Sequential Action Control for Models of Underactuated Underwater Vehicles in a Planar Ideal Fluid

Giorgos Mamakoukas, Malcolm A. MacIver, and Todd D. Murphey

Abstract— **This paper uses Sequential Action Control (SAC), a model-based method for control of non-linear systems, for fast, optimal trajectory-tracking tasks in the presence of fluid drift. Through the benchmark example of the kinematic car, it is shown that SAC outperforms a traditional offline projection - based optimization technique in terms of control effort and objective cost. Motivated by recent work on effort-efficient, sight-independent weakly electric fish, this papers also shows that SAC successfully provides control-optimal dynamics to perform short-range underwater maneuvers. Simulation results highlight SAC's robustness to different drift intensities and added mass properties, even when the effective fluid drift is not included in the controller's model.**

I. INTRODUCTION

This paper considers the problem of trajectory-tracking optimization in underwater dynamics. To reach places that are inaccessible to humans, underwater expeditions are executed by autonomous underwater vehicles (AUVs) for commercial, research, and rescue purposes (e.g. seabed exploration, marine life observation, oil spill detection, and Naval activities) [1]–[8]. To successfully perform these activities, AUVs need to consider dynamics, both of the vehicle and the surrounding fluid, and apply a series of control actions to move. Trajectory-tracking has been studied extensively, with respect to robustness to parameter uncertainty, disturbances, and underactuated performance [9]–[14].

Trajectory-tracking in fluid environments has additional difficulties. Under such conditions, AUVs face many challenges, such as limited battery power supply, low visibility, cluttered and turbulent environments, and strong, irregular, and unpredictable underwater drift [15]. Moreover, the fluid environment is not easy to model and its dynamics, affected by water pressure and drift, constantly change as robotic vehicles maneuver to different depths, and in and out of water currents. In general, underwater trajectory-tracking involves Navier-Stokes modeling with boundary conditions that are coupled with the vehicle motion; a very challenging problem requiring approximations for any real-time control approach [16]. The aforementioned issues of underwater tasks can destabilize optimization schemes that either do not run in real time or rely heavily on knowing the exact environmental parameters. Aside from flexibility in the model, successfully optimizing underwater movement requires algorithms that run on-line in order to respond to unexpected changes, as well as to incorporate feedback and apply optimal corrective actions. An algorithm with these traits is Sequential Action Control [17], [18].

SAC is an on-line, closed-loop predictive optimization control algorithm for non-linear systems. Unlike other optimization schemes that iteratively calculate optimal control actions, it computes an analytical closed-form expression for controls at each instant that best improve the stated objective. SAC's solutions have been shown to optimally improve a tracking objective with low execution time and minimal control effort [17]. Reference to optimal SAC controls throughout the paper is meant with respect to the time evolution of the objective function, unlike traditional optimal control methods that directly minimize the objective. Because of these traits, the SAC algorithm can be used to efficiently react to unexpected changes that may occur underwater and provide robustness to parameter variation, while also preserving control effort.

This paper expands the studies on SAC in the context of nonlinear dynamics of an underwater robot; it investigates the algorithm's performance and its robustness to different values of drift intensities and directions [17], [19], [20]. The authors first consider the kinematic car system with environmental drift. Simulations on the benchmark problem of the kinematic car illustrate that the performance of SAC is largely unaffected in the presence of velocity drift. SAC trajectories are compared to trajectories computed with a projection-based optimization scheme [21]; simulation results indicate that SAC produces comparable trajectories using less control effort and resulting in better objective cost.

The algorithm is further tested on a model with the kinematic car dynamics and the added mass parameters of the weakly electric fish, which has been shown to maneuver efficiently [22]. This system is referred to as the dynamic fish-robot throughout the paper. The algorithm's performance on short-range prey-catching scenarios using the animal's dynamics is near-optimal. Given that the fish maneuvers with minimal effort, this result could be of special importance to research inspired by the weakly fish and other systems to minimize power consumption without sacrificing performance. The primary contribution of this paper is showing that SAC is a suitable candidate that can directly address the aforementioned difficulties of trajectory-tracking in fluid environments.

The paper structure is as follows: Section II introduces the SAC algorithm and explains its benefits. Section III presents the application of SAC on the kinematic car and dynamic fish-robot systems and comments on the algorithm performance. Section IV summarizes the findings and reveals ideas for future work.

II. SEQUENTIAL ACTION CONTROL

Sequential Action Control is a closed-loop control algorithm that, in real-time, continuously computes a sequence of optimal infinitesimal controls and application times. During each cycle, SAC forward simulates the dynamics of the system along a user-specified time horizon T and analytically solves for the optimal infinitesimal control as a function of time. It then uses first-order sensitivity of the cost function to infinitesimal applications of the calculated control (called the mode insertion gradient in the hybrid systems literature [23], [24]) in order to determine the best time to act, i.e. the time at which the computed control would reduce the cost function the most. An overview of the SAC algorithm is presented next; for a complete description of the algorithm, refer to Reference [17].

Given dynamics

$$\dot{\vec{x}}(t) = \vec{g}(t, \vec{x}(t)) + H(t, \vec{x}(t)) \cdot \vec{u} \quad \forall t, \tag{1}$$

that are non-linear in state $\vec{x} : \mathbb{R} \mapsto \mathbb{R}^{n\times 1}$ and linear in control $\vec{u} : \mathbb{R} \mapsto \mathbb{R}^{m\times 1}$, SAC sequentially calculates infinitesimal control actions that are optimal with respect to decreasing at a specified rate α_d, not minimizing, a cost function of the form

$$J_1 = \int_{t_o}^{t_f} l_1(t, \vec{x}(t), \vec{u}(t))dt + m(\vec{x}(t_f)). \tag{2}$$

For tracking-objectives, the cost function takes the form

$$J_1 = \frac{1}{2}\int_{t_o}^{t_f} \|\vec{x}(t)-\vec{x}_d(t)\|_Q^2 dt + \frac{1}{2}\|\vec{x}(t_f)-\vec{x}_d(t_f)\|_{P_1}^2, \tag{3}$$

where $\vec{x}_d$ is the desired state-trajectory, and $Q = Q^T \geq 0$, $P_1 = P_1^T \geq 0$ are metrics on state error.

SAC considers two control-modes, $u_1(t)$ – fixed (typically zero) control – and $u_2(t)$, associated with default and optimal dynamics $\vec{f_1}$ and $\vec{f_2}$, respectively. Over the course of each horizon $[t_o, t_f]$, SAC injects optimal dynamics $\vec{f_2}$ for infinitesimal duration. That is, the algorithm switches from the default mode $\vec{f_1}$ to the optimal action mode $\vec{f_2}$ and back to $\vec{f_1}$. The optimal infinitesimal control $\vec{u}_2(t)$ at each time t has an analytical expression that minimizes the control magnitude and the mode insertion gradient to an infinitesimal application of $\vec{f_2}$ dynamics

$$u_2^*(t) = \min_{u_2(t)} \quad \frac{1}{2}(\frac{dJ_1}{d\lambda_i^+} - \alpha_d)^2 + \frac{1}{2}\|u_2(t)\|_R^2, \tag{4}$$

where $\alpha_d \in \mathbb{R}^-$ expresses a minimum (desired) sensitivity and R is a metric on control effort. The analytical solution of equation (4) is

$$u_2^*(t) = (\Lambda + R^T)^{-1}[\Lambda \vec{u}_1 + H^T\vec{\rho}\alpha_d], \tag{5}$$

where $\Lambda \triangleq H^T\vec{\rho}\ \vec{\rho}^T H$, and $\vec{\rho} : \mathbb{R} \mapsto \mathbb{R}^{n\times 1}$ is the adjoint (co-state) variable, given by:

$$\dot{\vec{\rho}} = -D_{\vec{x}} l_1(t, \vec{x}, \vec{u}_1)^T - D_{\vec{x}}\vec{f}(\vec{x}, \vec{u}_1)^T\vec{\rho} \tag{6a}$$

$$\vec{\rho}(t_f) = -D_{\vec{x}} m(\vec{x}(t_f))^T. \tag{6b}$$

The optimal control curve $u_2^*(t)$ returns the optimal control value as a function of time and enables SAC to compute the most effective time to act, i.e., one that minimizes a) control effort, b) the first-order sensitivity of the cost function to applications of $\vec{f_2}$, and c) the cost of waiting:

$$\tau_m = \min_t \quad \|u_2(t)\| + \frac{dJ_1}{d\lambda_i^+} + (t - t_o)^\beta. \tag{7}$$

Last, SAC considers finite controls that could improve the objective more than infinitesimal actions. Starting with an initial finite duration λ centered at τ_m, SAC iteratively reduces the duration via a backtracking line search until the objective improvement is above a specified value. SAC controls exist and are unique, as they are solutions to Tikhonov regularization problems [17]. The algorithm successively, every t_s seconds (sequencing rate), performs the set of computations presented in Algorithm 1.

Algorithm 1 Sequential Action Control

```
• Simulate dynamics $\vec{f_1}$ for $t \in (t_{curr}, t_{curr} + T)$[1]
• Compute initial tracking cost $J_{init,i}$ from equation (2)
• Analytically compute optimal control curve $\vec{u}_2^*(t)_i$
• Search for optimal time $\tau_{m,i}$ to enact infinitesimal
  control $\vec{u}_2(\tau_{m,i})$ from equation (7)
• Saturate control
• Perform Line Search to specify control duration $\lambda_i$,
  centered at $\tau_{m,i} : (\tau_{m,i} - \frac{\lambda_i}{2}, \tau_{m,i} + \frac{\lambda_i}{2})$.
```

The results in [17] show that SAC is promising for on-line optimization problems. Additionally, SAC is computationally very efficient, as it computes an analytical solution to a non-linear optimal control problem. As a result, it avoids the large computational cost involved in solving the $\frac{n\times n+n}{2}$ Riccati differential equations used by open-loop optimal control approaches for $\mathbb{R}^n$-state systems. Further, it readily imposes control constraints, and can avoid local solutions at which SQP algorithms stop prematurely (see [17]).

The computational cost for systems with multidimensional state and control space renders some optimization methods too slow to incorporate feedback and perform in real time. The speed of the algorithm and its ability to scale better with respect to state and control dimensions are the reasons to consider SAC appropriate for real-time applications.

III. APPLICATION TO SYSTEMS WITH DRIFT

In this section, we investigate applications of SAC on systems with drift. The goal is to illustrate the algorithm's ability to track trajectories:

– in non-dynamic/fluid environments
– underwater, in the presence of drift
– using dynamics with added mass in fluid.

The systems we consider are the 2D model of the kinematic car and an underwater model with the same dynamics and added mass parameters from the electric fish (dynamic fish-robot). These examples showcase SAC's general ability to

[1] $t_{curr} = i \times t_s$

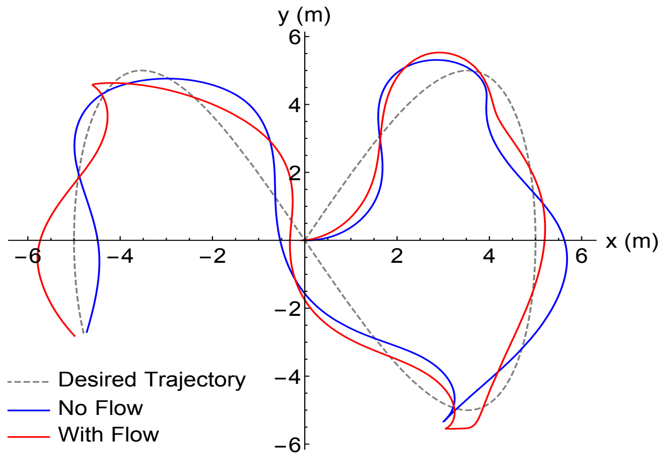


Fig. 1. A parametric plot of SAC-computed trajectories on tracking the desired trajectory (dotted line) using the kinematic car dynamics at a control sequencing frequency of 20 Hz. The performance of the control is tested against no drift and drift of -1 m/s $\hat{x}$. Although the time horizon used for the simulations is extremely short (T = 1 s), the performance of SAC remains largely unaffected by the presence of flow.

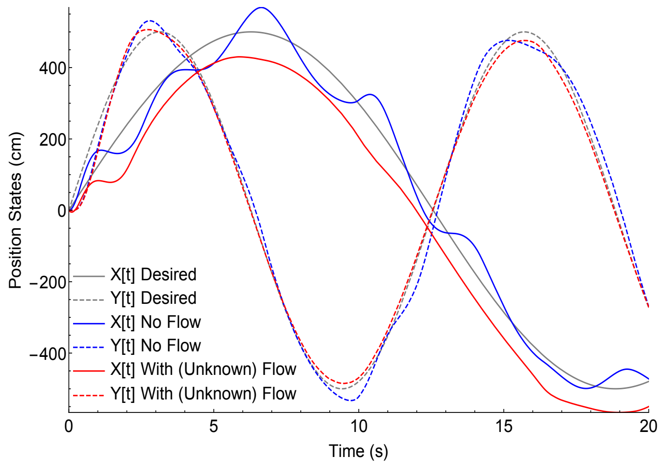


Fig. 2. The SAC algorithm tests the kinematic car system on tracking the reference (gray) signal in two ways. On the first (blue) run, SAC is tested in a non-fluid environment and on the second run (red) in a fluid environment with -1 m/s $\hat{x}$ flow. The simulation uses non-fluid dynamics (absence of drift effects) in both cases–that is, the controller does not know there is fluid drift on the second run. Dotted lines show the x-coordinates and solid ones the y- ones. The reference signal is marked with gray, the neutral-environment test with blue and the underwater one with red.

track trajectories in environments with fluid flow and, more specifically, control the kinematics of the actual systems. The kinematic model of the car tests the general trajectory - tracking ability of SAC in fluid environments. Application on the dynamic fish-robot serves as a stepping stone for ultimately testing SAC on a robotic fish system [25], [26].

Both examples present the same trajectory - tracking task with and without fluid drift. The results of the two cases are compared to show the effect of fluid environments on SAC. Underwater dynamics are simplified to constant fluid velocity drift in the system's state. To model the effect of drift, different intensities of fluid flow are used in both systems.

A. *Kinematic Car*

The kinematic car is a well studied example often used in the literature to measure tracking performance of optimization algorithms [27]–[29]. In this subsection, SAC is tested on the 2D underactuated model of the kinematic car with state $\vec{q} = (x, y, v, \theta, \dot{\theta})^T$, where v is the forward velocity of the car in the body frame, and control input $\vec{u} = (u_D, u_T)^T$ – drive and turn, respectively. The dynamics are modeled as

$$\vec{f}(\vec{x}, \vec{u}) = \begin{pmatrix} v \cdot cos\theta + \dot{x}_w \\ v \cdot sin\theta + \dot{y}_w \\ u_D - \eta_1 \cdot v \\ \dot{\theta} \\ u_T - \eta_2 \cdot \dot{\theta} \end{pmatrix}, \quad (8)$$

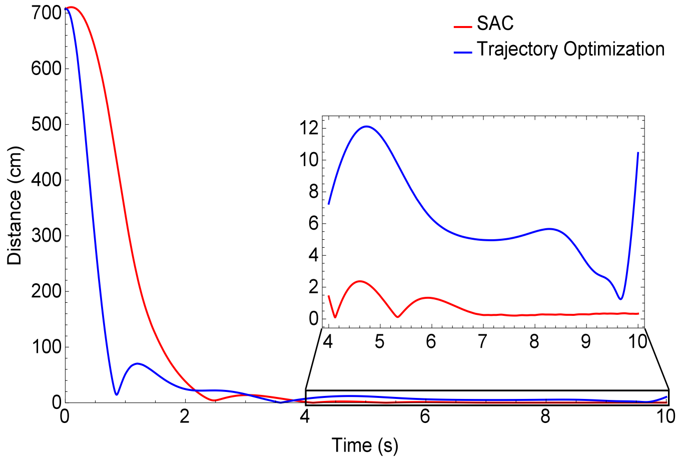
where $\dot{x}_w$ and $\dot{y}_w$ represents the fluid drift in the x- and y- world frame axes and $\eta_1 = 0.01$ 1/s and $\eta_2 = 0.03$ 1/s represent linear and rotational damping coefficients. In this example, SAC is applied to track the following desired trajectory:

$$\vec{q}_d = (5 \cdot sin\frac{t}{4},\ 5 \cdot sin\frac{t}{2},\ 0, \frac{\pi}{2},\ 0)^T. \quad (9)$$

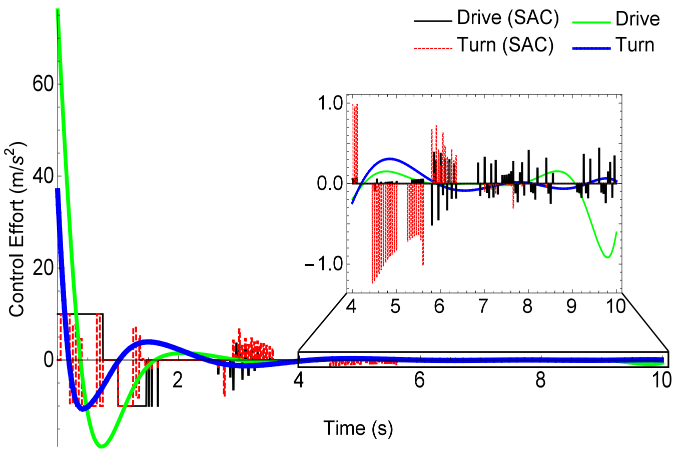
The parameters used in the simulations are t_s = 0.05 s (20 Hz rate), T = 1 s, Q = **Diag**[100, 100, 1/1000, 0, 1/100], P_1 = **Diag**[0, 0, 0, 0, 0], R = **Diag**$[10^{-8}, 10^{-8}]$, $\dot{x}_w$ = -1.0 m/s, $\dot{y}_w$ = 0. The system starts from initial conditions $\vec{q}_0 = (0, 0, 0, 0, 0)^T$ and the control remains constrained within the following limits: ($u_D \in [-10, +10]$ m/s^2, $u_T \in [-30, +30]$ rad/s^2).

The results of the simulation are presented in Fig. 1. The kinematic car stays within a couple centimeters of the desired trajectory, both in the absence and presence of fluid drift. In this benchmark example, fluid drift does not have a significant effect on tracking performance, despite the short time horizon used. Even though fluid dynamics have been simplified with constant velocity drift, SAC stays largely unaffected in its performance and is able to turn and drive the car to follow this changing track. The algorithm is also compared to a projection-based trajectory optimization method on the task of reaching a nearby location. As Fig. 3 shows, SAC is able to decrease the objective (distance to the target) with less control and better final error, despite the saturation limits applied on control.

The ability of SAC to perform well with fluid drift inspired further work. Specifically, the authors tested whether SAC could apply optimal corrective control, without knowledge of the existing drift effects. In simulation, SAC is tested in the context of two dynamics, the non-fluid $\vec{f}_{nf}$ and the fluid $\vec{f}_{real}$. Dynamics $\vec{f}_{nf}$ are the dynamics of the car $\vec{f}$ in the absence of drift ($\dot{x} = \dot{y} = 0$), whereas dynamics $\vec{f}_{real}$ are the actual dynamics of the environment ($\dot{x} = -1.0\ m/s, \dot{y} = 0$). In this way, SAC performs its computations, and applies control using $\vec{f}_{nf}$. The actual progression of the system states, however, occurs using $\vec{f}_{real}$ and the control that is



(a) Tracking error performance of SAC and Trajectory Optimization on the task of reaching a stationary nearby target.



(b) Controls produced by SAC and Trajectory Optimization on the task of reaching a stationary nearby target.

Fig. 3. SAC and the projection-based trajectory optimization scheme are tested on reaching a nearby target at (x,y) = (5 m, 5 m), starting from $\vec{q}_0 = \vec{0}$ and using the kinematic car dynamics in the presence of a -1.0 m/s $\hat{x}$ drift. As shown in the left figure, SAC satisfies saturation limits and exhibits better station keeping performance by remaining closer to the target (zoomed image). SAC outperforms Trajectory Optimization also in terms of the control efforts, which are measured by integrating control actions over application time: $\int_{t_0}^{t_f} u(t)dt$. Throughout the ten seconds of simulation, SAC uses 26.71 m/s$^2\cdot$ s and Trajectory Optimization uses 41.35 m/s$^2\cdot$ s. After the first two seconds, the integrated errors are 26.3 cm and 65.6 cm and the controls used are 3.63 m/s$^2\cdot$ s and 4.41 m/s$^2\cdot$ s for SAC and Trajectory Optimization, respectively. Control saturations used in SAC keep controls below 10 m/s^2, better resembling experimental actuation constraints.

computed for $\vec{f}_{nf}$. The results presented in Fig. 2 show that SAC performance does not significantly deteriorate "underwater" without knowledge of the true fluid dynamics. The controller tracks the desired y-state accurately, while the x-state is only slightly off throughout the simulation. As the figure shows, the car oscillates back and forth over the reference target in the non-fluid test (blue curves) due to control overshoot. This effect arises because of the very short time horizon that does not allow the controller to know the desired trajectory well in advance. A second simulation with twice the time horizon duration (T = 2 s) showed reductions in the oscillations around these reference x-state. Yet the results of Fig. 2 describe a more realistic scenario, in which the controller has limited information about the controller's future motion and becomes more reactive than predictive.

B. *Dynamic Fish-Robot*

Through the benchmark example of the kinematic car, SAC is shown to be capable of trajectory - tracking in the presence of fluid drift, whether or not it has any knowledge of the actual fluid dynamics. Its computational speed and robustness (evident in the results presented) are reasons to believe SAC can appropriately control underwater vehicles in real-time. Fish in general, and the weakly electric fish more specifically, are promising candidate system for underwater dynamical models in cluttered, dirty environments [22], [25], [30]–[32]. For this reason, this is the second system on which SAC is tested.

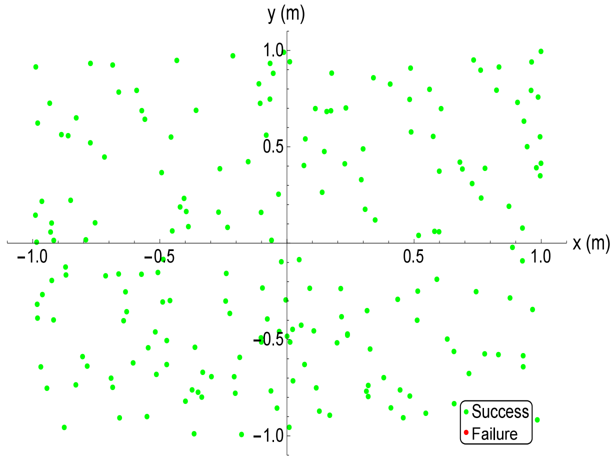
The weakly electric fish black ghost knifefish *Apteronotus albifrons* lives in dirty, turbulent waters and provides science with an example of optimal underwater motion [22], [31]. Its ability to navigate has already been studied and shown to be optimal in prey-tracking scenarios [22], providing a potentially useful design model for future AUVs.

Its ability to navigate in turbid water sensing by way of a self-generated electric field drew the attention of the scientific world which saw a model of how to extend underwater expeditions to low-visibility areas. As described in [33] and [34], the weakly electric fish use active electrosense to map its surroundings and catch its prey. It continually generates an oscillating electric field and, through thousands of electric sensors placed throughout its body, can sense the presence of objects around it. Objects that do not share the same conductivity as water and perturb the self-generated electric field of the fish cause voltage changes at the sensors that are processed by the animal.

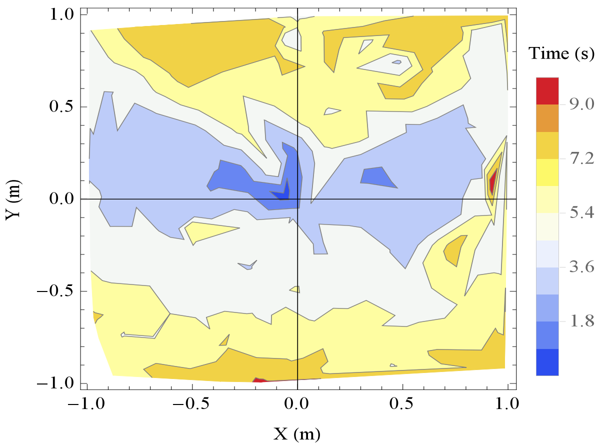
The dynamics of the underactuated electric fish are derived from Euler-Lagrange (EL) equations equivalent to Kirchhoff's equations [22]. With state $\vec{q} = (x, y, \theta, \dot{x}, \dot{y}, \dot{\theta})^T$ expressed in the world frame, and control input $\vec{u} = (u_D, u_T)^T$ – drive and turn, respectively – in the body frame, the model incorporates damping and control input in the body - frame and fluid drift in the world frame. Specifically, the fish is modeled as a rigid body with a generalized inertia matrix $I = \textbf{Diag}[m_1, m_2, m_3, j_1, j_2, j_3]$ and body-frame velocity V_b given in terms of $G_{wb}(x, y, \theta)$ (the transformation from the world to the body - frame) and $R(\theta)$ (the rotation matrix):[2]

$$V_b = (G_{wb}^{-1} \cdot \dot{G}_{wb})^{\vee} = \begin{pmatrix} R^T \dot{p} \\ \omega \end{pmatrix}, \tag{10a}$$

[2]The $^\vee$ operation on a 4x4 matrix is defined as $G^\vee = (G_{14}, G_{24}, G_{34}, G_{32}, G_{13}, G_{21})$, with G_{ij} defining the element of G in the i-th row, j-th column; $\dot{p} = (\dot{x}, \dot{y}, \dot{z})^T$.



(a) A success/failure map of nearby targets for the dynamic fish-robot. All targets are successfully reached within ten seconds of simulation time.



(b) Target locations are color-coded based on the simulation time it takes the dynamic robot-fish to reach them. Targets lying ahead (+$\hat{x}$) or behind (-$\hat{x}$) the system are reached the fastest. The asymmetry around the y-axis origin is due to the +0.1 m/s $\hat{y}$ drift.

Fig. 4. A map of target locations posed to the dynamic robot-fish system in the presence of +0.1 m/s $\hat{y}$ drift. Two hundred targets are randomly generated from a sample space of (x, y) = (1 m, 1 m) using Monte Carlo sampling. Success is defined by whether the robot-fish, always starting from an initial state of $\vec{q}_0 = \vec{0}$, is at the end of the simulaton (10 seconds) within 2 cm of the target, equal to half the longest dimension of the electric fish [22]. The only concern of the task is to approach the nearby targets and so zero weight is applied on the orientation θ of the system.

$$R(\theta) = \begin{pmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{10b}$$

$$G_{wb}(x, y, \theta) = \begin{pmatrix} \ddots & & & x \\ & R & & y \\ & & \ddots & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{10c}$$

The generalized inertia matrix I uses information about the physical limitations of motion and restorative forces in underwater environment. Parameters m_1, m_2, m_3 describe the added mass matrix (due to the volume of fluid accelerated by translations of the fish) and parameters j_1, j_2, j_3 refer to the added moment of inertia matrix (due to the volume of fluid accelerated by rotations). The values used are $(m_1, m_2, m_3) = (6.04, 17.31, 8.39)$ g, $(j_1, j_2, j_3) = (1.57, 27.78, 54.11)$ g cm^2 found in [22].

Assuming the body lies on a 2D plane, its potential energy (PE) is constant and its KE = $\frac{1}{2}V_b^T I V_b$ (invariant across transformations). The Lagrangian of the system is L $\triangleq$ KE - PE, the EL differential equations are:

$$\frac{\partial L}{\partial q_i} - \frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} = F_{ext}, \quad \text{for } i = 1, 2, 3, \tag{11a}$$

$$F_{ext} = R(\theta) \cdot \begin{pmatrix} u_D + D_1 \\ D_2 \\ u_T + D_3 \end{pmatrix}, \tag{11b}$$

where D_1, D_2, D_3 are the damping forces. The rotation matrix in equation (11) is used to convert the forces of damping and control in the body-frame. Solving equation (11) yields $\ddot{x}, \ddot{y}, \ddot{\theta}$. Water drift $\dot{x}_w, \dot{y}_w$ is added to the measured state - velocities $\dot{x}, \dot{y}$.

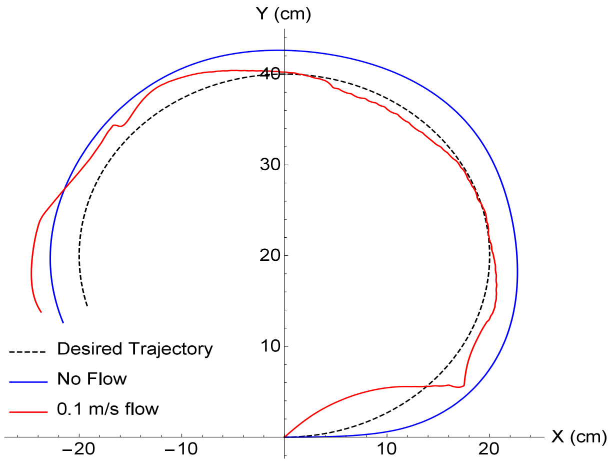
First, SAC is tested on reaching nearby targets in the presence of +0.1 m/s $\hat{y}$ drift. The results, presented in Fig. 4, show that SAC reaches all randomized targets. These results highlight the maneuverability of the dynamic robot-fish dynamics, which SAC can efficiently handle. The parameters used in the simulation are t_s = 0.05 s, T = 1 s, Q = **Diag**[1000, 1000, 0, 0.01, 0.01, 0.01], P_1 = **Diag**[10000, 10000, 0, 1, 1, 1], R = **Diag**[$10^{-3}, 10^{-6}$]. SAC is further applied on tracking the following desired trajectory for 20 seconds, with and without flow:

$$\vec{q}_d = (0.2 \cdot cos[\frac{t}{4} - \frac{\pi}{2}], 0.2 + 0.2 \cdot sin[\frac{t}{4} - \frac{\pi}{2}], 0, 0, 0, 0)^T. \tag{12}$$

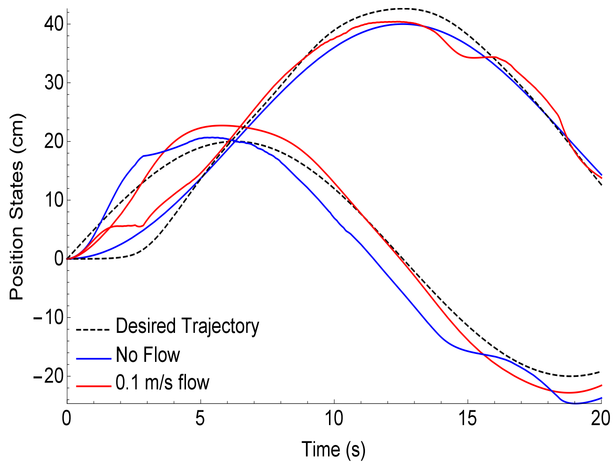
Starting from $\vec{q}_0 = (0,\ 0,\ 0,\ 0,\ 0,\ 0)^T$, SAC uses t_s = 0.05 s, T = 1 s, Q = **Diag**[1000, 1000, 0, 0.01, 0.01, 0.01], P_1 = **Diag**[10000, 10000, 0, 1, 1, 1], R = **Diag**[$10^{-3}, 10^{-6}$], and saturation constraints of 1 N on both control inputs. Simulation results are presented in Fig. 5. Not only does SAC successfully track the desired trajectory using the electric fish dynamics, but it also does reasonably well in the presence of fluid drift. The resulting underwater motion (with drift) is less smooth compared to the motion without drift, but is not worse in terms of the objective which is tracking the reference signal. Last, SAC is tested on the following trajectory - tracking task for a set of different fluid flow direction and intensities:

$$\vec{q}_d = (0.2 \cdot sin\frac{t}{4},\ 0.2 \cdot sin\frac{t}{2},\ 0,\ 0,\ 0,\ 0)^T. \tag{13}$$

Results are presented in Fig. 6 and show that SAC's ability to track the desired trajectory is robust to different fluid intensities. While these simulation results cannot guarantee SAC's success tracking trajectories underwater in general,

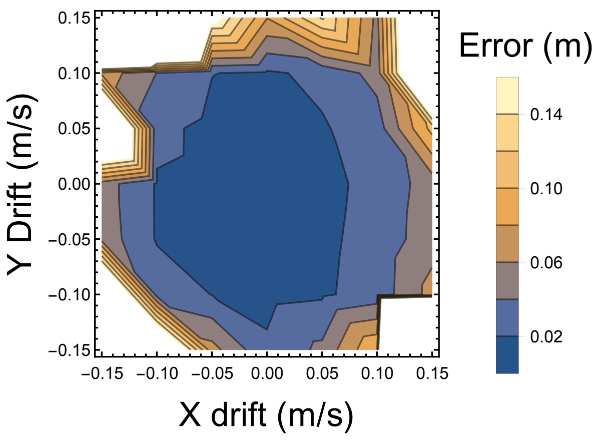


(a) A parametric plot of SAC-produced trajectories

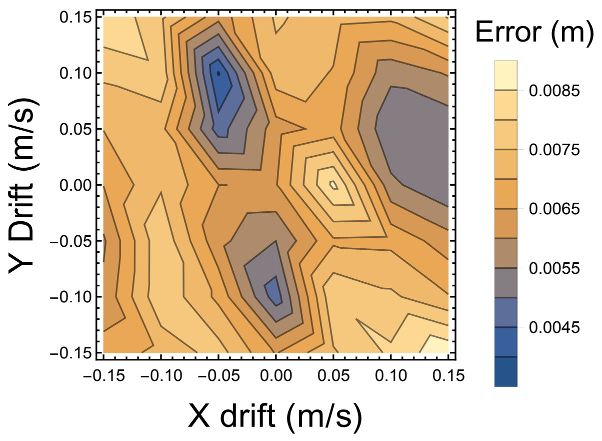


(b) Tracking error in the x-y states

Fig. 5. SAC is applied on the dynamic fish-robot model to track the desired trajectory at a control sequencing frequency of 20 Hz. The performance of the controller is tested against no drift and drift of +0.1 m/s $\hat{y}$. The computed trajectories are plotted against the reference signal. Although the time horizon used for the simulations is extremely short (T = 1 s), the performance of SAC remains largely unaffected by the presence of flow.



(a) Tracking error throughout simulation. The white region corresponds to large errors, not visible in the bar legend.



(b) Steady-state tracking error, as measured after the first five seconds of simulation.

Fig. 6. A contour plot on the effect of fluid drift intensity on trajectory - tracking performance for the dynamic system. The maps plot performance error as a function of fluid drift intensity (in both the x- and y- direction) and are generated from interpolating data for drift $\in$ (-0.15, 0.15) m/s sampled in steps of 0.05 m/s. Performance error is calculated as the integrated distance (in m) away from the desired trajectory throughout the simulation period (20 seconds). The majority of the error occurs in the first five seconds, until the controller catches up with the target. The right figure shows the error between 5-20 seconds. The desired trajectory is provided in 13 and has a total arc length of 1.52 m over the simulation period.

they suggest that SAC can be used for underwater tracking using the model of the knifefish and provides a promising basis for further exploration.

IV. CONCLUSIONS AND FUTURE WORK

This paper presents the ability of SAC to perform trajectory-tracking tasks in the presence of fluid drift. Because underwater environments are difficult to model accurately, several optimization schemes involve approximations in their models. Due to being offline or having a high computational cost, such schemes do not seem good candidates for real-time problems. On the other hand, the application results of this paper show SAC to be a control-efficient solution that is robust to fluid drift intensities, without sacrificing tracking performance. Hence, SAC seems a reasonable alternative for underwater trajectory-tracking.

The application tests in this paper provide only a limited number of investigated cases. Further, both the model of the kinematic car and the dynamic fish-robot avoid the complexity of three-dimensional navigation. Such concerns are the focus of future work.

REFERENCES

[1] D. R. Blidberg, "The development of autonomous underwater vehicles (AUV); a brief summary," in *IEEE ICRA*, vol. 4, 2001.

[2] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad hoc networks*, vol. 3, no. 3, pp. 257–279, 2005.

[3] J. J. Leonard, A. A. Bennett, C. M. Smith, H. Jacob, and S. Feder, "Autonomous underwater vehicle navigation," in *MIT Marine Robotics Laboratory Technical Memorandum*. Citeseer, 1998.

[4] A. Bowen, D. Yoerger, C. Taylor, R. McCabe, J. Howland, D. Gomez-Ibanez, J. Kinsey, M. Heintz, G. McDonald, D. Peters, B. Fletcher, C. Young, J. Buescher, L. Whitcomb, S. Martin, S. Webster, and M. Jakuba, "The Nereus hybrid underwater robotic vehicle for global ocean science operations to 11,000m depth," in *OCEANS 2008*, 2008, pp. 1–10.

[5] W. Zhang, S.-X. Guo, and K. Asaka, "A new type of hybrid fish-like microrobot," *International Journal of Automation and Computing*, vol. 3, no. 4, pp. 358–365, 2006.

[6] A. Murphy, M. Landamore, and R. Birmingham, "The role of autonomous underwater vehicles for marine search and rescue operations," *Underwater Technology*, vol. 27, no. 4, pp. 195–205, 2008.

[7] A. Birk, M. Pfingsthorn, and H. Bulow, "Advances in underwater mapping and their application potential for safety, security, and rescue robotics (SSRR)," in *Safety, Security, and Rescue Robotics (SSRR), 2012 IEEE International Symposium on*, 2012, pp. 1–3.

[8] P. Hagen, N. Storkersen, B.-E. Marthinsen, G. Sten, and K. Vestgard, "Military operations with hugin AUVs: lessons learned and the way ahead," in *Oceans 2005 - Europe*, vol. 2, 2005, pp. 810–813 Vol. 2.

[9] I. Kaminer, A. Pascoal, E. Hallberg, and C. Silvestre, "Trajectory tracking for autonomous vehicles: An integrated approach to guidance and control," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 1, pp. 29–38, 1998.

[10] A. Brezoescu, P. Castillo, R. Lozano, and F. Compiègne, "Straight-line path following in windy conditions," in *Conference on Unmanned Aerial Vehicle in Geomatics, Zurich, Switzerland*, 2011.

[11] P. Encarnaçao and A. Pascoal, "Combined trajectory tracking and path following: an application to the coordinated control of autonomous marine craft," in *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, vol. 1. IEEE, 2001, pp. 964–969.

[12] A. P. Aguiar and J. P. Hespanha, "Position tracking of underactuated vehicles," in *American Control Conference, 2003. Proceedings of the 2003*, vol. 3. IEEE, 2003, pp. 1988–1993.

[13] P. Encarnacao and A. Pascoal, "3D path following for autonomous underwater vehicle," in *Proc. 39 th IEEE Conference on Decision and Control*, 2000.

[14] F. Repoulias and E. Papadopoulos, "Trajectory planning and tracking control of underactuated AUVs," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 1610–1615.

[15] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad hoc networks*, vol. 3, no. 3, pp. 257–279, 2005.

[16] D. Smallwood, L. L. Whitcomb *et al.*, "Model-based dynamic positioning of underwater robotic vehicles: Theory and experiment," *IEEE Journal of Oceanic Engineering*, vol. 29, no. 1, pp. 169–186, 2004.

[17] A. Ansari and T. D. Murphey, "Sequential action control: Closed-form optimal control for nonlinear systems," *IEEE Transactions on Robotics*, Submitted, 2015.

[18] A. Ansari and T. D. Murphey, "Control-on-request: Short-burst assistive control for long time horizon improvement," in *American Control Conference*, 2015.

[19] A. Ansari, K. Flaßkamp, and T. D. Murphey, "Sequential action control for tracking of free invariant manifolds," in *Conference on Analysis and Design of Hybrid Systems*, In Press.

[20] A. D. Wilson, J. Schultz, A. Ansari, and T. D. Murphey, "Real-time trajectory synthesis for information maximization using sequential action control and least-squares estimation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, In Press.

[21] J. Hauser, "A projection operator approach to the optimization of trajectory functionals," in *IFAC world congress*, vol. 15, 2002, p. 310.

[22] C. M. Postlethwaite, T. M. Psemeneki, J. Selimkhanov, M. Silber, and M. A. MacIver, "Optimal movement in the prey strikes of weakly electric fish: a case study of the interplay of body plan and movement capability," *Journal of The Royal Society Interface*, vol. 6, no. 34, pp. 417–433, 2009.

[23] Y. Wardi, M. Egerstedt, and P. Twu, "A controlled-precision algorithm for mode-switching optimization," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. IEEE, 2012, pp. 713–718.

[24] T. M. Caldwell and T. D. Murphey, "Projection-based optimal mode scheduling," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*. IEEE, 2013, pp. 5307–5314.

[25] I. D. Neveln, Y. Bai, J. B. Snyder, J. R. Solberg, O. M. Curet, K. M. Lynch, and M. A. MacIver, "Biomimetic and bio-inspired robotics in electric fish research," *Journal of Experimental Biology*, vol. 216, no. Pt 13, pp. 2501–2514, Jul 2013.

[26] O. M. Curet, N. A. Patankar, G. V. Lauder, and M. A. MacIver, "Mechanical properties of a bio-inspired robotic knifefish with an undulatory propulsor," *Bioinspir. Biomim.*, vol. 6, no. 2, 2011.

[27] R. Pepy, A. Lambert, and H. Mounier, "Path planning using a dynamic vehicle model," in *Information and Communication Technologies, 2006. ICTTA'06. 2nd*, vol. 1. IEEE, 2006, pp. 781–786.

[28] A. De Luca, G. Oriolo, and C. Samson, "Feedback control of a nonholonomic car-like robot," in *Robot motion planning and control*. Springer, 1998, pp. 171–253.

[29] B. Kiss and E. Szádeczky-Kardoss, "On-line time-scaling control of a kinematic car with one input," in *Control & Automation, 2007. MED'07. Mediterranean Conference on*. IEEE, 2007, pp. 1–6.

[30] D. Babineau, J. E. Lewis, and A. Longtin, "Spatial acuity and prey detection in weakly electric fish," *PLoS Comput Biol*, vol. 3, no. 3, p. e38, 2007.

[31] M. MacIver, E. Fontaine, and J. W. Burdick, "Designing future underwater vehicles: Principles and mechanisms of the weakly electric fish," *IEEE Journal of Oceanic Engineering*, vol. 29, no. 3, pp. 651–659, 2004.

[32] M. Porez, V. Lebastard, A. J. Ijspeert, and F. Boyer, "Multi-physics model of an electric fish-like robot: Numerical aspects and application to obstacle avoidance," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 1901–1906.

[33] J. R. Solberg, K. M. Lynch, and M. A. MacIver, "Active electrolocation for underwater target localization," *International Journal of Robotics Research*, vol. 27, no. 5, pp. 529–548, 2008.

[34] J. B. Snyder, M. E. Nelson, J. W. Burdick, and M. A. MacIver, "Omnidirectional sensory and motor volumes in an electric fish," *PLoS Biology*, vol. 5, no. 11, pp. 2671–2683, 2007.