NULL-SPACE-BASED BEHAVIOR GUIDANCE OF PLANAR DUAL-ARM UVMS

Signe Moe[1], Gianluca Antonelli[2] and Kristin Y. Pettersen[1]

Abstract— **Dual-arm Underwater Vehicle-Manipulator Systems are able perform a variety of interventions tasks, and there are still many challenges related to making such vehicles autonomous. This paper presents a guidance method for generating reference trajectories for such a system by considering the main manipulator and the vehicle base as a leader unit and the secondary manipulator as a follower unit. The desired behavior of the system is expressed as different tasks, and the reference trajectories are calculated using pseudo-inverse Jacobian task matrices and the null-space-based method. The proposed method has been implemented for a particular planar UVMS and simulated for a defined set of tasks. Simulation results confirm the correctness of the proposed method.**

I. INTRODUCTION

Unmanned Underwater Vehicles (UUVs) play an increasingly important role within subsea exploration, including marine archeology, biology and the oil and gas sector. There exists two main types of UUVs: Remotely Operated Vehicles (ROVs) and Autonomous Underwater Vehicles (AUVs). ROVs are linked to a surface ship through a physical tether and are often fully/partly operated by a human operator, whereas AUVs are operated independently and are not tethered [1]. Although research is aiming to increase the degree of autonomy of UUVs, there are still many challenges related to this. AUV survey operations has a high degree of autonomy, while inspection and in particular intervention operations still require a human in the loop, at the very least as an observer that can intervene if necessary [2]. A typical UUV has a camera mounted on the body so the operator/human supervisor can observe what is happening. However, a fixed camera can never give the same viewing freedom and flexibility as that of a movable camera.

In case the mission requires interaction between the UUV and the environment, one or more manipulator arms can be attached to the vehicle body. This entire system is referred to as an Underwater Vehicle-Manipulator System (UVMS) [3]. Such vehicles have a much wider spectrum of possible missions since they are not limited to survey missions only, but can perform tasks such as sampling, retrieval of instruments, assembling and/or maintenance of underwater structures etc.

At a kinematic level, a UVMS can be considered as a manipulator arm mounted on a floating base. Thus, the theory also applies to other such systems, for instance a quadcopter or an UAV (Unmanned Aerial Vehicle) with one or more manipulator arms. A general floating base system without a manipulator arm has 6 Degrees of Freedom (DOFs): 3 for position and 3 for orientation. Adding a n-link manipulator arm results in a $6+n$ DOF system, which is said to be kinematically redundant if it possesses more DOFs than those required to perform a certain task [4]. A general manipulation task is specified in terms of end effector position and orientation, and as such an UVMS is always kinematically redundant because the necessary DOFs are provided by the vehicle itself. In this case, it will be the guidance system's task to calculate the desired vehicle position/velocity and manipulator angles/angular velocities based on the current system state and the desired position/orientation of the manipulator end effector. To do so, one must solve the inverse kinematics problem. The most common approach to this is to use a Jacobian-based method [5]-[8].

The "excess" DOFs can be utilized as a way to perform several tasks using Null-Space-Based (NSB) behavioral control [9]. Several different tasks and their corresponding forward kinematics and Jacobian matrices are defined in [3]. Examples of such secondary tasks are manipulability maximization, obstacle avoidance, joint limit avoidance, actuator power consumption, etc [10]-[13]. A task-priority framework has been successfully implemented within the TRIDENT EU FP7 project [14].

A two-manipulator system can use the two arms to cooperate and thereby perform more complex tasks and pick up larger/heavier objects. A lot of research has been done on fixed dual-arm systems regarding coordinated and cooperative control, leader/follower control, force control, collision detection and avoidance etc. A Jacobian based method is used to calculate the desired manipulator motion in [15]. This paper considers relative motion between the two end effectors rather than the motion relative to a world-fixed coordinate system as this is more intuitive. In [16] a leader-follower set-up between the two manipulators is proposed, where the reference of the leader manipulator is feed-forwarded to calculate the appropriate reference of the follower. A centralized impedance control strategy using force and moment measurements is considered in [17]. Here, the dynamics rather than the kinematics is regarded. However, the dynamics of a dual-arm UVMS is highly complex, non-linear and nearly impossible to model correctly without making simplifications and approximations. The kinematics, however, is straight forward and exactly defined, and thus a

[1]S.Moe and K.Y.Pettersen are with the Center for Autonomous Marine Operations and Systems (AMOS), at The Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway {signe.moe, kristin.y.pettersen}@itk.ntnu.no

[2]G.Antonelli is with the Department of Electrical and Information Engineering, University of Cassino and Southern Lazio, Cassino, Italy antonelli@unicas.it

kinematic approach may be better and more accurate.

Although there exists some literature on the guidance and control of a floating base with one manipulator, very little has been done for a two-manipulator system. A floating base dual-arm system is more complicated as the position and orientation of the base is subject to change. In [18], one arm is commanded to perform desired tasks while the other provides compensating motions to keep the base inertially fixed. However, this limits the system in several ways: The base is required to be stationary during intervention tasks, and the second manipulator can not be used for intervention tasks. A dual-arm free-floating base system in also considered in [19]. Here a control scheme for trajectory tracking of the end effectors is proposed. However, this approach does not control the position/orientation of the base itself, something that may be desirable for certain missions.

This paper proposes a NSB Jacobian-based guidance scheme for a dual-arm UVMS which calculates references for both the manipulator arms and the vehicle base itself, in order to implement several concurrent tasks. A variety of tasks can be implemented and included in a prioritized order. Furthermore, it is proposed to consider one manipulator and the vehicle base as a leader and the second manipulator as the follower. This division ensures that conflicting tasks do not attempt to move the vehicle base in different directions. As a further consequence of this, we choose to define the motion of the follower manipulator relative to the base instead of relative to the inertial system. One particular case that is considered in this paper is the case where one of the manipulators carries a camera, and its main objective is to always point the camera towards the end effector of the other manipulator which is to perform an intervention task, thereby offering the operator/supervisor a good view of the intervention operation. The base and the intervention manipulator then constitute the leader, and the camera holding manipulator is the follower.

This paper is organized as follows: Section II presents the vehicle kinematics and the proposed division of the leader-follower states. Section III describes the implemented tasks and the corresponding task Jacobians and Section IV contains the proposed guidance system based on the vehicle tasks. Finally, simulation results are given in Section V and conclusions and further work in Section VI.

II. VEHICLE KINEMATICS

A complete model of an UVMS consists of the kinematics and the kinetics. The kinematics is relatively straight forward and described in [3]-[20]. The forward kinematics of the manipulator arm(s) can be derived, for instance, by the Denavit-Hartenberg convention [21], and the Jacobian matrix describes the relationship between the end effector velocities and the angular velocities of the arm. However, the kinetics of such a vehicle is very complex, highly non-linear and contains several cross-terms because of the interaction between the vehicle body and the manipulator arm [3], [22]. In addition, the numeric values of the hydrodynamical parameters are difficult to identify precisely [23]-[25]. As such, model-based control of an UVMS is challenging.

In this paper a two-manipulator underwater vehicle in the plane is considered. The proposed approach can be followed for manipulators with any number of links. However, for simplicity of presentation we write the equations for the 3 and 2 links case. The vector $\boldsymbol{\eta_b} = [x_b, y_b, \psi_b]^T$ is the vehicle position and orientation relative to the inertial frame. Similarly, $\boldsymbol{\eta_1} = [x_{ee1}, y_{ee1}, \psi_{ee1}]^T$ and $\boldsymbol{\eta_2} = [x_{ee2}, y_{ee2}, \psi_{ee2}]^T$ describe the position and orientation of the manipulator end effectors in inertial frame, see Figure 1.

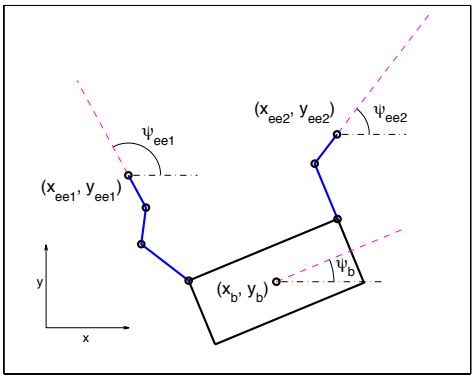


Fig. 1: Illustration of vehicle base with two manipulator arms with position and orientation of the vehicle body and the manipulator end effectors.

The vehicle's velocity is defined in body-frame as $\boldsymbol{\nu} = [u, v, r]^T$, where u and v are the linear velocities and r is the angular rate. Equation (1) describes the relationship between the velocities in the inertial and body frame:

$$\dot{\boldsymbol{\eta}}_{\boldsymbol{b}} = \boldsymbol{R}_b^i(\boldsymbol{\eta_b})\boldsymbol{\nu} = \begin{bmatrix} \cos(\psi_b) & -\sin(\psi_b) & 0 \\ \sin(\psi_b) & \cos(\psi_b) & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{\nu} \qquad (1)$$

Furthermore, the manipulator arms have joint angles $\boldsymbol{q_1} = [q_{11}, q_{12}, q_{13}]^T$ and $\boldsymbol{q_2} = [q_{21}, q_{22}]^T$ and joint angular velocities $\dot{\boldsymbol{q}}_{\boldsymbol{1}}$ and $\dot{\boldsymbol{q}}_{\boldsymbol{2}}$.

The forward kinematics of the manipulator arms are straight forward. Defining the vehicle length and height as L and H respectively and the length of the jth link of manipulator i as l_{ij}, the position and orientation of the manipulator arms in the inertial frame are given as follows:

$$\boldsymbol{\eta_1}(\boldsymbol{\eta_b}, \boldsymbol{q_1}) = \boldsymbol{\eta_b} + \boldsymbol{R}_b^i(\boldsymbol{\eta_b}) \cdot \underbrace{\begin{bmatrix} -\frac{L}{2} + l_{11}\cos(q_{11}) + l_{12}\cos(q_{11}+q_{12}) + l_{13}\cos(q_{11}+q_{12}+q_{13}) \\ \frac{H}{2} + l_{11}\sin(q_{11}) + l_{12}\sin(q_{11}+q_{12}) + l_{13}\sin(q_{11}+q_{12}+q_{13}) \\ q_{11}+q_{12}+q_{13} \end{bmatrix}}_{\triangleq \boldsymbol{k}_1(\boldsymbol{q_1})} \qquad (2)$$

$$\boldsymbol{\eta_2}(\boldsymbol{\eta_b}, \boldsymbol{q_2}) = \boldsymbol{\eta_b} + \boldsymbol{R}_b^i(\boldsymbol{\eta_b}) \underbrace{\begin{bmatrix} \frac{L}{2} + l_{21}\cos(q_{21}) + l_{22}\cos(q_{21}+q_{12}) \\ \frac{H}{2} + l_{21}\sin(q_{21}) + l_{22}\sin(q_{21}+q_{22}) \\ q_{21}+q_{22} \end{bmatrix}}_{\triangleq \boldsymbol{k}_2(\boldsymbol{q_2})} \qquad (3)$$

Similarly, the forward kinematics of the manipulator arms relative to the body frame is given as

$$\boldsymbol{\eta}_1^b = \boldsymbol{k}_1(\boldsymbol{q_1}) \tag{4}$$

$$\boldsymbol{\eta}_2^b = \boldsymbol{k}_2(\boldsymbol{q_2}). \tag{5}$$

When considering the control of a one-manipulator UVMS, it is common to consider the vector $\boldsymbol{\zeta} = [\boldsymbol{\nu}^T, \dot{\boldsymbol{q}}^T]^T$ where $\boldsymbol{q}$ is a n-dimensional vector containing the angles of the manipulator arm. The guidance system then aims to find desired values for $\boldsymbol{\zeta}$ that, if fulfilled, will result in the vehicle tasks (see Section III) being fulfilled. As such, one option for the two-manipulator UVMS considered in this paper is to choose $\boldsymbol{\zeta} = [\boldsymbol{\nu}^T, \dot{\boldsymbol{q_1}}^T, \dot{\boldsymbol{q_2}}^T]^T$. With this choice one would find references for the entire system as a whole since the two manipulators share one floating base. However, if the manipulators have conflicting tasks, they will try to pull the vehicle base in different directions. As such, this paper proposes to choose one of the manipulators as the main manipulator and any others as additional manipulators. These are only considered relative to the vehicle base and their tasks as such do not affect the desired velocity/position of the vehicle itself. Take for instance the case where the main task of manipulator two is to point the camera towards the main manipulator arm to provide the human operator/supervisor with a good visual input. To achieve this goal, only the joint angles of manipulator 2 should be changed: In particular, we do not want the position and orientation of the vehicle base to be affected in order to achieve this goal. Consequently, we propose to consider the two following independent vectors:

$$\boldsymbol{\zeta_1} = \begin{bmatrix} \boldsymbol{\nu} \\ \dot{\boldsymbol{q_1}} \end{bmatrix} \tag{6}$$

$$\boldsymbol{\zeta_2} = \dot{\boldsymbol{q_2}} \tag{7}$$

III. VEHICLE TASKS

This paper considers the NSB behavior control to perform several tasks at once. If the UVMS has more DOFs than those required to execute a given task the system is redundant with respect to that specific task and kinematic redundancy can be exploited to achieve additional tasks. The tasks are sorted by priority: The secondary task is given lower priority with respect to the primary task by projecting the relative actions through the null space of the primary task Jacobian. The tertiary task is given lower priority with respect to the secondary task by projecting the relative actions through the null space of the primary and secondary task Jacobian and so on (see Section IV).

To illustrate the general method proposed in this paper, a certain scenario has been implemented and simulated. The different tasks are defined in this section.

Task j for manipulator i is denoted σ_{ij} and the corresponding task Jacobian J_{ij}. In general for a two-manipulator UVMS modeled as proposed in this paper, for a m DOF task, σ_{1j} is a vector of length m and the corresponding Jacobian is a $m \times (n_b + n_1)$ matrix, and σ_{2j} is a vector of length m and the corresponding Jacobian is a $m \times n_2$ matrix, where n_b, n_1 and n_2 are the DOFs of the vehicle itself, Manipulator 1 and Manipulator 2, respectively. In this paper we will develop the equations for the particular case where $n_b = n_1 = 3$ and $n_2 = 2$. This implies that the main manipulator and vehicle can be given tasks that require up to 6 DOFs and the secondary manipulator can be given tasks that require 2 DOFs. For the particular case where manipulator 2 carries a camera and Manipulator 1 is for intervention, the following tasks are proposed:

Manipulator 1 - Task 1 - End effector trajectory and orientation - 3 DOFs: The end effector should track a given trajectory with a given orientation:

$$\boldsymbol{\sigma}_{11}(\boldsymbol{\eta_b}, \boldsymbol{q_1}) = \boldsymbol{\eta}_1(\boldsymbol{\eta_b}, \boldsymbol{q_1}) = \begin{bmatrix} x_{ee1} \\ y_{ee1} \\ \psi_{ee1} \end{bmatrix} \tag{8}$$

$$\boldsymbol{\sigma}_{11,d} = \boldsymbol{\eta}_{1,d} = \begin{bmatrix} x_{ee1,d} \\ y_{ee1,d} \\ \psi_{ee1,d} \end{bmatrix} \tag{9}$$

The expression for $\boldsymbol{\sigma}_{11}$ is given by (2), and the Jacobian

$$\dot{\boldsymbol{\sigma}}_{11} = \boldsymbol{J}_{11}(\boldsymbol{\eta_b}, \boldsymbol{q_1})\boldsymbol{\zeta_1} \tag{10}$$

is derived by taking the time-derivative of (2) and inserting (1).

Manipulator 1 - Task 2 - Orientation of vehicle - 1 DOF: The vehicle should have a constant orientation of $\psi_b = 0$.

$$\boldsymbol{\sigma}_{12}(\boldsymbol{\eta_b}) = \psi_b \tag{11}$$

$$\boldsymbol{\sigma}_{12,d} = 0 \tag{12}$$

$$\dot{\boldsymbol{\sigma}}_{12} = \dot{\psi}_b = r = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{\zeta_1} = \boldsymbol{J}_{12}\boldsymbol{\zeta_1} \tag{13}$$

Manipulator 1 - Task 3 - Vertical distance between the vehicle and the end effector - 1 DOF: The vertical distance between the vehicle center and the end effector should be constant and positive to ensure that the manipulator is operating over the vehicle and that the vehicle is not blocking the view of end effector 2.

$$\boldsymbol{\sigma}_{13}(\boldsymbol{\eta_b}, \boldsymbol{q_1}) = y_{ee1} - y_b \tag{14}$$

$$\boldsymbol{\sigma}_{13,d} = C \tag{15}$$

Similarly to Task 1, the Jacobian is derived by taking the time derivative of σ_{13} using (2) and (1).

$$\dot{\boldsymbol{\sigma}}_{13} = \boldsymbol{J}_{13}(\boldsymbol{\eta_b}, \boldsymbol{q_1})\boldsymbol{\zeta_1} \tag{16}$$

Manipulator 2 - Task 1 - Relative Field of View - 1 DOF: End effector 2 should always point towards end effector 1. In this case, the task is chosen as the error between the desired and actual direction of the manipulator, so the desired task value is 0:

$$\boldsymbol{\sigma}_{21}(\boldsymbol{\eta_b}, \boldsymbol{q_1}, \boldsymbol{q_2}) = \sqrt{(\boldsymbol{a}_{des} - \boldsymbol{a})^T(\boldsymbol{a}_{des} - \boldsymbol{a})} \tag{17}$$

$$\boldsymbol{\sigma}_{21,d} = 0 \tag{18}$$

where $\boldsymbol{a}_{des}$ and $\boldsymbol{a}$ are unit vectors illustrated in Figure 2 and are defined as

$$\boldsymbol{a}_{des} = \frac{1}{\sqrt{(x_{ee1}-x_{ee2})^2+(y_{ee1}-y_{ee2})^2}} \underbrace{\begin{bmatrix} x_{ee1}-x_{ee2} \\ y_{ee1}-y_{ee2} \\ 0 \end{bmatrix}}_{\triangleq \boldsymbol{p}_e} \tag{19}$$

$$\boldsymbol{a} = \begin{bmatrix} \cos(\psi_{ee2}) & -\sin(\psi_{ee2}) & 0 \\ \sin(\psi_{ee2}) & \cos(\psi_{ee2}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \tag{20}$$

The corresponding 1×2 Jacobian is defined in [3] as

$$\dot{\boldsymbol{\sigma}}_{21} = \boldsymbol{J}_{21}(\boldsymbol{\eta_b}, \boldsymbol{q_1}, \boldsymbol{q_2})\boldsymbol{\zeta_2} \tag{21}$$

$$\boldsymbol{J}_{21} = \frac{(\boldsymbol{a}_{des}-\boldsymbol{a})^T}{\|\boldsymbol{a}_{des}-\boldsymbol{a}\|}\left(-\boldsymbol{S}(\boldsymbol{a}_{des})\boldsymbol{S}(\boldsymbol{p}_e)^\dagger \boldsymbol{J}_p + \boldsymbol{S}(\boldsymbol{a})\boldsymbol{J}_o\right) \tag{22}$$

where $\boldsymbol{S}(\cdot)$ is the matrix cross product operator and $\boldsymbol{X}^\dagger$ denotes the Moore-Penrose inverse of the matrix $\boldsymbol{X}$. $\boldsymbol{J}_p$ and $\boldsymbol{J}_o$ denote the position and orientation Jacobian matrices of end effector 2. With the proposed choice of $\boldsymbol{\zeta_1}$ and $\boldsymbol{\zeta_2}$, the position and orientation is that of end effector 2 relative to the body frame. By taking the time derivative of (5) $\boldsymbol{J}_p$ and $\boldsymbol{J}_o$ can be derived:

$$\begin{aligned} \dot{\boldsymbol{\eta}}_{\boldsymbol{2}}^{b}(\boldsymbol{q_2}) &= \frac{\delta}{\delta t}\boldsymbol{k}_2(\boldsymbol{q_2}) \\ &= \begin{bmatrix} -l_{21}\sin(q_{21})\dot{q}_{21} - l_{22}\sin(q_{21}+q_{12})(\dot{q}_{21}+\dot{q}_{22}) \\ l_{21}\cos(q_{21})\dot{q}_{21} + l_{22}\cos(q_{21}+q_{22})(\dot{q}_{21}+\dot{q}_{22}) \\ \dot{q}_{21}+\dot{q}_{22} \end{bmatrix} \\ &= \begin{bmatrix} -l_{21}\sin(q_{21}) - l_{22}\sin(q_{21}+q_{12}) & -l_{22}\sin(q_{21}+q_{12}) \\ l_{21}\cos(q_{21}) + l_{22}\cos(q_{21}+q_{22}) & l_{22}\cos(q_{21}+q_{22}) \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{q}_{21} \\ \dot{q}_{22} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{J}_p \\ \boldsymbol{J}_o \end{bmatrix} \boldsymbol{\zeta_2} \end{aligned} \tag{23}$$

Remark 1: Note that the vectors $\boldsymbol{a}$ and $\boldsymbol{a}_{des}$ have been expanded to three-dimensional vectors even though movement in only considered in the plane. This is due to the fact that the matrix cross product operator is a 3×3 matrix, and the dimensions must fit to carry out the matrix multiplications. Similarly, (23) shows that $\boldsymbol{J}_p$ and $\boldsymbol{J}_o$ are 2×2 and 1×2 matrices, respectively. However, in (22) they are expanded to be 3×2 matrices by adding one zero row at the bottom of $\boldsymbol{J}_p$ and two zero rows at the top of $\boldsymbol{J}_o$.

Remark 2: The Jacobian includes a singularity that occurs when $\boldsymbol{a}_{des} = \boldsymbol{a}$. This is resolved in the implementation by dividing by a small number ε rather than the error norm if the norm is smaller than ε [3].

Manipulator 2 - Task 2 - First joint - 1 DOF: The first joint of manipulator 2 should be kept constant: $q_{21} = \pi/2$. This ensures that manipulator 2 operates over the vehicle, similarly to Task 3 for Manipulator 1.

$$\boldsymbol{\sigma}_{22}(\boldsymbol{q_2}) = q_{21} \tag{24}$$

$$\boldsymbol{\sigma}_{22,d} = \frac{\pi}{2} \tag{25}$$

$$\dot{\boldsymbol{\sigma}}_{22} = \dot{q}_{21} = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{\zeta_2} = \boldsymbol{J}_{22}\boldsymbol{\zeta_2} \tag{26}$$

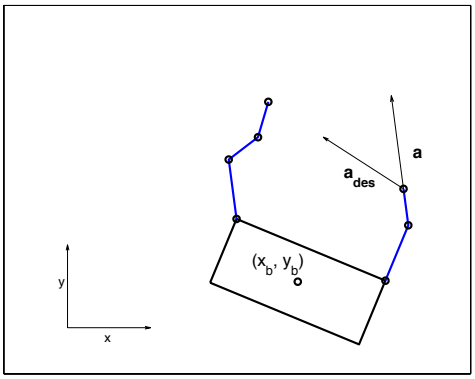


Fig. 2: Illustration of $\boldsymbol{a}_{des}$ and $\boldsymbol{a}$ for relative field of view.

IV. GUIDANCE SYSTEM

This section presents the guidance system that calculates the desired system trajectories. With the proposed division of $\boldsymbol{\zeta_1}$ and $\boldsymbol{\zeta_2}$ a general task and the corresponding Jacobian can be expressed as

$$\boldsymbol{\sigma}_1 = \boldsymbol{f}(\boldsymbol{\eta_b}, \boldsymbol{q_1}) \tag{27}$$

$$\dot{\boldsymbol{\sigma}}_1 = \boldsymbol{J}_1(\boldsymbol{\eta_b}, \boldsymbol{q_1})\boldsymbol{\zeta_1} \tag{28}$$

$$\boldsymbol{\sigma}_2 = \boldsymbol{g}(\boldsymbol{\eta_b}, \boldsymbol{q_1}, \boldsymbol{q_2}) \tag{29}$$

$$\dot{\boldsymbol{\sigma}}_2 = \boldsymbol{J}_2(\boldsymbol{\eta_b}, \boldsymbol{q_1}, \boldsymbol{q_2})\boldsymbol{\zeta_2}. \tag{30}$$

Note that tasks related to the vehicle/Manipulator 1 are completely independent on manipulator 2 ($\boldsymbol{q_2}$) due to the fact that this is considered the main manipulator/leader. Tasks related to manipulator 2, however, may generally depend on the configuration of the vehicle, Manipulator 1 and 2.

For a single task, the desired $\boldsymbol{\zeta}$ can be calculated as

$$\boldsymbol{\zeta}_{1,d} = \boldsymbol{J}_1^\dagger(\dot{\boldsymbol{\sigma}}_{1,d} + \boldsymbol{\Lambda}_1\boldsymbol{e}_1), \tag{31}$$

$$\boldsymbol{\zeta}_{2,d} = \boldsymbol{J}_2^\dagger(\dot{\boldsymbol{\sigma}}_{2,d} + \boldsymbol{\Lambda}_2\boldsymbol{e}_2), \tag{32}$$

where $\boldsymbol{\Lambda}$ is a positive definite gain matrix and $\boldsymbol{e} \triangleq \boldsymbol{\sigma}_d - \boldsymbol{\sigma}$. This is illustrated in Figure 3. Note that the calculated, desired state of the UVMS is used as feedback in the guidance system rather than the actual state of the UVMS. In other words, the guidance system is independent of the actual behavior of the UVMS and only generates trajectories that, if fulfilled, will result in the best possible achievement of the given tasks. These trajectories can be calculated off-line. Future work includes closing the loop completely, expanding the guidance system to take the current UVMS state into account and thereby providing a more robust result.

However, several tasks have been defined with a certain priority. As such, it is desirable to find a $\boldsymbol{\zeta}_{1,d}$ and $\boldsymbol{\zeta}_{2,d}$ that, if fulfilled, will result in achievement of all tasks. In case of conflicting tasks this might not be possible, in which case the goal is to find the best possible solution with respect to the defined priorities. This is done with the NSB method. Before adding the contribution of a lower-priority task to the overall desired vehicle velocity, it is projected onto the null space of

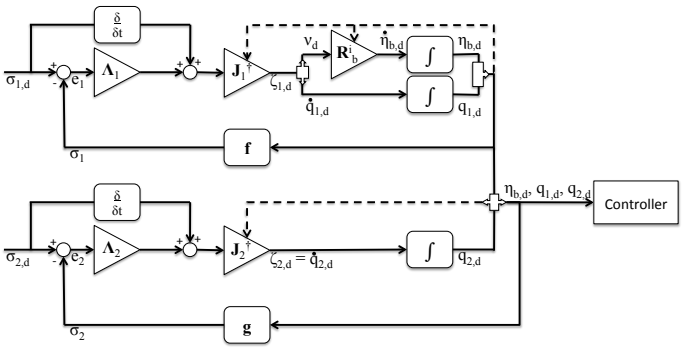


Fig. 3: Block diagram of the proposed guidance system when one task for the leader and one task for the follower is implemented ((31) and (32)). Triangular blocks are gains and rectangular blocks are functions. Based on the desired tasks a reference for $\boldsymbol{\eta_b}$, $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$ is calculated and sent to the controller system.

the immediately higher-priority task so as to remove velocity components that would conflict with it. The total proposed guidance law for the considered system is thus given by (33) and (34).

$$\boldsymbol{\zeta}_{1,d} = \boldsymbol{J}_{11}^{\dagger}(\dot{\boldsymbol{\sigma}}_{11,d} + \boldsymbol{\Lambda}_{11}\boldsymbol{e}_{11}) + (\boldsymbol{I} - \boldsymbol{J}_{11}^{\dagger}\boldsymbol{J}_{11})\boldsymbol{J}_{12}^{\dagger}(\dot{\boldsymbol{\sigma}}_{12,d} + \boldsymbol{\Lambda}_{12}\boldsymbol{e}_{12}) + (\boldsymbol{I} - \begin{bmatrix}\boldsymbol{J}_{11}\\ \boldsymbol{J}_{12}\end{bmatrix}^{\dagger}\begin{bmatrix}\boldsymbol{J}_{11}\\ \boldsymbol{J}_{12}\end{bmatrix})\boldsymbol{J}_{13}^{\dagger}(\dot{\boldsymbol{\sigma}}_{13,d} + \boldsymbol{\Lambda}_{13}\boldsymbol{e}_{13}) \quad (33)$$

$$\boldsymbol{\zeta}_{2,d} = \boldsymbol{J}_{21}^{\dagger}(\dot{\boldsymbol{\sigma}}_{21,d} + \boldsymbol{\Lambda}_{21}\boldsymbol{e}_{11}) + (\boldsymbol{I} - \boldsymbol{J}_{21}^{\dagger}\boldsymbol{J}_{21})\boldsymbol{J}_{22}^{\dagger}(\dot{\boldsymbol{\sigma}}_{22,d} + \boldsymbol{\Lambda}_{22}\boldsymbol{e}_{22}) \quad (34)$$

V. SIMULATION RESULTS

The UVMS described in Section II, the tasks listed in Section III and the guidance law (33) and (34) have been implemented using Matlab. This section presents simulation results of the UVMS behavior if the calculated desired states are tracked perfectly by the controller.

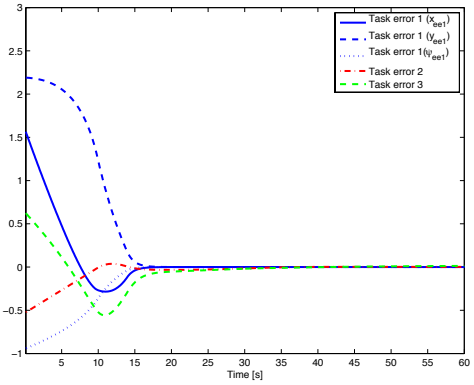
In the presented simulation, Manipulator 1 (task 1) has been tasked with tracking a straight line trajectory with a constant velocity between a defined start and end point. Furthermore, the desired orientation is constant and normal to the line. The desired vertical distance between the vehicle and end effector 1 (task 3) has been chosen as $\frac{H}{2} + \frac{2}{3}(l_{11} + l_{12} + l_{13})$. The guidance law error gains have been chosen as follows:

$$\boldsymbol{\Lambda}_{11} = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}, \boldsymbol{\Lambda}_{12} = 3, \boldsymbol{\Lambda}_{13} = 2, \boldsymbol{\Lambda}_{21} = 1, \boldsymbol{\Lambda}_{22} = 3 \quad (35)$$

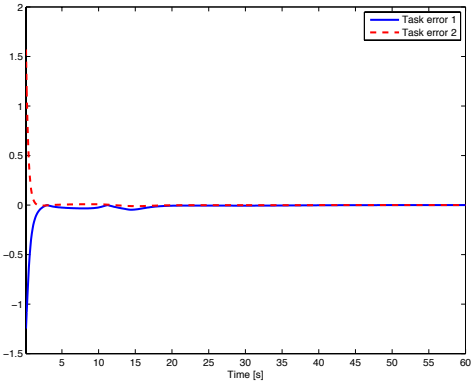
Finally, the UVMS has been implemented with a saturation on the linear and angular velocities. A method for ensuring that the saturation is not reached is described in [26]. This approach has not been implemented in this paper.

$$|u| \leq 3\frac{m}{s}, |v| \leq 2\frac{m}{s}, |r| \leq 3\frac{deg}{s}, |\dot{q}_{ij}| \leq 10\frac{deg}{s} \quad (36)$$

The simulation results are shown in Figures 4a-4d and confirm that the calculated reference values will in fact, if fulfilled, result in achievement of the implemented tasks. Furthermore, Figures 5a and 5b show that the task errors all converge to zero. Note that higher priority tasks converge before the tasks with lower priority.



(a) Task errors for Manipulator 1 and vehicle.

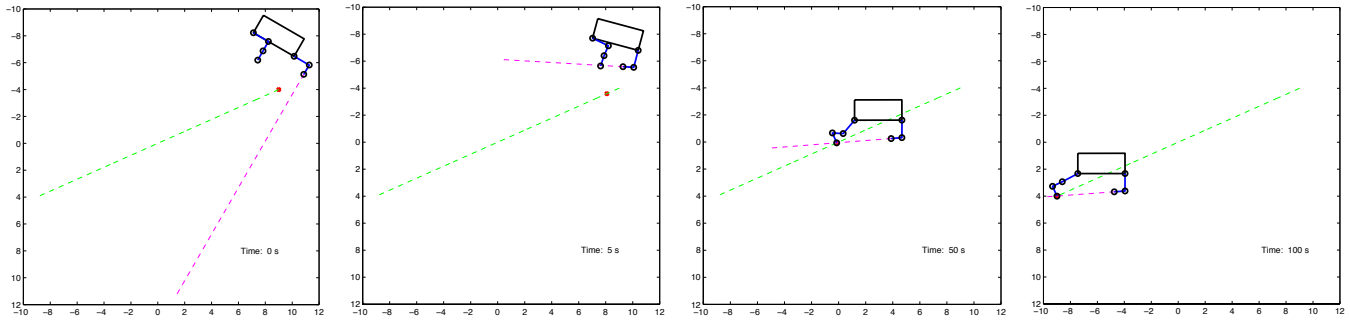


(b) Task errors for Manipulator 2.

VI. CONCLUSIONS

This paper presents a method for generating reference trajectories for a two-manipulator Underwater Vehicle Manipulator System by considering the main manipulator and the vehicle base as a leader unit and the secondary manipulator as a follower unit. This ensures that the two manipulators do not attempt to drive the vehicle base in opposing directions/velocities due to conflicting tasks. The references are calculated using the task Jacobian pseudoinverses and a Null-Spaced-Based control approach. The proposed approach has been implemented for a particular UVMS working in the plane and simulated for a defined set of tasks. In particular, the case where one manipulator is dedicated to intervention operations and the other is holding a camera to provide the operator/supervisor a good view of the intervention, has been considered. Relevant tasks for this scenario have been defined, and simulations illustrate the performance of the proposed guidance approach.

Further work includes extending the simulation to 3 dimensions, implementing coordination tasks where the two manipulator arms cooperate on a task, considering collection detection/avoidance in case the two arms have



(a) Initial state, time = 0 s. (b) Time = 5 s. (c) Time = 50 s. (d) Time = 100 s.

Fig. 4: Simulation results: The green dashed line shows the desired trajectory for end effector 1. The red dot is moving along the line and is the reference position. The purple dashed line shows the pointing direction of Manipulator 2. The results show that all the tasks are fulfilled during the simulation.

a joint workspace and testing in the field to validate the method with experimental results .

ACKNOWLEDGMENTS

This work was supported by the Research Council of Norway through the Center of Excellence funding scheme, project number 223254.

REFERENCES

[1] R. Christ and R. Wernli, *The ROV Manual: A User Guide for Remotely Operated Vehicles*. Elsevier Science, 2013.

[2] L. Santos, S. J., P. Sanz, and J. Garca, "Cognitive Skills Models: Towards Increasing Autonomy in Underwater Intervention Missions," in *Proc. 2013 International Conference on Intelligent Robots and Systems*, Tokyo, Japan, 2013.

[3] G. Antonelli, *Underwater Robots, Third Edition*. Springer International Publishing, 2014.

[4] G. Antonelli and S. Chiaverini, "Task-Priority Redundancy Resolution for Underwater Vehicle-Manipulator Systems," in *Proc. 1998 International Conference on Robotics and Automation*, Leuven, Belgium, 1998.

[5] C. Klein and C.-H. Huang, "Review of pseudoinverse control for use with kinematically redundant manipulators," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 13, no. 2, pp. 245–250, 1983.

[6] D. Nenchev, Y. Umetani, and K. Yoshida, "Analysis of a redundant free-flying spacecraft/manipulator system," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 1, pp. 1–6, 1992.

[7] F. Caccavale and B. Siciliano, "Kinematic control of redundant free-floating robotic systems," *Advanced Robotics*, vol. 15, no. 4, pp. 429–448, 2001.

[8] O. Egeland and K. Pettersen, "Free-floating robotic systems," in *Control Problems in Robotics and Automation*. Springer Berlin Heidelberg, 1998, vol. 230, pp. 119–134.

[9] G. Antonelli, F. Arrichiello, and S. Chiaverini, "The null-space-based behavioral control for autonomous robotic systems," *Intelligent Service Robotics*, vol. 1, no. 1, pp. 27–39, 2008.

[10] T. Yoshikawa, "Manipulability of Robotic Mechanisms," *The International Journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.

[11] O. Khatib, "A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation," *IEEE Journal of Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.

[12] E. Marchand, F. Chaumette, and A. Rizzo, "Using the task function approach to avoid robot joint limits and kinematic singularities in visual servoing," in *Proc. 1996 International Conference on Intelligent Robots and Systems*, Osaka, Japan, 1996.

[13] M. Vukobratovic and M. Kircanski, "A dynamic approach to nominal trajectory synthesis for redundant manipulators," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 14, no. 4, pp. 580–586, 1984.

[14] E. Simetti, G. Casalino, S. Torelli, A. Sperindé, and A. Turetta, "Floating Underwater Manipulation: Developed Control Methodology and Experimental Validation within the TRIDENT Project," *Journal of Field Robotics*, vol. 31, no. 3.

[15] C. Park and K. Park, "Design and kinematics analysis of dual arm robot manipulator for precision assembly," in *Proc. 6th IEEE International Conference on Industrial Informatics*, Daejeon, South Korea, 2008.

[16] R. Vidal, O. Shakernia, and S. Sastry, "Formation control of nonholonomic mobile robots with omnidirectional visual servoing and motion segmentation," in *Proc. IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, 2003.

[17] F. Caccavale, P. Chiacchio, A. Marino, and L. Villani, "Six-DOF Impedance Control of Dual-Arm Cooperative Manipulators," *Mechatronics, IEEE/ASME Transactions on*, vol. 13, no. 5, pp. 576–586, 2008.

[18] S. Agrawal and S. Shirumalla, "Planning motions of a dual-arm free-floating manipulator keeping the base inertially fixed," *Mechanism and Machine Theory*, vol. 30, no. 1, pp. 59 – 70, 1995.

[19] D. Huang and C. L., "Inverse Kinematic Control of Free-floating Space Robot System Based on a Mutual Mapping Neural Network," in *Proc. 7th World Congress on Intelligent Control and Automation*, Chongqing, China, 2008.

[20] J. Kim and W. Chung, "Dynamic Analysis and Active Damping Control for Underwater Vehicle-Manipulator Systems," in *Proc. 2003 International Offshore and Polar Engineering Conference*, Honolulu, Hawaii, 2003.

[21] M. Spong and S. Hutchinson, *Robot Modeling and Control*. Wiley, 2005.

[22] I. Schjølberg and T. Fossen, "Modelling and Control of Underwater Vehicle-Manipulator Systems," in *Proc. 3rd Conference on Marine Craft maneuvering and control*, Southampton, United Kingdon, 1994.

[23] M. Nomoto and M. Hattori, "A deep ROV "Dolphin 3K": Design and performance analysis," *IEEE Journal of Oceanic Engineering*, vol. 11, no. 3, pp. 373–391, 1986.

[24] D. Marco, A. Martins, and A. Healy, "Surge Motion Parameter Identification for the NPS Phoenix AUV," in *Proc. International Advanced Robotics Program*, Lafayette, Louisiana, USA, 1998.

[25] D. Smallwood and L. Whitcomb, "Adaptive identification of dynamically positioned underwater robotic vehicles," *IEEE Transactions on Control Systems Technology*, vol. 11, no. 4, pp. 505–515, 2003.

[26] G. Antonelli, G. Indiveri, and S. Chiaverini, "Prioritized Closed-Loop Inverse Kinematic Algorithms for Redundant Robotic Systems with Velocity Saturations," in *Proc. IEEE/RSJ International Conference on Intelligent RObots and Systems*, St. Louis, Montana, USA, 2009.