Visual-Inertial Guidance With a Plenoptic Camera for Autonomous Underwater Vehicles

Jeremiah Eisele, Zhuoyuan Song, *Member, IEEE*, Kevin Nelson, and Kamran Mohseni, *Senior Member, IEEE*

***Abstract*—This letter demonstrates the feasibility of near real-time, plenoptic-inertial navigation on a low-cost central processing unit (CPU). To enable real-time operation, a standard plenoptic camera was modeled as a system of stereo cameras and triangulation was used to estimate its pose from a minimal set of subaperture images. The relationship between distance and disparity for the simplified model was experimentally validated in an aquatic environment, using a first-generation Lytro camera, and a mean error of 2% of the target distance was obtained. This letter culminates with testing the proposed navigation system on an in-house developed, novel, biologically inspired, autonomous underwater vehicle (AUV), *CephaloBot*. The test consisted of the AUV rotating around a static object while maintaining a fixed separation distance. The mean position error from the test was 2.5% of the target distance. With the simplified plenoptic model, only 750 ms were required to process the raw plenoptic data and estimate position on an Intel i5 CPU. The processing delay was short enough that the delayed position measurements bounded the effects of sensor drift when fused with an inertial measurement unit using a delayed extended Kalman filter. This result demonstrates the feasibility of plenoptic-inertial navigation on a low-cost CPU.**

***Index Terms*—Marine robotics, visual-based navigation, localization.**

I. Introduction

AUTONOMOUS underwater vehicles (AUV) are increasingly relied upon to inspect critical underwater infrastructure and perform mapping and surveying tasks [1]. However, their autonomy and performance is clearly dependent upon their perception and navigation capabilities. Since electromagnetic communication signals are severely attenuated by water, the Global Positioning System (GPS) is not available to underwater robots [2]. Instead of GPS, AUVs typically perform localization by combining data from inertial, optical, and/or sonar sensors in a probabilistic fashion [2]. Although sonar has increased range compared to optical cameras, they are generally inferior with regards to resolution, cost, size, weight, and power consumption [3]. Given these advantages, the performance of optical cameras in aquatic environments is an important consideration when designing compact AUVs – especially those tasked with performing underwater inspection and surveying. Monocular cameras tend to be compact and low-cost, but are inherently scale ambiguous and require motion to detect distance [4]. Stereo (multi-camera) systems encode the scene's scale and distance, but they must be carefully aligned, synchronized, and calibrated [4]. Additionally, underwater optical systems are negatively affected by light-attenuation, backscatter , and refraction [5]. Fortunately, there is a relatively new type of optical system, known as a plenoptic (light-field) camera, that exhibits improved performance in many underwater conditions [5].

The plenoptic camera, named after the plenoptic function [6], is comparable in size to a monocular camera, and can detect scene distance and scale. The term "plenoptic camera" first appears in [7], where the authors utilized relay optics, a single main lens, and a 2-D lenticular array to sample the plenoptic function. The first hand-held plenoptic camera [8] resulted from modifying a single-lens reflex (SLR) camera by fixing a micro-lens array (MLA) directly in front of the sensor plane. The authors of [8] determined that the ideal separation distance between the MLA and sensor plane was one micro-lens focal length – this configuration of camera is known as the unfocused or standard plenoptic camera (SPC). An alternate configuration of the plenoptic camera was developed in [9], which has a variable separation distance between the MLA and sensor – this configuration is known as the focused plenoptic camera (FPC).

In recent years, the underwater computer vision community has investigated the benefits of plenoptic cameras in aquatic environments. In [5], [10], and [11], plenoptic image processing techniques were used to reduce the effects of turbidity, particulate, haze, and backscatter. Additionally, researchers have investigated the use of light-fields for plenoptic flow and visual odometry. In [12] and [13], the authors developed and experimentally validated plenoptic motion estimation and closed-form plenoptic odometry (plenoptic flow) algorithms, respectively. Unfortunately, real-time operation with a MLA-based camera would require specialized hardware to decode the raw plenoptic data at sufficient speeds. A direct plenoptic odometry algorithm for an FPC was developed in [14], but a GPU was required to achieve real-time performance.

Manuscript received December 14, 2018; accepted May 6, 2019. Date of publication May 23, 2019; date of current version June 6, 2019. This letter was recommended for publication by Associate Editor M. Dunbabin and Editor J. Roberts upon evaluation of the reviewers' comments. This work was supported in part by the Office of Naval Research and in part by the National Science Foundation. *(Corresponding author: Kamran Mohseni.)*

J. Eisele and K. Nelson are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: jeisele@ufl.edu; kjnelson@ufl.edu).

Z. Song was with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611 USA. He is now with the Department of Mechanical Engineering, University of Hawai'i, Honolulu, HI 96822 USA (e-mail: zsong@hawaii.edu).

K. Mohseni is with the Department of Mechanical and Aerospace Engineering, Department of Electrical and Computer Engineering, and the Institute for Networked Autonomous Systems, University of Florida, Gainesville, FL 32611 USA (e-mail: mohseni@ufl.edu).

Digital Object Identifier 10.1109/LRA.2019.2918677

Unlike traditional cameras, plenoptic cameras encode the intensity, position, and *angle* of light rays entering the camera body. Additionally, the data from an SPC can be configured into an array of sub-aperture images [8], [15], [16] each of which show the scene from a slightly shifted perspective. Because of this, plenoptic cameras are sometimes modeled as an array of "virtual cameras" [5]. According to [17] and [18], the *virtual* cameras' optical axes are either parallel or converging, depending upon whether the *physical* camera's objective lens is focused at optical infinity. The authors in [17] and [18] built a custom SPC and demonstrated that the relationship between object distance and parallax (disparity) has the same form as that found in traditional stereoscopy. However, the distance-disparity relationship in [17] requires the location of the SPC's principal planes and pupils to be known, and this information may not be available in off-the-shelf cameras.

The authors of [19] developed a distance-disparity relationship for an SPC to measure an object's distance. They utilized a first generation Lytro camera to perform experimental validation. However, the authors noted that their method was time-consuming and not automated, thereby limiting the usefulness of their method for real-time navigation purposes.

The use of plenoptic cameras in AUV navigation was first considered by our group in [20], where we focused on target acquisition with a plenoptic (light-field) camera and developed an optimal control strategy for simulation of AUV docking. The goal of this investigation is to design, implement, and experimentally validate a plenoptic navigation system that could operate in real-time on a low-cost central processing unit (CPU). This system could enable low-cost, compact AUVs to simultaneously localize and capture light-field images during inspection and surveying missions; thereby helping to overcome the challenges of underwater navigation and photography. Our approach was to reduce the plenoptic camera model to that of one or multiple stereo cameras and use triangulation to estimate the AUV's position from a minimal set of sub-aperture images. A disparity map was generated by performing block matching [21] and then imposing mutual information based semi-global matching [22]. The camera matrix for the stereo camera model was obtained through calibration with Zhang's method [23]. The disparity map was then converted to a depth-map using the disparity–distance relationship for a stereo camera. To compensate for the low signal-to-noise ratio (SNR) that is inherent to sub-aperture images [8], the target's mean distance was calculated by averaging over it's segmented depth-map. Although more accurate depth estimation techniques exist, even those categorized as "fast" [24] cannot perform in real-time without a GPU or other specialized hardware. Alternatively, we introduce a computationally inexpensive method of estimating pose for compact AUVs without specialized hardware. Additionally, if computational resources exist, significant performance improvements can be expected by incorporating more sophisticated depth estimation algorithms into the proposed approach. A full description of our localization algorithm is presented in Section (II-B). The proposed technique could yield position estimates of sufficient accuracy while utilizing only a small fraction of the available plenoptic data from an off-the-shelf camera. This technique will significantly reduce the processing cost and enable real-time navigation.

To further improve accuracy or reduce computational cost, we consider the suitability of fusing the position measurements with an inertial measurement unit (IMU) using a delayed extended Kalman filter (DEKF). Fusing the sensors would allow for less frequent processing of the plenoptic data. As we show, the plenoptic position estimates place an upper bound on the IMU sensor drift. Additionally, the root mean squared error (RMSE) of the fused state estimates increases with the processing delay of the visual position estimate. This result highlights the importance of reducing the delay period and validates the usefulness of our approach.

To the best of our knowledge, this letter is the first to: design, implement, and experimentally validate a computationally efficient plenoptic camera based navigation system on an AUV, and demonstrate the feasibility of 3-DOF state estimation when fused with an IMU; demonstrate that triangulation from a minimal set of low-resolution, low-SNR sub-aperture images is sufficient for real-time pose estimation in an aquatic environment.

II. Methodology

A. Plenoptic Camera Model

A plenoptic camera is a specialized camera which captures the position, direction, and intensity of incoming light rays [7], [8]. Instead of capturing a 2-D image, as monocular cameras do, a plenoptic camera captures a 4-D function $L(s,t,u,v)$ known as a light-field [16]. The light-field is commonly parameterized using the two-plane-parameterization [5], shown in Fig. 1a, where the $s-t$ and $u-v$ planes represent the spatial and angular coordinates of a light-ray, respectively. An idealized model of a plenoptic camera is shown in Fig. 1b, where Z is the distance from the main lens to an object, z is the distance from the micro-lens array to the object's virtual image, B is the distance from the micro-lens array to the main lens, b is the distance between the sensor plane and micro-lens array, d is the micro-lens pitch, f_1 is the micro-lens focal-length, f_2 is the main lens focal length, and D is the diameter of the main lens aperture.

The plenoptic camera model in Fig. 1b is identical to that of a standard monocular camera, except for the addition of a micro-lens array located at a distance b in front of the sensor plane. For an SPC, the distance b is fixed at the micro-lens focal length and the distance B is adjusted so that objects at the desired focus distance are imaged onto the micro-lens array. This is in contrast to monocular cameras where objects at the focus distance are focused onto the sensor array. The purpose of the micro-lens array is to separate incoming light rays by angle of incidence [8]. Behind each micro-lens is a 2-D array of pixels with u,v coordinates. Each pixel measures the light intensity from a particular direction. A sub-aperture image is formed by choosing a pixel from behind each micro-lens, where each pixel has the same angular (u,v) coordinates [8]. Each sub-aperture image, or view, shows the scene from a slightly different perspective. The perspective shift, or disparity, between sub-aperture images is a function of an object's distance from the camera. A general expression relating distance Z and disparity Δ_x for a plenoptic camera was derived in [19] as:

$$\frac{1}{Z} = \frac{1}{f_2} - \frac{1}{B} + \frac{bd\Delta_x}{B^2 q}. \tag{1}$$

Note that q is the pixel pitch. The optical axes of the virtual cameras are parallel when the main lens is focused at optical infinity [17]. Though this configuration reduces the depth of field, errors associated with underwater optical systems are also reduced – see section (III-A) for additional details. For this case, the distance B will be approximately equal to f_2. Returning to

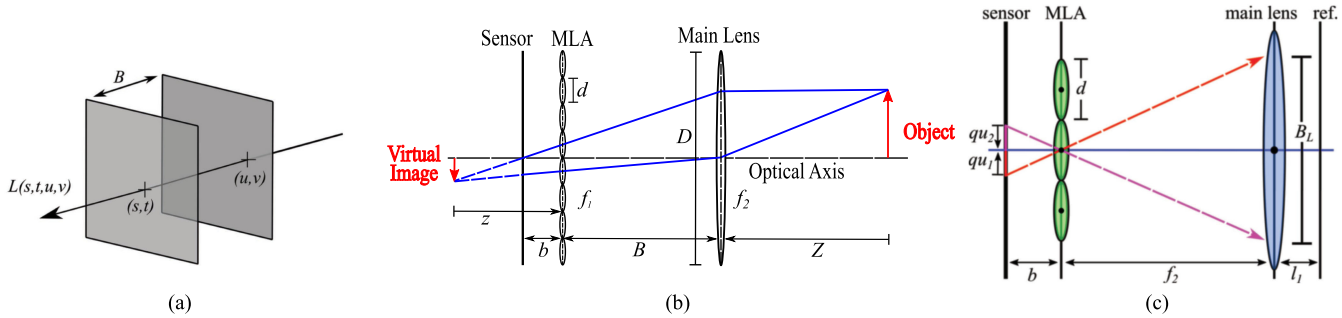

Fig. 1. (a) The two-plane parameterization of an incident light-ray at position (s, t) and angle (u, v). Two parallel planes, separated by an arbitrary distance B, can be used to describe the flow of light through a region of space. (b) Model for a micro-lens based plenoptic camera (adapted from [19, Fig. 9a]). (c) Illustration of the geometric relationship between a pair of virtual cameras and their stereo baseline (adapted from [25, Fig. 33]). Thin-lens approximation was adopted for the main lens.

(1), and substituting f_2 for B, the term $\frac{1}{f_2} - \frac{1}{B} = 0$. Equation (1) can then be rearranged to:

$$Z = f_2 \frac{f_2 q}{b} \frac{1}{d\Delta_x}. \tag{2}$$

Let B_L represent the baseline of the stereo camera model. By examining Fig. 1c, and applying the law of similar triangles, it is clear that $\frac{|qu_2|+|qu_1|}{b} = \frac{B_L}{f_2}$, where u_2 and u_1 are the angular coordinates of the right and left sub-aperture images in our stereo camera model. By letting $\Delta_u = u_2 - u_1$, the expression can be simplified to $B_L = f_2 q \Delta_u / b$.

From Fig. 1c, the height of each micro-lens is equal to d, and each micro-lens corresponds to a pixel in the sub-aperture image [8]. As such, the disparity between two sub-aperture images, in units of length, is $d\Delta_x$ m. As noted in [19], (1) applies to the case where $\Delta_u = 1$. In general, the q in (2) can be replaced with $q\Delta_u$, and then simplified to:

$$Z = \frac{f_2 B_L}{d\Delta_x}. \tag{3}$$

Equation (3) is the well known relationship between distance, focal length, and disparity for a stereo camera with parallel optical axes. It is important to note that (3) was derived from an idealized model with a thin lens. In practice, it is necessary to account for the unknown distance l_1 between some reference plane on the *physical camera* (see Fig. 1c) and the optical center of the virtual cameras. This step was incorporated into our camera calibration procedure.

B. Underwater Visual Localization

When navigating through areas that are in close proximity to the seabed or underwater structures, camera systems can be used to aid in AUV localization. The proposed algorithm was developed for the case where an AUV is maneuvering with respect to a single fixed object (e.g. AUV docking, or inspection tasks), but could be extended to more complex cases. Though not generally required, we assumed a priori knowledge of the target's color and geometry to simplify algorithm development. This assumption would likely be valid when performing AUV docking [20] or navigating around previously surveyed man-made structures (e.g. pilings, oil/gas pipelines). Additionally, the target was assumed to be distinguishable from the background, and static. Since we are using the plenoptic stereo model described in the previous section, we assume that the plenoptic data has been processed into a pair of sub-aperture images using the method from (II-C). We estimated the camera's pose using the following three steps: (i) Disparity and depth map creation; (ii) Image segmentation; (iii) Location of the target's centroid.

Distance can be determined by measuring the disparity between two sub-aperture images and then converting to distance using (3). Estimating disparity from a pair of rectified images is a well explored problem and we used the method from [22]. Given a depth map of the scene, the world coordinates are easily located using the pinhole camera model. The conversion from image to world coordinates requires the camera's intrinsic matrix, which was determined through calibration using the method developed in [23] – see section (III-B) for details. The transformation equation is $[X_c, Y_c, Z_c] = \frac{Z_c}{f}[x_i, y_i, f]$, where X_c, Y_c, Z_c are camera centric world points, x_i, y_i are image points, and f is the virtual camera's focal length. To separate the target from the background, the depth map was segmented using a combination of edge detection, color detection, and distance thresholding. From the segmented depth-map, the mean target distance was determined. Finally, a distinguishable feature was located on the target and it's geometry was used to adjust the measurement from the target's surface to its centroid. The result of this process is a position measurement $[X_c, Z_c]$ from the camera's optical center to the object's centroid.

In section (II-D), we describe the process of using a DEKF to fuse the plenoptic position measurements with inertial data from an IMU. Assuming independent heading and velocity observations are available, the AUV's full state vector is available at the filter's output.

C. Plenoptic Image Processing

The raw data that is captured by an SPC is not a four-dimensional light-field, instead it is a 2-D image of the scene projected through the main lens and micro-lens array. The process of converting the raw data into a light-field is known as decoding [25]. Relevant publications on this topic, as well as calibration and rectification of plenoptic cameras and images include [8], [25]–[27]. Since our objective was to use the plenoptic camera for navigation, we needed an image processing pipeline that was near real-time. The raw Lytro files are approximately 16 MB in size [27] and result in 100 sub-aperture images [25]. Since we only require a single pair of sup-aperture images, we sought to reduce the image processing tasks that operated on the

entire light-field. For this reason, we performed the following steps when converting the raw light-field data into a pair of sub-aperture images: (i) Decoding; (ii) Demosaicing; (iii) Vignetting correction; (iv) Extract an orthogonal subset from the (hexagonally packed) image data, apply median filtering, then rotate and stretch to form sub-aperture images; (v) Stereo rectification.

There are several open-source toolboxes for processing light-field data (e.g. [25] and [27]). Ultimately, we chose the "Lytro Compatible Library" [27] because it was capable of rapidly decoding the raw images and interfacing with the first generation Lytro camera. Once the raw plenoptic images were decoded, they were demosaiced, and vignetting correction was applied with the toolbox's built-in functions. The Lytro's micro-lens array is composed of approximately 328×380 micro-lenses in a hexagonal configuration [27]. This configuration results in non-orthogonal sampling of the scene. To obtain an orthogonal sampling, we opted to discard every other row of data – thereby forming an orthogonal subset. Although this approach significantly simplifies the decoding process and helps make real-time implementation on compact platforms possible, aliasing of scene content above the Nyquist frequency will result unless the raw data is appropriately (low-pass) filtered before sampling. When computational resources permit, one can consider alternative decoding techniques as described in [25] and [26]. The sub-aperture images were then constructed from an orthogonal subset, and vertically stretched by a factor of $\sqrt{3}$. Stretching corrects for the spatial distortion introduced by using an orthogonal subset and the unequal micro-lens density that is inherent to hexagonal configurations [25]. The sub-aperture images were selected from the middle row of available views ($v_1 = v_2 = 0$) and such that the baseline between them was maximized. Stereo rectification of the sub-aperture images was then performed to remove lens distortion and assure that correspondences between the stereo images were in the same row. The parameters required to perform the rectification were obtained previously during calibration. See section (III-A) for calibration details.

D. Sensor Fusion With a DEKF

One of the challenges with using commercial plenoptic cameras for real-time robotic applications is their processing delays. Although our proposed approach is able to extract the depth map in a relatively timely fashion by utilizing a subset of the available plenoptic data, the CPU processing delay prevents the Lytro camera from being relied upon as the sole navigation sensor. Instead of incorporating a GPU or other specialized hardware, we alleviate this issue by fusing the camera measurements with a low-cost IMU in a delayed fashion. Due to the unpredictable delay time, a generic DEKF algorithm introduced by Larsen *et al.* [28] was adopted due to its efficiency towards irregular delays.

The DEKF extrapolates the measurements of the slower sensor to the current time instant to fuse with the other faster sensors. This approach is computationally more efficient than maintaining multiple parallel filters running simultaneously for each sensor, and is desirable for situations with relatively large delays. The DEKF fuses the delayed sensor observation, obtained at time s, at current time step k by extrapolating it such that the measurement is given by $z_k^{\text{ext}} = z_s + C_k\hat{x}_k - C_s\hat{x}_s$, where z_s is the actual sensor observation at time s, C is the measurement Jacobian, and $\hat{x}$ is the estimated system state. The optimal Kalman gain for fusing the delayed sensor observations

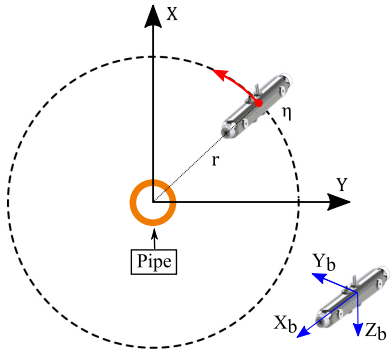

Fig. 2. The desired trajectory is such that the AUV faces the pipe (inertial origin) while *swaying* tangential to the dashed circle. The origin of the body-fixed coordinate system is at the AUV's geometric center (shown lower-right).

can be derived as $K_k = MP_sC_s^\top[C_sP_sC_s^\top + R_s]^{-1}$, where P is the state covariance, R is the covariance of the delayed sensor observation, and M contains the extrapolation information in the form $M = \prod_{i=0}^{N-1}(I - K_{k-i}C_{k-i})A_{k-i-1}$, with N being the number of delayed samples of the faster sensor and A being the motion Jacobian. Interested readers are referred to [28] for details on the derivation of the DEKF.

The DEKF was preferred over other alternative sensor fusion schemes due to its efficiency for real-time systems and low demand for processing power. Since there are no camera measurements fused in the delay period, the extrapolation method adopted by the DEKF is optimal. We note that the DEKF is only applicable when the delayed sensor measurements are fused in the correct order. Alternative sensor fusion schemes may be considered depending on knowledge about the system. For instance, the robust $\mathcal{H}_\infty$ filter [29] can be applied instead if system parameter uncertainties exist. For detailed reviews on various sensor fusion techniques for navigation and tracking, the interested readers are referred to [30] and [31].

E. Control of AUV

In this section, we present the controller development for the pipe survey maneuver depicted in Fig. 2. This maneuver leverages the fact that the platform is fully actuated in surge, sway, and yaw to rotate about the pipe at a fixed radius and with a constant angular velocity such that the pipe is centered in the camera's view. A PID controller was developed to track the trajectory and it is shown to be sufficient to provide tracking stability.

While various other nonlinear controllers could provide tracking stability, we chose to implement a PID controller for the simplicity of implementation and execution. The PID controller is straightforward to implement, gain tune, and debug, as it is a well-established method for stabilizing systems.

1) Desired Trajectory: We develop the trajectory around the pipe such that it is continuous and smooth. In polar coordinates, the trajectory is designed such that the radius, r, is constant and the angle around the object, $\theta(t)$, is changing linearly with time. That is, $r(t) = R$, and $\theta(t) = 2\pi t/T$, where R denotes the radius of the trajectory and T denotes the period of the trajectory. An illustration of the desired trajectory can be seen in Fig. 2. Converting the trajectory to the inertial frame of reference gives us $x_d(t) = R\cos(\theta(t)) + x_p$, $y_d(t) = R\sin(\theta(t)) + y_p$, $\psi_d(t) = \theta(t) + \pi$, where x_p is the x coordinate of the pipe, y_p

is the y coordinate of the pipe, and the desired trajectory of the vehicle can be written in vectorial notation as $\eta_d = [x_d, y_d, \psi_d]^T$ with x_d and y_d being inertial coordinates of the vehicle and ψ_d being the heading angle of the vehicle. It is easy to see that the desired trajectory and its derivatives are smooth, bounded, and continuously differentiable.

2) Controller Design and Stability Analysis: The dynamics of underwater vehicles are generally modeled by the following set of differential equations [32]:

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + G(\eta) = \tau, \tag{4a}$$

$$\dot{\eta} = J_\Theta(\eta)\nu, \tag{4b}$$

where the vector $\eta \in \mathbb{R}^n$ contains the position and orientation of the vehicle in the inertial frame, $\nu \in \mathbb{R}^n$ contains the linear and angular velocity of the vehicle expressed in the body-fixed frame. The term $M \in \mathbb{R}^{n\times n}$ represents a matrix containing the inertial terms of the vehicle. $C : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is a matrix containing the Coriolis/centrifugal terms of the vehicle. $D : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ represents a matrix containing the drag terms. $G(\eta) : \mathbb{R}^n \to \mathbb{R}^n$ represents the restoring forces acting on the vehicle. The vector $\tau \in \mathbb{R}^n$ denotes the control forces and moments. Finally, $J_\Theta : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ represents the velocity transformation from the inertial frame to the body-fixed frame.

Assumption 1: The vehicle is neutrally buoyant and roll and pitch stable and operating in 3-DOF. This assumption allows us to simplify the dynamics by treating $G(\eta) = 0$.

A nonlinear PID controller can be designed to stabilize the system:

$$\tau = J^{-1}(\eta)\Big(K_p\tilde{\eta} + K_d\dot{\tilde{\eta}} + K_i\int_0^t \tilde{\eta}dt\Big), \tag{5}$$

where $K_p, K_d, K_i \in R^{n\times n}$ are diagonal matrices of positive, constant gains.

Theorem 1: The controller given by (5) stabilizes the system (4), given sufficient gains K_p, K_d, and K_i.

The proof of Theorem 1 is well known in control literature and can be found in [32], for example. □

III. Experimental Validation and Results

A. Camera Calibration and Configuration

The effects of refraction on underwater imaging systems have been reported in [33] and [34] and can result in significant measurement error. It was reported in [33] that errors associated with flat-interface refraction are exacerbated if the optical axis of a camera is not perpendicular to its underwater housing's view-port. To obtain this configuration the plenoptic camera's objective lens was focused at infinity so that the optical axes of the virtual camera array were parallel. The camera was then placed directly against the waterproof housing's view-port such that main lens' optical axes was perpendicular to the view-port. In this position, the distance between the camera's optical center and the view-port is minimized which also reduces flat-interface refraction errors [33]. The camera calibration from [23] was performed "in-water" and "in-air". The best results were obtained by calibrating in-air and then applying a focal length adjustment according to [35].

Prior to calibration, the Lytro's zoom setting was adjusted to $1.5x$, the auto-focus was disabled, and it was manually focused at optical infinity. The stereo calibration was performed by capturing 20 images of a 7×9 checkerboard pattern with 25.4 mm squares. The distance from camera to checkerboard was approximately 500 mm. The plenoptic images were then processed according to the method described in section (II-C) to extract the sub-aperture images. Finally, the sub-aperture images were used to perform the calibration procedure described in [23] to obtain the camera matrices for the virtual stereo cameras. The camera matrix can be decomposed into the extrinsic and intrinsic matrices. The extrinsics relating the stereo cameras contain the stereo baseline between the virtual cameras. We found that the stereo baseline estimate derived from Fig. 1c was about 10% lower than the baseline obtained via calibration.

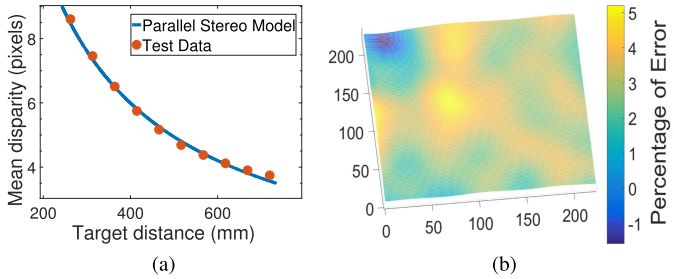

Fig. 3. (a) Experimental results comparing the theoretical and actual relationship between distance and disparity for our plenoptic stereo model. The mean error, as a percentage of distance, was 2% with a standard deviation of 1.4%. (b) A 3-D representation of the depth estimation error over the camera's field-of-view as a percentage of the planar target's distance. The camera to target distance was 619 mm.

B. Validation of Distance-Disparity Relationship

The distance-disparity relationship (3) for the plenoptic-stereo model, described in section (II-A), was experimentally validated in an aquatic environment with a 1st generation Lytro camera. The camera settings were adjusted according to section (III-A) and were not changed during the test. The camera was placed in a waterproof housing and attached to a rail in front of a checkerboard calibration pattern. The assembly was then submerged in water. A total of 16 images were taken as the distance from camera to target was adjusted from 152 mm to 914 mm, in 50.8 mm increments. The plenoptic images were processed according to the method detailed in section (II-C) to extract a pair of sub-aperture images. A disparity map of the target was created and the mean disparity was calculated at each distance. The dataset was then cropped to include only the data between 254 mm and 711 mm since the other measurements were determined to be outside the camera's depth range. Additionally, the distance from the virtual camera array to the front of the waterproof camera housing was determined to be 69 mm since this resulted in the smallest root mean squared error (RMSE) between the test data and the stereo model. A comparison of the theoretical and actual results are shown in Fig. 3a. The mean error, as a percentage of target distance, was 2% with a standard deviation of 1.4%.

Additionally, an evaluation of the depth estimation error over the camera's field-of-view was performed by examining the depth-map of a planar target. The test was conducted underwater with a planar calibration target located 619 mm from the camera. To minimize the influence of sensor noise, we captured four identical light-field images, processed them as described above,

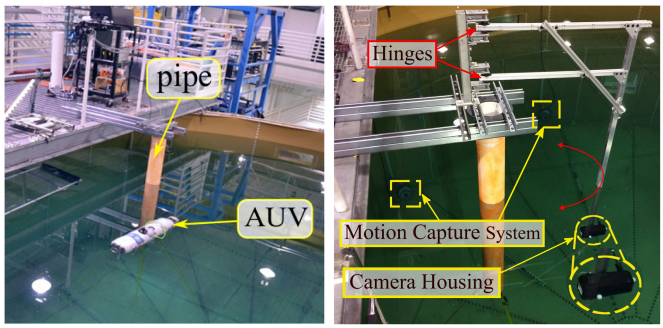

Fig. 4. (a) Vehicle testing tank, showing an orange pipe attached to the platform by Unistrut framing and our AUV in the water. (b) The experimental setup for a mock pipe-inspection test. The underwater camera ("close-up" shown bottom-right) faces the orange pipe as the arm is rotated 180°. Given independent heading measurements, the camera images are used to reconstruct its trajectory.

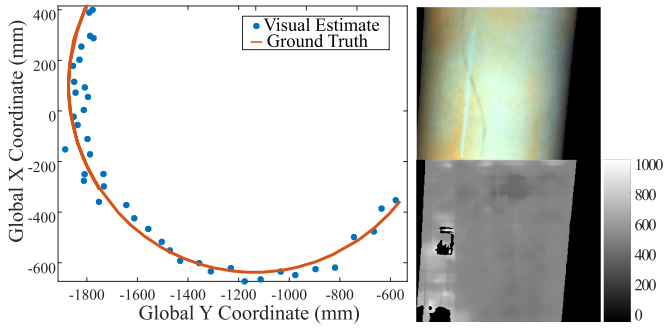

Fig. 5. (a) Results from the 2-DOF mock pipe-inspection test. The solid red line is the ground truth trajectory and the blue dots are plenoptic image based visual position estimates. The camera-to-pipe distance was approximately 650 mm. (b) Segmented image of pipe (top) and corresponding depth-map (bottom).

and combined their disparity maps. The result was converted to a depth-map and aggressively low-pass filtered. Finally, the error associated with each pixel in the depth-map was determined as a percentage of target distance. A 3-D representation of the depth-estimation error is shown in Fig. 3b. The majority of the depth-map was below 3% error, but some regions had error up to 5%. The mean error over the field-of-view was 3% with a standard deviation of 0.4%. These results appear to be in good agreement with our theory and suggest that the proposed method works well when the target is within the camera's depth range. The interested readers are referred to [36] for details regarding the depth range of plenoptic cameras.

C. Testing Facilities

Our lab is outfitted with a large, 225,000 L water tank which is used to perform vehicle testing. The tank is shown in Fig. 4a. A platform overhangs the tank which allows access to the water and a vertical section of orange pipe has been fixed to the platform using Unistrut framing. Six underwater motion capture cameras are located at the bottom of the tank and can be used to localize the vehicle.

D. 2-DOF Mock Pipe Inspection

To assess the performance of our visual navigation system, a pipe inspection experiment was designed to replicate the trajectory of our AUV CephaloBot performing the pipe survey maneuver devised in section (II-E). As shown in Fig. 4b, the experimental setup consists of a rotating mechanical arm mounted atop a vertical section of pipe. The arm is mounted so that it's axis of rotation is dislocated from the pipe's – creating 2-DOF motion between the camera and the pipe. A waterproof housing containing the camera is rigidly attached to the mechanical arm. The ground truth was provided by the motion capture system.

The test was performed by manually rotating the arm about the pipe at approximately 3°/s. Simultaneously, the Lytro's shutter was remotely triggered every 2.5 seconds. The data was processed according to the procedure outlined in section (II-B). A segmented sub-aperture image and depth map of the underwater pipe is shown in Fig. 5b.

The experimental results of the trajectory estimates are shown in Fig. 5a as well as the ground truth for comparison. During this test, the mean distance from camera to target was 650 mm, with a standard deviation of 32 mm. The mean position error was 4.7% of the target distance, with a standard deviation of 2.4%.

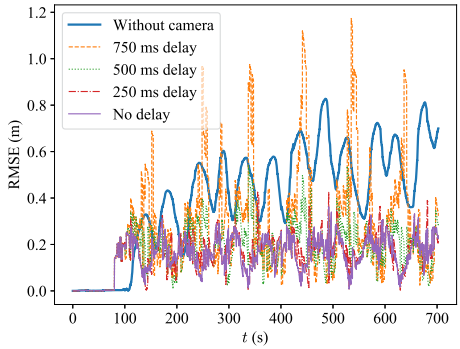

Fig. 6. The RMSE of the camera location in cases not using camera feedback and using camera feedback with 750 ms, 500 ms, 250 ms, 0 ms delay. Independent heading and velocity feedbacks are provided in all cases.

E. Sensor Fusion Results

To demonstrate the feasibility of AUV state estimation from a fused plenoptic-inertial sensor, a BNO055 IMU was rigidly attached to a Lytro camera and the assembly was affixed to the rotating frame shown in Fig. 4b. During the test, the arm was rotated back and forth six times through 180° of rotation. The state vector to be estimated comprises camera location, camera velocity, and camera heading angle. The camera experiences rotational motions as illustrated in Fig. 4b. The IMU samples at approximately 100 Hz while the camera feedback has approximately 2.5 s intervals between consecutive frames. Fig. 6 shows the location estimation errors under five fusion conditions, i.e., not using camera feedbacks, using camera feedbacks with 750 ms, 500 ms, 250 ms, and 0 ms delay. In all cases, real-time feedbacks of velocity and heading are provided. As shown by the results, without image-processing delays, the camera feedback introduces an upper bound to the localization error of the IMU. This bound increases as the image processing delay increases.

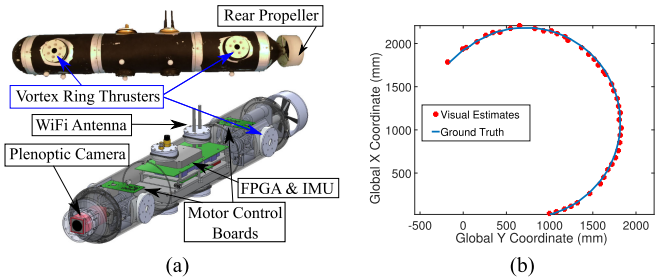

Fig. 7. (a) Our autonomous underwater vehicle, CephaloBot. The plenoptic camera, vortex ring thrusters, and other major components are indicated. The AUV is 1.12 m long and 0.152 m in diameter. (b) Results from the plenoptic navigation system showing the estimated trajectory of the AUV performing a pipe surveying test. The ground truth is shown for comparison.

F. AUV "CephaloBot"

The pipe surveying maneuver was performed on our custom AUV, CephaloBot [37]. CephaloBot is a torpedo shaped AUV designed and manufactured by our group and shown in Fig. 7a. CephaloBot is 1.12 m (44 in) in length and 0.152 m (6 in) in diameter. CephaloBot has a rear propeller which can generate surge control forces and four vortex ring thrusters (VRTs) which provide sway and yaw control forces and moments. Thus, CephaloBot is fully actuated in surge, sway, and yaw which allows the vehicle to implement the control law, eq. (5), and perform the pipe surveying maneuver. In [37], we describe in further detail the design of CephaloBot and its subsystems. A similar design concept has been applied towards several recent iterations of compact AUV prototypes [38], [39].

CephaloBot's four VRTs are custom actuators that were designed and manufactured in-house. These actuators are biologically inspired by the locomotion of jellyfish and cephalopods and provide thrust by successively ingesting and expelling jets of water from an internal cavity via small openings in the hull. The thrusters produce a positive flux of impulse energy even though there is a zero net mass flux over a full pulsation cycle. Additional details on modeling the vortex ring thruster dynamics can be found in our previous studies [40]–[43].

G. AUV Testing

To test the performance of the navigation system under 3-DOF motion, the system was integrated into our AUV CephaloBot. As shown in Fig. 7a, the plenoptic camera was mounted in the AUV's nosecone to enable visual localization relative to an underwater pipe. The test was designed to emulate an AUV surveying a vertical section of a pipeline. State feedback from the motion capture system was fed into CephaloBot's nonlinear PID controller in order to maintain the desired trajectory. See section (II-E) for controller details. This test was performed at the water's surface in order to maintain WiFi communication with a remote PC that was hosting the control algorithm.

The vehicle was placed in the tank facing the pipe at a distance of approximately 2 m. At the start of the test, the AUV approached the pipe until it was about 640 mm from it. At this point, the vehicle performed station-keeping until directed to begin the pipe inspection. Upon command, the AUV traveled along the desired trajectory at a rate of about $3^\circ/\text{s}$. Once it had traveled 180° about the pipe, the test ended. Throughout the test, the camera captured images at the maximum rate of one image per 2.5 s. Results from this test are shown in Fig. 7b and Table I.

TABLE I
SUMMARY OF AUV TEST RESULTS

	Distance	Error X_b, Y_b (%)	Std X_b, Y_b (%)	Time Delay
2DOF	650 mm	4.7, n.a.	2.4, n.a.	n.a.
3DOF	640 mm	1.8, 1.8	1.2, 2	750 ms

The mean distance from the camera's face to the pipe's center was 640 mm with a standard deviation of 50.8 mm. The mean position error in X_b and Y_b was 1.8% and 1.8%, of the target distance, respectively. The standard deviation in X_b and Y_b was 1.2% and 2%, respectively. The 2-DOF and 3-DOF test results are summarized in Table I.

H. Discussion

Our test results show that a single pair of sub-aperture images is sufficient for estimating the pose of an AUV from a known inspection target. Additionally, the trajectory of an AUV undergoing 3-DOF motion, can be recovered provided independent heading and velocity measurements are available. The total time required to process the raw plenoptic data and measure the AUV's position was 750 ms when using a PC with an i5-6200U processor operating at 2.3 GHz. As shown in Fig. 6, this processing delay is small enough to enable fusion of the measurements with an IMU. Fusing the sensors with a DEKF results in a bounded output, thereby overcoming the sensor drift inherent to many IMU's, and reducing the frequency that the plenoptic data must be processed.

Though our results indicate the feasibility of our approach, their are numerous depth-estimation techniques for light-fields that are far more accurate. However, these approaches are either time consuming or require significant computational resources that compact AUVs are unlikely to possess. This suggests that there is value in our approach which strikes a balance between measurement accuracy and computational cost. Another point to consider is that while the proposed navigation system only requires a single pair of sub-aperture images, the raw light-field data would presumably be recorded. This would enable AUVs with minimal resources to capture light-field data during underwater inspection tasks. The raw data can then be processed off-line using plenoptic image processing techniques that reduce the effects of particulate, backscatter, and low-light in underwater imagery. As CPUs advance, one could expect that additional data from the light-field could be processed in real-time to improve the navigation system's performance.

IV. CONCLUSION

This letter developed and experimentally validated a plenoptic navigation system that was capable of near real-time operation on a low-cost CPU. To reduce image processing delays, the plenoptic camera was modeled as pairs of virtual cameras thereby enabling position estimation via triangulation from a minimal set of sub-aperture images. We fused the delayed visual position measurements with an IMU and obtained a bounded output – demonstrating that our measurement delay was small enough for real-time implementation of our approach. Our technique is simple, fast, and could form the basis of more advanced methods. This result demonstrates the feasibility of our proposed system and helps pave the way for plenoptic-inertial navigation on AUVs without specialized hardware.

REFERENCES

[1] A. Lazinica, *Mobile Robots: Towards New Applications*. London, U.K.: I-Tech Education and Publishing, 2006, doi: 10.1109/ICEngTechnol.2012.6396150

[2] L. Paull, S. Saeedi, M. Seto, and H. Li, "AUV navigation and localization: A review," *IEEE J. Oceanic Eng.*, vol. 39, no. 1, pp. 131–149, Jan. 2014.

[3] F. Bonin-font, A. Ortiz, and G. Oliver, "Visual navigation for mobile robots: A survey," *J. Intell. Robot. Syst.*, vol. 53, no. 3, pp. 263–296, Dec. 2008.

[4] M. O. Aqel, M. H. Marhaban, M. I. Saripan, and N. B. Ismail, "Review of visual odometry: Types, approaches, challenges, and applications," *SpringerPlus*, vol. 5, no. 1, 2016, Art. no. 1897.

[5] D. G. Dansereau, "Plenoptic signal processing for robust vision in field robotics," Ph.D. dissertation, School Aerospace Mech. Mechatronic Engr., Univ. Sydney, Camperdown, NSW, Australia, 2014.

[6] E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision," in *Proc. Comput. Models Vis. Process.*, 1991, pp. 3–20.

[7] E. H. Adelson and J. Y. A. Wang, "Single lens stereo with a plenoptic camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 99–106, Feb. 1992.

[8] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, "Light field photography with a hand-held plenoptic camera," Tech. Rep. CSTR 2005-02, Stanford Comput. Sci., Stanford, CA, USA, 2005.

[9] T. Georgiev and A. Lumsdaine, "Focused plenoptic camera and rendering," *J. Electron. Imag.*, vol. 19, no. 2, pp. 1–11, 2010.

[10] K. Skinner and M. Johnson-Roberson, "Underwater image dehazing with a light field camera," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 62–69.

[11] V. Tian, Z. Murez, T. Cui, Z. Zhang, D. Kriegman, and R. Ramamoorthi, "Depth and image restoration from light field in a scattering medium," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2401–2410.

[12] F. Dong, S.-H. Ieng, X. Savatier, R. Etienne-Cummings, and R. Benosman, "Plenoptic cameras in real-time robotics," *Int. J. Robot. Res.*, vol. 32, no. 2, pp. 206–217, 2013.

[13] D. G. Dansereau, I. Mahon, O. Pizarro, and S. B. Williams, "Plenoptic flow: Closed-form visual odometry for light field cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, San Francisco, CA, USA, Sep. 2011, pp. 4455–4462.

[14] N. Zeller, F. Quint, and U. Stilla, "From the calibration of a light-field camera to direct plenoptic odometry," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 7, pp. 1004–1019, Oct. 2017.

[15] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. 23rd Annu. Int. Conf. Comput. Graph. Interactive Techn.*, New York, NY, USA, May 1996, pp. 31–42.

[16] C. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proc. 23rd Annu. Int. Conf. Comput. Graph. Interactive Techn.*, New Orleans, LA, USA, Aug. 1996, pp. 43–54.

[17] C. Hahne, A. Aggoun, V. Velisavljevic, S. Fiebig, and M. Pesch, "Baseline and triangulation geometry in a standard plenoptic camera," *Int. J. Comput. Vis.*, vol. 126, no. 1, pp. 21–35, 2018.

[18] C. Hahne, A. Aggoun, S. Haxha, V. Velisavljevic, and J. C. J. Fernandez, "Light field geometry of a standard plenoptic camera," *Opt. Express*, vol. 22, no. 22, pp. 26659–26673, 2014.

[19] P. Yang *et al.*, "Close-range photogrammetry with light field camera: From disparity map to absolute distance," *Appl. Opt.*, vol. 55, no. 27, pp. 7477–7486, 2016.

[20] Z. Song and K. Mohseni, "Automated autonomous underwater vehicle docking with a light-field camera," in *Proc. MTS/IEEE OCEANS Conf.*, Anchorage, AK, USA, Sep. 18–21 2017, pp. 1–8.

[21] K. Konolige, "Small vision systems: Hardware and implementation," in *Robot. Res.*, Y. Shirai and S. Hirose, Eds. London, U.K.: Springer, 1998, pp. 203–212.

[22] H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, USA, Jun. 2005, no. 2, pp. 807–814.

[23] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.

[24] R. Ferreira and N. Gonçalves, "Accurate and fast micro lenses depth maps from a 3D point cloud in light field cameras," in *Proc. 23rd Int. Conf. Pattern Recognit.*, 2016, pp. 1893–1898.

[25] D. G. Dansereau, O. Pizarro, and S. B. Williams, "Decoding, calibration and rectification for lenselet-based plenoptic cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 1027–1034.

[26] D. Cho, M. Lee, S. Kim, and Y. W. Tai, "Modeling the calibration pipeline of the lytro camera for high quality light-field image reconstruction," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, Dec. 2013, pp. 3280–3287.

[27] J. Kucera, "Computational photography of light-field camera and application to panoramic photography," M.S. thesis, Dept. Software Comput. Sci. Edu., Charles Univ., Prague, Czech Republic, 2014.

[28] T. Larsen, N. Andersen, O. Ravn, and N. Poulsen, "Incorporation of time delayed measurements in a discrete-time Kalman filter," in *Proc. 37th IEEE Conf. Decision Control*, Tampa, FL, USA, Dec. 1998, pp. 3972–3977, vol. 4.

[29] R. M. Palhares, C. E. de Souza, and P. L. Dias Peres, "Robust $\mathcal{H}_\infty$ filtering for uncertain discrete-time state-delayed systems," *IEEE Trans. Signal Process.*, vol. 49, no. 8, pp. 1696–1703, Aug. 2001.

[30] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation: Theory Algorithms and Software*. Hoboken, NJ USA: Wiley, 2004.

[31] D. Smith and S. Singh, "Approaches to multisensor data fusion in target tracking: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 12, pp. 1696–1710, Dec. 2006.

[32] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Hoboken, NJ USA: Wiley, 2011.

[33] T. Treibitz, Y. Schechner, and H. Singh, "Flat refractive geometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 51–65, Jan. 2012.

[34] J. Gedge, M. Gong, and Y.-H. Yang, "Refractive epipolar geometry for underwater stereo matching," in *Proc. Canadian Conf. Comput. Robot Vis.*, St. Johns, NL, Canada, May 2011, pp. 146–152.

[35] J.-M. Lavest, G. Rives, and J.-T. Laprest, "Dry camera calibration for underwater applications," *Mach. Vis. Appl.*, vol. 13, pp. 245–253, 2003.

[36] S. Zhu, A. Lai, K. Eaton, P. Jin, and L. Gao, "On the fundamental comparison between unfocused and focused light field cameras," *Appl. Opt.*, vol. 57, no. 1, pp. A1–A11, 2018.

[37] M. Krieg, P. Klein, R. Hodgkinson, and K. Mohseni, "A hybrid class underwater vehicle: Bioinspired propulsion, embedded system, and acoustic communication and localization system," *Marine Technol. Soc. J.: Special Edition Biomimetics Marine Technol.*, vol. 45, no. 4, pp. 153–164, 2011.

[38] Z. Song *et al.*, "A compact autonomous underwater vehicle with cephalopod-inspired propulsion," *Marine Technol. Soc. J.*, vol. 50, no. 5, pp. 88–101, 2016.

[39] Z. Song, M. Krieg, and K. Mohseni, "Development of a compact autonomous underwater vehicle for hierarchical multi-agent cooperation," in *Proc. MTS/IEEE OCEANS Conf.*, Charleston, SC, USA, Oct. 2018, pp. 1–7.

[40] M. Krieg and K. Mohseni, "Thrust characterization of pulsatile vortex ring generators for locomotion of underwater robots," *IEEE J. Oceanic Eng.*, vol. 33, no. 2, pp. 123–132, Jan. 2008.

[41] M. Krieg and K. Mohseni, "Dynamic modeling and control of biologically inspired vortex ring thrusters for underwater robot locomotion," *IEEE Trans. Robot.*, vol. 26, no. 3, pp. 542–554, Jun. 2010.

[42] M. Krieg and K. Mohseni, "Pressure and work analysis of unsteady, deformable, axisymmetric, jet producing cavity bodies," *J. Fluid Mech.*, vol. 769, pp. 337–368, 2015.

[43] M. Krieg and K. Mohseni, "Modelling circulation, impulse and kinetic energy of starting jets with non-zero radial velocity," *J. Fluid Mech.*, vol. 719, pp. 488–526, 2013.