greatly among the different systems, and is subject to a number of external noise sources. In order to best utilize this information it is processed using a mathematical model for the motion to estimate the movement vector for the immediate future. The timescale to be estimated into the future depends on the manipulation system's ability to react to changes in movement. The vehicle model used was the simplified control model described in [12]. Together with the input from [13] an estimator was created, yielding the necessary prediction timeframe of roughly one second. Since in the ROV simulator at the DFKI only translational movements can be simulated, the vehicle model was further simplified to a final 3DOF: surge(x), sway(y) and heave(z).

The resulting motion equation is given by equation 1. \mathbf{P}_e represents the estimated position, \mathbf{P}_t is the sensed position, \mathbf{V}_t and \mathbf{A}_t are the momentary speed/acceleration. μ and λ are two parameters which can be used to adjust the temporal shift of the estimated position. Even though this model appears to be comparably simple, it is fit to estimate a pure translational continuous movement with high precision.

$$\begin{aligned} \mathbf{P}_e &= \mathbf{P}_t + \mu \mathbf{V}_t \Delta t + \lambda \mathbf{A}_t \Delta t^2 \\ \mathbf{P}_t &= \begin{bmatrix} x \\ y \\ z \end{bmatrix} \\ \mathbf{V}_t &= \begin{bmatrix} x_{t-1} - x_t \\ y_{t-1} - y_t \\ z_{t-1} - z_t \end{bmatrix} \\ \mathbf{A}_t &= \begin{bmatrix} v_{xt-1} - v_{xt} \\ v_{yt-1} - v_{yt} \\ v_{zt-1} - v_{zt} \end{bmatrix} \end{aligned} \quad (1)$$

In the next step the manipulator compensation movement has to be calculated. The estimated movement vector $\mathbf{P}_e(t)$ is subtracted from the initial position \mathbf{P}_0 , which results in the compensation vector \mathbf{C} . This vector is added to the original manipulator position \mathbf{M}_0 to receive the compensated cartesian manipulator position \mathbf{M}_t . The whole equation for \mathbf{M}_t is given in 2. The computation of inverse kinematics (which has been shown to work in realtime for the Orion7P by the authors before, see [11]) for \mathbf{M}_t results in the joint space position $\mathbf{J}_t = [\theta_0 \ \theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ \theta_5]^T$. This last step is necessary since the manipulator can only be controlled in joint space. The new joint space position \mathbf{J}_t is given to the manipulator.

$$\mathbf{M}_t = \mathbf{M}_0 + \mathbf{C} = \mathbf{M}_0 + (\mathbf{P}_0 - \mathbf{P}_e(t)) \quad (2)$$

The final step involves moving the manipulator to the desired position. For this purpose the current desired position is approached continuously, and maintained as long as it is valid. A speed controller is used, which controls the joints to move with a speed proportional to their current angle delta. This results in a smooth but fast approach to the desired compensation position of the manipulator. A linear PD-controller is used as speed controller.

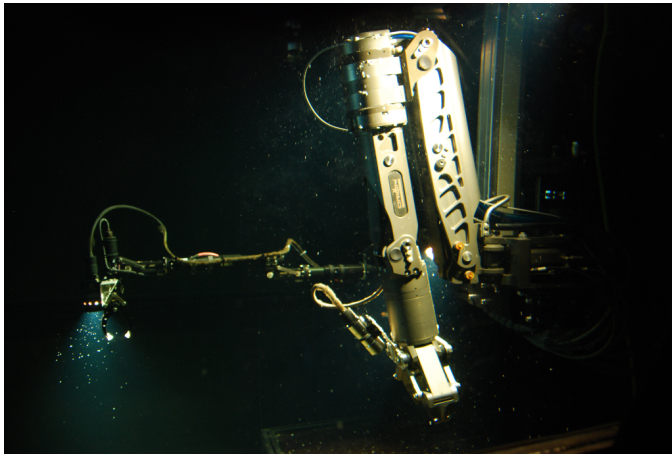


Fig. 2. The experimental setup.

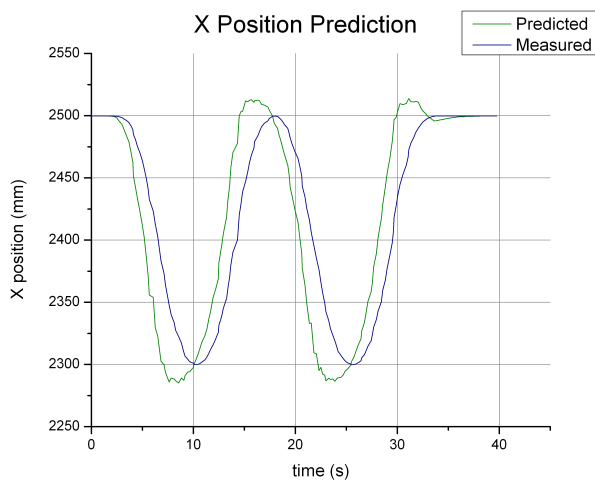


Fig. 3. Measured movement curve for the x axis (blue) and predicted movement (green).

The algorithm is straightforward but expandable to more complex vehicle models (e.g. considering roll, pitch and yaw movements) without overall changes. It results in a considerably dampened end-effector movement when compared to ROV movement. In a noise-free ideal environment the end-effector seems to stand still while the rest of the system moves.

IV. EXPERIMENTAL RESULTS

A number of experiments were conducted at the underwater testbed of the DFKI using the hardware ROV simulator. It consists of a 3d gantry crane installed over a $20m^3$ water basin. Attached to this gantry crane is the hydraulic underwater manipulator Orion7P, a number of cameras, light spots and other instruments commonly found on work-class ROV systems. The setup tries to resemble the front section of a ROV, which can be positioned with high accuracy in all three translational directions by the gantry crane. An image of the system showing the Orion7P and the tip of the gantry crane can be seen in figure 2. The experiments involved moving

TABLE I
COMPARISON OF COMPENSATION PERFORMANCE FOR DIFFERENT MOVEMENT PATTERNS AND SPEEDS

Movement radius (x/y/z)[cm]	Speed [cm/s]	Deviation [cm]
20 / 20 / 0	20	x 7.74
		y 6.85
		z 2.02
20 / 20 / 10	10	x 3.3
		y 2.53
		z 0.99
20 / 20 / 0	10	x 3.55
		y 2.9
		y 0.59
20 / 20 / 0	5	x 1.49
		y 1.69
		y 0.27
20 / 20 / 10	5	x 1.66
		y 1.62
		y 0.37
10 / 10 / 10	10	x 2.97
		y 2.73
		y 2.56
10 / 10 / 10	5	x 1.59
		y 1.27
		y 0.98
10 / 10 / 10	2	x 0.8
		y 0.5
		y 0.33

the gantry crane in a sinuoid trajectory resembling ROV movement. The sinoid trajectory can be considered as a 'worst-case' scenario, due to the constant acceleration and deceleration of the overall system. Therefore it is a valuable benchmark for system performance. Amplitude and Frequency for the testing trajectories have been selected to be in the same order of magnitude and shape as were used by [13]. Movement radii between 0 and 20 cm with speeds of 2-20 cm/s were used. The gantry crane is able to move at much higher speeds, but such movements seem inadequate for the selected scenarios, where speeds below 10 cm/s are expected.

The position update frequency was estimated to be rather low, since most of the previously described movement measurement methods have low sampling rates. The consequence is that the estimation will not be able to handle high-frequency movements or extreme amplitude changes. This is not considered to be a major problem, since the use-case for this algorithm does not allow for more extreme movement profiles. For the experiments an update rate of 5 Hz was used.

The performance of the position predictor is shown in figure 3. As expected it shows some overshoot in areas of movement direction inversion and some ripple noise. The performance strongly resembles the results obtained in [13], supporting the correctness of our simplified prediction model. The ripple noise mainly originates in the acceleration part of equation 1, which led to the idea of filtering the acceleration using an infinite-response filter with a window width of 5. This strongly reduced the ripple without harming the dynamic properties of the controller. The experimental results for the complete compensation algorithm are summarized in table III. All movement trials were conducted using the same, previously optimized set of μ/λ parameters. As expected, the performance

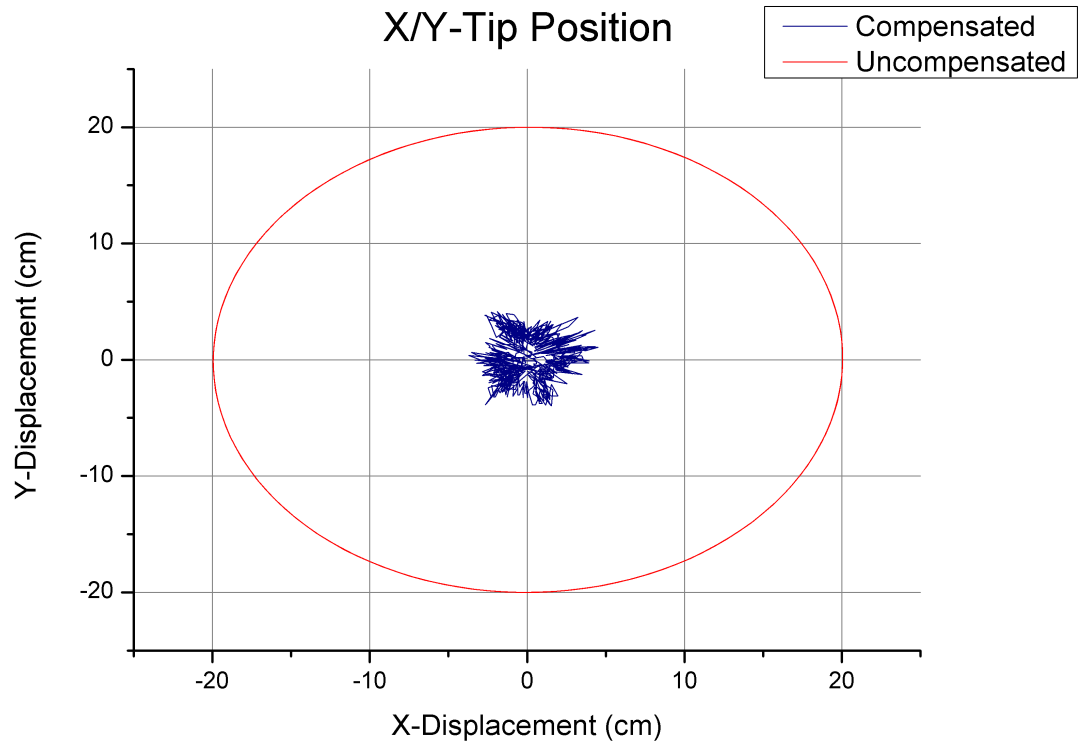


Fig. 4. Plot of the tip displacement while compensating a 20/20/0 cm movement with a 5 cm/s speed. Overlaid is the movement without compensation.

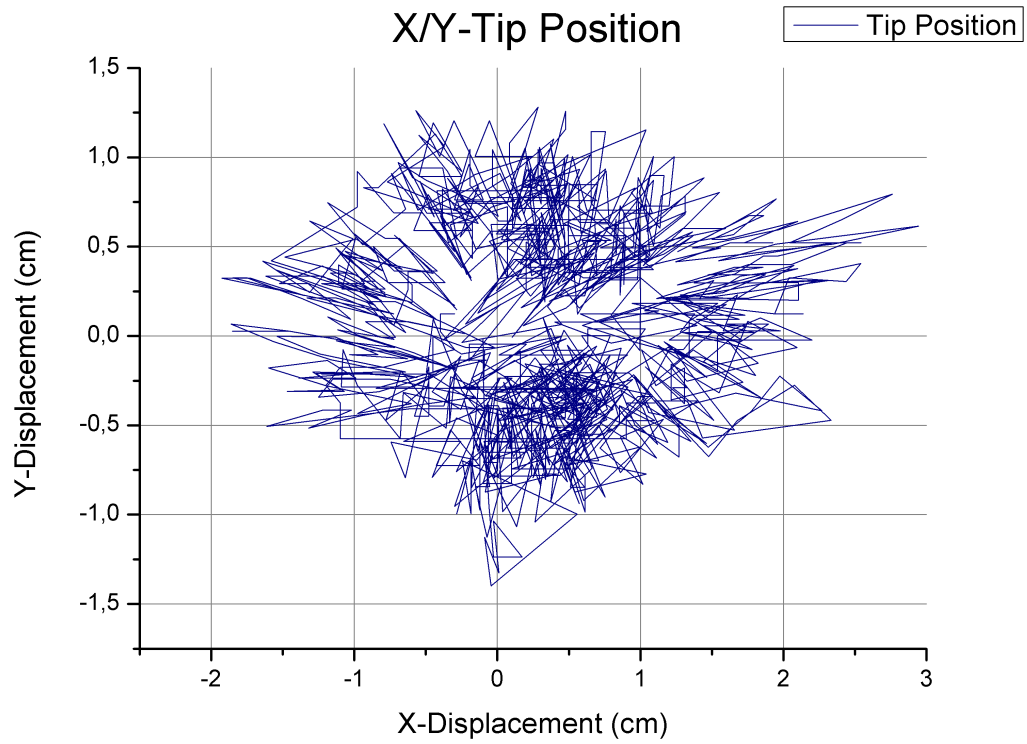


Fig. 5. Plot of the tip displacement while compensating a 10/10/10 cm movement with a 2 cm/s speed

X/Y/Z-Tip Position

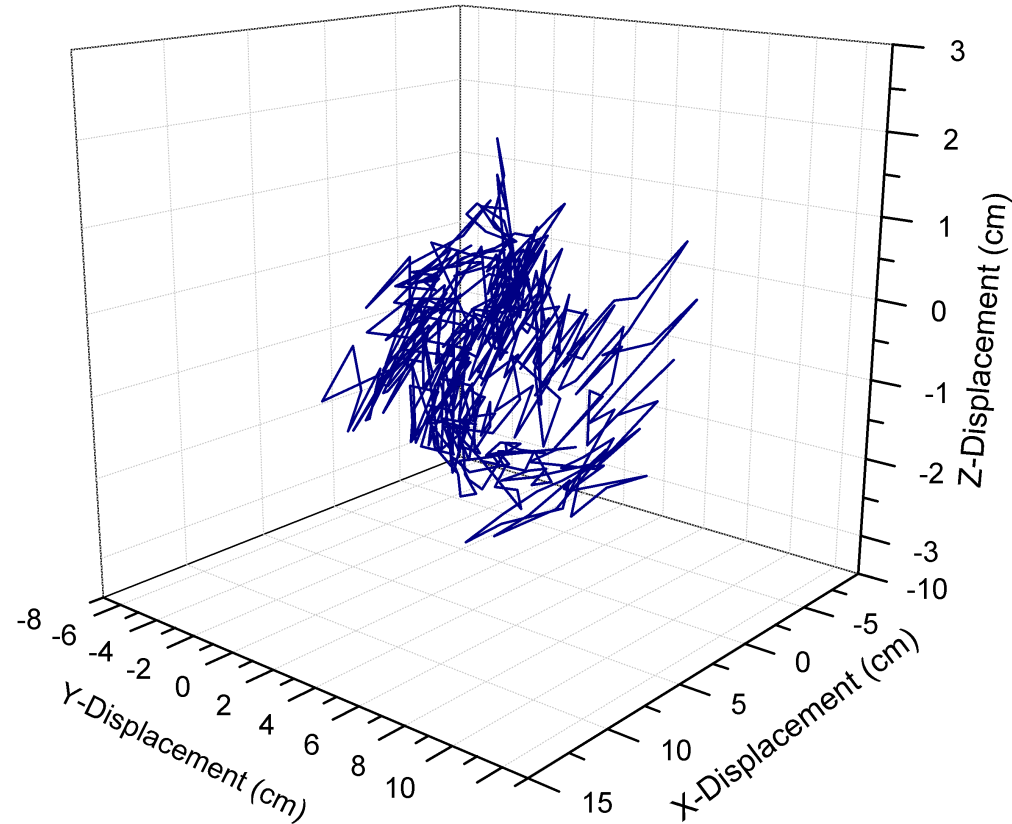


Fig. 6. 3d plot of the tip displacement while compensating a 20/20/10 cm movement with a 10 cm/s speed

of the motion compensation is directly dependant on movement speed. Interestingly the movement magnitude seems to affect the performance as well, while speed has much stronger impact than magnitude. This is attributed to the fact, that for the higher magnitudes the manipulator reaches the boundaries of its workspace, resulting in extreme joint positions and the decreased accuracy attributed to such configurations. For a reasonable speed of 5 cm/s and movement magnitudes between 20 and 10 cm a position deviation lower than 1.7 cm could be achieved, which can even be reduced to below-cm accuracy when operating at 2 cm/s. These accuracies lie well inside the movement accuracies of the Orion7P manipulation system, and are not expected to be significantly improvable without modifications to the system. In any case it is not possible with the supplied master-arm to move an ORION 7P with such precision. The tip position deviation is shown in figures 5 and 6. In order to give a scale for comparison, figure 4 shows the compensated tip position vs. the uncompensated tip position.

V. CONCLUSION

A complete ROV movement compensation algorithm was implemented, and its effectiveness shown in a number of experiments on a hydraulic deep-sea manipulator. The resulting movement compensation accuracy of under 1 cm is unparalleled for an unmodified ROV-manipulation system. The robustness of the proposed algorithm was tested by greatly increasing the movement speed beyond the expected magnitudes without increasing the compensation deviation disproportionately.

In order to compare the performance of the compensation system to the performance of human operators, a series of trials involving a number of differently proficient manipulator operators is planned. It is expected to provide valuable comparison data. Future work will include testing the compensation algorithm with real ROV trajectories obtained by a FOG-system (fiber-optic gyro). A number of real system tests are also scheduled, where the system will run on an actual work-class ROV. The movement model will be adapted to incorporate roll, pitch and yaw movement prior to these trials. Future work will also include the coupling of a ROV station keeping algorithm with our proposed movement compensation method,

in order to further improve the effectiveness of this operator support system. Since the Orion7P manipulator used in our ROV simulator has a wrist-camera, another idea to increase the accuracy of the motion compensation is to incorporate movement information obtained using optical-flow based algorithms on the wrist-cam image. This should improve the performance especially at higher speeds or during movements which the vehicle movement model does not account for.

ACKNOWLEDGMENT

The authors would like to thank all colleagues at the DFKI Bremen for their support and feedback. The work presented here is part of the CManipulator project, funded by the German Ministry of Economics (BMWI), grant No. 03SX231.

REFERENCES

- [1] M. W. Dunnigan and G. T. Russell, "Reduction of the dynamic coupling between a manipulator and roV using variable structure control," in *CONTROL94. Conference Publication No. 389.0, MTS/ IEEE Oceans*. IEEE, April 1994.
- [2] J.-H. Ryu, D.-S. Kwon, and P.-M. Lee, "Control of underwater manipulators mounted on an roV using base force information," in *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*. IEEE, may 2001.
- [3] G. Antonelli, *Underwater Robots*, B. Siciliano, O. Khatib, and F. Groen, Eds. Springer, 2006.
- [4] G. Casalino, D. Angeletti, T. Bozzo, and G. Marani, "Dextreous underwater object manipulation via multirobot cooperating systems," in *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, May 2001, pp. 3220–3225.
- [5] J. Yuh, S. Choi, C. Ikehara, G. Kim, G. McMurty, M. Ghasemi-Nejhad, N. Sarkar, and K. Sugihara, "Design of a semi-autonomous underwater vehicle for intervention missions (sauvim)," Apr 1998, pp. 63–68.
- [6] G. Marani, J. Yuh, and S. Choi, *Advances in Unmanned Marine Vehicles*. IET, 2005, ch. 10: Autonomous manipulation for an intervention AUV, pp. 217–237.
- [7] G. Marani, S. K. Choi, and J. Yuh, "Underwater autonomous manipulation for intervention missions auvs," *Ocean Engineering*, vol. 36, no. 1, pp. 15 – 23, 2009, aUV. [Online]. Available: <http://www.sciencedirect.com/science/article/B6V4F-4T7F5PD-1/2/ce310fd84710cc86d5ffd871b95edd70>
- [8] E. Marchand, F. Chaumette, F. Spindler, and M. Perrier, "Controlling an uninstrumented manipulator by visual servoing," in *The International Journal of Robotics Research*. SAGE Publications, July 2002, pp. 635–647.
- [9] S. Robotics, *Orion7P Technical Manual*, 1st ed., Schilling Robotics, Davis, CA, 2007.
- [10] L. Christensen, P. Kampmann, M. Hildebrandt, J. Albiez, and F. Kirchner, "Hardware roV simulation facility for the evaluation of novel underwater manipulation techniques," May 2009.
- [11] M. Hildebrandt, J. Albiez, and F. Kirchner, "Computer-based control of deep-sea manipulators," *OCEANS 2008 - MTS/IEEE Kobe Techno-Ocean*, pp. 1–6, April 2008.
- [12] T. I. Fossen, *Marine Control Systems - Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*. Trondheim, Norway: Marine Cybernetics, 2002.
- [13] D. A. Smallwood and L. L. Whitcomb, "Model-based dynamic positioning of underwater robotic vehicles: Theory and experiment," *IEEE Journal of Oceanic Engineering*, 2004.