Exploration of the applicability of probabilistic inference for learning control in underactuated autonomous underwater vehicles

Wilmer Ariza Ramirez[1] · Zhi Quan Leong[1] · Hung Duc Nguyen[1] · Shantha Gamini Jayasinghe[1]

Abstract

Underwater vehicles are employed in the exploration of dynamic environments where tuning of a specific controller for each task would be time-consuming and unreliable as the controller depends on calculated mathematical coefficients in idealised conditions. For such a case, learning task from experience can be a useful alternative. This paper explores the capability of probabilistic inference learning to control autonomous underwater vehicles that can be used for different tasks without re-programming the controller. Probabilistic inference learning uses a Gaussian process model of the real vehicle to learn the correct policy with a small number of real field experiments. The use of probabilistic reinforcement learning looks for a simple implementation of controllers without the burden of coefficients calculation, controller tuning or system identification. A series of computational simulations were employed to test the applicability of model-based reinforcement learning in underwater vehicles. Three simulation scenarios were evaluated: waypoint tracking, depth control and 3D path tracking control. The 3D path tracking is done by coupling together a line-of-sight law with probabilistic inference for learning control. As a comparison study LOS-PILCO algorithm can perform better than a robust LOS-PID. The results show that probabilistic model-based reinforcement learning can be a deployable solution to motion control of underactuated AUVs as it can generate capable policies with minimum quantity of episodes.

Keywords PILCO · LOS · Underwater vehicle · Path tracking · Reinforcement learning

1 Introduction

Autonomous underwater vehicles play an important role in the exploration of the seas. This exploration is primarily driven by commercial, military and scientific needs. In this context, the proper and correct navigation of the vehicle is a key requirement. Motion controllers that are used for navigating AUVs can be classified in four basic strategies: point stabilization (Dong et al. 2015), trajectory tracking (Alonge et al. 2001), path following (Breivik and Fossen 2005) and

✉ Wilmer Ariza Ramirez
  wilmer.arizaramirez@utas.edu.au

  Zhi Quan Leong
  Zhi.Leong@utas.edu.au

  Hung Duc Nguyen
  h.d.nguyen@utas.edu.au

  Shantha Gamini Jayasinghe
  shantha.jayasinghe@utas.edu.au

[1] Australian Maritime College, University of Tasmania, Newnham, Australia

path tracking (Xiang et al. 2011). Point stabilization controllers stabilize a vehicle to the desired goal posture from an initial configuration (Forouzantabar et al. 2012). Trajectory tracking controllers use a virtual vehicle to generate a reference trajectory that has an associated time required to be employed by the real vehicle (Do et al. 2004). In the case of path following the vehicle is forced to pursue the desired path without temporal specifications. In the case of path following controller, it usually employs the line-of-sight (LOS) coupled with another controller to minimize the error between the obtained geometric references and the vehicle variables. The final strategy is path tracking, which combines trajectory tracking and path following by the introduction of a virtual time parameter to force the vehicle to complete the path whitin a specific time.

The difficulties of controlling underwater vehicles arises due to non-linear and time-varying dynamics of underwater vehicles, uncertainties in its hydrodynamic coefficients and disturbance in the environment (e.g. ocean currents). Furthermore, all complexities are exacerbated for the controller in underactuated vehicles (Forouzantabar et al. 2012); underac-

tuated vehicles have more degrees of freedom to be controlled than surfaces of control. Nevertheless, this configuration is more prevalent as it is the most energy efficient design for travelling at high speeds (Xiang et al. 2015).

Waypoint tracking is the most common methodology to control a vehicle, e.g. commercial vehicles such as Gavia (Roper 2007) and REMUS (Holsen 2015) use this methodology. Waypoint tracking is directing the vehicle to approximate to a series of specific target points. The vehicle calculates the required direction to which the vehicle should be directed and upon arriving at the proximities of a point is given a new target. Like many industries, PID is the most common methodology to control the vehicle orientation and speed. However, there is research to employ more robust options than PID. Rout and Subudhi (2017) employs a NARMAX model from the vehicle and a constrained self-tuning controller to direct the vehicle to the respective target. Other methodologies employ a combination of LOS for waypoint (Ataei and Yousefi-Koma 2015) and a standard controller or backstepping techniques to minimize the error between the vehicle position and the desired position (Saravanakumar and Asokan 2011).

In the case of LOS, commercially LOS-PID and LOS-Fuzzy controllers are employed as their implementation are simpler and more accepted in the industry (Yu et al. 2016; Xiang et al. 2017; Wang et al. 2017). Other research as Caharija et al. (2012) used a LOS guidance law with two integrators and three feedback controllers to compensate for external unknown perturbation such as ocean current which is one of the weaknesses for methodologies as PID/Fuzzy controllers. Yang et al. (2016) have used an alternative methodology of a grey prediction to obtain the next AUV position in advance and then use LOS to calculate the desired angles, such that if there is environmental interference, the vehicle will not be affected.

In the search of more robust controllers, nonlinear control techniques had been explored. Repoulias and Papadopoulos (2007) have designed a horizontal path following controller based on Lyapunov stability theorem and backstepping method. In Liang et al. (2015), a method consisted of Lyapunov stability theorem and feedback gain backstepping reduce the complexity of the controller and improve adjustability of the parameters. Another methodology proposes a global path following for AUV based on the same coordinates to achieve global asymptotic stability of the following error (Gao et al. 2010). Following the research of backstepping, Liang et al. (2016) have adopted fuzzy backstepping sliding mode control to overcome non-linearities, uncertainties and external disturbances.

However, the aforementioned research in controls are focused to provide a more robust path following performance. The controller still requires the calibration of parameters or specific design of observers to identify the unknown parameters of the dynamic model. A methodology to overcome this is the use of machine learning algorithms. The most prominent algorithm in machine learning is neural networks. In particular, the research to control underwater vehicles had focused on the use of machine learning algorithms to recognize uncertainties. Chu and Zhu (2015), and Bian et al. (2012) have designed a combined version of control law for the convergence of the kinematic model and an adaptive backstepping sliding control based in radial basis function (RBF) neural network to identify the unknown parameters of the dynamic model. In a similar way, Wang et al. (2018) have reduce the backstepping complexity by the inclusion of a second-order filter to obtain the derivatives of the virtual controller and filter high-frequency measurement noise, and coupled the filter with an RBF neural network that compensates for vehicle uncertainties.

Although the practicality of machine learning has been largely to identify uncertainties in AUV control, some machine learning algorithms are capable of doing more such as controlling the vehicle directly. Recently, there has been increasing research efforts on the use of reinforcement learning to generate policies to control underwater vehicles and robots in general. An example of machine learning control can be seen in (Kawano 2005), where reinforcement learning (RL) based on the Markov decision process (MDP) was employed to produce a policy capable of controlling a vehicle around an obstacle with a minimum cost. In the case of path-following, reinforcement learning had been applied to path following of ships. In Shen and Guo (2016), an actor-critic multilayer perception reinforcement learning is used to reduce the tracking error to zero. Deep reinforcement learning has also been proposed as a possible solution for the tracking problem. Yu et al. (2017) employed two neural networks. The primary neural network selects the action and the secondary evaluates whether the produced action is valid; with further modification through a deep deterministic policy gradient. Another application used continuous actor-critic learning automaton algorithm to teach an AUV to follow a pipeline (Fjerdingen et al. 2010), considering the improved performance in search of the policy of this algorithm the number of episodes over the platform can be over the hundreds.

RL can be divided into two methodologies: model-based methods and model-free methods, such as Q-learning (Watkins 1989) or TD-learning (Sutton and Barto 2011). The application of path-following control based in traditional RL such as Q-learning (Gaskett et al. 1999) is highly complex and difficult as a high quantity of experiments is required to acquire data and test each policy iteration. The additional difficulties in underwater vehicles are vehicle safety, maximum time underwater and computational power. RL for a system with low-dimensional state spaces and fairly favourable dynamics can require thousands of trials to arrive at the

appropriate policies (Deisenroth et al. 2015; Yu et al. 2017). Carlucho et al. (2018) explored the application of deep RL over the data obtained from the navigation system and a deterministic policy gradient algorithm to find a policy control velocities and rotation for a fully actuated AUV. Similarly, Knudsen et al. (2019) applied a deep RL algorithm coupled with a proportional derivative controller to maintain a constant position and orientation in respect to a reference point.

Model-based RL methods are more efficient than model-free methods in searching for a useful policy, as the policy is searched over a model and not the real platform. However, their accuracy can suffer severely from model errors. A solution to address the model errors is the use of probabilistic models to express its uncertainty. An application of model-free methodologies that use a probabilistic methodology was propose by De Paula and Acosta (2015). Their methodology use a on-line selective reinforcement learning approach combined with Gaussian Process (GP) regression for learning reference tracking control policies given no prior knowledge of the dynamical system. Deisenroth et al. (2015) have proposed Probabilistic Inference for Learning Control (PILCO), which is a model-based policy search method. The probabilistic model uses non-parametric Gaussian processes (GPs) to characterise the model uncertainty and the policy improvement is based on analytic policy gradients which employs deterministic approximate interference techniques. Due to probabilistic modelling and inference approach, PILCO can achieve higher learning efficiency than other methods in continuous state-action domains and, hence, is directly applicable to complex mechanical systems, such as robots.

In this paper, the authors explore the applicability of PILCO to control underactuated AUVs by a series of simulations with different objectives and target values. The selected vehicle for the simulation is MULLAYA. MULLAYA is an underactuated AUV designed by the Defence Science and Technology Group of Australia as a research platform. The main algorithms required by MULLAYA for deployment and test of software and hardware are: depth control, waypoint tracking and path following. The main goals of our implementation of reinforcement learning with PILCO are:

– Minimum quantities of episodes over the platform;
– Small test time over the platform;
– Minimum quantity of variables to be predicted by the GP; and
– Vehicle safety.

2 Underwater vehicle mathematical model

In Fossen (1994) it was shown that the non-linear dynamic equations of motion of an underwater vehicle can be
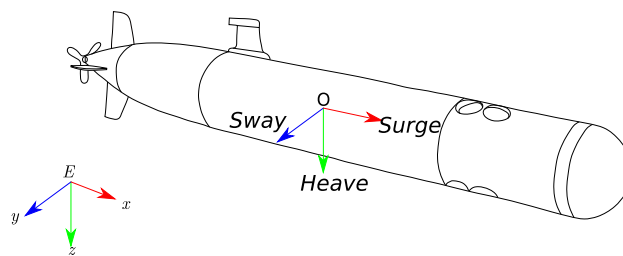

Fig. 1 AUV different reference frames, vehicle frame is equal to the centre of buoyancy

expressed in vector notation defined by a state vector composed by the vector v of velocities on the body frame of the form $[u, v, w, p, q, r]^T$ and the vector η of position in the Earth fixed frame (Fig. 1) of the form $[\xi, \eta, \zeta, \phi, \theta, \psi]^T$ such that

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\,\mathbf{v} + \mathbf{D}(\mathbf{v})\,\mathbf{v} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \qquad (1)$$

with the kinematic equation

$$\dot{\boldsymbol{\eta}} = \mathbf{J}(\boldsymbol{\eta})\,\mathbf{v} \qquad (2)$$

where η position and orientation of the vehicle in Earth-fixed frame, $\mathbf{v}$ linear and angular vehicle velocity in body fixed frame, $\dot{\mathbf{v}}$ linear and angular vehicle acceleration in body fixed frame, $\mathbf{M}$ matrix of inertial terms, $\mathbf{C}(\mathbf{v})$ matrix of Coriolis and centripetal terms, $\mathbf{D}(\mathbf{v})$ matrix consisting of damping or drag terms, $\mathbf{g}(\boldsymbol{\eta})$ vector of restoring forces and moments due to gravity and buoyancy, $\boldsymbol{\tau}$ vector of control and external forces, and $\mathbf{J}(\boldsymbol{\eta})$ rotation matrix that converts velocity in a body fixed frame v to an Earth fixed frame velocity $\dot{\eta}$.

Equation (1) can be expanded into a more general equation of motion as has been shown in Prestero (2001) and Gertler and Hagen (1967). The result of the expansion will be a system of six equation with 73 hydrodynamic coefficients. However, for a complete model the control surfaces must be modelled. In a general case, the resulting forces and moments of a control surface (thrusters and fins) can be expressed as (Fossen 1994)

$$
\begin{aligned}
F_{prop} &= -K_{fprop}\,|n|\,n \\
M_{prop} &= -K_{mprop}\,|n|\,n
\end{aligned}
\qquad (3)
$$

$$
\begin{aligned}
L_{fin} &= K_{L|\delta_{\text{fin}}}\delta_{\text{fin}}v_e^2 \\
M_{fin} &= K_{M|\delta_{\text{fin}}}\delta_{\text{fin}}v_e^2
\end{aligned}
\qquad (4)
$$

A more accurate thruster model can be found in (Kim and Chung 2006) with the inclusion of the motor model and fluid dynamics. However, in this study, the linear parametric model from Fossen (1994) is used, the model employed

do not consider the effect of fins over the total thruster force.

3 LOS guidance law mathematical background

This section describes the mathematical background of the 3D guidance law employ in the present study for control of an underactuated AUV. In this study, it was decided to employ the LOS proposed in Xiang et al. (2017) as it is an extension of the work of Fossen (1994).

If the vehicle kinematic is represented by its spatial position $p(t) \triangleq [x(t), y(t), z(t)]^T$ and its velocity is represented by $v(t) \triangleq \dot{p}(t) \in \mathbb{R}^3$ state is related to the $\{I\}$ frame. Also, the speed is represented by $U(t) \triangleq |v(t)| = \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2 + \dot{z}(t)^2} > 0$. The steering of the underactuated AUV is characterized by the azimuth angle χ and elevation υ (Fig. 2).

$$\begin{cases} \chi = \text{atan2}(\dot{y}, \dot{x}) \\ \upsilon = \arctan\left(\frac{-\dot{z}}{\sqrt{\dot{x}^2+\dot{y}^2}}\right) \end{cases} \quad (5)$$

If its consider a continuously path parametrized by a scalar variable $\varpi \in \mathbb{R}$, the position of a point over the path is represented by $p_p(\varpi) \in \mathbb{R}^3$ (Fig.3). Similarly, the orientation of the point can be defined as

$$\begin{cases} \psi_p = \text{atan2}(y'_p, x'_p) \\ \theta_p = \arctan\left(\frac{-z'_p}{\sqrt{x'^2_p+y'^2_p}}\right) \end{cases} \quad (6)$$
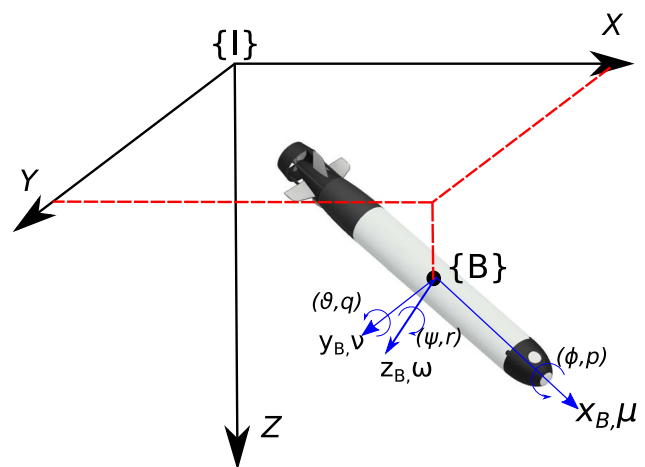

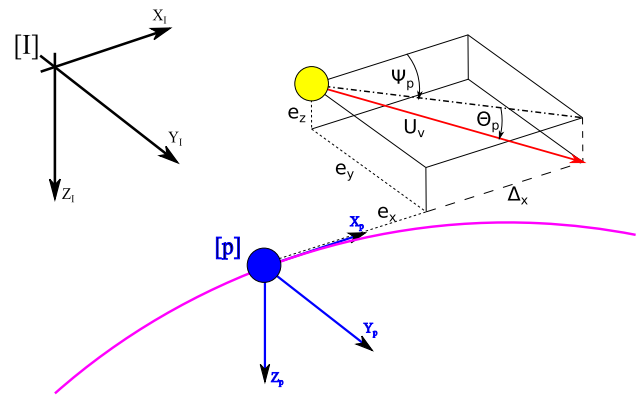**Fig. 2** LOS reference frames

Fig. 3 LOS main variables representation

where $x'_p = dx_p/d\varpi$, $y'_p = dy_p/d\varpi$ and $z'_p = dz_p/d\varpi$. Hence the tracking error is expressed as:

$$\varepsilon = [x_e, y_e, z_e]^T = \mathbf{R}_F^T (p - p_p) \quad (7)$$

where $\mathbf{R}_F^T := \mathbf{R}_z(\psi_p)\mathbf{R}_y(\theta_p)$

$$\mathbf{R}_y = \begin{bmatrix} \cos(\psi_p) & -\sin(\psi_p) & 0 \\ \sin(\psi_p) & \cos(\psi_p) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8)$$

$$\mathbf{R}_z = \begin{bmatrix} \cos(\theta_p) & 0 & \sin(\theta_p) \\ 0 & 1 & 0 \\ -\sin(\theta_p) & 0 & \cos(\theta_p) \end{bmatrix} \quad (9)$$

If the following Lyapunov control function positive definitive is

$$V_\varepsilon = \frac{1}{2}\varepsilon^T \varepsilon \quad (10)$$

The derivative of Eq. (10) can be written as

$$\begin{aligned} V_\varepsilon = x_e \left(U_d \cos(\psi_r)\cos(\theta_r) - U_p\right) \\ + y_e U_d \sin(\psi_r)\cos(\theta_r) - z_e U_d \sin(\theta_r) \end{aligned} \quad (11)$$

where U_d is the desired composite speed of the AUV. The auxiliary control input of the virtual point P_p is chosen as:

$$U_p = U_d \cos(\psi_r)\cos(\theta_r) + k_x x_e \quad (12)$$

where the steering angles are:

$$\begin{cases} \psi_r = \arctan\left(\frac{-k_y y_e}{\Delta_y}\right) \\ \theta_r = \arctan\left(\frac{k_z z_e}{\Delta_z}\right) \end{cases} \quad (13)$$

If the guidance variables Δ_y and $\Delta_z > 0$, the control gains k_x, k_y, k_z are positive constants. If Eqs. (12) and (13) are substituted into Eq. (11) and considering the relationship among

inertial frame $\{I\}$, flow frame $\{W\}$ and path frame $\{F\}$, the desired azimuth angle υ_d and elevation angle χ_d can be written as (Breivik and Fossen 2009):

$$
\begin{aligned}
\upsilon_d &= \arcsin(\sin\theta_p \cos\psi_r \cos\theta_r + \cos\theta_p \cos\theta_r) \\
\chi_d &= atan2(\chi_{dy}, \chi_{dx})
\end{aligned}
\tag{14}
$$

where

$$
\begin{aligned}
\chi_{dy} &= \cos\psi_p \sin\psi_r \cos\theta_r - \sin\psi_p \sin\theta_p \sin\theta_r \\
&\quad + \sin\psi_p \cos\theta_p \cos\psi_p \cos\theta_r
\end{aligned}
\tag{15}
$$

$$
\begin{aligned}
\chi_{dx} &= -\sin\psi_p \sin\psi_r \cos\theta_r - \cos\psi_p \sin\theta_p \sin\theta_r \\
&\quad + \cos\psi_p \cos\theta_p \cos\psi_p \cos\theta_r
\end{aligned}
\tag{16}
$$

In order to transform the path following to path tracking the path was defined over time together with Eqs. (14) and (12). This was to produce not only the desire angles but also the required speed at each time instance. The steering error vector can be expressed as $[e_\mu, e_\upsilon, e_\chi]$ where $e_\mu = \mu_d - \mu_v$, $e_\upsilon = \upsilon_d - \upsilon_v$ and $e_\chi = \chi_d - \chi_v$.

4 Probabilistic inference for learning control (PILCO)

Adaptive controllers that consider uncertainty of the model parameters usully are done by the addition an extra term in the cost function to compensate for it. Here, the uncertainty of the model parameters is penalized to improve the model parameter estimation. PILCO is based in the modelling of model bias and averaging over model uncertainty, where model uncertainty is treated as temporally uncorrelated noise. However, PILCO do not require sampling methods for planning, or is it restricted to specific scenarios.

PILCO algorithm (Deisenroth et al. 2015; Deisenroth and Rasmussen 2011) (Algorithm 1) employs GPs as the base for policy search. A GP can be defined by a mean function $m(\cdot)$ and a positive definitive covariance function $k(\cdot, \cdot)$ commonly known as kernel. Usually a prior mean function $m \equiv 0$ and an exponentiated quadratic kernel (Eq. (17)) are employed. This kernel only has two parameters to learn, l that determines the length of the 'wiggles' in the function and σ^2 which determines the average distance of the function away from its mean (Duvenaud 2014).

$$
k_{SE}(x, x') = \sigma^2 \exp\left(-\frac{(x-x')^2}{2\ell^2}\right)
\tag{17}
$$

Given n training inputs $X = [x_1, \ldots, x_n]$ and corresponding training targets $Y = [y_1, \ldots, y_n]$, the posterior GP hyperparameters l and σ^2 are learned by evidence maximization (Rasmussen 2004). The posterior predictive distribution $p(f_*|x_*)$ of the function value $f_* = f(x_*)$ for a test input

x_* is Gaussian with mean and variance

$$
m_f(x_*) = \mathbf{k}_*^T \left(\mathbf{K} + \sigma_\varepsilon^2 \mathbf{I}\right)^{-1} y
$$

$$
\sigma_f^2(x_*) = k_{**} - \mathbf{k}_*^T \left(\mathbf{K} + \sigma_\varepsilon^2 \mathbf{I}\right)^{-1} k_* + \sigma_\varepsilon^2
\tag{18}
$$

```
[1] init: Sample controller parameters θ ∼ N(0, I);
[2] Apply random control signals and record data.;
[3] repeat
[4]     Learn probabilistic (GP) dynamics model;
[5]     repeat
[6]         Approximate inference for policy evaluation;
[7]         Gradient-based policy improvement;
[8]         Update parameters θ;
        until Convergence;
[9]     return θ*;
[10]    Set π* ← π(θ*);
[11]    Apply π* to system and record data;
    until Task Learned;
```
Algorithm 1: PILCO Algorithm

To find a policy π^* which converges to the desired target, PILCO builds a probabilistic GP dynamics model. This model will be the base for the deterministic approximate inference and policy evaluation, followed by the analytic computation of the policy gradients $\partial J^\pi(\theta)/\partial\theta$ for policy improvement. The policy π is improved based on the gradient information $\partial J^\pi(\theta)/\partial\theta$.

In our application, multiple GPs were employed as the evolving model to be use in the reinforced learning process. Each GPs employs a nonlinear autoregressive network with exogenous inputs (NARX) structure composed of a input vector X which contains the delayed signal $Y_p(t-1)$ and the delayed command signal $u(t-1)$ that predicts output vector Y_p.

Figure 4 show the general configuration employed for the application of PILCO to MULLAYA AUV. The desire state vector η_d is compare to the vehicle state η. At interaction $k = 0$ a random generator will generate three random signals for the control surfaces (propeller, elevator and rudder) to capture an initial GPs model. The nonparametric GPs model is composed of n GPs model of single input single output that model the relation between vector of input X and the measured output from the vehicle. The policy search algorithm produce a control policy by the minimization of the cost function over the fitted GPs model. The resultant policy is tested at $k = 1$ over the real robot.
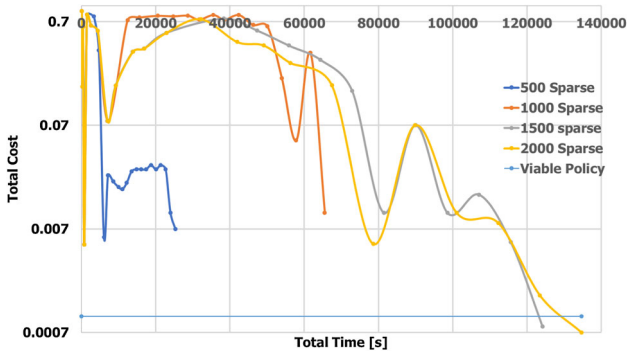
5 Simulation setup

The underwater vehicle should be able to switch between different controllers in a single mission depending on the
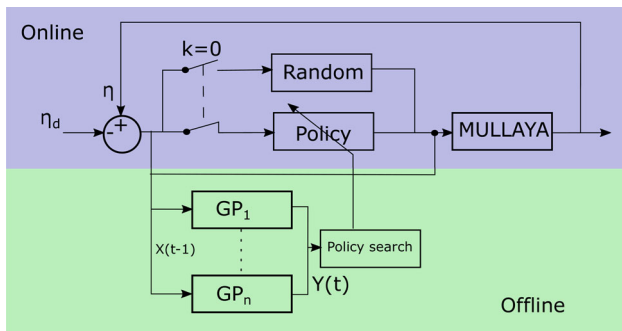
Fig. 4 General diagram of our implementation of PILCO for control of MULLAYA AUV

Fig. 5 Cost function evolution over running time with different sparse point numbers at 23 Trials

task to be solved at a specific time. The possible variants of controllers that an AUV can use in a mission are: follow bottom, depth control, follow pipe, go-to-point, path tracking, path following, etc. A series of different simulations were designed to evaluate the capability of PILCO to control an underactuated AUV within the three different evaluation scenarios: way-point-tracking, Depth Control and Path-Tracking. These three scenarios were chosen because they were the main task required to deploy a mission planning algorithm for MULLAYA AUV.

The number of sparse points for each simulation was established by running simulations with 500, 1000, 1500 and 2000 sparse points and plotting the average cost of the full experiment in respect to the total running time. Figure 5 show the result of total cost for path following policy. 1500 sparse points for our simulation show the best results to obtain a viable policy with the lower running time.

5.1 Vehicle model

The vehicle model employed in this research is derived from semi-empirical calculation of the coefficients for the vehicle MULLAYA. The vehicle is controlled by a single propeller, a pair of elevator fins and a pair of rudder fins. General specifications of the vehicle are given in Table 1. The coefficients

Table 1 MULLAYA AUV particulars

Property	Value	Unit
Length	1.56	m
Diameter	150	mm
Max RPM	3000	RPM
Weight	239.364	N
Buoyancy	246.2	N

Table 2 MULLAYA AUV coefficients employed in simulations

Coeff.	Result	Coeff.	Result	Coeff.	Result
Xuu	− 2.8	Zrp	0.68	Nuv	−30.88
Xwq	− 29.6	Yuudr	12.12	Npq	−5.06
Xqq	− 0.68	Nuudr	− 7.51	Ixx	0.083
Xvr	29.6	Zuuds	− 12.12	Iyy	3.08
Xrr	− 4.95	Kpp	− 1.30E−01	Izz	3.08
Yvv	− 95.37	Kpdot	1.09E− 02	Nwp	0.66
Yrr	− 2.45	Mww	6.96	Nur	− 5.31
Yuv	− 32.9	Mqq	− 135.04	Xudot	− 0.51
Ywp	29.64	Mrp	5.1	Yvdot	− 29.64
Yur	7	Muq	− 5.34	Nvdot	0.66
Ypq	0.68	Muw	27.16	Mwdot	0.66
Zww	− 95.37	Mwdot	− 0.68	Mqdot	− 5.05
Zqq	2.45	Mvp	− 0.68	Zqdot	− 4.94
Zuw	− 32.9	Muuds	− 7.738	Zwdot	− 29.64
Zuq	− 7	Nvv	6.96	Yrdot	4.95
Zvp	− 29.64	Nrr	− 135.03	Nrdot	5.05

were calculated with the same technique of (Prestero 2001) and the obtained coefficients are presented in Table 2. The assumption of our mathematical model are the same as (Prestero 2001) with the exception of the thruster model. External forces as ocean currents were not modelled. However, noise was added to the sensor measurement of the navigation system. As a engineering research platform, the vehicle will undergo multiple transformations over time in shape and internal engineering. This constant change requires the needs of constant update of the controller for the vehicle. Possible future modification can include vectorized propulsion systems and buoyancy controllers.

5.2 Waypoint tracking

The first type of controller evaluated for the underwater vehicle is a waypoint tracking controller. These types of controllers can be employed to transfer the vehicles between location were more specific controllers are employed or if can be work with the results of a path planning law that convert a desire path in waypoints. In this case, it is desired to control the vehicle velocity, azimuth and elevation such that
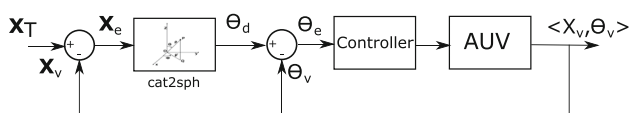
Fig. 6 Control system block for waypoint tracking control with PID and RL

the vehicle moves towards a specific location. The methodology used in this simulation is similar to the one employed in Bi et al. (2014) but extended to 3D. If $_v$ is the vehicle state vector $[X_v, Y_v, Z_v, \phi_v, \theta_v, \psi_v]$ that can be divided in position vector $\mathbf{X_v}$ and orientation vector $`_v$ on earth frame and the target point is expressed as the vector $\mathbf{X_T} = [X_d, Y_d, Z_d]$ the angles of the vector between the current position and the desired position can be expressed as

$$\psi_d = \tan^{-1}\left(\frac{Y_d - Y_v}{X_d - X_v}\right)$$

$$\theta_d = \tan^{-1}\left(\frac{\sqrt{(Y_d - Y_v)^2 + (X_d - X_v)^2}}{Z_d - Z_v}\right) \quad (19)$$

and the vector of angle errors can be expressed as the difference between the vehicle orientation and the desire orientation, i.e $`_e = [e_\psi, e_\theta, e_u]$, where $e_\psi = \psi_d - \psi_v$ and $e_\theta = \theta_d - \theta_v$. The target of the policy will be to minimize the error angles to zero and the surge speed error to zero. In the proposed simulation a surge speed of 1.2 m/s was set as the desired speed with an initial position of [0, 0, 0] ,an orientation of [0, 0, $\pi/4$] and an initial surge speed of 0.5 m/s. The first 12 s of real model simulation was employed to learn the policy with a target point with coordinates [40, 40, 10]. A constraint to limit the vehicle turn $< 180°$ was applied to the process of policy testing over the vehicle model. A second simulation with two target points [30, 30, 10] and [90, 100, 10] is employed to check the viability of the policy. A total of 1000 sparse points was located as the limiter from which a sparse model will be employ. The policy and simulation were executed at 5 Hz. A noise with variance of $\sigma^2 = 0.005$ was employed in the simulation and a variance of $\sigma^2 = 0.2$ was employed for the start position of the vehicle. When the vehicle arrives at 3 m of the objective the vehicle will be given the secondary target as the new objective. If the vehicle arrives at 1 m of the final target the simulation will stop. These switching points are place in accordance with the usual parameters of a mission planner (Thompson et al. 2020) (Fig. 6).

5.3 Depth control

A secondary scenario is explored to evaluate the controller's capability to keep a specific depth while traveling between location. This is an extension to the previous search of a

policy, i.e. waypoint. In this scenario, we want the vehicle to arrive at a specific depth before going to the desired location. The policy will minimize the vector of error to zero with the difference that the vector of errors is $`_e = [e_\psi, e_\theta, e_u, e_z]$, where $e_z = z_d - z_v$ similar to the waypoint tracking. The vehicle starts from the same initial position of the waypoint tracking scenario [0, 0, 0]. the proposed training target was the vector [40, 40, 5] and the selected test target points are [30, 30, 5] and [90, 100, 5]. A total of 1500 sparse points was located as the limiter from which a sparse model will be employed. The policy and simulation were executed at 5 Hz. A noise of $\sigma^2 = 0.005$ was employed for the simulation and a $\sigma^2 = 0.2$ was employed for the start position of the vehicle. When the vehicle arrives at 3 m of the objective the vehicle will be given the secondary target as the new objective. I If the vehicle arrives at 1 m of the final target the simulation will stop.

5.4 Path tracking

With the aim to test the capability of PILCO for path tracking, the scenario consists of a single policy to control propeller force, elevator force and rudder force of the vehicle. The decision to learn the force and not to control direct the RPM and angle of the fins was to be able to compare the policy to a standard controller from the literature. In the design of the authors path tracking simulation, the equation presented in section Sect. 3. Figure 7 shows the block diagram of the control implementation whereby in the case of PILCO the policy will evolve based on the learned GPs. The target of all learned policies is to reduce the vector of errors $[e_\mu, e_\upsilon, e_\chi]$ to zero. A LOS-PID was used as a performance comparison. The LOS-PID (Fig. 7) with the exact coefficient of the model was coded in the same ways as (Xiang et al. 2017) with the inclusion of measurement noise. The initial position of the vehicle was established as [60, 3, 1] with an initial orientation [0, 0, $3\pi/4$]. A total of 1500 sparse points was located as the limiter from which a sparse model will be employed. A noise with $\sigma^2 = 0.001$ was employed throughout the simulation and a random variation of the start point with a variance of $\sigma^2 = 0.2$ was employed. Higher values of σ were not possible for the LOS-PID to be comparable as it was not able to overcome higher values of noise. Both the PID and PILCO policy were executed at a frequency of 10 Hz. A helix path (Fig. 8) was parametrized as is show in Eq. (20).

$$X_{helix} = 60 \, \cos(0.02618 \, w_{ramp}(t))$$
$$Y_{helix} = 60 \, \sin(0.02618 \, w_{ramp}(t))$$
$$Z_{helix} = 2 + 2 \, w_{ramp}(t)/200 \quad (20)$$
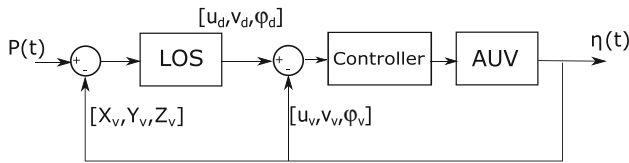
Fig. 7 Control systems block for LOS

Fig. 8 Spiral path to be followed

Fig. 9 Real model waypoint tracking data example for GPs learning
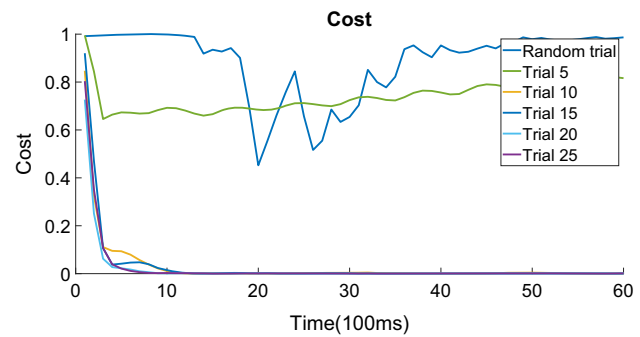
Fig. 10 Waypoint tracking cost function evolution over trial with real vehicle model
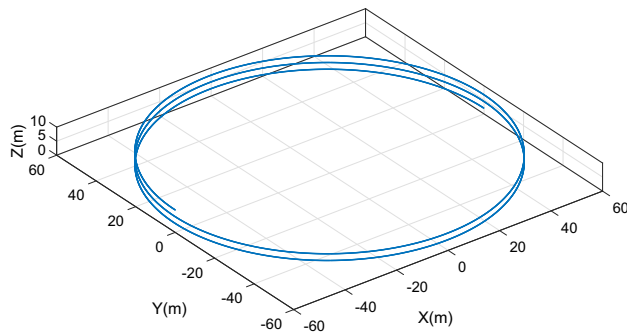
where $w_{ramp}(t)$ is a function over time with a slope m. The control learning of the vehicle was done over the first 20 s at a frequency of 10 Hz.

6 Results

6.1 Waypoint tracking

As evident in Fig. 11, the learning of a policy for waypoint tracking of an AUV is possible with the application of PILCO. However, reinforcement learning does not require the calibration of parameters, but rather, requires the tuning of the parameters of the **Q** matrix from the cost function. For the cost function design, the surge speed was given a higher importance in the cost function than the other error vectors. This escalation of each target in the learning policy was needed as an underactuated AUV needs to get to a specific speed to be able to dive and navigate with a more linear model. An example of the data employed to create the GPs model can be seen in Fig. 9.

The authors simulations showed that the required number of episodes over the platform was under 25 to obtain a usable policy to control the vehicle and be able to arrive at all three targets. Figure 10 shows the evolution of the cost function over time for the platform. The cost function drops rapidly to a low value as the vehicle learns to control the surge speed and from there, how to control its orientation. The decision to do this is based in that the control surfaces as fin require a minimum speed to be useful.

The result from the final policy selected is shown in Fig. 11; plotting against the results from the PID controller
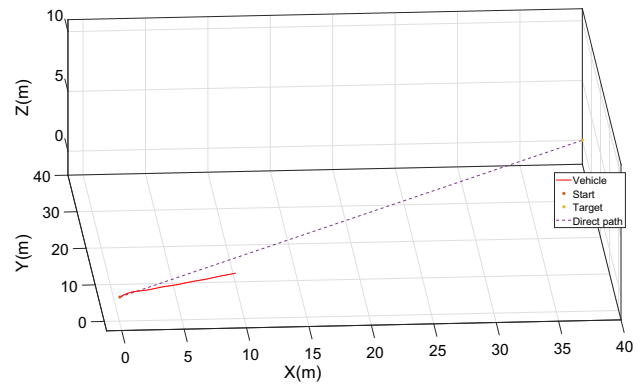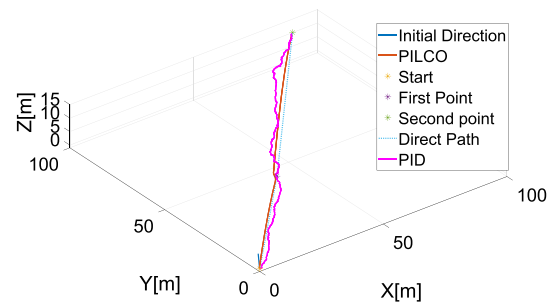

Fig. 11 3D view Waypoint tracking comparative result between PILCO and PID Results

for the same scenario. Both PID and PILCO arrive near to both test point. However, PILCO shows a more direct route taken towards the targets. The PID controller took 1201 cycles to arrive equivalent to 240.2 s and the PILCO policy took 569 cycles equivalent to 113.8 s. PILCO has an advantage over a simple PID as PILCO has learned the policy over a noisy platform(or enviroment) and the PID does not have any component to compensate for the noise in the measurement. For PID to compensates for the noise an observer will have to be designed and implemented. However, this will increase the complexity of the controller and the need for more design and calibration time for the PID controller. The results show that the waypoint tracking with a PILCO controller can be a

viable option as the tuning of the controller coefficients are practically zero. The robustness of the PILCO policy is higher as the platform will start the learning from different angles and can overcome noise in the measurement. The policy is also updated in case of failure. A down point of PILCO in the simulated scenarios is that the controller tries to always have the same behaviour, e.g. if the vehicle dives a little before directing to the first point after the second point the policy will try to repeat the same behaviour.

6.2 Depth control

The simulation of simultaneous waypoint tracking, and depth control of underwater vehicles shows that PILCO can learn a complex policy to control the vehicle. An example of the training data used for learning the vehicle GPs model can be seen in Fig. 12. The quality of the learning of the GPs model is directly related to the quality of the learning of the policy. After 20 episodes episodes over the platform a good quality GPs model has been learned such that a policy can be learnt. The evolution of the cost function over time can be seen in Fig. 13. In similar way to the standard waypoint tracking, the cost evolves very fast to a low value and then small changes are done in the policy learning. This small change is produced by the scaling of each of the targets to be obtained.
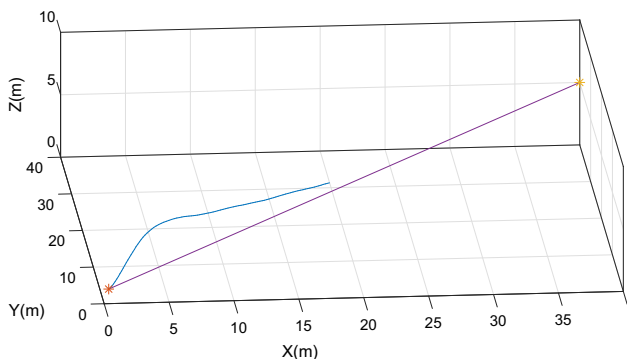

Fig. 12 Real model waypoint tracking and depth control data example for GPs learning
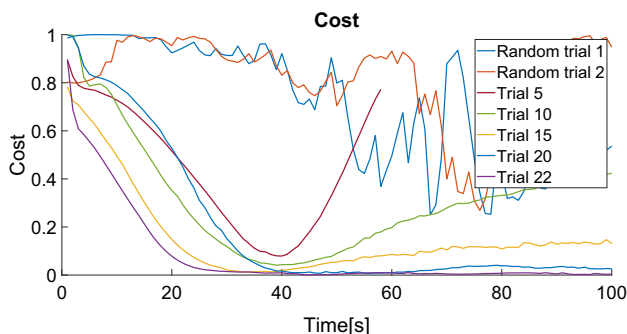

Fig. 13 Waypoint tracking and depth control cost function evolution over real vehicle model
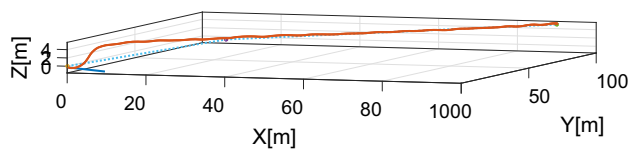

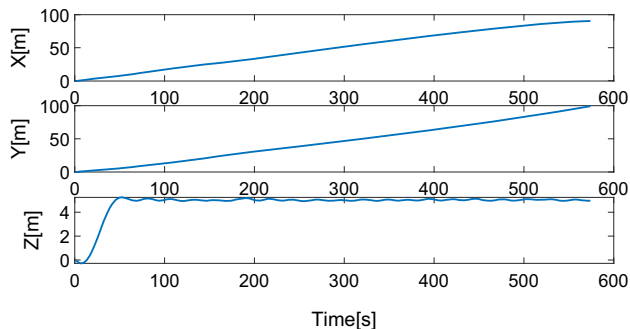**Fig. 14** 3D view Waypoint tracking and depth control PILCO results

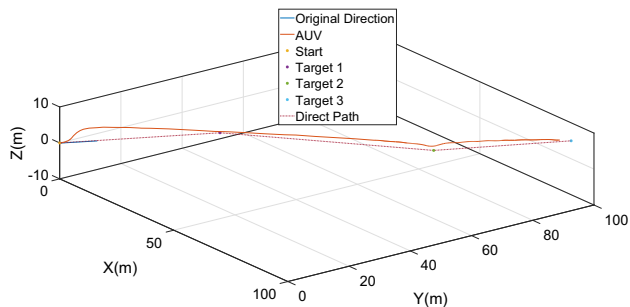**Fig. 15** $[X, Y, Z]$ Waypoint tracking and depth control PILCO results

Fig. 16 3D Waypoint tracking and depth control PILCO results with a target outside of the control depth
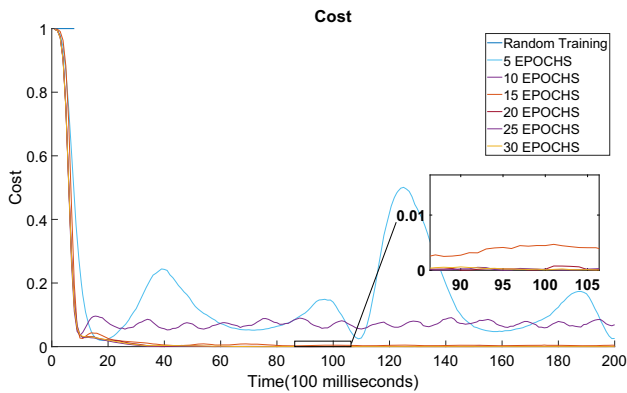
Figure 14 shows the results of policy 22. The policy is capable of keeping the depth at near to 5 m and at the same time keep all four errors at near zero value. Usually, the task described here will be done with two controllers a depth controller and a azimuth controller. As we are applying a single controller with two targets(elevation, depth) that at the start go on different paths. The $[X, Y, Z]$ results from the simulation are presented in Fig. 15. The policy produces a different behaviour than normal controllers, if by operator error a target was setup with a different depth than the desire depth the learned policy will try to push the vehicle to the target disregarding the depth by a small quantity such the target can be completed this can be observed in Fig. 16 . Three targets were setups as [30, 30, 5], [70, 70, 3], and [90, 100, 5] the second target is outside of the desire depth of the policy.

6.3 Path tracking

The training evolution of the GPs model in the direction of a better policy allows PILCO to search for a better policy
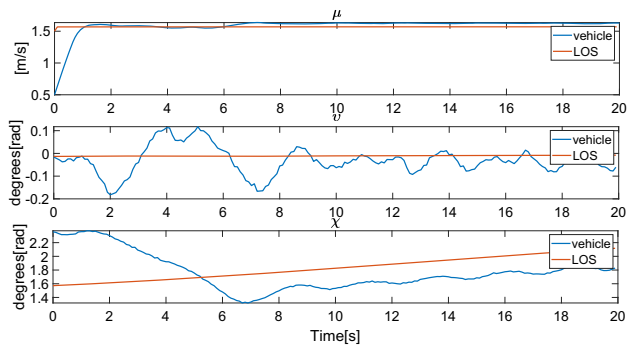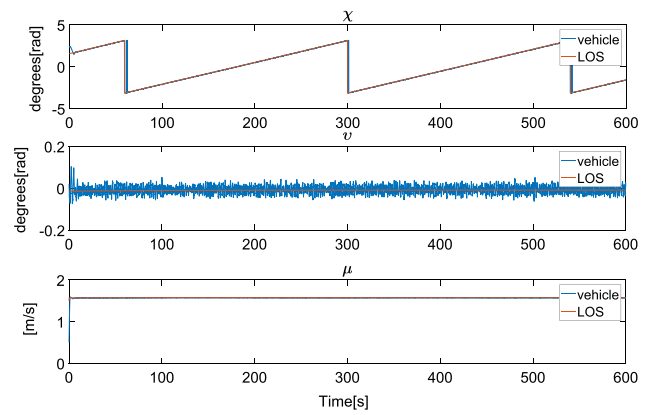
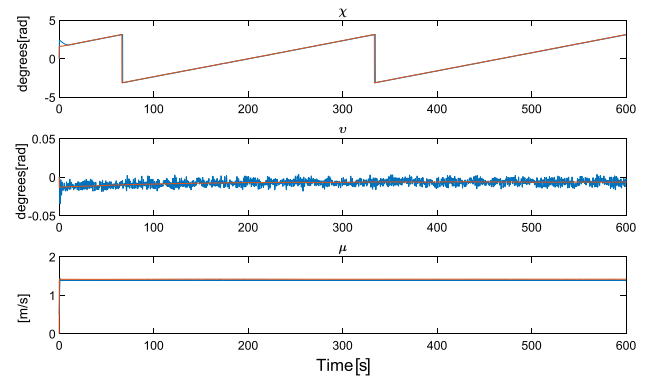**Fig. 17** Sample of data used for learning of GPs model employed for gradient descend learning of policy

Fig. 18 Cost evolution for each policy test over AUV real mathematical model

Fig. 19 Desire LOS angles and speed and vehicle LOS angles and speed produced by PID controller


Fig. 20 Desire LOS angles and speed and vehicle LOS angles and speed produced by PILCO policy

**Table 3** RMSE results from LOS-PID and LOS-PILCO

RMSE	Value
LOS-PID	1.214
LOS-PILCO	0.89

after each iteration with the real model (Fig. 18). Figure 17 presents an example of the data used for training of the GPs model that is later used for policy search. The reinforcement learning simulation to learn a LOS controller with PILCO shows that the reinforcement learning algorithm can learn to follow the desired path based on the desire angles produced by a LOS law (Fig. 18). The policy can perform better than a LOS-PID controller that is without noise compensation algorithms. The results from the PID to follow the LOS law can be seen in Fig. 19, without noise the PID can stabilize himself very fast but with minimum noise in the depth sensor the χ angle cannot be completely stable and the controller will fluctuate. In the same way, PILCO controller (Fig. 20) reacts to the noise of the measurement but can follow the desired path in a better way that the LOS-PID controller implemented.

The measurement of the RMSE between the desire vehicle position and the vehicle position for both LOS-PID and LOS-PILCO is presented in Table 3. LOS-PILCO is shown to out-perform the PID controller in the reduction of error between the LOS law and the vehicle position. Not only in the accuracy of placing the vehicle in the correct position but in the speed of deployment of a controller to follow LOS will LOS-PILCO will require less tuning with field tests. Another

advantage of PILCO is the absence of knowledge from the vehicle coefficients and not need to use other vehicles variables a surge, heave and sway speeds.

Figures 21 and 22 present the comparative plots of position in X, Y, Z and the 3D path taken by the vehicle with LOS-PID controller and LOS-PILCO policy. Both controllers shows their ability to direct the AUV to the desired path, with the PILCO policy showing a better performance after episode 20. The first 60 s of the produced signals from the selected PILCO policy in comparison to the signals generated by the PID controller can be seen in Fig. 23. All the commanded signals are in the range of response of the actuators and the noise level of both are in the expected range. PILCO and PID commanded signal for propeller are very similar in their behaviour. In contrast, the commanded signals of PILCO tend to response in a more energetic and harmonic form to the platform disturbance.
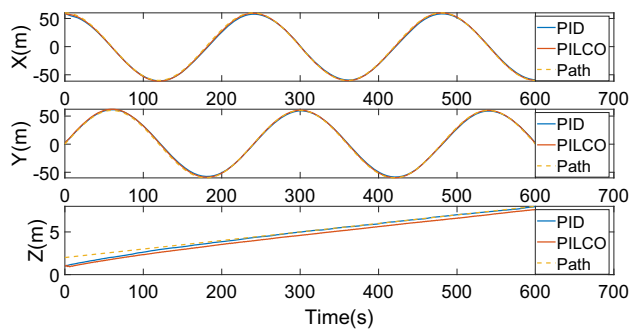
Fig. 21 X,Y,Z comparison of vehicle position controlled with LOS-PID and LOS-PILCO policy over time for spiral path
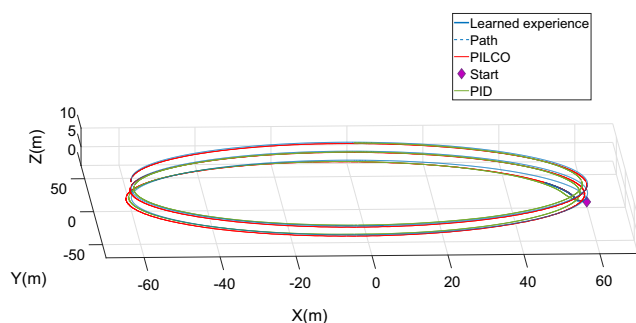
Fig. 22 3D comparison of vehicle controlled with PID and PILCO controller
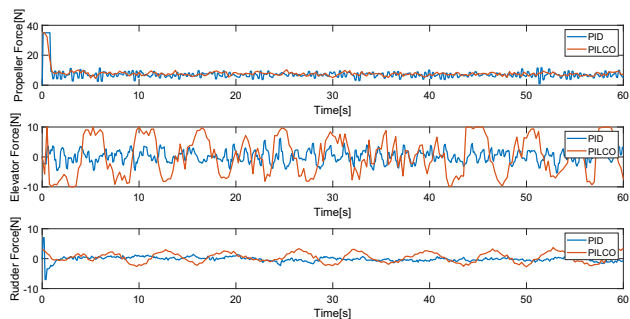
Fig. 23 Comparison of vehicle command between PID controller and PILCO policy

7 Conclusions

This paper investigates the applicability of PILCO algorithms and its requirements to learn policies to control underactuated AUVs. Three sets of simulations were designed to evaluate the capability of PILCO for waypoint tracking, depth control and path tracking. The simulations shown that a simple waypoint tracking control can be learnt in a small quantity of experiments over the real vehicle, the performance of the learnt policy was compared with a PID controller which is over-performed by the policy as the policy obtained is learned over the platform and the non-parametric model includes noise.

In a similar way, a depth control policy was learned by mixing the waypoint tracking objective with the depth objective. However, in this research the learning of a policy to only control depth was not successful as the GPs model to allow the learning of a policy requires more information. Nevertheless, in the simulation of depth control it had been shown that a policy of depth control can be obtained by the learning of simultaneous waypoint tracking and depth control. The combination of objectives gives us a more intuitive behaviour, similar to what a human will do with the proposed objectives.

In the case of the proposed LOS-PILCO methodology, it has been shown that PILCO is a viable option to learn a policy to minimize the error between a LOS law and the vehicle position. The PILCO policy is shown to perform equally and sometimes better than a PID. The RMSE shows that PILCO can obtain better performance and the long period of simulation shows that the learned policy can constantly minimize the error to the desired value.

In the real deployment of PILCO in MULLAYA AUV, the full PILCO algorithm will not be deployed as the MULLAYA AUV currently only possesses a dual core processor and pre-test have showed that running the full learning process is too computational expensive. The integration of the full learning process will be reassessed in the next hardware phase of the vehicle. The policy will be the only deployment to the vehicle and the logged data from the inertial navigation data will be the source for the improvement of the GPs model and the learning of the new policy. GPs models are less sensitive to the noise as their kernel include a function of smoothness of the predicted data. However, this may be not enough to compensate for the highly noisy underwater environment. In the case of noise overpassing the predicted limits, the presented model with exponentiated quadratic kernel can be modified to include the convolution of kernels adding a white noise kernel for noise modelling. Other algorithms can over perform a policy obtained by PILCO to control underwater vehicles. However, PILCO can be deployed in a short time as the need of building a mathematical model of long hours of testing can be reduced. In the case of MULLAYA AUV is a solution for the constant shape change and deployment of new technology. Further exploration of other model based algorithms for reinforced learning such as model based deep reinforced learning need to be explored.

PILCO algorithm has shown that is applicable to an underactuated AUV. In the simulations were consider limits that are a requirement for safety of the vehicle as maximum depth and maximum angles. The limits imposed to the platform do not limited the capability of PILCO to learn a viable policy. In the design of the cost function, the vehicle shape and type of actuators force the selection of values of the cost function such that forward speed has to represent a higher cost than the specific vehicle angles or position. Model based reinforce-

ment learning for underactuated AUV shows to be a solution motion control of underwater vehicles and can be a solution to control bioinspired vechicles which are more dynamically complex to describe with a mathematical model.

References

Alonge, F., D'Ippolito, F., & Raimondi, F. M. (2001). Trajectory tracking of underactuated underwater vehicles. In *Proceedings of the 40th IEEE conference on decision and control (Cat. No.01CH37228)* (Vol. 5, pp. 4421–4426). https://doi.org/10.1109/CDC.2001.980898.

Ataei, M., & Yousefi-Koma, A. (2015). Three-dimensional optimal path planning for waypoint guidance of an autonomous underwater vehicle. *Robotics and Autonomous Systems*, *67*, 23–32.

Bi, S., Niu, C., Cai, Y., Zhang, L., & Zhang, H. (2014). A waypoint-tracking controller for a bionic autonomous underwater vehicle with two pectoral fins. *Advanced Robotics*, *28*(10), 673–681. https://doi.org/10.1080/01691864.2014.888373.

Bian, X., Zhou, J., Yan, Z., & Jia, H . (2012). Adaptive neural network control system of path following for AUVs. In *2012 Proceedings of IEEE Southeastcon* (pp. 1–5). https://doi.org/10.1109/SECon.2012.6196982.

Breivik, M., & Fossen, T. I. (2005). Guidance-based path following for autonomous underwater vehicles. In *Oceans 2005* (Vol. 1–3, pp. 2807–2814).

Breivik, M., & Fossen, T. I. (2009). Guidance laws for autonomous underwater vehicles, chap 4. In A. V. Inzartsev (Ed.), *Underwater vehicles*. Rijeka: IntechOpen. https://doi.org/10.5772/6696.

Caharija, W., Pettersen, K. Y., Gravdahl, J. T., & Borhaug, E. (2012). Path following of underactuated autonomous underwater vehicles in the presence of ocean currents. In *2012 IEEE 51st annual conference on decision and control (CDC)* (pp. 528–535).

Carlucho, I., Paula, M. D., Wang, S., Petillot, Y., & Acosta, G. G. (2018). Adaptive low-level control of autonomous underwater vehicles using deep reinforcement learning. *Robotics and Autonomous Systems*, *107*, 71–86. https://doi.org/10.1016/j.robot.2018.05.016.

Chu, Z., & Zhu, D. (2015). 3d path-following control for autonomous underwater vehicle based on adaptive backstepping sliding mode. In *2015 IEEE international conference on information and automation* (pp. 1143–1147). https://doi.org/10.1109/ICInfA.2015.7279458

De Paula, M., & Acosta, G. G. (2015). Trajectory tracking algorithm for autonomous vehicles using adaptive reinforcement learning. In *OCEANS 2015 - MTS/IEEE Washington* (pp. 1–8). https://doi.org/10.23919/OCEANS.2015.7401861.

Deisenroth, M., & Rasmussen, C. E. (2011). Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th international conference on machine learning (ICML-11)* (pp. 465–472).

Deisenroth, M. P., Fox, D., & Rasmussen, C. E. (2015). Gaussian processes for data-efficient learning in robotics and control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(2), 408–423. https://doi.org/10.1109/TPAMI.2013.218.

Do, K. D., Pan, J., & Jiang, Z. P. (2004). Robust and adaptive path following for underactuated autonomous underwater vehicles. *Ocean Engineering*, *31*(16), 1967–1997. https://doi.org/10.1016/j.oceaneng.2004.04.006.

Dong, Z., Wan, L., Li, Y., Liu, T., Zhuang, J., & Zhang, G. (2015). Point stabilization for an underactuated auv in the presence of ocean currents. *International Journal of Advanced Robotic Systems*, *12*(7), 100. https://doi.org/10.5772/61037.

Duvenaud, D. (2014). Automatic model construction with gaussian processes. Ph.D. thesis., University of Cambridge.

Fjerdingen, S. A., Kyrkjebø, E., & Transeth, A. A. (2010). Auv pipeline following using reinforcement learning. In *Robotics (ISR), 2010 41st international symposium on and 2010 6th German conference on robotics (ROBOTIK), VDE* (pp. 1–8).

Forouzantabar, A., Gholami, B., & Azadi, M. (2012). Adaptive neural network control of autonomous underwater vehicles. *World Academy of Science, Engineering and Technology*, *6*(7), 304–309.

Fossen, T. I. (1994). *Guidance and control of ocean vehicles* (Vol. 199). New York: Wiley.

Gao, J., Yan, W., Zhao, N., & Xu, D. (2010). Global path following control for unmanned underwater vehicles. In *2010 29th Chinese Control Conference (CCC)* (pp. 3188–3192). IEEE.

Gaskett, C., Wettergreen, D., & Zelinsky, A., et al. (1999). Reinforcement learning applied to the control of an autonomous underwater vehicle. In *Proceedings of the Australian conference on robotics and automation (AuCRA99)*.

Gertler, M., & Hagen, G. R. (1967). Standard equations of motion for submarine simulation. David W Taylor Naval Ship Research and Development Center Bethesda MD: Tech. rep.

Holsen, S. A. (2015). Dune: Unified navigation environment for the remus 100 auv-implementation, simulator development, and field experiments. Master's thesis, NTNU.

Kawano, H. (2005). Method for applying reinforcement learning to motion planning and control of under-actuated underwater vehicle in unknown non-uniform sea flow. In *2005 IEEE/RSJ international conference on intelligent robots and systems* (pp. 996–1002). https://doi.org/10.1109/IROS.2005.1544973.

Kim, J., & Chung, W. K. (2006). Accurate and practical thruster modeling for underwater vehicles. *Ocean Engineering*, *33*(5–6), 566–586.

Knudsen, K. B., Nielsen, M. C., & Schjólberg, I. (2019). Deep learning for station keeping of AUVs. In *Oceans 2019 MTS/IEEE Seattle* (pp. 1–6). https://doi.org/10.23919/OCEANS40490.2019.8962598.

Liang, X., You, Y., Su, L., Li, W., & Zhang, J. (2015). Path following control for underactuated auv based on feedback gain backstepping. *Technical Gazette*, *22*(4), 829–835.

Liang, X., Wan, L., Blake, J. I., Shenoi, R. A., & Townsend, N. (2016). Path following of an underactuated auv based on fuzzy backstepping sliding mode control. *International Journal of Advanced Robotic Systems*, *13*(3), 122. https://doi.org/10.5772/64065.

Prestero, T. (2001). Verification of a six-degree of freedom simulation model for the remus autonomous underwater vehicle. Thesis.

Rasmussen, C. E. (2004). Gaussian processes in machine learning. In O. Bousquet, U. von Luxburg, & G. Rätsch (Eds.), *Advanced lectures on machine learning. ML 2003. Lecture notes in computer science* (Vol. 3176). Berlin, Heidelberg: Springer.

Repoulias, F., & Papadopoulos, E. (2007). Planar trajectory planning and tracking control design for underactuated AUVs. *Ocean Engineering*, *34*(11), 1650–1667. https://doi.org/10.1016/j.oceaneng.2006.11.007.

Roper, C. (2007). Using the Gavia Auv system to locate and document munitions dumped at sea. Online.

Rout, R., & Subudhi, B. (2017). Narmax self-tuning controller for line-of-sight-based waypoint tracking for an autonomous underwater vehicle. *IEEE Transactions on Control Systems Technology*, *25*(4), 1529–1536. https://doi.org/10.1109/TCST.2016.2613969.

Saravanakumar, S., & Asokan, T. (2011). Waypoint guidance based planar path following and obstacle avoidance of autonomous underwater vehicle. In *ICINCO (2)* (pp. 191–198).

Shen, H., & Guo, C. (2016). Path-following control of underactuated ships using actor-critic reinforcement learning with MLP neural networks. In *2016 6th international conference on information science and technology (ICIST)* (pp. 317–321). https://doi.org/10.1109/ICIST.2016.7483431

Sutton, R. S., & Barto, A. G. (2011). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.

Thompson, F., Galeazzi, R., & Guihen, D. (2020). Field trials of an energy-aware mission planner implemented on an autonomous surface vehicle. *Journal of Field Robotics*,. https://doi.org/10.1002/rob.21942.

Wang, J., Wang, C., Wei, Y., & Zhang, C. (2018). Three-dimensional path following of an underactuated auv based on neuro-adaptive command filtered backstepping control. *IEEE Access*,. https://doi.org/10.1109/ACCESS.2018.2883081.

Wang, S., Shen, Y., Sha, Q., Li, G., Jiang, J., Wan, J., Yan, T., & He, B. (2017). Nonlinear path following of autonomous underwater vehicle considering uncertainty. In *2017 IEEE Underwater Technology (UT)* (pp. 1–4). https://doi.org/10.1109/UT.2017.7890302.

Watkins, C. J. C. H. (1989). Learning from delayed rewards. Ph.D thesis, King's College, Cambridge.

Xiang, X., Lapierre, L., Liu, C., & Jouvencel, B. (2011). Path tracking: Combined path following and trajectory tracking for autonomous underwater vehicles. In *2011 IEEE/RSJ international conference on intelligent robots and systems* (pp. 3558–3563). https://doi.org/10.1109/IROS.2011.6094949.

Xiang, X., Lapierre, L., & Jouvencel, B. (2015). Smooth transition of auv motion control: From fully-actuated to under-actuated configuration. *Robotics and Autonomous Systems*, *67*, 14–22. https://doi.org/10.1016/j.robot.2014.09.024. (advances in Autonomous Underwater Robotics).

Xiang, X., Yu, C., & Zhang, Q. (2017). Robust fuzzy 3d path following for autonomous underwater vehicle subject to uncertainties. *Computers and Operations Research*, *84*, 165–177. https://doi.org/10.1016/j.cor.2016.09.017.

Yang, X., Shen, Y., Wang, K., Sha, Q., He, B., & Yan, T. (2016). Path following for an autonomous underwater vehicle using gp-los. In *OCEANS 2016-Shanghai* (pp. 1–5). IEEE.

Yu, C., Xiang, X., & Dai, J. (2016). 3d path following for under-actuated auv via nonlinear fuzzy controller. In *OCEANS 2016 - Shanghai* (pp 1–7). https://doi.org/10.1109/OCEANSAP.2016.7485727.

Yu, R., Shi, Z., Huang, C., Li, T., & Ma, Q. (2017). Deep reinforcement learning based optimal trajectory tracking control of autonomous underwater vehicle. In *2017 36th Chinese Control Conference (CCC)* (pp. 4958–4965). IEEE.

Wilmer Ariza Ramirez received the B.S. degree in mechatronic engineering from Universidad San Buenaventura, Bogota, Colombia, in 2009 and the M.Eng. degree in advances manufacturing technology from Swinburne University of Technology, Melbourne, Australia, in 2013. He is currently pursuing the Ph.D. degree in Maritime Engineering at University of Tasmania, Australian maritime College, Launceston, Australia His research interests include simultaneous localization and mapping, navigation, path planning, intelligent systems, and nonlinear control with the use of machine learning algorithms for underwater vehicles.

Zhi Quan Leong received the BEng in Ocean Engineering and Ph.D. degree in maritime engineering from the Australian Maritime College, University of Tasmania, Launceston, Australia. He is currently a Research Fellow (DSTG) at the Australian Maritime College, and Lecture of Hydrodynamics, Computational Fluid Dynamics at the university of Tasmania, Launceston, Australia. He is currently involved in an international research team led by Australia's Defence Science and Technology Group to refine the methods that will be used to test the hydrodynamic performance of underwater vehicles. His research interests include Ship and Platform Hydrodynamics, Naval Architecture, Marine Engineering, Ocean Engineering, Special Vehicles, Maritime Engineering, Navigation and Position Fixing.

Hung Duc Nguyen received the bachelor's degree in ship navigation engineering from the Vietnam Maritime University, Haiphong, Vietnam, and the master's and Ph.D. degrees in marine systems engineering from the Tokyo University of Marine Science and Technology, Tokyo, Japan. He is currently a Lecturer of marine control engineering with the National Centre for Maritime Engineering and Hydrodynamics, Australian Maritime College, University of Tasmania, Launceston, TAS, Australia. His current research interests include guidance, navigation and control of marine vehicles and offshore structures, ship maneuvering dynamics and manoeuvrability, GPS/GNSS and inertial navigation systems, linear and non-linear control systems, modeling and recursive/stochastic estimation of marine and offshore control systems, fault detection and monitoring based on modeling and estimation, instrumentation and process control, sea-keeping and roll stabilization systems, and marine robots and control applications in marine vehicles including underwater vehicles

Shantha Gamini Jayasinghe received his B.Sc. degree in Electronics and Telecommunication Engineering from the University of Moratuwa, Sri Lanka, in 2003, and Ph.D. Degree in Electrical Engineering from the Nanyang Technological University, Singapore, in 2013. From 2011 to 2015 he worked as an Electrical Systems Engineer at the Rolls Royce Advanced Technology Centre in Singapore. He is currently a Senior Lecturer in Maritime Electrical Engineering at Australian Maritime Collage, University of Tasmania, Australia. He has authored or co-authored over 100 international journal and conference papers, two books and two book chapters. He holds eight U.S./U.K. patents. His research interests include power electronic converters, renewable energy technologies, grid integration of energy systems, shipboard power systems, electric propulsion and control systems.