Automatic workspace analysis and vehicle adaptation for hydraulic underwater manipulators

Conference Paper · November 2009

DOI: 10.23919/OCEANS.2009.5422068 · Source: IEEE Xplore

4 authors:


Jan Christian Albiez
Kraken Robotik GmbH, Bremen, Germany
107 PUBLICATIONS **1,096** CITATIONS
SEE PROFILE


Marc Hildebrandt
Deutsches Forschungszentrum für Künstliche Intelligenz
29 PUBLICATIONS **276** CITATIONS
SEE PROFILE


Jochen Kerdels
FernUniversität in Hagen
36 PUBLICATIONS **131** CITATIONS
SEE PROFILE


Frank Kirchner
Universität Bremen
313 PUBLICATIONS **2,696** CITATIONS
SEE PROFILE

Automatic workspace analysis and vehicle adaptation for hydraulic underwater manipulators

Jan Albiez* , Marc Hildebrandt*, Jochen Kerdels* and Frank Kirchner*
*Underwater Robotics Department
DFKI Bremen, Robotics Innovation Center
Germany, 28359 Bremen
Email: jan.albiez@dfki.de

***Abstract*— In this paper we describe a methodology to represent the workingspace properties of an underwater manipulator with respect to tele-operation tasks. This representation takes the typical operation procedure for intervention tasks into account. The information gained from the representation can be used to automatically analyse the actual state of a manipulator with respect to the remaining dexterity of the manipulator.**

The workingspace representation is formally introduced and exemplary used on an ORION 7P from Schilling Robotics. The second half of the paper deals with an algorithm that is able to calculate online the distance between the actual position and the border to regions with less dexterity. The information gained by this algorithm is then used as a signal to an operator or as the basis for motion commands to the ROV carrying the manipulator.

I. INTRODUCTION

The main tool of modern work-class ROVs are manipulators. These manipulators are usually controlled by an operator via a master-slave-system. The only knowledge an operator has about the movements the arm is able to perform in its current configuration is based on the views of the ROV's cameras and his experience. There is no direct feedback to him, wether he can reach a desired position, and how much he can rotate the wrist at this position to do the task he is aiming for. This is a serious limitation considering the fact, that one of the primary tasks of modern ROV deployment is intervention work. Intervention in these cases consists of more an more task like opening fixtures, plugging connections etc. A strategy to circumvent the problem of the manipulators unknown movement capabilities is to fix the arm to a defined position and move the vehicle instead. This is done especially in cases where the ROV can move freely in the water column, despite of modern ROVs capabilities of automatic station keeping.

In robotics the ability of a robotic arm to reach a certain position in his working space and to be able to orient its tool is called *dexterity*. The part of the working space which allows the movement in all euclidian degrees of freedom is called the *full-dexterous workspace*. As the workspace is a multi-dimensional space, the calculation of the full-dexterous workspace is a hard tasks computational wise, and classical robot control algorithms try not to calculate it directly and rely on numerical methods instead (e.g. [1]).

For tele-operation this task is even more harder. Instead of using pre-calculated or online adapted trajectories, the arm movement is controlled by an human operator. Any boundary conditions arising from the planned trajectory of computer controlled system and minimising the computational effort are not available when a human controls the system. The only way to come up with something similar would be a system which predicts the movement the operator intents, which would add an additional source of inaccuracy to the whole system. Additionally, currently used master-slave systems mainly rely on pure angular coupling between master and slave and avoid the calculation of the inverse and direct kinematic solutions, which again are needed for the above mentioned workspace calculation algorithms.

The work presented in this paper deals with a system allowing to pass information about the current limits of the robots arm dexterity to an operator or the control-system of the ROV. At first we give a general introduction into the definition of workspaces and present an overview of current methods to analyse them. Then we present a new tele-operation centered representation of the workspace, which allows a quantification of the manipulator's dexterity at a given position. This representation is used to analyse the workspace of an ORION 7P from Schilling Robotics. The second part of the paper deals with an real-time capable algorithm using the new workspace representation to give the operator online information about the current dexterity of his manipulator.

II. WORKSPACE CALCULATION & REPRESENTATION

A. Introduction

The workspace of a robot arm is the set of all points the robot end-effector (commonly also denoted as *Tool Center Point (TCP)*) can reach. The working space can be represented either in angular or Cartesian space. In angular representation the working space has a dimension equal to the number of joints of the robot and represents all possible angular configurations. The Cartesian representation has the dimension of six, and represents all possible cartesian positions and rotations the TCP of the robot can reach. Formally this is described as:

$$WS_{eucl} \subset \Re^6 \quad (1)$$

$$WS_{joint} \subset \Re^n \text{ , } n = \#\text{joints} \quad (2)$$

Fig. 1. The ORION 7P mounted on the ROV simulation rig at the DFKI underwater testbed. This system is used to verify the methods introduced in this paper.

The main problem of any workspace analysis is the high dimensionality, even the main manipulators of ROVs have usually 6 DoF. Due to this fact most common workspace analysis methods concentrate on the space around the actual TCP or the position in the current trajectory.

B. Approaches

In most robotics application the main reason to know more about the workspace of a robot is for trajectory planning, obstacle avoidance and force control. Most commonly used methods are based on the online calculation of the Jacobian matrix (see e.g. [1]). The ellipsoids of operational velocities, which determine the movement space of the TCP, are the square roots of the eigenvalues λ_i determined by solving $det(JJ^T - \lambda I)$. By these method a manipulation index can be extracted which defines the "Ability to arbitrarily change the position and orientation of the robot end-effector" [4].

The calculation of the Jacobian proves in itself to be a numerically extreme hard problem, which is normally solved locally. [5] describes the most classical algorithm and is the basis for most force-controlled systems today. In [6] a simplistic and numerical stable approach for calculating the Transpose Jacobian is presented, which uses feedback linearisation and a local movement history. [7], [8], [9] and [10] all use an Jacobian calculation in some way or the other.

Furthermore several graphical methods exist, most of them using some sort of binary trees or bounding boxes to set up a hierarchical model to reduce complexity [11].

C. Workspace in Tele-Operation

The requirements for a workspace representation differ between tele-manipulation with ROVs and computer control. From a tele-operation point of view, with respect to the state-of-the-art ROV robot arm, the following requirements can be derived:

- Dexterity-representation of the arm at a certain position
- Distance to a virtual border where the robot arm looses its dexterity with respect to the current working direction
- Online calculation of the points above
- "Ideal" working area: Points with the maximum distance to regions in the working space where the arm movement is seriously constricted

Taking into account the abilities of the currently used arms and the established procedures in the offshore industry, the following problems arise:

- The sensor feedback of most deep-sea arms is limited to joint-angle data. Calculating the Jacobian requires angluar speed which can only be acquired by numerical derivation. This destabilises the online calculation of the Jacobian.
- Most systems don't have more than qualitative force measurement.
- The arms of work class ROVs are hydraulically activated with a limited range of movement. None of the currently used system has full euclidian dexterity.
- The position accuracy of the arms is low compared to land-based systems and the movement accuracy of the ROV carrying the arm is not very high due to environmental constraints (e.g. weight and size of the ROV, hydrodynamic resistance of the umbilical). This means that land-based approaches of mobile manipulators can not be transfered easily.
- Acceptance of computer control in the offshore industry is very low. A workspace representation must seamlessly integrate with a human operator.

In the following we introduce a workspace representation which takes these requirements and problems into account.

D. A Semi-Dexterous Workspace Representation

To limit the influences on the overall system dynamic, the speed of the arm movement during intervention tasks is very slow. This is even more true compared to the calculation power of todays computer systems. It is therefore feasible to limit the workspace representation to the static case. More problematic is the limited movement range of the arms. Aiming for the fully dexterous case is pointless since it only covers a fraction of the WS_{eucl}, if it exists at all.

When analysing the movements of a manipulator during intervention tasks, it is remarkable that there is always a main manipulation direction. The ROV is placed at a certain angle in front of the target, the arm is moved more closely to the target with the wrist approaching from an opportune direction and then the designated task is solved. During the task itself the orientation of the wrist only changes in a limited scope. This is especially true for opening fasteners, picking up objects, connecting plugs or operating scientific equipment. This fact can be used as basis to define a subset of the overall workingspace WS which is more suited to ROVs tele-manipulators:

Semi-Dexterous Workspace: The semi-dexterous workspace $SDWS_{\vec{d}}$ is defined as the subset of WS_{eucl}, which allows *euclidian orientation* $\vec{d} \in \Re^3$ of the TCP-Frame with respect to a given direction and an *angular variation* $\hat{d} \in \Re^3$ around this direction.

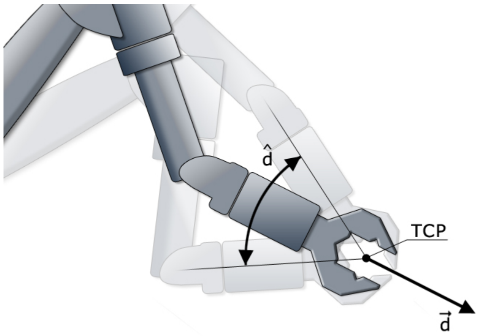


Fig. 2. Diagram of $\vec{d}$ and $\hat{d}$ for a 2D case.

In Fig. 2 $\vec{d}$ and $\hat{d}$ are shown for a two dimensional case.

The ideal maximum angular variation $\hat{d}_{max}$ is $\pm 90^\circ$ for all euclidian angles, which would result in a half-sphere. It is clear, that this is not the case for most commercially available underwater manipulators.

Looking more closely at the implication of $\hat{d}$ two things are obvious:

1) $\hat{d}$ is not constant for all points in $SDWS_{\vec{d}}$.
2) $|\hat{d}|$ denotes the amount of movement the TCP can do at a given position with respect to $\vec{d}$.

$|\hat{d}|$ gives us the possibility to define a *degree of dexterity* for every point in a $SDWS$:

Rating: The rating $r = \text{rating}(\vec{p}) \in \Re$ of a point $\vec{p} \in SDWS_{\vec{d}}$ is defined as $|\hat{d}_{max}|$ at $\vec{p}$. It denotes the maximum angular movement of the TCP-Frame with respect to $\vec{d}$.

With $SDWS$, $\vec{d}$ and r the most important tools for giving a qualitative and quantitative information about the dexterity of a ROV manipulator are defined. Considering an operator conducting an intervention task has a system that allows him to query these values whenever he likes, it is possible for him to estimate if the arm has enough dexterity to fulfill this task.

In Praxis the continuous notation of $\vec{d}$ is not feasible. More practically is a system which uses the main working direction as a basis, since in most cases the vehicle is place somewhere in front of the target. The main indicator for the working direction is the general orientation of the wrist with respect to the manipulator base:

Front The wrist is oriented in the direction of the vehicle
Down The wrist is oriented downwards (e.g. the sea-floor)
Up The wrist is oriented upwards (e.g. the target is above the vehicle)
Left The wrist is oriented to the left (target left to the vehicle)
Right The wist is oriented to the right

These main directions can be combined to get values in between, e.g. *FrontDown* or *LeftDown*. The main working directions depend heavily on the used system configuration and the position of the arm with respect to the vehicle. For the DFKI's single arm ORION system (Fig. 1, [13]) these are *Front*, *FrontDown* and *Down*. Using the "classical" setup in conjunction with an second manipulator the directions would be more oriented towards this arm. Fig. 3 depicts the described combination for the side view.

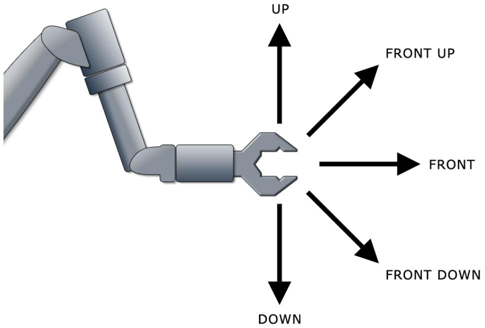


Fig. 3. The primary working directions *Up*, *Front* and *Down* together with two combinations, seen from the side.

Knowing the dexterity of an arm during the manipulation task covers only half of the problem. It would be better to know beforehand, while moving the ROV, where to position the arm in an optimal way. When the operation starts from an arm position that allows for a maximum cartesian movement while till having a desired dexterity, movements of the ROV can be limited.

For a given r and the corresponding $SDWS$ these position can be calculated easily using the geometric mean:

$$\vec{p}_r = \frac{\sum \vec{p} \in A_r}{|A_r|} \text{ with } A_r = \{\vec{p}|\text{rating}(\vec{p}) < r\} \qquad (3)$$

$\vec{p}_r$ can be calculated for every desired r, depending on the amount of dexterity the operator wishes to have. Due to the fact, that $DSWS$ is already oriented in the main working direction, we even don't have to care about to much "unused" space in the opposite direction.

This representation can now be used to analyse the workspace of a robotic arm and generate all the information needed by an operator. In the next section we will show this for an ORION 7P system.

III. WORKSPACE ANALYSIS ORION 7P

The ORION 7P is a widely used hydraulically actuated arm manufactured by Schilling Robotics [12]. It has 6 euclidian degrees of freedom and is configured as a position controlled master-slave system. In the underwater lab of the DFKI Robotics Innovation Center an ORION 7P is mounted on a 3D gantry crane together with a mock-up of a ROV-front (Fig. 1). This setup is used to develop computer control methods to support operators during intervention tasks ([13], [14], [3], [15]). The ORION 7P is symmetrically configured, which means that the 4th rotational degree of freedom (forearm roll) can move equally to the left and right, resulting in a forward oriented main movement. We use this ORION to show exemplarily the results of a workspace representation and analysis as described above.

TABLE I

PARAMETERS FOR THE ORION 7P $SDWS$ CREATION.

Cartesian	**Min**	**Max**
X	$-10cm$	$180cm$
Y	$-140cm$	$140cm$
Z	$-155cm$	$120cm$
Resolution		$0.5cm$
Angular	**Min**	**Max**
Angle 1	$-45\,^\circ$	$45\,^\circ$
Angle 2	$-45\,^\circ$	$45\,^\circ$
Angle-Resolution		$1\,^\circ$
Maximum Possible Rating		8182

Due to the used ORION 7P configuration the most interesting $SDWS$ are for $SDWS_{Front}$, $SDWS_{FrontDown}$ and $SDWS_{Down}$. The wrist roll angle of the ORION can continuously rotate $360\,^\circ$ and was therefore omitted from the rating calculation. Angle 1 and Angle 2 are the remaining two angles describing the respective $SDWS$.

We implemented a multi-threaded software which uses the fast inverse kinematic solution presented in [2] to scan a 3D bounding box, calculating the rating for every position. The workspace can be saved after the creation and the data used for the online adaptation. The workspace itself is discretised.

Since the real dimension of the respective $SD-WS$ was prior unknown, we used the parameters shown in table I. As the resolution for the cartesian and angular discretisation we used values which are below the precision achievable by the ORION 7P master-controller. The calculations have been done offline on an INTEL XEON ?? dual processor system using all 16 virtual cores. Each calculation lasted approximately $3.5h$.

In Fig. 4 the angular range of the ORION 7P is depicted. These graphics show the "real" workingspace of the arm, without respect to dexterity. In Fig 5 two analysis of the $SDWS$. Depicted are $SDWS_{Front}$ and $SDWS_{FrontDown}$. The graphs show the ratings in form of a color gradient. Looking at these graphs it is quite obvious why a automatic dexterity control system is needed. The show how much the dexterity of the system degrades inside its working space. For an operator this proves to be a serious limitation.

IV. ONLINE ADAPTATION

In this section we will introduce an automatic system which allows an operator to get information about the current dexterity and to move the ROV accordingly, using the results of the previous analysis.

A. Border Distance

Using the precalculated workspace rating and a given TCP position $\vec{p}$ we can calculate the distance s_v to a virtual border $\vec{v}$ where the robot arm looses its dexterity with respect to the current working direction and a given minimum threshold m_t as follows:

$$s_v = |\vec{v} - \vec{p}|$$

The algorithm used to determine the virtual border $\vec{v}$ has to be able to extract this information in real-time from the precalculated workspace rating data. Our approach uses a *bisection* algorithm which has a logarithmic runtime in relation to the maximum distance between the given TCP position and the outmost border $\vec{o}$ of the working space and the choosen precision ϵ for the given direction. The algorithm is depicted in figure 6 and subsequently described in pseudo-code:

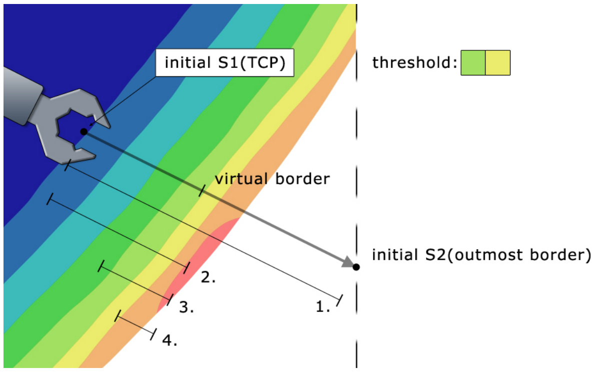


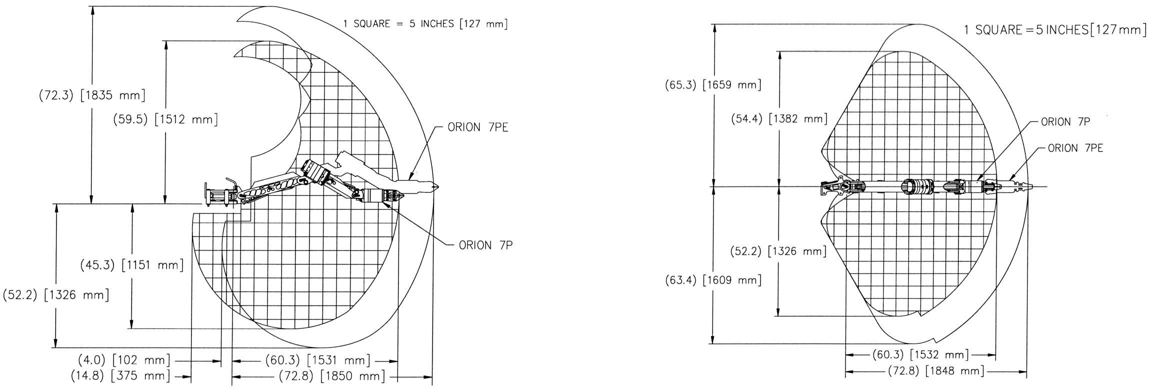
Fig. 6. Depiction of the used bisection algorithm with initial points s1 (TCP) and s2 (outmost border). Four consecutive intervals are shown and labeled respectively. The used threshold is a value corresponding to the transition between light green and yellow. The result of the algorithm is the virtual border point.

```
s1 := p;
s2 := o;
while precision > e do begin
  sm := (s2 + s1) / 2;
  if rating at sm > m_t then
    s1 := sm;
  else
    s2 := sm;
end
return v := (s2 + s1) / 2;
```

The algorithm uses the assumption that the rating space of the given manipulator is smooth and "sufficiently" convex. If this precondition is not met, a linear preprocessing step is needed to find the initial positions `s1` and `s2` for the bisection algorithm. This is done by performing a linear scan in the current working direction starting at the given TCP position until the particular rating falls below a given threshold. The last two points of this scan are then used as the initial points `s1` and `s2` for the bisection algorithm. The resolution of this preprocessing scan depends on the characteristics of the given manipulator system.

B. Operator Interface

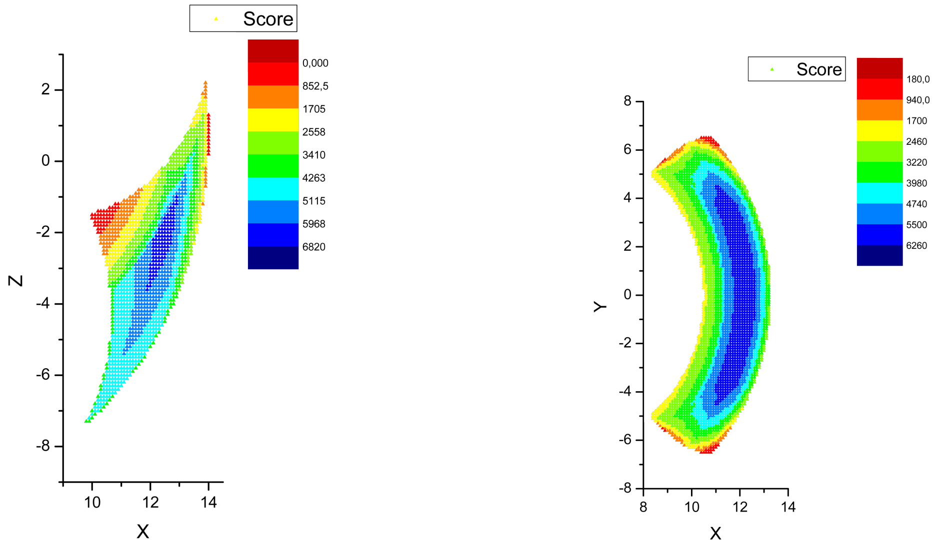
In our system setup the computer control of the ORION 7P is located between the master and the slave controller (Fig. 7). On the one hand this setup allows for the full control of the slave arm by the computer based controller, e.g. for autonomous grasping of objects. On the other hand the setup allows for the continous supervision of the manual control done by an operator using the master arm controller. Using these supervision capabilities the computer continously



(a) ORION angular workingspace XZ plane

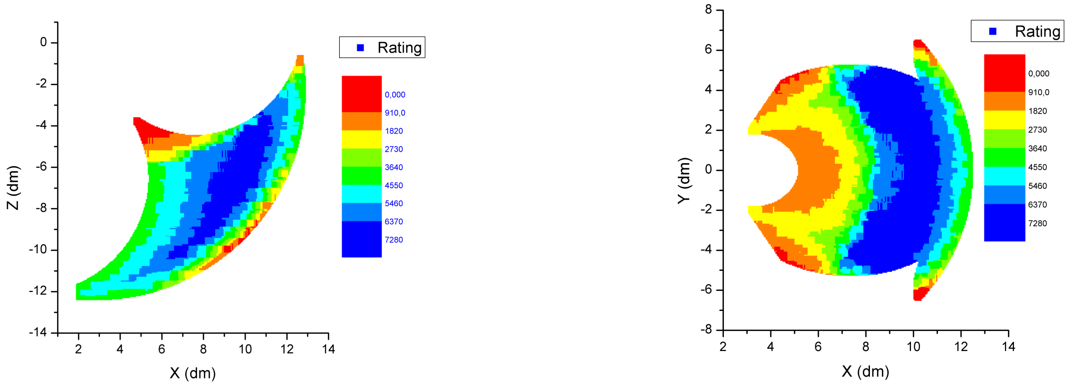
(b) ORION angular workingspace XY plane

Fig. 4. Motion ranges of the ORION 7P [12]



(a) DSW_{Down} XZ plane, $y = 0.0$

(b) DSW_{Down} XY plane, $y = 0.0$



(c) $DSW_{FrontDown}$ XZ plane, $y = 0.0$

(d) $DSW_{FrontDown}$ XY plane, $y = 0.0$

Fig. 5. DSW_{Down} and $DSW_{FrontDown}$ of the ORION 7P

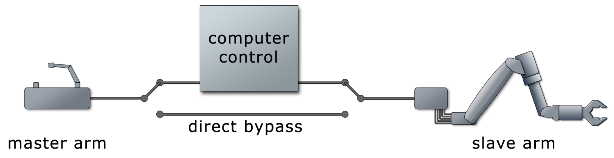


Fig. 7. The system setup used for the operator interface (see also [3]). The computer control is set between the master arm and the slave and can be disabled for security reasons at all times.

calculates the distance to the workspace border in the current working direction and provides this information to the operator.

Using the motion compensation capability of our system (see [3]) it is also possible to use this information to navigate the host system, i.e. the ROV on which the manipulator is mounted, into a better manipulation position while keeping the TCP in place. This mode of operation virtually increases the operating range and workspace of the used manipulator by utilising the movement capabilities of the host system in a semi-autonomous manner. With this setup one operator can control two systems, the host vehicle and the manipulator, simultaneously in a very intuitive manner.

V. CONCLUSION

In this paper a new form of workspace representation, focusing on the demands of a tele-operated system, was presented. The results of the modeling were used to create an automatic online adaption scheme, which enables an operator to get information about the current dexterity, the ranges were the dexterity doesn't exceed a given limit and the possibility for automatic movement.

One of the main drawbacks of the actual system is, that r is dimensionless, and that we currently do not hold information about points which can be reached, but without higher dexterity. We plan to improve the system in the future trying to solve this problem and we will conduct more experiments in our tank to get operator feedback for improvements.

ACKNOWLEDGMENT

The authors would like to thank all colleagues at the DFKI Robotics Innovation Center for their support and feedback. The work presented here is part of the CManipulator project, funded by the German Ministry of Economics and Technology (BMWI), grant No. 03SX231 and the VI-Bot project funded by the German Ministry of Education and Science (BMBF), grant No. 01IW07003.

REFERENCES

[1] J. J. Craig, *Introduction to Robotics Mechanics and Control*, 3rd ed. Pearson Prentice Hall, 2005.

[2] 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.

[3] M. Hildebrandt, L. Christensen, J. Kerdels, J. Albiez, and F. Kirchner, "Realtime motion compensation for rov-based teleoperated underwater manipulators," in *Proceedings of the 2009 IEEE OCEANS Conference*, May 2009.

[4] P. H. Chang, "Analysis and control of robot manipulators with kinematic redundancy," Ph.D. dissertation, 1987.

[5] K. O., "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 43–53, 1987.

[6] S. A. A. Moosaviana and E. Papadopoulos, "Modified transpose jacobian control of robotic systems," *Automatica*, vol. 43, pp. 1226–1233, July 2007.

[7] G. Antonelli, *Underwater Robots*, 2nd ed., ser. Springer Tracts in Advanced Robotics, B. Siciliano, O. Khatib, and F. Groen, Eds. Springer, 2006, vol. 2.

[8] 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.

[9] D. C. H. Kyrki, V.; Kragic, "New shortest-path approaches to visual servoing," in *Proceedings of the IIEEE/RSJ on Intelligent Robots and Systems Conference (IROS 2004)*, vol. 1, May 2004, pp. 349–354.

[10] 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.

[11] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, *Robotics: control, sensing, vision, and intelligence*. New York, NY, USA: McGraw-Hill, Inc., 1987.

[12] S. Robotics, *Orion7P Technical Manual*, 1st ed., Schilling Robotics, Davis, CA, 2007.

[13] D. Spenneberg, J. Albiez, F. Kirchner, J. Kerdels, and S. Fechner, "Cmanipulator: An autonomous dual manipulator project for underwater inspection and maintenance," in *Proceedings of OMAE 2007 ASME 2007 International Conference on Offshore and Mechanics and Arctic Engineering*, June 2007.

[14] J. Kerdels, J. Albiez, and F. Kirchner, "A robust vision-based hover control for rov," *OCEANS 2008 - MTS/IEEE Kobe Techno-Ocean*, pp. 1–7, April 2008.

[15] L. Christensen, P. Kampmann, M. Hildebrandt, J. Albiez, and F. Kirchner, "Hardware rov simulation facility for the evaluation of novel underwater manipulation techniques," in *Proceedings of the 2009 IEEE OCEANS Conference*, May 2009.