# Inverse Kinematic Control of a Free-Floating Underwater Manipulator Using the Generalized Jacobian Matrix

Morten F. Amundsen, Jørgen Sverdrup-Thygeson, Eleni Kelasidi, Kristin Y. Pettersen
Norwegian University of Science and Technology (NTNU)
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics
Trondheim, Norway
Email: {morten.f.amundsen, jorgen.sverdrup-thygeson, eleni.kelasidi, kristin.y.pettersen}@ntnu.no

***Abstract*—Traditional control of robot manipulators assume that the base of the manipulator is fixed in the environment. However, underwater vehicle-manipulator systems are usually floating freely. Using traditional fixed-base manipulator control for floating systems requires thruster actuation and a station-keeping algorithm capable of stabilizing the manipulator base. This paper describes a method for underwater manipulation that needs no stationkeeping, and therefore no thrusters. For vehicles with thrusters, the method presented can reduce the energy that the thrusters use on vehicle stabilization. The method is based on inverse kinematics using the generalized Jacobian matrix, a generalization of the traditional fixed-base manipulator Jacobian, formerly used for spacecraft manipulators. The generalized Jacobian matrix makes it possible to control the manipulator end-effector in inertial coordinates while the base remains passive and unactuated. The proposed control method is verified through simulations of an underwater swimming manipulator, using the Vortex Studio software. We compare the performance of the method in open-loop and with position feedback in closed-loop. The simulations show that underwater manipulation without base actuation is possible and that accuracy improves with access to position estimates.**

## I. Introduction

Underwater robots exist in numerous forms and have many applications. Some examples are seafloor mapping and geological sampling in research and science; construction, inspection, maintenance and repair of subsea installations in the oil and gas industry; and search and disposal of mines in the military [1].

Underwater intervention requires a vehicle equipped with a robotic manipulator arm. In the most general sense, such vehicles are termed underwater vehicle-manipulator systems (UVMSs) [2]. Traditional UVMSs and intervention-autonomous underwater vehicle (AUV) [3] resemble remotely operated vehicles (ROVs) with robotic arms attached. More recent designs include snake-like robots where the flexible body itself acts as the manipulator. Examples are underwater snake robots (USRs) [4] and underwater swimming manipulators (USMs) [5]. The USM is a new class of underwater robots that combines a bio-inspired snake-like appearance with thruster actuators [5].

Fig. 1. The Eelume USM, an example of an underwater snake robot. It is approximately 3.3 m long and weighs 80 kg.

The control of freely floating UVMSs is complicated by the many forces acting on them. The most significant forces are:

1) **Reaction forces:** Moving the joints of the manipulator will induce reaction forces that disturb the position and orientation of the manipulator base [6]. The disturbance occurs because the base is floating freely in the water. By contrast, the base of an industrial manipulator is firmly fixed to its environment and is not disturbed by the joint motion. Figure 2 illustrates the effects of joint motion and reaction forces on fixed-base and floating-base manipulators.
2) **Hydrostatic forces:** Gravity "pulls" the center of mass (COM) down, and buoyancy "pushes" the center of buoyancy (COB) up. The COM and COB of an underwater manipulator are not necessarily aligned, and this causes rotational hydrostatic restoring forces [7, Ch. 4]. Because the COM and COB locations depend on the joint configuration, the resulting restoring forces also depend on the joint configuration.

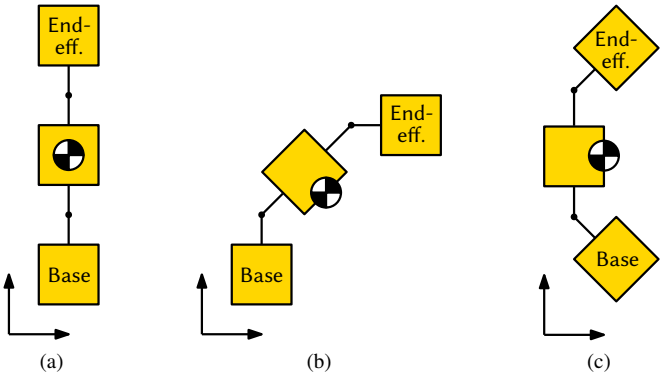



Fig. 2. Differences between fixed-base and floating-base manipulation: (a) shows the initial configuration, and (b) and (c) shows approximate final configurations after identical joint motion. In (b), the base is fixed, and this causes the total COM to move. In (c), the base is floating, and reaction forces on the base cause it to move, while the location of the COM remains the same.

3) **Hydrodynamic forces:** The manipulator moves through water, and is subject to hydrodynamic effects such as drag forces and increased apparent inertia due to added mass [8, Ch. 10].

The effects of the forces listed above are present for all underwater manipulators, but not to the same extent. Typical UVMSs that consist of a large base, or "body", with a smaller manipulator attached to it, will not be strongly affected by reaction forces and hydrostatic forces, due to the large inertia of this base, compared to the manipulator. ROVs and AUVs designed for intervention are typical examples. Snake-like underwater vehicles such as USRs and USMs are more strongly affected. These robots have a base link that is only a small fraction of the total vehicle's inertia and size. As a result, reaction forces and hydrostatic forces will affect them much more strongly than typical ROVs and AUVs. It is therefore vital to compensate for these effects when using USRs and USMs for manipulation tasks.

Today's control schemes often use inverse kinematics based on the *fixed-base* Jacobian matrix, with the consequence that manipulation is only accurate if the base of the vehicle is held stationary. A stationkeeping algorithm for the base is thus required, as well as thrusters to actuate its control inputs. Conversely, this paper examines control based on the generalized Jacobian matrix (GJM), which avoids the need for base stationkeeping.

The GJM was originally developed for manipulators mounted on free-floating vehicles in the near-vacuum of space, which experience negligible external forces [6], [9]–[11]. Conservation of momentum applies to these systems. Underwater vehicles are subject hydrodynamic effects, and the assumption of no external forces does not hold. The aim of this paper, however, is to investigate whether GJM-based control is feasible, which depends on the significance of the hydrodynamic effects.

This paper will investigate how to achieve accurate and flexible underwater manipulation for underwater manipulators, such as USRs and USMs. Previous UVMS and USM control has used thrusters for stationkeeping during manipulation [12]. This paper contrasts earlier approaches by presenting a solution that does not rely on thruster actuation. Avoiding or reducing thruster usage will save power and extend possible mission duration for battery-powered vehicles. The control system presented can be applied to any UVMSs.

The paper is organized as follows: In Section II, the GJM-based control scheme is introduced. In Section III the simulation environment is presented, and followed by the results of the simulations of the control scheme. Section IV draws conclusions based on the simulation results.

## II. Inverse Kinematic Control

### A. The generalized Jacobian matrix

The GJM provides a map between the joint velocities $\dot{\boldsymbol{q}} \in \mathbb{R}^n$ and the velocity twist $\boldsymbol{\nu} = (\boldsymbol{v}^{\mathrm{T}}, \boldsymbol{\omega}^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^6$ of the end-effector in the inertial frame,

$$\boldsymbol{\nu} = \hat{\boldsymbol{J}}(\boldsymbol{q}, \boldsymbol{R})\, \dot{\boldsymbol{q}}, \tag{1}$$

where $n$ is the number of joints, $\hat{\boldsymbol{J}} \in \mathbb{R}^{6\times n}$ is the GJM, $\boldsymbol{q} \in \mathbb{R}^n$ are the joint positions, and $\boldsymbol{R} \in SO(3)$ is the rotation from the inertial frame to the base link. The symbols $\boldsymbol{v}$ and $\boldsymbol{\omega}$ denote respectively the linear and angular velocity components of the velocity twist vector. Equation (1) contrasts the fixed-base manipulator Jacobian, which maps from joint velocities to end-effector velocity twist, expressed in a vehicle-fixed frame. By expressing the velocities in the inertial frame, the GJM takes into account the rotation and displacement of the end-effector that occurs due to the conservation of momentum.

A complete derivation of the GJM is given in [10]. The kinematic parameters of the vehicle, its link masses and inertia matrices, the joint configuration, and the orientation of the base frame, are required to define the GJM. For implementation, rotary encoders can measure the joint configuration, and the base orientation can be estimated based on measurements from an inertial measurement unit (IMU).

### B. Singularities of the generalized Jacobian matrix

A manipulator is said to be in a singular configuration when its Jacobian is singular, or rank deficient. When the Jacobian is singular, some of its task-space velocities are impossible to reach, and its inverse mapping is not well-defined. In the neighborhood of a singularity, very large joint velocities may be required to reach the desired task space velocities. Large joint velocities can cause spurious and unpredictable motion and may exceed the physical ability of the joint actuators.

Nearly all manipulators have *kinematic singularities*, which are singularities due to its kinematic structure. They can occur at the workspace boundary, and at points where joint axes align. The GJM also exhibits *dynamic singularities*, which depend on the dynamic properties of the vehicle. Kinematic and dynamic singularities reflect physical limitations and are only avoidable by physically staying away from such singular configurations.

USMs and USRs are subject to more significant reaction forces than typical free-floating manipulators, such as spacecraft manipulators and ROVs. This is because the base link of a snake robot is small—roughly the same size and mass as the other links. On the other hand, ROVs and satellite manipulators have base links that are usually much heavier than the entire manipulator. An example is the "Engineering Test Satellite VII", which is a $140\,\mathrm{kg}$ arm mounted on a $2550\,\mathrm{kg}$ base, which has been used for in-orbit GJM experiments [11].

The small relative inertia of the base makes the workspace of a free-floating snake robot small in comparison to its kinematic dimensions, when stationkeeping is not applied. Singularities at workspace boundary and inside the workspace are rarely far away, and to assure precise manipulation, it is necessary to avoid them in a way that minimizes the resulting tracking errors. The next section discusses inverse kinematic methods that take this into account.

### C. Inverse kinematics

As we have discussed, (1) maps from joint velocities to end-effector velocities. To control the manipulator end-effector, it is necessary to define the inverse mapping: calculating the joint velocities needed to achieve the desired end-effector velocity. Because $\hat{\boldsymbol{J}}$ is not generally invertible, this is a nontrivial problem, and its various solutions are termed inverse kinematic methods. In this section, we discuss a singularity robust inverse kinematic control method [13]–[15]. The singularity robustness makes this method suitable for UVMS control. The inverse of (1) can be written

$$\dot{\boldsymbol{q}} = \boldsymbol{f}(\hat{\boldsymbol{J}})\boldsymbol{\nu}. \tag{2}$$

Damped least-squares inverse kinematics [14] is a solution of (2) that attempts to avoid singularities by damping the joint velocities in near-singular configurations. It is the solution that minimizes

$$\left\| \boldsymbol{\nu} - \hat{\boldsymbol{J}}\dot{\boldsymbol{q}} \right\|^2 + \lambda^2 \|\dot{\boldsymbol{q}}\|^2 \tag{3}$$

with respect to the joint velocities $\dot{\boldsymbol{q}}$. The solution is

$$\dot{\boldsymbol{q}} = \left( \hat{\boldsymbol{J}}^{\mathrm{T}}\hat{\boldsymbol{J}} + \lambda^2 \mathbf{I} \right)^{-1} \hat{\boldsymbol{J}}^{\mathrm{T}}\boldsymbol{\nu}, \tag{4}$$

or equivalently [16],

$$\dot{\boldsymbol{q}} = \hat{\boldsymbol{J}}^{\mathrm{T}} \left( \hat{\boldsymbol{J}}\hat{\boldsymbol{J}}^{\mathrm{T}} + \lambda^2 \mathbf{I} \right)^{-1} \boldsymbol{\nu}. \tag{5}$$

Equation (4) is preferable as it requires fewer operations to compute, assuming $n > 6$ [16].

The damping term $\lambda^2\|\dot{\boldsymbol{q}}\|^2$ in (3) is added to damp down joint motion when the manipulator is near a singular configuration. A good choice for the damping factor $\lambda^2$, according to [17], is

$$\lambda^2 = \begin{cases} 0 & \text{when } \sigma_{\min} \geq \epsilon \\ \left[ 1 - \left( \frac{\sigma_{\min}}{\epsilon} \right)^2 \right] \lambda_{\max}^2 & \text{otherwise,} \end{cases} \tag{6}$$

where $\sigma_{\min}$ is the smallest singular value of $\hat{\boldsymbol{J}}$, $\epsilon$ is the singular value threshold below which the damping becomes active, and $\lambda_{\max}$ is the maximum damping factor. With (6), the solution is a pure task velocity error minimization when the manipulator is sufficiently far away from any singularities. As it approaches a singular configuration, the joint velocity damping term will dominate.

### D. Position feedback

Various methods exist for direct position measurements underwater. Global navigation satellite systems (GNSSs) are unavailable underwater, but long baseline (LBL), a similar method based on triangulation with distances to acoustic transponders, is possible. Another alternative is ultra-short baseline (USBL), using a sonar array for position measurements [18]. The drawback of acoustic underwater position measurement methods is that they require externally installed sensor equipment. This drawback does not apply to camera-based simultaneous localization and mapping (SLAM) methods, which is a viable approach in situations with good visibility [19].

For ocean operations without position measuring equipment, accurate estimates of the end-effector position may be hard to obtain. Using an IMU, one can estimate the position by dead-reckoning, which combines and integrates velocity and acceleration level measurements, but the estimate will drift. If only acceleration measurements are available, the drift error will be quadratic in time, and with velocity measurements, it will be linear in time [20]. Estimators based on double integration are usually not sufficiently accurate over time to be used for precision tasks. For this reason, Doppler velocity log (DVL) measurements are often used in inertial navigation systems (INSs) [18]. These indirect estimation methods are not perfect but are often the only available methods for ocean operations away from man-made structures.

A simple method for incorporating position measurements into the manipulator control scheme is to use closed-loop inverse kinematics (CLIK) [21]. CLIK was developed to remove the accumulated integration error from joint trajectory reconstruction [16], but is here used to also account for position errors that occur during underwater GJM-based manipulation. A first-order CLIK algorithm, adapted from [16] to give position but not orientation feedback, is

$$\dot{\boldsymbol{q}} = \hat{\boldsymbol{J}}^{\dagger} \begin{pmatrix} \boldsymbol{v} + \boldsymbol{K}(\boldsymbol{p} - \hat{\boldsymbol{p}}) \\ \boldsymbol{\omega} \end{pmatrix} + \left( \mathbf{I} - \hat{\boldsymbol{J}}^{\dagger}\hat{\boldsymbol{J}} \right) \dot{\boldsymbol{q}}_0, \tag{7}$$

where $\hat{\boldsymbol{J}}^{\dagger}$ is the pseudoinverse of the GJM, $\boldsymbol{K}$ is a constant, positive definite gain matrix, $\boldsymbol{p}$ is the end-effector position in the inertial frame according to the reference trajectory, and $\hat{\boldsymbol{p}}$ is a measurement or estimate of $\boldsymbol{p}$. The second term can optionally be used to exploit the redundancy of a redundant manipulator. To allow us to use the CLIK algorithm with any inverse kinematic method, we extract the velocity twist command

$$\boldsymbol{\nu}_c = \begin{pmatrix} \boldsymbol{v} + \boldsymbol{K}(\boldsymbol{p} - \hat{\boldsymbol{p}}) \\ \boldsymbol{\omega} \end{pmatrix}. \tag{8}$$

We can then combine it with e.g. (4) by inserting $\boldsymbol{\nu}_c$ for $\boldsymbol{\nu}$. A diagram of the closed-loop system, combining position

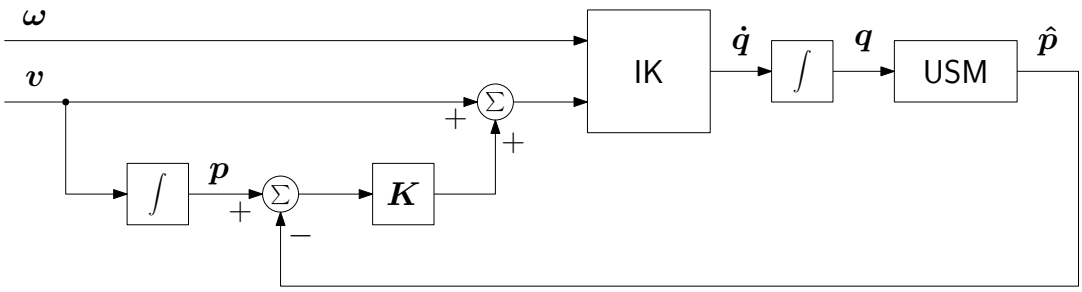



Fig. 3. Diagram of the closed-loop inverse kinematics control system, using position feedback.

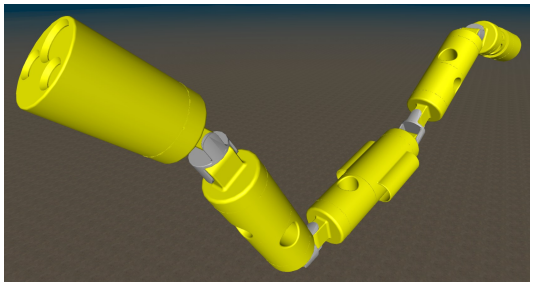


Fig. 4. The USM simulation model, as seen in Vortex

feedback (8) with damped least-square inverse kinematics (4), is displayed in Figure 3.

One aim of this paper is to illustrate the performance differences between open-loop and closed-loop GJM control. This allows us to discuss the capabilities of the open-loop system, and how much it will benefit from position feedback. A thorough investigation of different feedback algorithms is suggested as future work.

## III. Simulations

This section illustrates the potential of GJM based control. We compare two simulation scenarios: Open-loop inverse kinematics, and closed-loop inverse kinematics with position feedback. In the open-loop case, inverse kinematic control is done without correction based on measurements, using (4). In the closed-loop case, position measurements are used to correct the trajectory, by substituting $\boldsymbol{\nu}_c$ from (8) for $\boldsymbol{\nu}$ in (4).

Both scenarios simulate 3 degree of freedom (DOF) linear motion control of the end-effector in inertial space, which is relevant for many use cases, such as positioning a camera gimbal, and light intervention with a multi-DOF end-effector tool. Any combination of the 6 spatial DOFs is possible to control when using the GJM, by extracting the appropriate sub-matrix. The reference trajectories are given as linear velocities

$$\boldsymbol{v}_d \equiv (u_d, v_d, w_d)^{\mathrm{T}}, \tag{9}$$

and as linear positions

$$\boldsymbol{p}_d \equiv (x_d, y_d, z_d)^{\mathrm{T}}, \tag{10}$$

where the positions references are calculated by integrating the velocity references over time,

$$\boldsymbol{p}_d(t) \equiv \int_0^t \boldsymbol{v}_d(t)\,\mathrm{d}t. \tag{11}$$

The reference velocity and position trajectories are identical in both simulation scenarios, to allow fair comparison. The trajectory corresponds to pulling the end-effector backwards, to the right, and up, starting from a nearly outstretched configuration.

### A. Simulation environment

The simulations are run in MATLAB/Simulink [22] and Vortex Studio [23]. The control system is implemented in Simulink, while the dynamics are simulated in Vortex. Vortex performs hydrodynamic simulation and visualization, based on a link-by-link model of a USM, where the kinematic and dynamic properties are specified for each link. The Vortex model developed for these simulations is a continuation of the model presented in [5]. In particular, the model presented in [5] has been extended for links of varying masses and lengths. The number of thrusters has also been increased. The model used in this paper is shown in Figure 4.

### B. Open-loop inverse kinematics

This section investigates inverse kinematic GJM-based control in open-loop, using the scheme in (4). Figure 5 compares the desired and measured end-effector velocity of an example trajectory, along with a comparison of the measured position versus the integrals of the velocity references. The reference trajectory is as described earlier in this section, and Figure 6 shows snapshots of the vehicle configurations throughout the maneuver.

This simulation scenario investigates the extent to which the GJM is applicable for underwater manipulator control, despite the hydrodynamic effects that are not accounted for.

The end-effector velocities follow the trajectory with only small deviations, no larger than approximately $0.02\,\mathrm{cm/s}$. The errors are mostly found in the sway and heave velocities during the steady-state segment of the trajectory. This is presumably due to the hydrodynamics, in particular, the tendency of the vehicle to "swim" when moving its joints. A small component of the transient error can also be attributed to the joint motor dynamics, which are modeled as a first-order system with a $100\,\mathrm{ms}$ time constant.

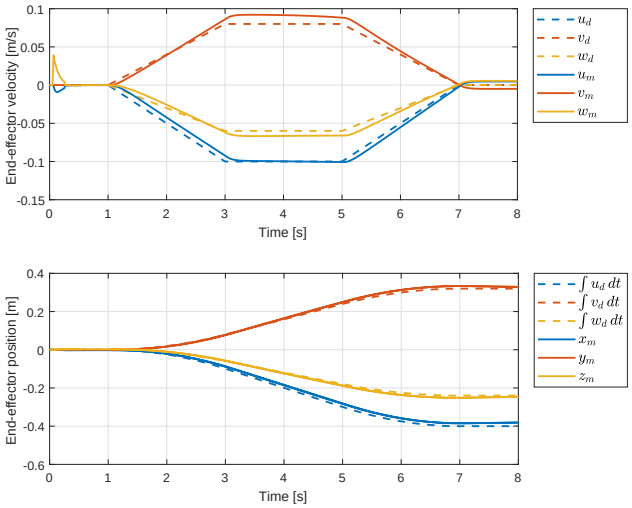



Fig. 5. End-effector velocity (above) and position trajectories (below) for the open-loop maneuver. Dashed lines represent the reference trajectory, and solid lines the simulation output. The lower sub-figure compares the integrated velocity trajectory to the measured end-effector positions.

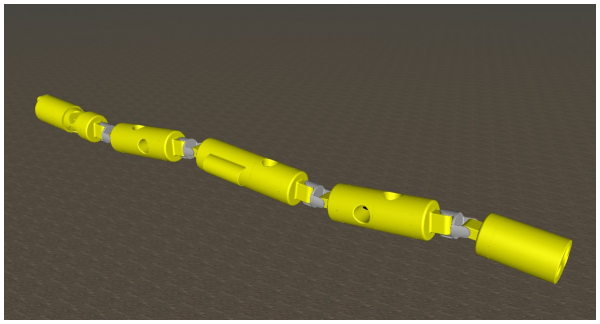
Figure 5 reveals that the surge velocity $u_m$ follows its reference $u_d$ more closely than the sway and heave velocities. The negative surge velocity corresponds to moving the end-effector backward. Under GJM control, the tail must move in the opposite direction to cancel the reaction forces of the end-effector's motion. The hydrodynamic forces caused by the motion of the end-effector and the tail will, therefore, be roughly symmetric and opposite. This gives a small total hydrodynamic disturbance, providing greater precision in the surge direction. On the other hand, the sway and heave velocities of the end-effector do not create a symmetric motion with canceling hydrodynamic effects.

The particular application dictates the level of precision needed for underwater manipulation. If the vehicle is piloted by a human, using a joystick or other device to give end-effector velocity commands, the primary concern is that the resulting end-effector velocities must be proportional to the velocity references. The pilot can easily account for errors in magnitude: If they notice that the end-effector moves too fast or slow, they can adjust the joystick position accordingly. A human pilot will also typically have access to a live camera feed, which allows the pilot to perform manual position feedback, by monitoring the position of the end-effector on the video.

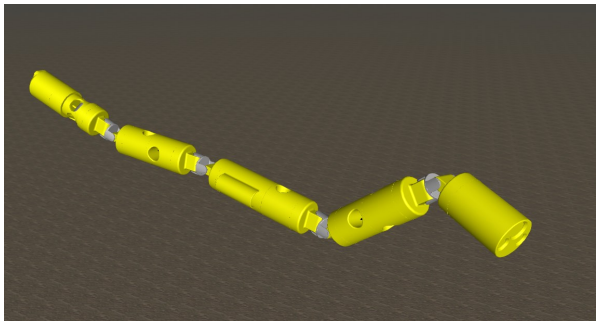
The norm of the position deviation after the maneuver (at $t = 8\,\mathrm{s}$) is $2.2\,\mathrm{cm}$, an accuracy that is sufficient for teleoperation and autonomous operation. However, the position error in open-loop is not corrected and will accumulate over time, so the applicability for autonomous tasks depends on the required time horizon. For teleoperation, the operator will be able to manually correct for position errors over time.

### C. Closed-loop inverse kinematics

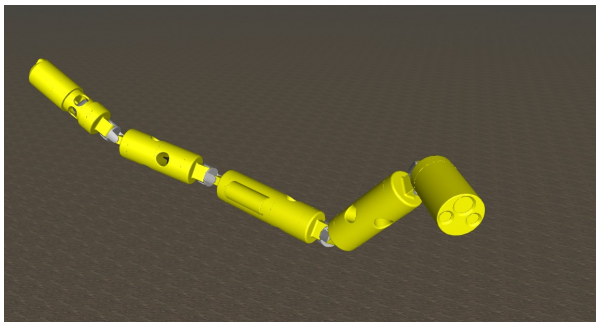
The closed-loop simulation uses CLIK by combining (4) and (8), with $\boldsymbol{K} = 3 \cdot \mathbf{I}_{3\times 3}$. As shown in Figure 7, the position tracking is very accurate throughout the entire maneuver. The



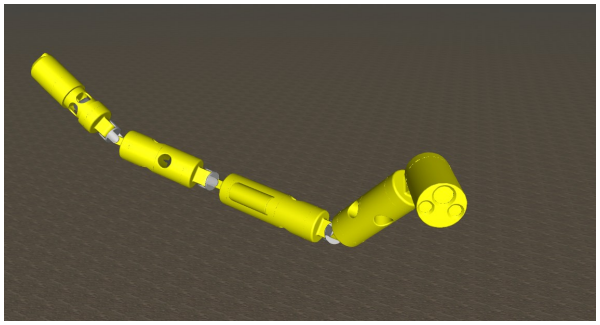
(a) $t \approx 1$



(b) $t \approx 3$



(c) $t \approx 5$



(d) $t \approx 7$

Fig. 6. Snapshots of the vehicle configurations throughout the open-loop maneuver, at even intervals of the simulation time $t$.

velocities are also close to their reference trajectories, but with some deviation, particularly some transients visible at the beginnings and ends of the velocity ramps. This behavior is expected, as changes in velocity are used to fulfill the position tracking objective, through the position error term in (8).

In order for the system to navigate with full autonomy, the requirements on control accuracy are much stricter than

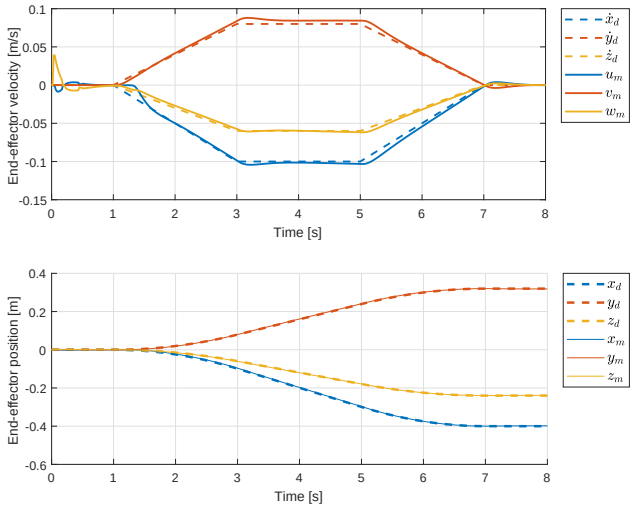



Fig. 7. End-effector velocity (above) and position trajectories (below) for the closed-loop maneuver. Dashed lines represent the reference trajectory, and solid lines the simulation output. The upper sub-figure compares the derivative of the position trajectory to the measured end-effector velocity.

for teleoperation, as no human is present to detect and correct unexpected errors. For intervention tasks, such as valve turning, the end-effector position accuracy needs to be on the order of centimeters (compare for instance the valves used in [24]). The closed-loop simulation shows that the position tracking accuracy is significantly improved compared to the open-loop case. The norm of the position error at the end of the simulation is on the order of millimeters, visible in Figure 7. This accuracy shows the potential for using the proposed controller to perform teleoperation and autonomous tasks, without restriction on the time horizon.

## IV. Conclusions

We have proposed a control system for free-floating underwater manipulators. The control system differs from earlier approaches in that it uses the GJM for inverse kinematic control, which allows us to leave the base of the manipulator unactuated. A result of the unactuated base is that the power normally used to carry out stationkeeping with thrusters can be reduced or eliminated. The system has been shown through simulations to have good performance, illustrated with simulations of an example maneuver.

Inverse kinematics with the GJM without feedback is sufficient for teleoperation. Expanding the control system with position feedback increases accuracy and therefore eases the operator's workload. For autonomous operation, position feedback is preferable, as it accounts for the errors that occur due to the unmodeled hydrodynamic effects, and allows centimeter precision.

The simulations demonstrate that GJM based end-effector control of underwater manipulators is feasible, and that centimeter level accuracy is possible when position estimates are available.

## References


[1] J. Yuh, "Design and control of autonomous underwater robots: A survey," *Autonomous Robots*, no. 1, pp. 7–24.

[2] I. Schjølberg and T. I. Fossen, "Modelling and control of underwater vehicle-manipulator systems," in *Proceedings of the 3rd Conference on Marine Craft Maneuvering and Control*, pp. 45–57.

[3] P. Ridao, M. Carreras, D. Ribas, P. J. Sanz, and G. Oliver, "Intervention AUVs: The Next Challenge," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 12 146–12 159, 2014.

[4] E. Kelasidi, P. Liljeback, K. Y. Pettersen, and J. T. Gravdahl, "Innovation in Underwater Robots: Biologically Inspired Swimming Snake Robots," *IEEE Robotics and Automation Magazine*, no. 1, pp. 44–62, mar.

[5] J. Sverdrup-Thygeson, E. Kelasidi, K. Y. Pettersen, and J. T. Gravdahl, "The Underwater Swimming Manipulator – A Bio-Inspired AUV," in *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)*. Tokyo, Japan: IEEE, pp. 387–395.

[6] Y. Umetani and K. Yoshida, "Resolved Motion Rate Control of Space Manipulators with Generalized Jacobian Matrix," *IEEE Transactions on Robotics and Automation*, no. 3, pp. 303–314, jun.

[7] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Wiley, apr 2011.

[8] P. J. From, J. T. Gravdahl, and K. Y. Pettersen, *Vehicle-Manipulator Systems*, 1st ed., ser. Advances in Industrial Control. London: Springer, 2014.

[9] Y. Umetani and K. Yoshida, "Continuous path control of space manipulators mounted on OMV," *Acta Astronautica*, no. 12, pp. 981–986, dec.

[10] K. Yoshida and Y. Umetani, "Control of Space Manipulators with Generalized Jacobian Matrix," in *Space robotics: Dynamics and control*, Y. Xu and T. Kanade, Eds. Springer, 1993, ch. 7, pp. 165–204.

[11] K. Yoshida, "Engineering Test Satellite VII Flight Experiments for Space Robot Dynamics and Control: Theories on Laboratory Test Beds Ten Years Ago, Now in Orbit," *The International Journal of Robotics Research*, no. 5, pp. 321–335, may.

[12] J. Sverdrup-Thygeson, E. Kelasidi, K. Y. Pettersen, and J. T. Gravdahl, "A control framework for biologically inspired underwater swimming manipulators equipped with thrusters," in *Proc. 10th IFAC Conference on Control Applications in Marine Systems (CAMS)*, vol. 49, no. 23, Trondheim, Norway, 2016, pp. 89–96.

[13] Y. Nakamura and H. Hanafusa, "Inverse Kinematic Solutions With Singularity Robustness for Robot Manipulator Control," *Journal of Dynamic Systems, Measurement, and Control*, no. 3, p. 163.

[14] C. W. Wampler, "Manipulator Inverse Kinematic Solutions Based on Vector Formulations and Damped Least-Squares Methods," *IEEE Transactions on Systems, Man and Cybernetics*, no. 1, pp. 93–101, jan.

[15] A. A. Maciejewski and C. A. Klein, "Numerical filtering for the operation of robotic manipulators through kinematically singular configurations," *Journal of Robotic Systems*, vol. 5, no. 6, pp. 527–552, 1988.

[16] S. Chiaverini, G. Oriolo, and A. A. Maciejewski, "Redundant Robots," in *Springer Handbook of Robotics*, 2nd ed., B. Siciliano and O. Khatib, Eds. Springer International Publishing, ch. 10, pp. 221–242.

[17] S. Chiaverini, O. Egeland, and R. K. Kanestrøm, "Achieving user-defined accuracy with damped least-squares inverse kinematics," in *Fifth International Conference on Advanced Robotics Robots in Unstructured Environments*, pp. 672–677.

[18] J. C. Kinsey, R. M. Eustice, and L. L. Whitcomb, "A survey of underwater vehicle navigation: recent advances and new challenges," *Proc. of MCMC, Lisbon, 2006*, no. May, 2006.

[19] R. M. Eustice, H. Singh, J. Leonard, M. Walter, and R. Ballard, "Visually Navigating the RMS Titanic with SLAM Information Filters," *Robotics: Science and Systems*, pp. 57–64, 2005.

[20] Y. K. Thong, M. S. Woolfson, J. A. Crowe, B. R. Hayes-Gill, and D. A. Jones, "Numerical double integration of acceleration measurements in noise," *Measurement: Journal of the International Measurement Confederation*, vol. 36, no. 1, pp. 73–92, 2004.

[21] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, "Closed-Loop Inverse Kinematics Schemes for Constrained Redundant Manipulators with Task Space Augmentation and Task Priority Strategy," *The International Journal of Robotics Research*, pp. 410–425.

[22] MathWorks Inc., "MATLAB Robotics System Toolbox Release 2017a," Natick, MA, 2017.

[23] CM Labs Simulations Inc., "Vortex Studio Release 6.8.1," Montreal, Canada.

[24] P. Cieslak, P. Ridao, and M. Giergiel, "Autonomous underwater panel operation by GIRONA500 UVMS: A practical approach to autonomous underwater manipulation," *Proceedings - IEEE International Conference on Robotics and Automation*, no. June, pp. 529–536, may.