Trajectory Tracking of Underactuated Sea Vessels With Uncertain Dynamics: An Integral Reinforcement Learning Approach

Mohammed Abouheaf, Wail Gueaieb, Md Suruz Miah, and Davide Spinello

***Abstract*—Underactuated systems like sea vessels have degrees of motion that are insufficiently matched by a set of independent actuation forces. In addition, the underlying trajectory-tracking control problems grow in complexity in order to decide the optimal rudder and thrust control signals. This enforces several difficult-to-solve constraints that are associated with the error dynamical equations using classical optimal tracking and adaptive control approaches. An online machine learning mechanism based on integral reinforcement learning is proposed to find a solution for a class of nonlinear tracking problems with partial prior knowledge of the system dynamics. The actuation forces are decided using innovative forms of temporal difference equations relevant to the vessel's surge and angular velocities. The solution is implemented using an online value iteration process which is realized by employing means of the adaptive critics and gradient descent approaches. The adaptive learning mechanism exhibited well-functioning and interactive features in react to different desired reference-tracking scenarios.**

***Index Terms*—Approximate Dynamic Programming, Integral Reinforcement Learning, Adaptive Critics, Underactuated Vessels**

I. Introduction

The trajectory-tracking problem is a sub-class of the optimal tracking problems where error dynamical equations are derived to solve this category of problems. A considerable number of solution methods are either offline or rely on complicated adaptive structures. The complexity escalates when it is desired to control underactuated high order systems [1]–[3]. This work proposes a data-driven machine learning approach that fuses measurements into control strategies for underactuated sea vessels. This approach makes use of the tracking error signals of the orientation of the vessel to compensate for the thrust and rudder forces without actually solving any error dynamical equations or solving complex adaptive control laws. This work combines ideas from Reinforcement Leaning (RL) and optimal control theory to propose reference-tracking control mechanism for a class of underactuated mechanical systems.

Numerous control approaches, such as backstepping, sliding mode, nonlinear adaptive controllers, among others, have been proposed in the literature to control the trajectories of unmanned vessels [1]–[3]. Among them, approximate dynamic programming approaches are used to tackle the action-state curses of dimensionality associated with the dynamic programming methods. They are employed to solve the optimal control problems using different forms of Bellman equations to optimize the underlying cost functions [4], [5]. The solution of the Hamilton-Jacobi-Bellman (HJB) of a dynamical system, not only provides an optimal control solution but also forms a temporal difference optimization setup that is employed by machine learning environments [6]–[8]. The optimal solution is found by applying Bellman optimality conditions to derive the optimal strategy. Bellman equations may be arranged into various temporal difference forms that can be realized using reinforcement learning approaches [9].

The RL mechanisms allow dynamical systems to select their control strategies in a dynamic learning environment to transit to better states that maximize the sum of cumulative rewards [10]–[12]. Integral Reinforcement Learning (IRL) approaches are employed to solve the differential graphical games and adaptive control problems for linear and nonlinear systems in [13]–[16]. Policy and value iteration methods provide different realizations for the IRL solutions. These are used to approximate the best strategies-to-follow, by minimizing a performance index of the states or dynamic positions [8]. Policy iteration method necessitates an initial admissible policy and may employ least-square approximations to arrive at certain solutions [17]. Value iteration provides a non-decreasing sequence of solving value functions bounded above by the optimal solution [18]. While in policy iteration, the solving value functions are non-increasing and bounded below by the value at the optimal strategy. The adaptive critics neural network tools are used to approximate the strategy and the associated value using actor and critic structures [19]–[22].

The main contributions can be explained as follows; First, a machine learning process is presented to design a reference-tracking mechanism for an underactuated sea vessel. This approach does not use any error dynamical equations or need to employ complicated adaptive control laws. Second, it makes benefit of the online measurements, or simply the tracking errors, to decide the control strategies. Finally, a value iteration process is developed to update the rudder and thrust control strategies using means of adaptive critics.

The remaining of the paper is structured as follows. Section II explains the dynamics of an underactuated sea vessel

Mohammed Abouheaf and Wail Gueaieb are with School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Ontario, Canada. e-mail: {mabouhea,wgueaieb}@uOttawa.ca

Md Suruz Miah is with Department of Electrical and Computer Engineering, Bradley University, Peoria, Illinois, USA. e-mail: smiah@bradley.edu

Davide Spinello is with Department of Mechanical Engineering, University of Ottawa, Ottawa, Ontario, Canada. e-mail: dspinell@uOttawa.ca

This is the postscript version of the published paper. (doi: 10.1109/SMC42975.2020.9283399)

followed by laying out the mathematical framework of the trajectory-tracking problem. Sections III and IV present the integral reinforcement learning solution and the associated adaptive critics implementation. The analysis of the simulation results and final concluding remarks are introduced in Sections V and VI, respectively.

II. Underactuated Sea Vessel: Problem Setup

The dynamics of an underactuated ship are defined by [1]

$$\dot{x} = v\,\cos(\psi) - u\,\sin(\psi) \tag{1a}$$
$$\dot{y} = v\,\sin(\psi) + u\,\cos(\psi) \tag{1b}$$
$$\dot{\psi} = r \tag{1c}$$
$$\dot{v} = \frac{m_{22}}{m_{11}}\,u\,r - \frac{d_{11}}{m_{11}}\,v + \frac{1}{m_{11}}\,c_F \tag{1d}$$
$$\dot{u} = -\frac{m_{11}}{m_{22}}\,v\,r - \frac{d_{22}}{m_{22}}\,u \tag{1e}$$
$$\dot{r} = \frac{m_{11}-m_{22}}{m_{33}}uv - \frac{d_{33}}{m_{33}}\,r + \frac{1}{m_{33}}\,c_R, \tag{1f}$$

where (x, y) and ψ represent the vessel's position and orientation; v, r, and u denote the surge velocity, yaw angular velocity, and sway velocity, respectively; m_{11}, m_{22}, and m_{33}, represent inertia masses; d_{11}, d_{22}, and d_{33}, represent drag coefficients; c_F and c_R are the thrust and rudder control forces, respectively. Fig. 1 depicts the kinematic parameters of an underactuated vessel with respect to the world coordinate system along with a desired (reference) Cartesian trajectory $(x^{\text{ref}}(t), y^{\text{ref}}(t)),\ t \geq 0$, generated by an independent command generator of the reference mission.

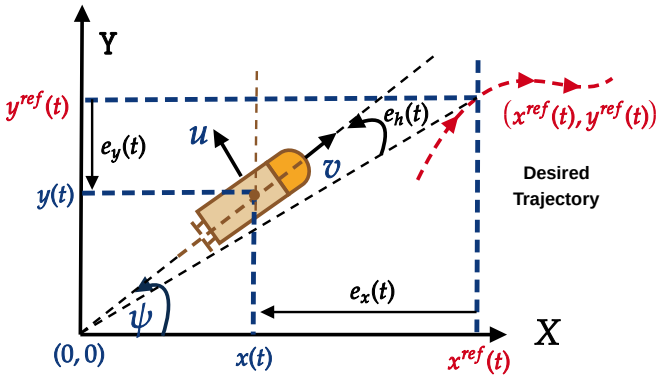

Fig. 1. Kinematics of an underactuated sea vessel in following a reference (desired) trajectory $(x^{\text{ref}}(t), y^{\text{ref}}(t)),\ t \geq 0$.

The objective is to steer the vessel towards the desired reference trajectory so that the tracking errors $e_x(t) = x(t) - x^{\text{ref}}(t)$ and $e_y(t) = y(t) - y^{\text{ref}}(t)$ converge to zero as $t \to \infty$. This task can be formulated as an optimal control problem aiming at calculating the optimal thrust and rudder control forces, c_F^* and c_R^*, respectively. Adopting such formulation would lead to nonlinear control laws. To simplify the problem, and so the associated machine learning technique, we will now introduce the concepts of tangential and bearing corrections.

A. Linear Velocity Optimization: Tangential Corrections

Define the velocity scalar adjustments along the x and y directions as $c_x(t) = \boldsymbol{K}_x(t)\boldsymbol{E}_x(t)$ and $c_y(t) = \boldsymbol{K}_y(t)\boldsymbol{E}_y(t)$, respectively, with $\boldsymbol{E}_x(t) = [\,e_x(t)\ \ e_x(t-\Delta)\ \ e_x(t-2\Delta)\,]^T \in \mathbb{R}^{3\times 1}$ and $\boldsymbol{E}_y(t) = [\,e_y(t)\ \ e_y(t-\Delta)\ \ e_y(t-2\Delta)\,]^T \in \mathbb{R}^{3\times 1}$, $\boldsymbol{K}_x(t) \in \mathbb{R}^{1\times 3}$ and $\boldsymbol{K}_y(t) \in \mathbb{R}^{1\times 3}$ are sub-control feedback gains to be later determined using a machine learning process, and Δ is a sampling time. The tangential adjustment in the surge velocity v can be written as $c_v(t) = \sqrt{c_x^2(t) + c_y^2(t)}/\Delta - v(t-\Delta)$. This form discounts the old velocity instances and adds a velocity adjustment. The objective of such formulation is to let $c_v(t) \to 0$ (or simply $\dot{v} \to 0$) as $t \to \infty$ so that the surge velocity dynamics have the form $\dot{v} = c_v$. Equating to (1d), we get the thrust force c_F as

$$c_F = m_{11}\,c_v - m_{22}\,u\,r + d_{11}\,v. \tag{2}$$

As such, we can design a controller to minimize objective cost functions U_x and U_y in the x and y directions, respectively, where $U_i\left(\boldsymbol{E}_i(t), c_i(t)\right) = \frac{1}{2}\left(\boldsymbol{E}_i^T(t)\boldsymbol{Q}_i\boldsymbol{E}_i(t) + R_i[c_i(t)]^2\right)$, $i \in \{x, y\}$, for some design parameters $\boldsymbol{Q}_i > \boldsymbol{0} \in \mathbb{R}^{3\times 3}$ and $R_i > 0 \in \mathbb{R}$. When referring to a matrix, the notation "$> \boldsymbol{0}$" refers to a positive definite matrix. The performance index associated to the long-run cost is defined by

$$J_i = \int_0^\infty U_i\left(\boldsymbol{E}_i(\zeta), c_i(\zeta)\right)\,d\zeta. \tag{3}$$

B. Angular Velocity Optimization: Bearing Correction

The tracking error corresponding to the vessel's orientation is $e_h(t) = \underline{/x(t) + jy(t)} - \underline{/x^{\text{ref}}(t) + jy^{\text{ref}}(t)}$, where $\underline{/\cdot}$ denotes the phase of the complex quantity $(\cdot)$. Define the control law $c_h(t) = \tilde{c}_h(t) + \underline{/c_x + jc_y}$ with bearing adjustment $\tilde{c}_h(t) = \boldsymbol{K}_h(t)\boldsymbol{E}_h(t)$, where $\boldsymbol{E}_h(t) = [\,e_h(t)\ \ e_h(t-\Delta)\ \ e_h(t-2\Delta)\,]^T \in \mathbb{R}^{3\times 1}$ and the gain $\boldsymbol{K}_h(t) \in \mathbb{R}^{1\times 3}$ is to be determined later. This step takes into consideration the orientation error and the bearing corrections as per the surge velocity tangential adjustments c_x and c_y. It enables the adoption of a heading control law $\psi(t) = \underline{/x(t) + jy(t)} + c_h(t) \in \left[\frac{-\pi}{2}, \frac{\pi}{2}\right]$ rad. Using (1c) and (1f) yields

$$\dot{r} = \ddot{\psi} = \frac{m_{11}-m_{22}}{m_{33}}\,u\,v - \frac{d_{33}}{m_{33}}\,r + \frac{1}{m_{33}}\,c_R. \tag{4}$$

Applying Euler's approximation, the angular acceleration $\ddot{\psi}$ is estimated by $\ddot{\psi} \approx (\psi(t) - 2\psi(t-\Delta) + \psi(t-2\Delta))/\Delta^2$. Equating to (4), $(\psi(t) - 2\psi(t-\Delta) + \psi(t-2\Delta))/\Delta^2 = \frac{m_{11}-m_{22}}{m_{33}}\,u\,v - \frac{d_{33}}{m_{33}}\,r + \frac{1}{m_{33}}\,c_R$. Therefore, the rudder control force can be written as

$$\begin{aligned} c_R &= \frac{\psi(t) - 2\psi(t-\Delta) + \psi(t-2\Delta)}{\Delta^2/m_{33}} \\ &- (m_{11}-m_{22})\,u\,v + d_{33}\,r. \end{aligned} \tag{5}$$

The final optimization goal is then to minimize the following objective function associated to the bearing adjustment:

$$U_h\left(\boldsymbol{E}_h(t), c_h(t)\right) = \frac{1}{2}\left(\boldsymbol{E}_h^T(t)\,\boldsymbol{Q}_h\,\boldsymbol{E}_h(t) + R_h\,[c_h(t)]^2\right),$$

where $\boldsymbol{Q}_h > \mathbf{0} \in \mathbb{R}^{3\times 3}$ and $R_h > 0 \in \mathbb{R}$. The associated performance index is defined by

$$J_h = \int_0^{\infty} U_h\left(\boldsymbol{E}_h(\zeta), c_h(\zeta)\right)\, d\zeta. \tag{6}$$

With the introduction of the tangential and bearing corrections, the trajectory-tracking optimization problem is reduced to minimizing the indices J_x, J_y, and J_h. It is important to point out that such a formulation avoids the explicit use of the full sixth-order nonlinear dynamics (1). The control signals c_v and c_h will be later calculated using a reinforcement learning process. These sub-control signals will be calculated in an online fashion based on location feedback measurements, such as GPS data, for instance. They are also employed implicitly into the computation of the thrust and rudder actuation forces c_F and c_R. It is also worth mentioning that the term $\underline{/c_x + jc_y}$ may be omitted from the calculation of the control law $c_h(t)$. However, including it can speed up the convergence process.

III. Optimal Control Solution

We now introduce the optimal control solution using integral forms of Bellman equations within a value iteration framework. The conditions of optimality are found using Bellman optimization principles [7]. The optimization problem finds the optimal control strategies by minimizing the performance indices J_x, J_y, and J_h, with respect to the control signals c_x, c_y, and c_h, respectively.

A. Integral Bellman Equations

The solving value structure for each segment of the optimization problem (i.e., searching for the optimal strategies c_x, c_y, and c_h) is motivated by the linear quadratic forms of the underlying cost functions. Hence, the solution for each segment is quadratic in the recent error records and sub-control signals such that

$$J_i \equiv V_i(\boldsymbol{E}_i, c_i) = \frac{1}{2}[\boldsymbol{E}_i^T \quad c_i^T]\ \boldsymbol{H}_i \begin{bmatrix} \boldsymbol{E}_i \\ c_i \end{bmatrix}, \tag{7}$$

where $i \in \{x, y, h\}$, matrix $\boldsymbol{H}_i$, at each direction x, y, and h, has a symmetric structure $\boldsymbol{H}_i = \begin{bmatrix} \boldsymbol{H}_{\boldsymbol{E}_i\boldsymbol{E}_i} & \boldsymbol{H}_{\boldsymbol{E}_i c_i} \\ \boldsymbol{H}_{c_i\boldsymbol{E}_i} & \boldsymbol{H}_{c_i c_i} \end{bmatrix} > \mathbf{0}$.

Equating the performance indices (7) to (3) and (6), yields the following integral Bellman equations

$$\begin{aligned} V_i(\boldsymbol{E}_i(t)\,,\, c_i(t)) = &\int_t^{t+\Delta} U_i\left(\boldsymbol{E}_i(\zeta)\,,\, c_i(\zeta)\right)\, d\zeta \\ &+ V_i(\boldsymbol{E}_i(t+\Delta)\,,\, c_i(t+\Delta)), i \in \{x, y, h\}. \end{aligned} \tag{8}$$

Applying the optimality conditions on these integral Bellman equations leads to the following optimal control strategies [7]:

$$c_i^* = \arg\min_{c_i} \left(V_i(\boldsymbol{E}_i(t), c_i(t))\right), i \in \{x, y, h\}.$$

Hence, each optimal strategy is evaluated such that $\boldsymbol{H}_{c_i c_i} c_i(t) + \boldsymbol{H}_{c_i \boldsymbol{E}_i} \boldsymbol{E}_i(t) = 0$. The solution is a model-free optimal control strategy

$$c_i^*(t) = -\left[\boldsymbol{H}_{c_i c_i}^{-1}\, \boldsymbol{H}_{c_i \boldsymbol{E}_i}\right] \boldsymbol{E}_i\,(t). \tag{9}$$

Applying each optimal control strategy (i.e., c_x^*, c_y^*, and c_h^*) into the respective integral Bellman equation (8) yields the following integral Bellman optimality equations

$$\begin{aligned} V_i^*(\boldsymbol{E}_i(t), c_i^*(t)) = &\int_t^{t+\Delta} U_i\left(\boldsymbol{E}_i(\zeta), c_i^*(\zeta)\right)\, d\zeta + \\ &V_i^*(\boldsymbol{E}_i(t+\Delta), c_i^*(t+\Delta)), i \in \{x, y, h\}. \end{aligned} \tag{10}$$

The simultaneous solution of these integral Bellman optimality equations yields a trajectory-tracking for the sea vessel.

B. Value Iteration Solution

A value iteration algorithm is introduced to implement the online IRL solution for the Bellman optimality equations (10). This process does not require any initial admissible policies. It finds the optimal strategies using nondecreasing and upper-bounded sequence of solving value functions. The process is summarized in Algorithm 1. It is important to mention that the value iteration process employs partial knowledge about the dynamics of the sea vessel. It is proven to generally converge by generating a sequence of non-decreasing value functions that are upper-bounded by the optimal value [23].

Algorithm 1 Online Value Iteration Mechanism

Input:

Initial solving matrices $\mathbf{H}_i^1$, $i \in \{x, y, h\}$

Initial tracking error vectors $\mathbf{E}_i^1$ and strategies c_i^1

Error threshold γ

Convergence time window L

Maximum number of learning iterations N_T

Output:

Optimal solving matrices $\mathbf{H}_i^*, i \in \{x, y, h\}$

for $\ell = 1$ to N_T **do**
  Calculate the cost value $\int_t^{t+\Delta} U_i\left(\boldsymbol{E}_i^\ell(\zeta)\,,\, c_i^\ell(\zeta)\right)\, d\zeta$
  Find $\mathbf{E}_i^\ell(t+\Delta)$ and $c_i^\ell(t+\Delta)$ using (1) and (9)
  Determine $V_i^\ell(\boldsymbol{E}_i^\ell(t+\Delta)\,,\, c_i^\ell(t+\Delta))$ using (7)
  Evaluate $V_i^{\ell+1}(\dots)$ using

$$\begin{aligned} V_i^{\ell+1}(\dots) = &\int_t^{t+\Delta} U_i\left(\boldsymbol{E}_i^\ell(\zeta)\,,\, c_i^\ell(\zeta)\right)\, d\zeta \\ &+ V_i^\ell(\boldsymbol{E}_i(t+\Delta)\,,\, c_i^\ell(t+\Delta)) \end{aligned}$$

  ▷ The policies c_i and solving values V_i will be later implemented using an adaptive critics scheme
  if $\ell > L$ **and** $\left\|\mathbf{H}_i^{\ell+1-j} - \mathbf{H}_i^{\ell-j}\right\| \le \gamma$, $\forall j \in \{0, 1, \dots, L\}$, **then**
    $\mathbf{H}_i^* \leftarrow \mathbf{H}_i^{\ell+1}$
  end if
end for
return Optimal solving matrices $\mathbf{H}_i^*$

IV. Adaptive Critics Realization

We adopt actor-critic structures in the form of neural network approximators employed by the IRL controller. The best strategies are approximated using actor networks, while the values of these strategies are approximated by means of critic networks [9], [24], [25]. Herein, each solving value function is approximated using a critic neural network defined by

$$\hat{V}_i(\boldsymbol{E}_i, \hat{c}_i) = \frac{1}{2}[\boldsymbol{E}_i^T \;\; \hat{c}_i^T] \; \boldsymbol{W_i} \; \begin{bmatrix} \boldsymbol{E}_i \\ \hat{c}_i \end{bmatrix}, \tag{11}$$

where $\boldsymbol{W}_i$, $i \in \{x, y, h\}$ are the critic approximation weights for the solving value functions $\hat{V}_i$ (i.e., one for each of the three adaptive learning control loops). The structures of the critic networks are motivated by those of the functions $\hat{V}_i$. Similarly, the optimal strategies are approximated such that

$$\hat{c}_i = \boldsymbol{\Omega_i} \boldsymbol{E}_i, \tag{12}$$

where $\boldsymbol{\Omega}_i, i \in \{x, y, h\}$ are network approximation weights.

The adaption process of the adaptive critics weights employs a gradient descent approach. The target values which are used to tune the different critic weights are given by

$$\tilde{V}_i = \int_t^{t+\Delta} U_i \left(\boldsymbol{E}_i(\zeta), \hat{c}_i(\zeta)\right) d\zeta + \hat{V}_i(\boldsymbol{E}_i(t+\Delta), \hat{c}_i(t+\Delta)). \tag{13}$$

They express the desired value functions of the approximations $\hat{V}_i$, $i \in \{x, y, h\}$. Similarly, the desired values of the approximated optimal strategies $\hat{c}_i$, that are used to tune the actor weights, are defined by

$$\tilde{c}_i = -\left[\boldsymbol{W}_{\hat{c}_i \hat{c}_i}^{-1} \boldsymbol{W}_{\hat{c}_i \boldsymbol{E}_i}\right] \boldsymbol{E}_i, \quad i \in \{x, y, h\}. \tag{14}$$

The approximation error of each critic network is defined by $\varepsilon_i^{Critic} = \frac{1}{2}\left(\hat{V}_i(\boldsymbol{E}_i, \hat{c}_i) - \tilde{V}_i\right)^2$, while the approximation error of each actor network is given by $\varepsilon_i^{Actor} = \frac{1}{2}\left(\hat{c}_i - \tilde{c}_i\right)^2$. The neural network weights are tuned through a gradient descent approach. It yields the following update rules for the critic approximation weights $\boldsymbol{W}_i$, $i \in \{x, y, h\}$:

$$\boldsymbol{W}_i^{\ell+1} = \boldsymbol{W}_i^{\ell} - \eta_c \left(\left(\hat{V}_i(\boldsymbol{E}_i, \hat{c}_i) - \tilde{V}_i\right) \begin{bmatrix} \boldsymbol{E}_i \\ \hat{c}_i \end{bmatrix} [\boldsymbol{E}_i^T \; \hat{c}_i^T] \right)^{\ell}, \tag{15}$$

where $0 < \eta_c < 1$ is a critic-network learning rate. Similarly, the update laws for the actor weights are

$$\boldsymbol{\Omega}_i^{\ell+1} = \boldsymbol{\Omega}_i^{\ell} - \eta_a \left((\hat{c}_i - \tilde{\boldsymbol{c}}_i) \, \boldsymbol{E}_i^T \right)^{\ell}, \tag{16}$$

where $0 < \eta_a < 1$ is an actor-network learning rate. The actor and critic weights are tuned online using a value iteration process as detailed out in Algorithm 2.

V. Simulations and Results

The usefulness of the proposed IRL-based adaptive learning approach is verified using two simulation scenarios. In the first scenario, the sea vessel tracks a linear reference trajectory with a constant speed. In the second scenario, the vessel follows a dynamic reference trajectory with varying linear and angular velocities. We use the same dynamic parameters of the vessel as in [1]: $m_{11} = 19\,\text{kg}$, $d_{11} = 4\,\text{kg/s}$, $m_{22} = 35.2\,\text{kg}$, $d_{22} = 1\,\text{kg/s}$, $m_{33} = 4.2\,\text{kg·m}^2$, and $d_{33} = 10\,\text{kg·m}^2\text{/s}$. The parameters of the learning environment are chosen as $\Delta = 0.1\,\text{s}$, $\eta_c = 0.005$, $\eta_a = 0.1$. The weighting matrices are set to $\boldsymbol{Q}_i = \boldsymbol{I}_{3\times 3}$ and $R_i = 1$, $i \in \{x, y, h\}$, where $\boldsymbol{I}$ is the identity matrix.

Algorithm 2 Adaptive Critics Implementation

Input:

Initial neural network weights $\mathbf{W}_i^1$ and $\mathbf{\Omega}_i^1, i \in \{x, y, h\}$

Initial tracking error vectors $\mathbf{E}_i^1$ and strategies $\hat{c}_i^1$

Error threshold γ

Convergence time window L

Maximum number of learning iterations N_T

Output:

Tuned neural network weights $\mathbf{W}_i^*$ and $\mathbf{\Omega}_i^*, i \in \{x, y, h\}$

for $\ell = 1$ to N_T **do**
  Calculate the cost value $\int_t^{t+\Delta} U_i \left(\boldsymbol{E}_i^{\ell}(\zeta), \hat{c}_i^{\ell}(\zeta)\right) d\zeta$
  Find $\mathbf{E}_i^{\ell}(t+\Delta)$ and $\hat{c}_i^{\ell}(t+\Delta)$ using (1) and (12)
  Compute $\hat{V}_i^{\ell}(\boldsymbol{E}_i^{\ell}(t+\Delta), \hat{c}_i^{\ell}(t+\Delta))$ using (11)
  Determine $\tilde{V}_i$ and $\tilde{c}_i$ using (13) and (14), respectively
  Update the critic and actor weights using (15) and (16), respectively
  if $\ell > L$ **and** $\left\|\mathbf{W}_i^{\ell+1-j} - \mathbf{W}_i^{\ell-j}\right\| \leq \gamma, \; \forall j \in \{0, 1, \ldots, L\}$, **then**
    $\mathbf{W}_i^* \leftarrow \mathbf{W}_i^{\ell+1}$
  end if
  if $\ell > L$ **and** $\left\|\mathbf{\Omega}_i^{\ell+1-j} - \mathbf{\Omega}_i^{\ell-j}\right\| \leq \gamma, \; \forall j \in \{0, 1, \ldots, L\}$, **then**
    $\mathbf{\Omega}_i^* \leftarrow \mathbf{\Omega}_i^{\ell+1}$
  end if
end for
return Tuned weights $\mathbf{W}_i^*$ and $\mathbf{\Omega}_i^*$

A. Scenario 1: Linear Trajectory with a Constant Speed

In this scenario, the vessel is set to follow a trajectory defined by $v^{\text{ref}}(t) = 9\,\text{m/s}$ and $\psi^{\text{ref}}(t) = 0.5\,\text{rad}$, $\forall t \geq 0$, with initial conditions $(x^{\text{ref}}(0), y^{\text{ref}}(0)) = (40, 60)$ and $(x(0), y(0)) = (-100, -100)$. Fig. 2 reveals the convergence of the variations in the critic weights. The values of the thrust and rudder control forces are shown in Fig. 3. They are synchronized with the critic weights adaptation results. The vessel starts off by picking up speed till it latches to the reference trajectory in less than $50\,\text{s}$. This is clearly seen in Figs. 4 and 5.

B. Scenario 2: Reference Trajectory with Time-Varying Linear and Angular Velocities

In this experiment, the vessel is set to track a challenging trajectory defined by $v^{\text{ref}}(t) = 9 + 0.2\,n(t)\,(15\cos(5\pi\Delta t) + 5\sin(5\pi\Delta t))$ and $\psi^{\text{ref}}(t) = 0.5\,n(t)\cos(20\pi\Delta t)$. The parameter $n(t)$ is a random variable drawn from a Gaussian distribution $\sim \mathcal{N}(0, 1)$. The initial conditions are taken as

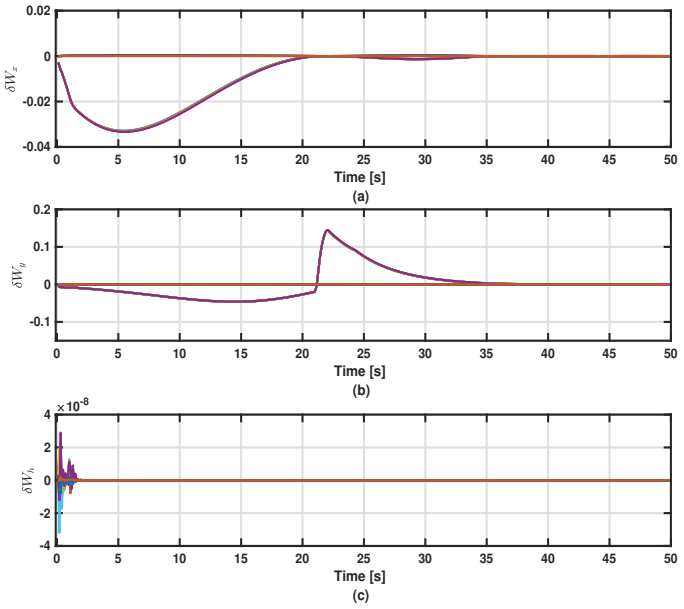

Fig. 2. Scenario 1: variations in critic weights, (a) δW_x, (b) δW_y, and (c) δW_h.

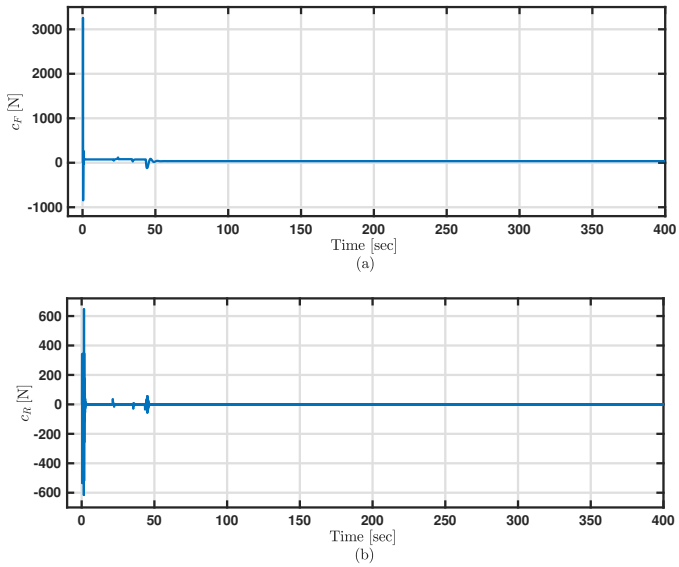

Fig. 3. Scenario 1: (a) Thrust control force c_F, (b) Rudder control force c_R.

$(x^{\text{ref}}(0), y^{\text{ref}}(0)) = (40, 60)$ and $(x(0), y(0)) = (30, 60)$. The thrust and rudder forces are shown in Fig. 6. The adaptive learning process enabled the vessel to track the rapidly varying sinusoidal trajectory, as illustrated in Figs. 7 and 8. The figures unveil the ability of the adaptive learning process to adjust to the high maneuverability enforced by the reference trajectory.

VI. Conclusion

The paper introduces an online integral reinforcement learning mechanism to control an underactuated sea vessel. The solution employs a value iteration process that uses an integral form of Bellman equation. This approach does not employ any traditional error dynamics equations which typically result in hard-to-implement control policies. It rather makes use of measurements relevant to the position of the vessel to make

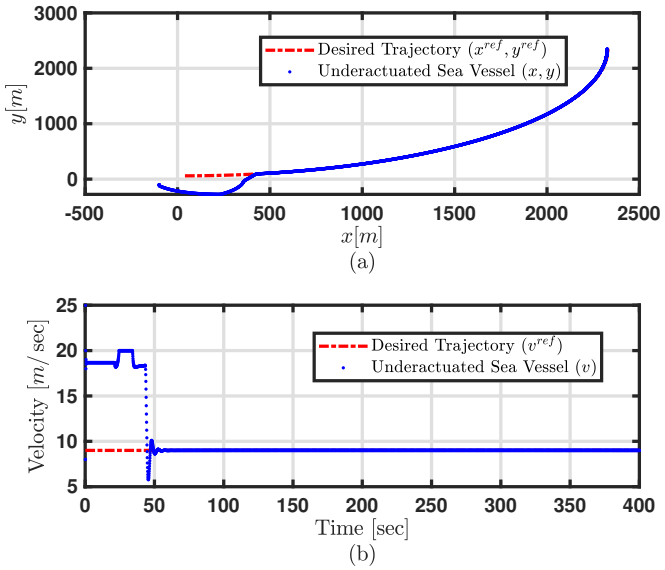

Fig. 4. Scenario 1: (a) Position phase plan, (b) Linear velocity.

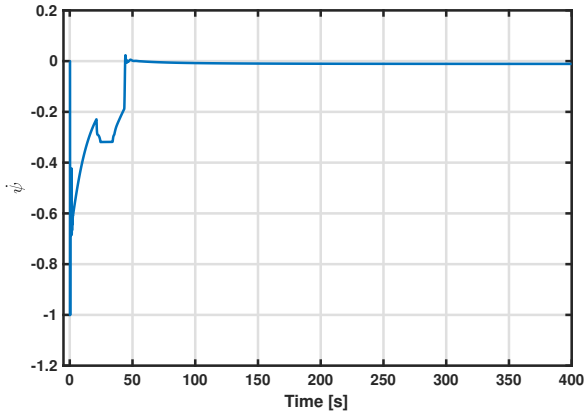

Fig. 5. Scenario 1: Angular velocity.

optimal control decisions. Further, it relaxes the dependence of the solution on recognizing the complete dynamical model of the vessel by suggesting intermediate model-free control strategies. The thrust and rudder actuation forces of the vessel are set to be explicit functions of such strategies. The adaptive critics are then used to implement the integral reinforcement learning solution by approximating the underlying optimal strategies and their associated values. The performance of the adaptive learning process is validated using two test cases.

References

[1] Fuguang Ding, Yuanhui Wang, and Yong Wang, "Trajectory-tracking controller design of underactuated surface vessels," in *OCEANS'11 MTS/IEEE KONA*, Sep. 2011, pp. 1–5.

[2] T. Li, R. Zhao, C. L. P. Chen, L. Fang, and C. Liu, "Finite-time formation control of under-actuated ships using nonlinear sliding mode control," *IEEE Transactions on Cybernetics*, vol. 48, no. 11, pp. 3243–3253, Nov 2018.

[3] N. Wang and X. Pan, "Path following of autonomous underactuated ships: A translation–rotation cascade control approach," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 6, pp. 2583–2593, Dec 2019.

[4] P. Werbos, "Beyond regression: New tools for prediction and analysis in the behavior sciences," Ph.D. dissertation, Harvard University, 1974.

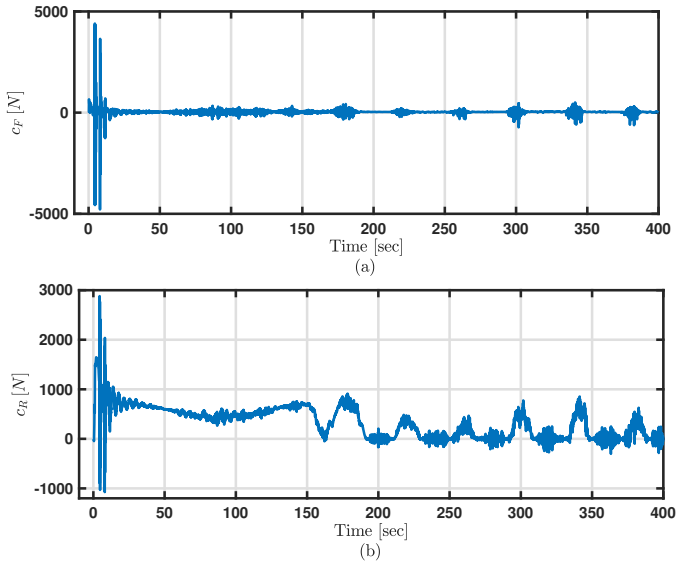

Fig. 6. Scenario 2: (a) Thrust control force c_F, (b) Rudder control force c_R.

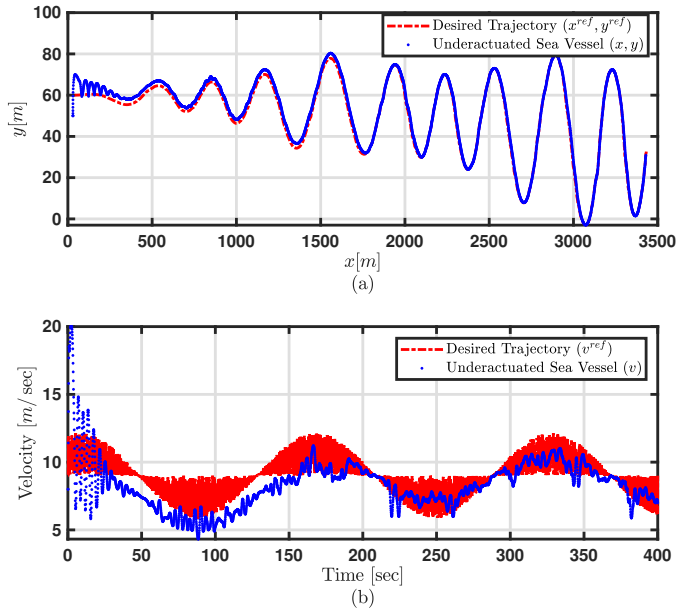

Fig. 7. Scenario 2: (a) Position phase plan, (b) Linear velocity.

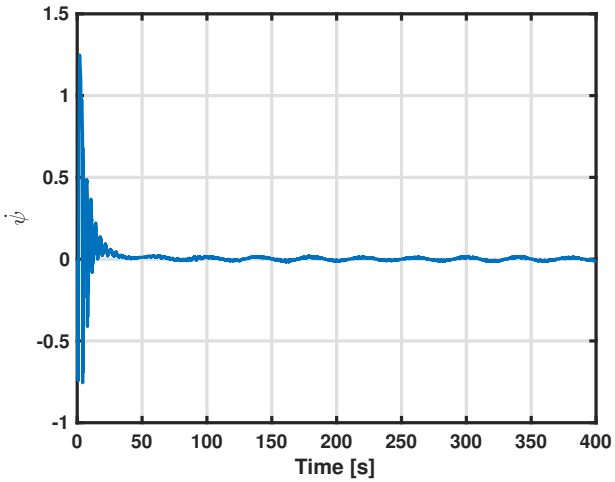

Fig. 8. Scenario 2: Angular velocity.

[5] D. P. Bertsekas and J. N. Tsitsiklis, "Neuro-dynamic programming: An overview," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 1, Dec 1995, pp. 560–564.

[6] A. Bryson, "Optimal control-1950 to 1985," *IEEE Control Systems*, vol. 16, no. 3, pp. 26–33, 1996.

[7] F. Lewis, D. Vrabie, and V. Syrmos, *Optimal Control*, 3rd ed. New York, USA: John Wiley, 2012.

[8] M. Abouheaf, F. Lewis, M. Mahmoud, and D. Mikulski, "Discrete-time dynamic graphical games: Model-free reinforcement learning solution," *Control Theory and Technology*, vol. 13, no. 1, pp. 55–69, 2015.

[9] M. Abouheaf and F. Lewis, *Dynamic Graphical Games: Online Adaptive Learning Solutions Using Approximate Dynamic Programming*. World Scientific, 2014, ch. Chapter 1, pp. 1–48.

[10] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed., ser. Second. Massachusetts: MIT Press, 1998.

[11] M. Abouheaf and W. Gueaieb, "Multi-agent reinforcement learning approach based on reduced value function approximations," in *2017 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*, Oct 2017, pp. 111–116.

[12] M. I. Abouheaf, F. L. Lewis, and M. S. Mahmoud, "Model-free adaptive learning solutions for discrete-time dynamic graphical games," in *53rd IEEE Conference on Decision and Control*, Dec 2014, pp. 3578–3583.

[13] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. L. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration," *Automatica*, vol. 45, no. 2, pp. 477–484, feb 2009.

[14] M. I. Abouheaf, F. L. Lewis, and M. S. Mahmoud, "Differential graphical games: Policy iteration solutions and coupled riccati formulation," in *2014 European Control Conference (ECC)*, June 2014, pp. 1594–1599.

[15] J. Y. Lee, J. B. Park, and Y. H. Choi, "Integral reinforcement learning for continuous-time input-affine nonlinear systems with simultaneous invariant explorations," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 916–932, 2015.

[16] M. Abouheaf, W. Gueaieb, and A. Sharaf, "Load frequency regulation for multi-area power system using integral reinforcement learning," *IET Generation, Transmission Distribution*, vol. 13, no. 19, pp. 4311–4323, 2019.

[17] M. Abouheaf and M. Mahmoud, "Policy iteration and coupled riccati solutions for dynamic graphical games," *International Journal of Digital Signals and Smart Systems*, vol. 1, no. 2, pp. 143–162, 2017.

[18] M. Abouheaf, F. Lewis, K. Vamvoudakis, S. Haesaert, and R. Babuska, "Multi-agent discrete-time graphical games and reinforcement learning solutions," *Automatica*, vol. 50, no. 12, pp. 3038–3053, 2014.

[19] D. Bertsekas and J. Tsitsiklis, *Neuro-Dynamic Programming*, 1st ed. Massachusetts: Athena Scientific, 1996.

[20] L. Busoniu, R. Babuska, and B. D. Schutter, "A comprehensive survey of multi-agent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, 2008.

[21] R. Cui, C. Yang, Y. Li, and S. Sharma, "Adaptive neural network control of auvs with control input nonlinearities using reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 6, pp. 1019–1029, June 2017.

[22] M. Abouheaf, N. Q. Mailhot, W. Gueaieb, and D. Spinello, "Guidance mechanism for flexible-wing aircraft using measurement-interfaced machine-learning platform," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 7, pp. 4637–4648, 2020.

[23] M. Abouheaf and W. Gueaieb, "Model-free adaptive control approach using integral reinforcement learning," in *2019 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, June 2019, pp. 1–7.

[24] B. Widrow, N. K. Gupta, and S. Maitra, "Punish/reward: Learning with a critic in adaptive threshold systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 5, pp. 455–465, 1973.

[25] D. Prokhorov and D. Wunsch, "Adaptive critic designs," *IEEE Transactions on Neural Networks*, vol. 8, no. 5, pp. 997–1007, Sep. 1997.