Simulation of the Underwater Vehicle “Snookie”: Navigating like a Fish

Stefan Sosnowski, Jan-Moritz P. Franosch, Licong Zhang, Yimin Nie,
Kolja Kühnlenz, Sandra Hirche, J. Leo van Hemmen

***Abstract*— Underwater vehicle “Snookie” will serve as a test platform to investigate how fish detect, localize, and avoid objects under water. Fish measure local water velocity changes through their mechanosensory lateral-line system. Obstacles change water velocity on the skin of the fish. Fish use these tiny velocity changes to detect and navigate around objects. Snookie shall mimic its biological archetype by using hot-thermistor anemometry to measure water velocities around its nose.**

We have built a simulation environment to test Snookie’s behavior. The simulation contains a movement model for Snookie, given the motor forces. The water velocities on Snookie’s nose are calculated approximately and change when Snookie comes near to a wall or cylindrical obstacles. Water velocity changes induce voltage changes in the simulated sensors that Snookie uses to detect obstacles and react accordingly.

***Index Terms*— submarine simulation, artificial lateral line, autonomous underwater vehicle (AUV), object avoidance**

I. INTRODUCTION

Fish often use their lateral-line system as their only means of navigation, especially under poor visual conditions. An example is the blind Mexican cave fish that, though without vision, is able to navigate around objects. It appears to “perceive” the objects as it passes by [1]–[3]. It does so by detecting tiny water velocity changes on its skin that occur when the fish passes nearby obstacles.

Fields of possible applications for an underwater vehicle equipped with an artificial lateral-line system could be turbid water with very poor vision as in canalization, pits filled up with water, swamped buildings, and deep sea. Even under good visual conditions, the lateral-line system may prove useful to avoid collisions and to relieve some burden from a camera or sonar system. Future applications of an underwater vehicle equipped with a lateral-line system could therefore be surveying and mapping of waters where humans cannot dive, e.g., because of poor vision, narrow space, or danger of collapse. Object detection by a lateral-line system requires only passive sensors, which has the advantage that sensors do not interfere with each other, e.g., in a swarm of vehicles.

Underwater vehicle Snookie has been built to mimic fish navigation; cf. Fig. 1. It shall avoid objects by measuring water velocity at different positions on its nose. Artificial lateral-line systems are under research [4], but have not yet been used as a robotic sensory system. Instead of using mechanoreceptors, Snookie will use hot-thermistor anemometry to measure water velocities. The novel sensor concept has been verified for its object detection capability in past experiments with a sphere dragged through the water in the vicinity of a wall [5].

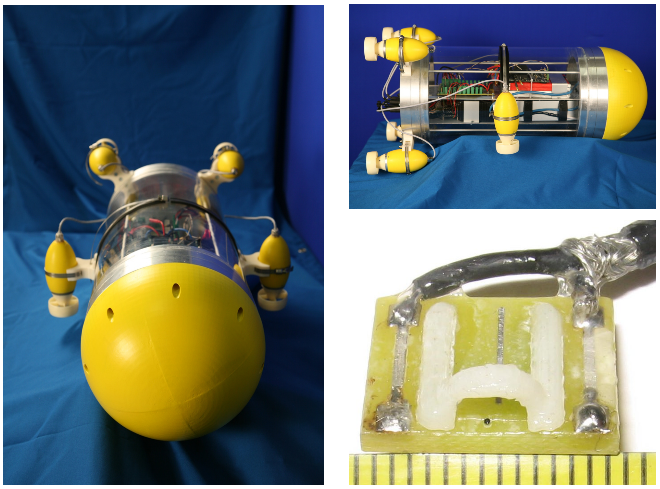


Fig. 1. Prototype of underwater vehicle “Snookie”, here without the semi-sphere on its back. Water velocity sensors will be mounted onto the yellow nose. Lower right: Velocity sensor consisting of a thermistor (small black dot, diameter 0.36 mm) sitting under a protective bow. The thermistor is heated by a current and simultaneously, power dissipation is measured, which in turn depends on water velocity.

For stationary objects, the lateral-line system only works in a very close-up range to the objects to be detected, say, in a distance of about one length of the fish’s body [6]. Moving objects, however, often generate wakes and mark their trajectories through a vortex street [7] - a strategy which could be pursued by a technical system as well.

The contribution of this paper is the approach to utilize a lateral-line sensor for object avoidance in a simulated environment. The simulation serves as a test bed for control strategies and object-detection algorithms by simulating Snookie’s movement, water velocity changes due to the presence of obstacles, voltage changes of velocity sensors mounted on Snookie’s hull as well as Snookie’s evasive reactions. The simulation environment has been custom built because most existing submarine simulations [8] are specifically designed for one type of vehicle or lack the possibility to integrate water velocity calculations. C++ source code is available on request.

The remainder of this paper is organized as follows. First, we introduce the various components of underwater

S. Sosnowski, L. Zhang, K. Kühnlenz and S. Hirche are with Institute of Automatic Control Engineering, Technische Universität München, 80290 München, Germany (sosnowski@tum.de)

Jan-Moritz P. Franosch, Yimin Nie and J. Leo van Hemmen are with Physik Department T35, Technische Universität München, 85747 Garching bei München, Germany (Jan-Moritz.Franosch@ph.tum.de)

robot Snookie as far as they are necessary to understand the simulation requirements. Then models for the water velocity sensors and for vehicle movement are described, and simulated sensor output is compared with experiments. Finally, we report on Snookie's simulated object avoidance behavior.

II. THE UNDERWATER VEHICLE "SNOOKIE"

Snookie has approximately cigar form, a diameter of $25\,\mathrm{cm}$ and an overall length of $74\,\mathrm{cm}$. Six water-tight encapsulated thrusters driven by brushed DC-motors generate thrust up to $7\,\mathrm{N}$ each. Four motors face in forward direction and can thus control forward speed, yaw and pitch; cf. Fig. 1. Two motors generate upward and downward forces and thus directly control depth as well as roll. The motors enable direct control over 5 out of 6 degrees of freedom, making the robot highly maneuverable.

A $60\,\mathrm{MHz}$ ARM7 processor drives the main processing board, which is equipped with an inertia measurement unit and a pressure sensor (Ascending Technologies). The inertia measurement unit fuses and preprocesses sensor data from three MEMS gyroscopes, a three-axis acceleration sensor, a three-axis magnetometer and the pressure sensor. The command unit therefore gets stable and reliable angular and translational data.

The following section describes the sensor model.

III. HOT THERMISTOR SENSOR MODEL

If a thermistor has temperature T, the surrounding water has temperature T_∞ and moves with velocity v, then thermal energy dissipation P_ϑ at the thermistor is approximately [9]–[11]

$$P_\vartheta(v,T) = [A + Bv^n](T - T_\infty) \tag{1}$$

where A, B and n are constants depending on size and shape of the thermistor and the surrounding medium. Thus, the faster water velocity v, the higher is energy dissipation P_ϑ and therefore it can be used to measure water velocity.

The sensors contain very small glass-bead thermistors with $0.36\,\mathrm{mm}$ diameter (Honeywell 111 Series); cf Fig. 1. Power dissipation of these thermistors, when mounted on a small PCB board, fits well the above power law with $n = 0.34$, $A = 1.03\,\mathrm{mW/K}$, and $B = 0.74\,\mathrm{mW/[K(m/s)^n]}$ [12]. The same parameters have been used for simulations.

Given the thermal capacity C of the thermistor and the electrical power P_{el}, the temperature change of the thermistor is

$$C\frac{\mathrm{d}T}{\mathrm{d}t} = P_{el} - P_\vartheta(v,T)\,. \tag{2}$$

In independent measurements, the thermal capacity of the thermistor has been determined as $C = 9 \cdot 10^{-5}$ J/K [13].

Given the current I through the thermistor and its resistance R_ϑ, dissipated electrical power equals

$$P_{el} = I^2 R_\vartheta\,. \tag{3}$$

The resistance R_ϑ of the thermistor depends in turn on its temperature. Given the resistance R_0 at temperature T_0, the resistance R_ϑ of a thermistor is about [14], [15]

$$R_\vartheta = R_0 \exp\left[\beta_\vartheta \cdot \left(\frac{1}{T_0} - \frac{1}{T}\right)\right] \tag{4}$$

where the constant $\beta_\vartheta \approx -3090\,\mathrm{K}$ and $R(20°\mathrm{C}) \approx 2\,k\Omega$ for the type of thermistor used here. In the thermistor simulation a constant current of $I = 19\,\mathrm{mA}$ heats the thermistor, which causes a temperature of about $70°\mathrm{C}$ in experiments [12] as well as in simulations for water velocity $v = 0$. For each time step, given the current temperature T of the thermistor, the simulation first calculates the resistance R_ϑ (4) of the thermistor, the electrical power P_{el} (3), and the dissipated thermal power P_ϑ (1) and then integrates (2) by applying one Euler integration step with a time step of $0.01\,\mathrm{ms}$. The sensor output, namely, the voltage U at the sensor, is then

$$U = \frac{R_\vartheta}{I}\,. \tag{5}$$

To calculate the voltage at the thermistor sensors, we need the water velocity v at each point on Snookie's hull where a sensor sits. The next section describes how to calculate water velocity at Snookie's hull, given Snookie's velocity.

IV. MODEL OF THE FLOW AROUND THE VEHICLE

Say Snookie propagates with constant velocity $\mathbf{V}$. As we are only interested in the flow around Snookie's nose where the water velocity sensors are located, we approximate the whole vehicle by a sphere with radius $a = 12.5\,\mathrm{cm}$. At the Reynolds numbers we are interested in ($\mathrm{Re} = 2aV/\nu \approx 26000$ for velocity $V = 0.1\,\mathrm{m/s}$ with kinematic viscosity ν of water) the boundary layer is very thin. The thickness B of the boundary layer on Snookie's nose is less than $B \approx a/\sqrt{\mathrm{Re}} \approx 1\,\mathrm{mm}$ [16]. Thus we are in the Euler flow regime outside the sphere and can neglect viscosity there. A sphere with radius a moving with velocity $\mathbf{V}$ in a nonviscous fluid generates a so-called dipole velocity field. If the sphere is at the origin of the coordinate system, the water velocity $\mathbf{v}$ at position $\mathbf{r}$ generated by the sphere is [17]

$$\mathbf{v}(\mathbf{r},\mathbf{V}) = \frac{a^3}{2|\mathbf{r}|^5}[3(\mathbf{V}\cdot\mathbf{r})\mathbf{r} - |\mathbf{r}|^2\mathbf{V}]\,.$$

This velocity field fulfills the continuity equation

$$\partial_x v_x + \partial_y v_y + \partial_z v_z = 0$$

as well as the Euler boundary condition that fluid at the boundary moves with the same velocity as the boundary in direction perpendicular to the boundary.

Now let Snookie approach a wall. An infinite wall introduces another boundary condition, viz., that the fluid at the wall cannot move perpendicularly to the wall. As described in [5], we can fulfill the additional boundary condition by introducing another "mirror" sphere; cf. Fig. 2. We get the mirror sphere by mirroring the original's sphere position and velocity by using the wall as a mirror. Because of mirror symmetry, the velocity field generated by both spheres at the wall is then parallel to the wall, as it should be according to the Euler boundary conditions.

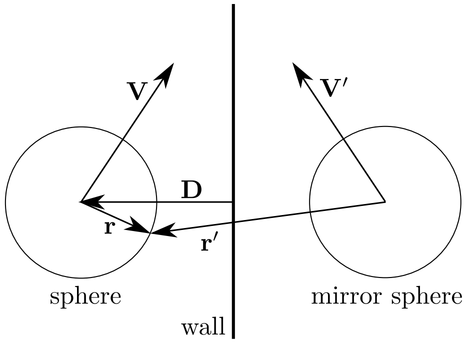


Fig. 2. Snookie, here simplified to be a sphere, moves with velocity $\mathbf{V}$ towards a wall at distance $|\mathbf{D}|$. The mirror sphere moving with velocity $\mathbf{V}'$ serves to fulfill the boundary condition at the wall. A water velocity sensor sits at position $\mathbf{r}$ on the sphere's surface.

Let the original sphere be at the origin of the coordinate system and let the wall be at distance $|\mathbf{D}|$, with the vector $\mathbf{D}$ pointing from the wall to the center of the sphere, perpendicularly to the wall. If the velocity of the sphere is $\mathbf{V}$, then the velocity of the mirror sphere is $\mathbf{V}' = \mathbf{V} - 2\mathbf{D}(\mathbf{D}\cdot\mathbf{V})/|\mathbf{D}|^2$. The center of the mirror sphere is at $-2\mathbf{D}$.

The velocities caused by the moving sphere and by its mirror sphere add up linearly so that the overall water velocity at position $\mathbf{r}$, in the laboratory coordinate system, is

$$\mathbf{v} = \mathbf{v}(\mathbf{r}, \mathbf{V}) + \mathbf{v}(2\mathbf{D} + \mathbf{r}, \mathbf{V}') \,. \tag{6}$$

As $\mathbf{v}$ is not necessarily tangential to the surface, this velocity has still to be projected onto the surface to get an estimate of the velocity at the surface. As the thermistor sensors sit under a protective bow, each sensor has a preferred direction $\mathbf{s}$, with $|\mathbf{s}| = 1$ and $\mathbf{s}$ parallel to the surface of the sphere. In addition, the sphere itself is moving with velocity $\mathbf{V}$. Thus the water velocity v_s measured by a sensor on the surface of the vehicle is

$$v_s = (\mathbf{v} - \mathbf{V}) \cdot \mathbf{s} \,. \tag{7}$$

This approach is an approximation insofar as the boundary condition at the surface of the first sphere is disturbed by the presence of the second sphere and thus boundary conditions on the spheres are only satisfied approximately. The approximation is exact for the limit case $a \to 0$ or $|\mathbf{D}| \to \infty$. The approximation becomes an exact solution by using an infinite series of mirror spheres [17].

In the case when Snookie is passing a cylindrical object, we have approximated the surface of the object by a wall parallel to the cylinder surface. This approximation is good if the radius of the cylinder is large enough. For approximations that are accurate even for small obstacles and account for fish bodies other than spheres, see [18]. Moreover, for the case of a rotationally symmetric fish-like body approaching a wall and gliding alongside a wall, approximations for the resulting flow exist [19], [20]. In our simulations, however, we decided to use the simplest approximations described above as these were easy to implement and describe the general case of arbitrary attack angles between Snookie and wall. Moreover, in our case with relatively high Reynolds numbers $\mathrm{Re} \approx 26000$ one can expect turbulence approximately where the half-spherical nose is attached to the cylindrical body of Snookie [16, Figure 14.15h]. Hence approximations that take into account body shapes other than spherical by simultaneously assuming rotation-free Euler flow like [19], [20] would be inadequate in our case anyway.

To calculate the water velocity at Snookie's hull using (7) and (6), we of course need Snookie's velocity $\mathbf{V}$. The next section describes how the simulation calculates Snookie's movement given the motor forces acting on Snookie.

V. SIMULATION OF VEHICLE MOVEMENT

A. Drag Force on a Steadily Moving Object

Let ϱ be the density of an incompressible fluid, V the velocity of the moving object, A the frontal area, and F_d the drag force, then the drag coefficient

$$c_d = \frac{F_d}{\frac{1}{2}\varrho V^2 A} \tag{8}$$

depends only on the Reynolds number $\mathrm{Re} = Vl/\nu$ where l is a characteristic length of the object (e.g., its diameter) and ν is the dynamic viscosity [21]. For a sphere $c_d \approx 0.3$ between $\mathrm{Re} \approx 1000$ and $\mathrm{Re} \approx 100000$ [21].

Of course Snookie does not only move steadily, but should also decelerate and turn to avoid objects. Thus the simulation has to predict the outcome of so-called zero-velocity maneuvers where inertial forces dominate. Inertial forces not only stem from accelerating Snookie's body itself, but also from accelerating the water around it. Inertial forces dominate over viscous forces if $V^2 \ll \dot{V}l$ [21].

Snookie is equipped with a total propulsive force of $F_{\mathrm{tot}} = 4 \times 7\,\mathrm{N}$. Its radius is $a = 12.5\,\mathrm{cm}$ and its overall length, together with the front and back half-spheres, is $L = 74\,\mathrm{cm}$. Thus, neglecting the motors, with neutral buoyancy its mass is $m \approx \varrho\pi[4a^3/3 + a^2(L - 2a)] = 32\,\mathrm{kg}$. With the acceleration $\dot{V}$ being of the order of $F_{\mathrm{tot}}/m = 0.9\,\mathrm{m/s}$ and setting $l = a$, inertial forces dominate for vehicle velocities $V \ll 0.3\,\mathrm{m/s}$, thus inertial forces clearly have to be taken into account when simulating Snookie's object-avoidance behavior. The next section describes how to calculate forces acting on a body that accelerates in a nonviscous fluid.

B. Drag Force on an Accelerating Object

With pressure p, surface S of the vehicle, normal vector $\mathbf{n}$ of the surface and vector $\mathbf{r}$ describing points on the surface, the force $\mathbf{F}$ that the *fluid* exerts on the moving object and the moment $\mathbf{M}$ are

$$\begin{aligned} \mathbf{F} &= \oint_S p\mathbf{n}\,\mathrm{d}S \,, \\ \mathbf{M} &= \oint_S p(\mathbf{r} \times \mathbf{n})\,\mathrm{d}S \,. \end{aligned}$$

To calculate forces arising from acceleration, we neglect viscosity. In a rotation-free nonviscous fluid, the time-dependent Bernoulli equation

$$p + \varrho\frac{\partial\Phi}{\partial t} + \frac{1}{2}\varrho\mathbf{v}^2 = \mathrm{const.}$$

holds and the velocity $\mathbf{v}$ can be expressed by a velocity potential Φ through $\mathbf{v} = -\operatorname{grad}\Phi$ so that

$$\mathbf{F} = -\varrho \oint_S \left[\frac{\partial \Phi}{\partial t} + \frac{1}{2}\left(\operatorname{grad}\Phi\right)^2\right]\mathbf{n}\,\mathrm{d}S\,.$$

After some calculations [21] one obtains

$$\begin{aligned}\mathbf{F} &= \varrho\frac{\mathrm{d}}{\mathrm{d}t}\oint_S \Phi\mathbf{n}\,\mathrm{d}S\,,\\ \mathbf{M} &= \varrho\frac{\mathrm{d}}{\mathrm{d}t}\oint_S \Phi(\mathbf{r}\times\mathbf{n})\,\mathrm{d}S\,.\end{aligned}$$

The overall velocity potential

$$\Phi(\mathbf{r},t) = \sum_{i=1}^{6}\mathcal{U}_i(t)\varphi_i(\mathbf{r})$$

caused by a moving and turning object as illustrated in Fig. 3 is linear in the velocity $\mathbf{V} =: (\mathcal{U}_1,\mathcal{U}_2,\mathcal{U}_3)$ and the angular rotation $\mathbf{\Omega} =: (\mathcal{U}_4,\mathcal{U}_5,\mathcal{U}_6)$ of the vehicle around a point fixed in the body. Provided that $\mathbf{V}$ is the velocity of the object in a coordinate system that *rotates with the body*, force and moment then become

$$\begin{aligned}\mathbf{F} &= \sum_{i=1}^{6}\Bigg[\varrho\frac{\mathrm{d}\mathcal{U}_i}{\mathrm{d}t}\oint_S \varphi_i\mathbf{n}\,\mathrm{d}S \qquad (9)\\ &\qquad +\varrho\,\mathcal{U}_i\mathbf{\Omega}\times\oint_S\varphi_i\mathbf{n}\,\mathrm{d}S\Bigg]\\ \mathbf{M} &= \sum_{i=1}^{6}\Bigg[\varrho\,\mathcal{U}_i\mathbf{V}\times\oint_S\varphi_i\mathbf{n}\,\mathrm{d}S - \varrho\frac{\mathrm{d}\mathcal{U}_i}{\mathrm{d}t}\oint_S(\varphi_i\times\mathbf{n})\,\mathrm{d}S\\ &\qquad -\varrho\,\mathcal{U}_i\mathbf{\Omega}\times\oint_S\varphi_i(\mathbf{r}\times\mathbf{n})\,\mathrm{d}S\Bigg]\,. \qquad (10)\end{aligned}$$

The moment is to be taken relative to a fixed point in the body, which is assumed to be the origin $\mathbf{0}$.

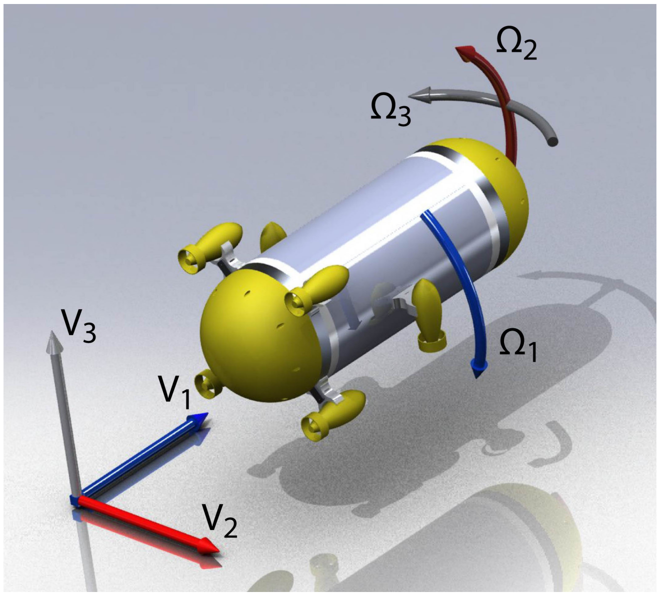


Fig. 3. Underwater vehicle Snookie moving with velocity $\mathbf{V} = (V_1, V_2, V_2)$ and turning with angular velocity $\mathbf{\Omega} = (\Omega_1, \Omega_2, \Omega_3)$.

The assumptions for deriving (9) and (10) are the following [21]. First, the fluid is ideal and irrotational. Second, the body is rigid and does not change shape. Third, the fluid is unbounded and of infinite extent. In our case, the fluid can taken to be ideal only for low Snookie velocities $\mathbf{V}$. We will discuss viscous corrections later on. The second assumption is fulfilled because indeed Snookie does not change shape. The third assumption is fulfilled when Snookie is maneuvering in sufficient depth and sufficiently far away from walls or other obstacles.

With the added-mass tensor [21]

$$m_{ij} := -\varrho\oint_S \varphi_i\frac{\partial\varphi_j}{\partial\mathbf{n}}\,\mathrm{d}S \qquad (11)$$

where $m_{ij} = m_{ji}$, $m_{ij} > 0$, and ε_{ijk} being the fully antisymmetric tensor, equations (9) and (10) can also be written ($i = 1,\ldots,6$; $j,k,l = 1,\ldots,3$)

$$\begin{aligned}F_j &= -\sum_i m_{ji}\dot{\mathcal{U}}_i - \sum_{ikl}\varepsilon_{jkl}m_{li}\mathcal{U}_i\Omega_k\,,\\ M_j &= -\sum_i m_{j+3,i}\dot{\mathcal{U}}_i - \sum_{ikl}\varepsilon_{jkl}m_{l+3,i}\mathcal{U}_i\Omega_k\\ &\quad -\sum_{ikl}\varepsilon_{jkl}m_{li}\mathcal{U}_i\mathcal{U}_k\,.\end{aligned}$$

If $\mathbf{V}'$ is the velocity of the moving object in a *fixed* coordinate system and $\mathbf{V}$ is the velocity of the moving object in a *body-fixed* coordinate system, then

$$\dot{\mathbf{V}}' = \dot{\mathbf{V}} + \mathbf{\Omega}\times\mathbf{V}\,.$$

If ϱ_B is the mass density of the body, which is a function of position inside the body, the inertial force is the change of momentum

$$\mathbf{F} = \frac{\mathrm{d}}{\mathrm{d}t}\int_B \varrho_B(\mathbf{V}' + \mathbf{\Omega}\times\mathbf{r}')\,\mathrm{d}^3r'$$

and the moment is the change of angular momentum

$$\mathbf{M} = \frac{\mathrm{d}}{\mathrm{d}t}\int_B \varrho_B\mathbf{r}'\times(\mathbf{V}' + \mathbf{\Omega}\times\mathbf{r}')\,\mathrm{d}^3r'\,.$$

These equations may be written ($i = 1,\ldots,6$; $j,k,l = 1,\ldots,3$)

$$\begin{aligned}F_j &= \sum_i M_{ji}\dot{\mathcal{U}}_i + \sum_{ikl}\varepsilon_{jkl}M_{li}\mathcal{U}_i\Omega_k\,,\\ M_j &= \sum_i M_{j+3,i}\dot{\mathcal{U}}_i + \sum_{ikl}\varepsilon_{jkl}M_{l+3,i}\mathcal{U}_i\Omega_k\\ &\quad +\sum_{ikl}\varepsilon_{jkl}M_{li}\mathcal{U}_i\mathcal{U}_k\end{aligned}$$

where the $\mathcal{U}_i$ with $1 \leq i \leq 6$, are again the velocities and angular velocities in a *body-fixed* coordinate system. If $\mathbf{c}$ is the center of gravity and I_{ij} are the moments of inertia, the “mass matrix” M is

$$M = \begin{pmatrix} m\mathbf{1} & -m\mathbf{c}\times\\ m\mathbf{c}\times & I\end{pmatrix}$$

where m is the mass of the body, $\mathbf{1}$ is the 3×3 identity matrix, and the matrix

$$\mathbf{c}\times := \begin{pmatrix} 0 & -c_3 & c_2 \\ c_3 & 0 & -c_1 \\ -c_2 & c_1 & 0 \end{pmatrix}$$

is defined so that $(\mathbf{c}\times)\mathbf{x} = \mathbf{c} \times \mathbf{x}$. The equations of motion for an object in an ideal fluid are therefore ($i = 1, \ldots, 6$, $j, k, l = 1, \ldots, 3$) [21]

$$\begin{aligned} F_j &= \sum_i \mu_{ji}\dot{\mathcal{U}}_i + \sum_{ikl} \varepsilon_{jkl}\mu_{li}\mathcal{U}_i\Omega_k \,, && (12) \\ M_j &= \sum_i \mu_{j+3,i}\dot{\mathcal{U}}_i + \sum_{ikl} \varepsilon_{jkl}\mu_{l+3,i}\mathcal{U}_i\Omega_k && (13) \\ &\quad + \sum_{ikl} \varepsilon_{jkl}\mu_{li}\mathcal{U}_i\mathcal{U}_k \end{aligned}$$

where the so-called virtual masses are

$$\mu_{ij} = M_{ij} + m_{ij} \,. \qquad (14)$$

C. Total Drag Force on a Moving Object

In a viscous fluid, we have to take into account additional viscous forces acting on the vehicle. To avoid a full hydrodynamic simulation by exploiting simplicity and computational efficiency, the simulation accounts for viscous drag forces using (8) and thus setting the forward drag force equal to

$$F_{\mathrm{fd}} = -\frac{1}{2}\varrho c_{\mathrm{fd}} A_f |V_1| V_1 \qquad (15)$$

and taking the sideward drag force to be

$$\mathbf{F}_{\mathrm{sd}} = -\frac{1}{2}\varrho c_{\mathrm{sd}} A_s \begin{pmatrix} V_2 \\ V_3 \end{pmatrix} \sqrt{V_2^2 + V_3^2} \qquad (16)$$

where c_{fd} and c_{sd} are the forward and sideward drag coefficients and $A_f = a^2\pi$ and $A_s = a^2\pi + 2a(L - 2a)$ the respective cross-sections with $L = 74\,\mathrm{cm}$ being the overall length and $a = 12.5\,\mathrm{cm}$ the radius of the vehicle. In absence of measurement data and better assumptions, the authors have set forward and sideward drag coefficients to be $c_{\mathrm{fd}} = c_{\mathrm{sd}} = 0.3$, i.e., to those of a sphere.

The resulting system of equations (12), (13), (15), and (16) is equivalent to the "revised standard equations of motion" [8], [22], [23] for submarines, except that viscous damping of angular velocities and viscous coupling of transversal velocities to angular velocities are neglected as no good estimates for the respective coefficients exist yet. In future experiments, it is planned to determine the coefficients of the system of equations experimentally. Of course the assumption that viscous drag forces just add linearly to inertial forces and that viscous forces can be decomposed into forward and sideward forces as described above is only an approximation. The decomposition is correct in the extreme cases that the vehicle moves straightforward ($V_2 = V_3 = 0$) or sidewards and up or down ($V_1 = 0$). The equations of motion described above are, however, widely used in marine hydrodynamics and can explain ship maneuvers fairly well [21].

The next section describes how the simulation of Snookie's aquatic motion works by solving the system of differential equations (12) and (13).

D. Simulation of Movement

For the simulations, it is assumed that the inertia measurement unit provides the controller with exact depth, 3-dimensional orientation, and angular velocity. To simulate the movement of the vehicle, during each time step the simulation computes data coming from a simulated inertial measurement unit, that emulates the data structure of the real hardware. The data are the input to the closed-loop position and orientation control. The closed-loop position and orientation control consists of separate proportional and derivative (PD) controllers for the roll, pitch, and yaw axis (Ω_1,Ω_2 and Ω_3), the depth and the forward velocity (V_1, V_2, V_3) as described in detail in [24]; cf. Fig. 3. The resulting control output is fed into the motor controllers in the same format as in the real hardware, thus the control parameter sets found in the simulation can be used for the hardware as well. The resulting forward and reverse motor forces directly depend on the motor commands. For more accurate simulations, the authors intend to include a motor model later. Using the motor forces, the simulation calculates the forces F_i and the moments M_i acting on the vehicle. Then the simulation code solves the linear system of equations (12) and (13) for $\dot{\mathcal{U}} = (\dot{V}_1, \dot{V}_2, \dot{V}_3, \dot{\Omega}_1, \dot{\Omega}_2, \dot{\Omega}_3)$.

The linear and angular accelerations $\dot{\mathcal{U}}$ are defined with respect to the body-fixed coordinate system in Fig. 3, thus integrating the accelerations ($\dot{V}_1, \dot{V}_2, \dot{V}_3$) leads to a velocity $\mathbf{V}$, which has to be rotated accordingly to get the current velocity in the world coordinate system. Similarly, integrating the angular accelerations ($\dot{\Omega}_1, \dot{\Omega}_2, \dot{\Omega}_3$) leads to the current angular velocities ($\Omega_1, \Omega_2, \Omega_3$) in a body-fixed coordinate system. Thus the current change of angular orientation of the vehicle has to be calculated by first rotating ($\Omega_1, \Omega_2, \Omega_3$) into the world coordinate system while using the current angular orientation of the vehicle.

Specifically, let the columns of the rotation matrix $E = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ be the unit vectors that point into forward, sideward and upward direction; cf. Fig. 3. Then in each time step with duration Δt, the center of mass position $\mathbf{c}$ is updated according to

$$\mathbf{c}(t + \Delta t) = \mathbf{c}(t) + E(t)\mathbf{V}(t)\,\Delta t \,,$$

the velocity $\mathbf{V}$ according to

$$\mathbf{V}(t + \Delta t) = \mathbf{V}(t) + \dot{\mathbf{V}}(t)\,\Delta t \,,$$

the angular velocity $\mathbf{\Omega}$ according to

$$\mathbf{\Omega}(t + \Delta t) = \mathbf{\Omega}(t) + \dot{\mathbf{\Omega}}(t)\,\Delta t \,,$$

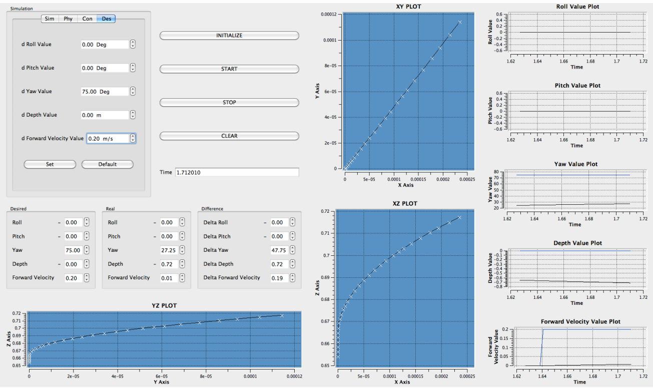
and the columns $\mathbf{e}_i$, $i = 1, \ldots, 3$, of the rotation matrix E according to

$$\mathbf{e}_i(t + \Delta t) = \mathbf{e}_i(t) + E(t)\mathbf{\Omega}(t)\,\Delta t \times \mathbf{e}_i(t) \,.$$

For $\Delta t \to 0$, the rotation matrix stays orthogonal, as it should, but due to $\Delta t > 0$, it does not. Thus the simulation re-orthogonalizes E by setting $\mathbf{e}_i$ to $\mathbf{e}_i/|\mathbf{e}_i|$ and $\mathbf{e}_3$ to $\mathbf{e}_1 \times \mathbf{e}_2$.

We do not know the added-mass tensor m_{ij} (11) for Snookie's body, but for an ellipsoid of revolution (spheroid), it can by calculated analytically [21]. Thus we have approximated Snookie's shape by an ellipsoid of revolution with the same length and diameter as Snookie. For an ellipsoid of revolution around the x-axis with length $L = 2b$ and maximum diameter $2a$, $m_{11} = \frac{4}{3}\pi\varrho ba^2 m'_{11}$, $m_{22} = m_{33} = \frac{4}{3}\pi\varrho ba^2 m'_{22}$, $m_{44} = 0$ and $m_{55} = m_{66} = \frac{4}{15}\pi ba^2(a^2 + b^2)m'_{55}$. The m'_{ij} depend on the ratio b/a and are given in [21, Figure 4.8].

Simulation data are displayed in real time on the screen and parameters (e.g., desired bearing and depth, controller gains) can be adjusted during the simulation with a graphical user interface. Fig. 4 shows a screen shot of the simulator.


Simulation

d Roll Value 0.00 Deg
d Pitch Value 0.00 Deg
d Yaw Value 75.00 Deg
d Depth Value 0.00 m
d Forward Velocity Value 0.20 m/s

Time 1.712010

Desired		Real		Difference	
Roll	- 0.00	Roll	- 0.00	Delta Roll	- 0.00
Pitch	- 0.00	Pitch	- 0.00	Delta Pitch	- 0.00
Yaw	75.00	Yaw	27.25	Delta Yaw	47.75
Depth	- 0.00	Depth	- 0.72	Delta Depth	0.72
Forward Velocity	0.20	Forward Velocity	0.01	Delta Forward Velocity	0.19

Fig. 4. A screen shot of the submarine simulation program. It is programmed in C++ and uses Qt for the graphical user interface. The graphs with blue background display the current position of Snookie in the three main coordinate plains. The graphs in the right column show time-dependent roll, pitch, yaw, depth and forward velocity. The user can adjust desired values, e.g., depth and velocity, online during the simulation and can tune various control parameters.

E. Estimation of Maneuvering Capabilities

As the water velocity detectors on Snookie's hull are near-range sensors and can only detect obstacles that are about one nose radius away, Snookie has to be able to maneuver very fast. Ship are known to maneuver very slowly, so why should Snookie be able to avoid obstacles that are detected right in front of its nose? The reason is that Snookie lives underwater and that every buoyancy-neutral underwater vehicle can stop in about one body length.

In the following, we calculate the range d that an underwater vehicle with motor force F needs for stopping in a worst case. Given the virtual mass μ (14) of the vehicle and its velocity V, the accelerating force from (12) is $F = \mu\dot{V}$. We neglect viscous forces here, which help the vehicle to stop even faster. With the density ϱ of the fluid and the radius a and length L of the vehicle, its virtual mass is $\mu \sim \varrho a^2 L$. The range the vehicle needs to decelerate from speed V is $d \sim V^2/\dot{V}$. For the fastest possible speed V, according to (8),

$$F \sim \frac{1}{2}\varrho c_d V^2 a^2 \,,$$

thus the distance d the vehicle needs to stop,

$$d \sim \frac{F}{\varrho c_d a^2}\frac{\varrho a^2 L}{F} = \frac{L}{c_d} \,,$$

is independent of motor force F and front section area a^2. Thus, independently of motor force, the distance an underwater vehicle needs to stop is proportional to its length. As Snookie is far longer than wide, it has to use excess motor force to stop in time. We therefore run Snookie with $0.1\,\mathrm{m/s}$ when doing object avoidance although its maximum speed is faster than $1\,\mathrm{m/s}$ according to simulations.

VI. WALL-DETECTION AND AVOIDANCE ALGORITHM

When Snookie approaches a wall, it has to first detect that a wall is present, then estimate where the wall is, and finally start its avoidance maneuver. During the avoidance maneuver, self-movement disturbs velocity measurement, thus no update of wall position data is possible. Therefore, Snookie has to remember the wall position after it has detected a wall and then perform the wall avoidance maneuver without feedback. In the current simulations, Snookie uses eight sensors positioned as indicated in Fig. 5.

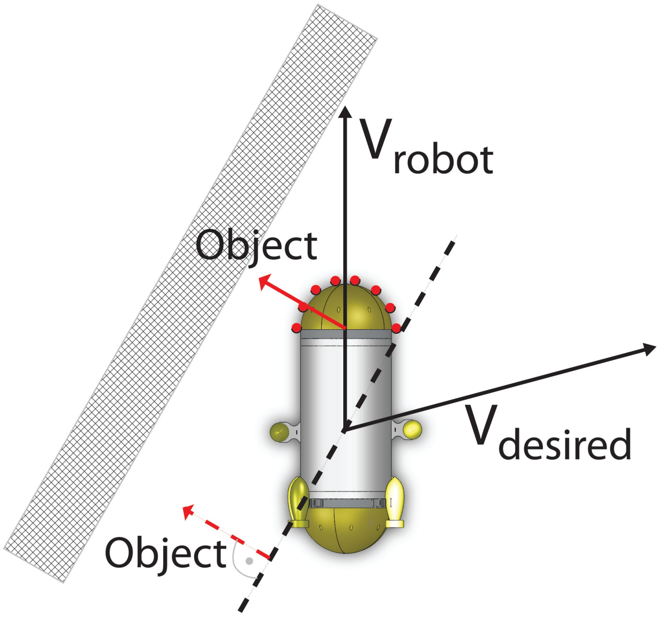


Fig. 5. When Snookie detects a wall, it estimates the direction to the object (red arrow) by a population vector depending on the voltages at its sensors (red dots). The new desired velocity $\mathbf{V}_{\mathrm{desired}}$ is computed by mirroring Snookie's current velocity $\mathbf{V}_{\mathrm{robot}}$ at the estimated plane parallel to the wall (dashed black).

The signal to the sensors is exponentially high-pass filtered with a time constant of $2\,\mathrm{s}$ since very slow changes of water velocity and thus voltage probably do not indicate the presence of a wall. A double criterion has been used to detect a wall. First, the voltage difference between any two sensors must exceed $1\,\mathrm{mV}$. Second, the square root of the sum of squares of the high-pass filtered voltages $\bar{U}_i$, $1 \leq i \leq 8$, of the sensors must exceed $1\,\mathrm{mV}$. As soon as both criteria are fulfilled, the algorithm estimates the direction to the wall by a so-called population vector code [25]. Estimated wall direction $\mathbf{D}_{\mathrm{est}}$ is the sum of the positions $\mathbf{r}_i$ of the sensors relative to the center of the nose, weighted by the high-pass filtered voltage $\bar{U}_i$ of each sensor,

$$\mathbf{D}_{\text{est}} = \sum_{i=1}^{8} \bar{U}_i \mathbf{r}_i \, .$$

The resulting vector $\mathbf{D}_{\text{est}}$ roughly indicates the direction to the wall. New desired velocity is computed by mirroring the current velocity at the estimated surface of the wall that is perpendicular to $\mathbf{D}_{\text{est}}$; cf. Fig. 5.

Snookie remembers desired velocity to use it later on during its maneuvers. First, after detecting that a wall is present, Snookie stops as fast as it can by switching its motors to full reverse speed. When it reaches a backwards velocity of 0.5 m/s, the desired value of forward velocity is set to zero. Desired bearing is set equal to the direction of the value calculated for the new velocity before so that, if Snookie approached the wall before its maneuver, it is now turning away from the wall. When Snookie has reached its new bearing, the absolute value of desired forward velocity is reset to 0.1 m/s.

VII. RESULTS OF OBSTACLE DETECTION SIMULATIONS

Experiments have been done where a sphere with a thermistor sensor mounted on it was dragged past a cylindrical obstacle as in Fig. 6, and the voltage at the thermistor sensor was recorded [12].

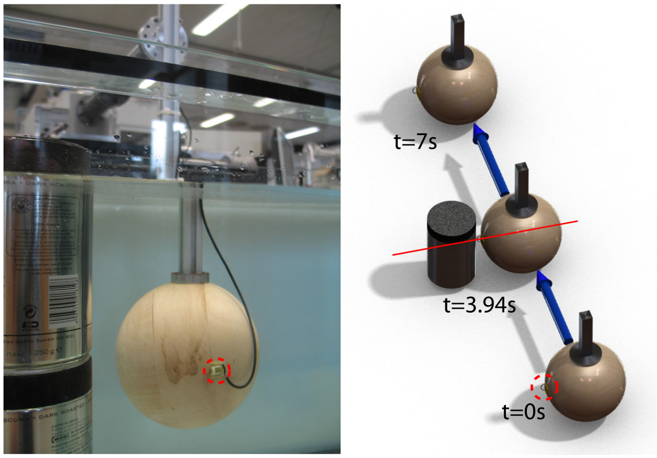


Fig. 6. Experimental setup used for object detection. A sphere (diameter 15 cm, velocity 10 cm/s) with the bow-protected thermistor sensor (dashed red circle) is dragged along a linear axis past a cylindrical object (diameter 9 cm). The minimal distance between sensor and cylinder is 1.5 cm and is reached at the position indicated by the red line.

The simulation started with the same parameters as in the experiment that lead to a voltage response shown in Fig. 7. The overall voltage is significantly higher than the measured voltage stemming from inaccuracies in the model equations (1), (6), and (7). The simulated *change* of voltage turned out to be much lower than the change of voltage actually measured when passing a cylindrical obstacle. This suggests that the actual water velocity increase due to presence of objects is even higher than assumed in the simulation, which would make obstacle avoidance easier. Further experiments have to be carried out to clarify the issue.

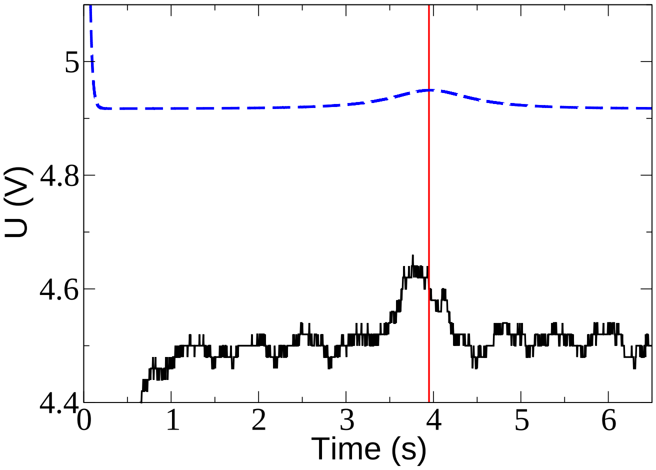


Fig. 7. Solid black line: Measured voltage response U of the thermistor sensor of Fig. 1 with the experimental setup of Fig. 6. The minimal distance between sensor and cylinder is reached at the time indicated by the red line. The body is clearly detectable by the present apparatus (voltage increases in the vicinity of the red line) although there is still some noise mainly due to the power source and a non-constant velocity of the linear axis. The bow-protected sensor has a resistance $R_0 = 1790\,\Omega$ and was heated by a constant current of 18 mA. Dashed blue: Voltage response U of the simulated sensor as given by (5) with the parameters matching the experiment.

Fig. 8 shows simulated voltage responses of the sensors when Snookie passes an object at different distances.

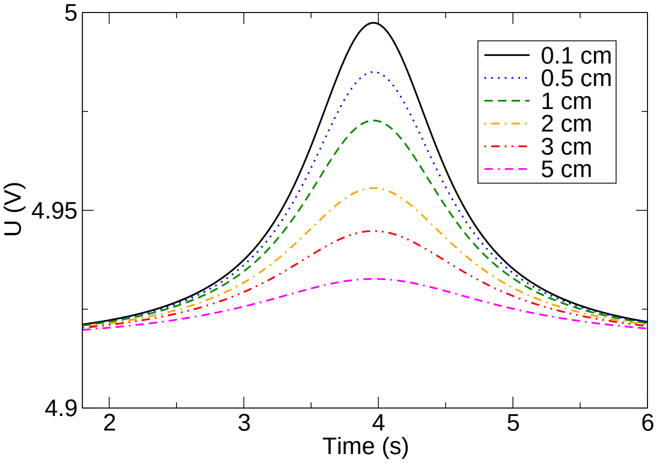


Fig. 8. Simulated voltage responses U of Snookie's left and right sensor when Snookie is passing by a cylindrical obstacle as in Fig. 6 in different distances. The minimal distance of the sensor to the surface of the cylindrical obstacle is indicated in the legend. Snookie itself is simulated by a sphere with the same radius as Snookie's nose (12.5 cm).

Fig. 9 shows an example simulation where Snookie is heading towards a wall, detects that a wall is near, and then avoids the wall. To simulate Snookie's behavior, the simulation solved the equations of motion (12) and (13) and calculated the water velocities on Snookie's hull (6), (7), given the distance $\mathbf{D}$ to the wall. The voltage response (5) of eight equidistant water velocity sensors mounted at the left- and right-hand side of Snookie's nose has been calculated.

Snookie acted according to the wall-detection and avoidance algorithm described above and managed to avoid the wall.

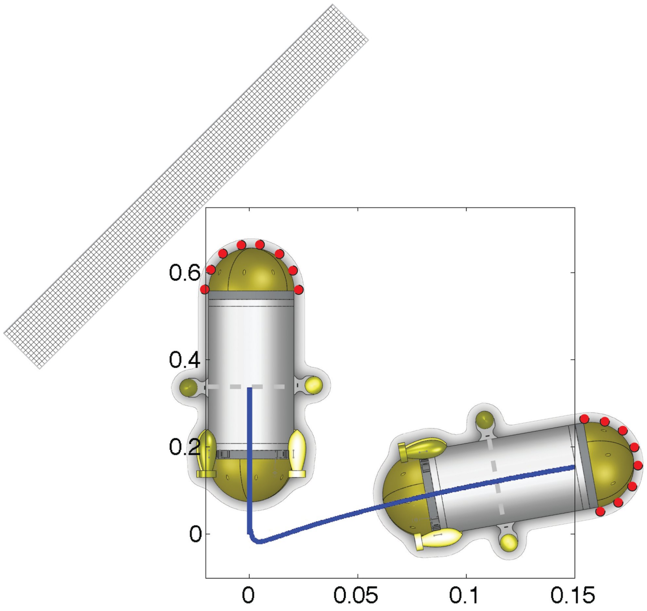


Fig. 9. Illustration of object avoidance. The blue curve indicates Snookie's path during a simulation run (ideal sensory data, no environmental noise) where Snookie is heading with 0.1 m/s towards a wall with an incidence angle of 45°. The robot starts at (0,0), detects the wall, comes to a full stop after 0.34 m, drives backwards and turns to avoid the wall.

VIII. Conclusion and Future Work

By simulating the whole vehicle together with the sensors on a physical level in a closed-loop wall-avoidance scenario, we have seen that object avoidance is feasible using the technique of measuring water velocities on an underwater vehicle's hull. The sensors have the necessary short time constants and suitable sensitivity. Snookie's main controllers keep depth and bearing stable and smoothly react to new set values. Snookie has sufficient maneuvering capabilities to stop in time and avoid walls.

Further experiments have to be performed to identify missing parameters in Snookie's equation of motion. Underwater recordings of sensory data will be performed to test and improve the physical modeling of the influence of walls and cylindrical obstacles on water velocity. Also, the performance of the wall detection and direction estimation algorithm has to be investigated systematically. When object detection and avoidance is indeed possible, another area of research will be how Snookie can map its environment and keep track of its position relative to the obstacles detected.

IX. Acknowledgments

The authors thank all students who have contributed to the autonomous underwater vehicle project. This work is supported in part within the DFG excellence initiative research cluster *Cognition for Technical Systems – CoTeSys*, see also `http://www.cotesys.org`, and by the Bernstein Center for Computational Neuroscience (BCCN) Munich.

References

[1] T. Teyke, "Collision with and avoidance of obstacles by blind cave fish *Anoptichthys jordani* (*Characidae*)," *Journal of Comparative Physiology A*, vol. 157, pp. 837–843, 1985.

[2] H. Abdel-Latif, E. S. Hassan, and C. von Campenhausen, "Sensory performance of blind Mexican cave fish after destruction of the canal neuromasts," *Naturwissenschaften*, vol. 77, pp. 237–239, 1990.

[3] E. S. Hassan, H. Abdel-Latif, and R. Biebricher, "Studies on the effects of Ca++ of the blind Mexican cave fish and Co++ on the swimming behavior," *J. Comp. Physiol. A*, vol. 171, pp. 413–419, 1992.

[4] Y. Yang, N. Nguyen, N. Chen, M. Lockwood, C. Tucker, H. Hu, H. Bleckmann, C. Liu, and D. L. Jones, "Artificial lateral line with biomimetic neuromasts to emulate fish sensing," *Bioinspir. Biomim.*, vol. 5, p. 016001, 2010.

[5] N. Martiny, S. Sosnowski, K. Kühnlenz, S. Hirche, Y. Nie, J.-M. P. Franosch, and J. L. van Hemmen, "Design of a lateral-line sensor for an autonomous underwater vehicle," in *8th IFAC International Conference on Maneuvering and Control of Marine Craft, MCMC 2009*, Sao Paulo, Brazil, 2009, pp. 292–297.

[6] A. B. Sichert, R. Bamler, and J. L. van Hemmen, "Hydrodynamic object recognition: When multipoles count," *Phys. Rev. Lett*, vol. 102, p. 058104, 2009.

[7] J.-M. P. Franosch, H. J. A. Hagedorn, J. Goulet, J. Engelmann, and L. J. van Hemmen, "Wake tracking and the detection of vortex rings by the canal lateral line of fish," *Phys. Rev. Lett.*, vol. 103, p. 078102, 2009.

[8] O. Matsebe, M. Kumile, C, and N. S. Tlale, "A review of virtual simulators for autonomous underwater vehicles (AUVs)," in *NGCUV 2008*, Lakeside Hotel, Killaloe, Ireland, 2008.

[9] H. H. Bruun, *Hot wire anemometry*. Oxford: Oxford Univ. Press, 1996.

[10] I. C. Itsweire and K. N. Helland, "A high-performance low-cost constant-temperature hot-wire anemometer," *Journal of Physics E-Scientific Instruments*, vol. 16, no. 6, pp. 549–553, 1983.

[11] H. Strickert, *Hitzdraht- und Hitzfilmanemometrie*. Berlin: VEB Verlag Technik, 1974.

[12] J.-M. P. Franosch, S. Sosnowski, N. Kuheniri Chami, S. Hirche, and J. L. van Hemmen, "Biomimetic lateral-line system for underwater vehicles," 2010, abstract submitted to the IEEE Sensors 2010 Conference, Hawaii, Nov. 1-4. 2010.

[13] B. Schröfel, "Design and development of a constant-temperature-circuit for fluid flow detection on a diving robot," Master's thesis, Technical University of Munich, 2009.

[14] D. E. Murphy and R. E. Sparks, "A thermistor anemometer for measurement of low fluid velocities," *Industrial & Engineering Chemistry Fundamentals*, pp. 642–645, 1968.

[15] B. Mauconduit and M. Trinite, "New low-speed anemometer with constant temperature thermistor," *J. Phys. E: Sci. Instrum.*, vol. 9, pp. 1091–1096, 1976.

[16] R. L. Panton, *Incompressible Flow*, 3rd ed. Hoboken: Wiley, 2005.

[17] H. Lamb, *Hydrodynamics*, 6th ed. Cambridge: Cambridge University Press, 1932.

[18] E. S. Hassan, "Mathematical analysis of the stimulus for the lateral line organ," *Biol. Cybern.*, vol. 52, no. 1, pp. 23–36, 1985.

[19] ——, "Mathematical description of the stimuli to the lateral line system derived from a three dimensional flow field analysis. I. The case of moving in open water and of gliding toward a plane surface," *Biol. Cybern.*, vol. 66, pp. 443–452, 1992.

[20] ——, "Mathematical description of the stimuli to the lateral line system derived from a three dimensional flow field analysis. II. The case of gliding alongside or above a plane surface," *Biol. Cybern.*, vol. 66, pp. 453–461, 1992.

[21] J. N. Newman, *Marine Hydrodynamics*. Cambridge: MIT Press, 1977.

[22] J. Feldman, "DTNSRDC revised standard submarine equations of motion," DTNSRDC, Tech. Rep., 1979, DTNSRDC/SPD-0393-09.

[23] R. I. Hickey, "Submarine motion simulation including zero speed and propeller race effects," Master's thesis, Massachusetts Institute of Technology, 1990.

[24] I. Vasilescu, C. Detweiler, M. Doniec, D. Gurdan, S. Sosnowski, J. Stumpf, and D. Rus, "AMOUR V: A hovering energy efficient underwater robot capable of dynamic payloads," *The International Journal of Robotics Research OnlineFirst*, vol. 29, pp. 547–570, 2010.

[25] J. L. van Hemmen and A. B. Schwartz, "Population vector code: a geometric universal as actuator," *Biol. Cybern.*, vol. 98, pp. 509–518, 2008.