PAPER • **OPEN ACCESS**

Controller parameter tuning using actor-critic algorithm

To cite this article: Ayman Elshabrawy M Ahmed 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **610** 012054

IOP Conf. Series: Materials Science and Engineering **610** (2019) 012054 doi:10.1088/1757-899X/610/1/012054

Controller parameter tuning using actor-critic algorithm

Ayman Elshabrawy M Ahmed[1]

[1] Research Center, - Egyptian Armed Forces, Egypt

Email: a.shabrawy45@gmail.com

Abstract. In this paper a new reinforcement learning strategy is used for on-line tuning the control system of the aerodynamic missile. Aerodynamics missile automatic control system's mission is to overcome the missile's flight various disturbances encountered in the process of precise and real-time control of missiles attitude. Reinforcement learning algorithm (RL) is used to tune a PID controller to replace "gain schedule" Technique usually used. The result shows that RL with the new reward function is able to optimize the PID parameters with advantage over old method in terms of convergence speed and smaller overshoot

Keywords: Reinforcement learning; Gain schedule; , Missile control

1. Introduction

Due to its simplicity, reliability, and the clear relationship between its parameters and the system response specifications, the conventional PID control is still the most popular design approach used in the field of real-time control, even with the presence of the modern good complicated control scheme, for instance, adaptive control, neural control, fuzzy control, etc. It is well known that a well-tuned PID controller is able to achieve an excellent performance. However, it suffers the main disadvantage of resulting in a poor performance whenever the plants are subjected to some kind of disturbance, or when the plants have a high-order, non-linear structure. In flight control systems, a class of PID controllers which uses the gain scheduling method are widely used. This method uses flight height, speed, or attack angle as schedule variables, to interpolate in a pre-given gain scheduling table, to ensure behavior requirements in different flight conditions. However, the establishment of a gain scheduling table is a complicated task, especially if the missiles have wide flight scope and high maneuverability, or the dynamics have many uncertain parameters, as in the case of a large missile. In this paper, we deal with the control problem of missile systems with input unmodeled dynamic in pitch channel.

Reinforcement learning controllers are bio-inspired and based on the idea of learning from experience coupled with the principle of reward and punishment borrowed from living things (human and animal)[9]. Contradictory to supervised learning that is normally used in neural network, Reinforcement learning uses, unsupervised learning based on the trial and error routine since it is a directed learning technique based on interaction with the environment. RL framework can be seen in Figure 1. It consists of system Environment and control Agent. In the environment, there is a certain policy which produces a certain state and associated reward for each action. The agent receives a scalar "reward" from the environment, which gives the agent an indication of the quality of that action. The main goal of the agent is to maximize the total accumulated reward, also called the return. By following a given policy and processing the rewards, the agent can build estimates of the return. The function

representing this estimated return is known as the value function. By using this past experience, the agent decides which future action to take to increase the reward.

Several other types of RL Method have been also presented. They can be divided into two main groups, on-line learning and off-line learning. In this research, an Actor-Critic method is used which is an online RL technique[9]. The advantage of using the Actor-Critic method is to compute continuous actions without the need for optimization procedures on a value function. The critic's importance is the estimate of the expected return allows for the actor to update with gradients that have lower variance, speeding up the learning process. This will be discussed in more detail in Section 3. This paper is arranged as follows: Section 2 states the model of the aerodynamic missile and the controller goal; Section 3 briefly explains the basics of reinforcement learning and describes the actor-critic controller structure and some mathematic fundamentals and presents the design and numerical experiment for the adaptive autopilot; and section 5 concludes with the result.

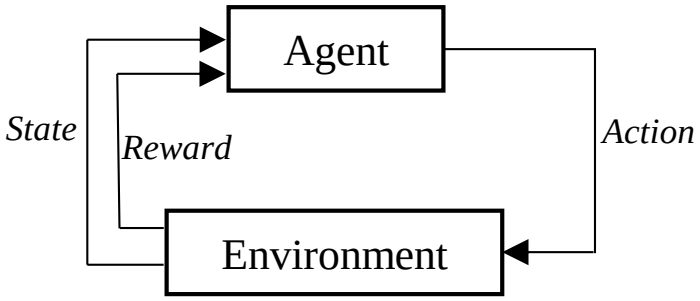

Figure 1. Reinforcement learning framework [9]

2. Modeling for aerodynamic Type missile

A block diagram of typical missile autopilot found in air-to-air and surface-to-air missiles is shown in Figure 2[2]. As the missile is symmetrical with respect to Y and Z axes, the pitch and yaw autopilots are the same. Skid-to-turn configuration is adopted for this analysis.

The pitch autopilot employs a rate feedback to damp the short-period oscillations. There are two paths shown for the rate feedback signal, one is for the boost phase (B), the other for the sustainer and coast phases (S). The design of autopilot in the sustain phase only will be discussed in this research since the dramatic change in the dynamics coefficient which we are interested in happens in this phase. As the accelerometer is not located at the missile center of gravity (Cg), the accelerometer will sense both the normal acceleration of the missile Cg (a_z) and the tangential acceleration due to a pitch angular acceleration ($\dot{\theta}$). The short period approximation for equation of motion becomes [2]

$$\left(\frac{mU}{Sq}s - C_{z_\alpha}\right)\acute{\alpha}(s) + \left(-\frac{mU}{Sq}s\right)\theta(s) = C_{z_{\delta_e}}\delta_e(s) \tag{1}$$

$$\left(C_{m_\alpha}\right)\acute{\alpha}(s) + \left(\frac{I_y}{Sqd}s^2\right)\theta(s) = C_{m_{\delta_e}}\delta_e(s) \tag{2}$$

where m is missile mass, d is missile diameter, S is missile cross-sectional area, α is angle of attack, δ_c is canard angle, θ is pitch angle, U is linear velocity in OX axis, u is change in linear velocity in OX axis, q is dynamic pressure, C_{z_α} is variation of Z force with angle of attack, C_w is gravity, Θ is angle between horizontal and OX axis measured in vertical plane, $C_{m_{\delta_c}}$ is change pitching moment due to change in canard angle, $C_{z_{\delta_c}}$ is change in force in Z direction due to change in canard angle, $C_{m_{\dot{\alpha}}}$ is downwash lag on moment created by wings , C_{m_α} is change in pitching moment due to a change in angle of attack, I_y is moment of inertia in OY axis and C_{m_q} is effect on pitching moment due to a pitch rate

Figure 2 is a simplified block diagram for Pitch autopilot,

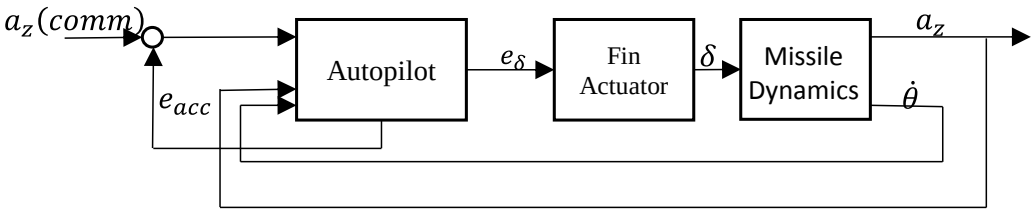

Figure 2. Pitch channel autopilot

The missile is a time variant system that continuously changes its parameter during its flight path. This is due to the change of physical properties and flight conditions. The physical properties of interest are the mass, moment of inertia and the center of gravity (Cg) location, which are functions of the fuel used. The flight conditions (altitude and velocity) determine the Mach number which is needed to calculate the missile stability derivative since it determines the center of pressure (Cp) and its relation to Cg. Table (1) shows the variation in the transfer function due to this change.

Table 1. Missile transfer function in different interval of time

t(sec)	T(f)
3	$\frac{\dot{\theta}(s)}{\delta_t(s)} = \frac{-106.47(s+0.418)}{s^2+0.644s+86.4}$
5	$\frac{\dot{\theta}(s)}{\delta_t(s)} = \frac{-279.61(s+0.775)}{s^2+0.95s+116.87}$
12	$\frac{\dot{\theta}(s)}{\delta_t(s)} = \frac{-369.4(s+0.94)}{s^2+1.098s+126.4}$
20	$\frac{\dot{\theta}(s)}{\delta_t(s)} = \frac{-469.6(s+1.2)}{s^2+1.27s+72.25}$
26	$\frac{\dot{\theta}(s)}{\delta_t(s)} = \frac{-247.7(s+0.64)}{s^2+0.764s+95.46}$
27	$\frac{\dot{\theta}(s)}{\delta_t(s)} = \frac{-224.75(s+0.603)}{s^2+0.726s+91.4}$

The classical method for designing an autopilot for such a system is to choose the most suitable operating point and set the controller gain based on it. However, this gain did not give the optimum performance in the entire trajectory. The most used solution for this is to have a different controller for every segment of trajectory, which is typically called “gain scheduling”. The other solution proposed in this research is to introduce a PID controller for the first interval and use a reinforcement learning algorithm to change the controller gain by learning the changes of model dynamics.

3. PID controller design concept

The RL module proposed is an actor-critic learning control architecture, which was early studied in [9][1] and had been effectively applied to several difficult problems[3][4][7]. In the RL module, there are three components which are, the actor network, the critic network and the reward function, as can be seen in figure 4.

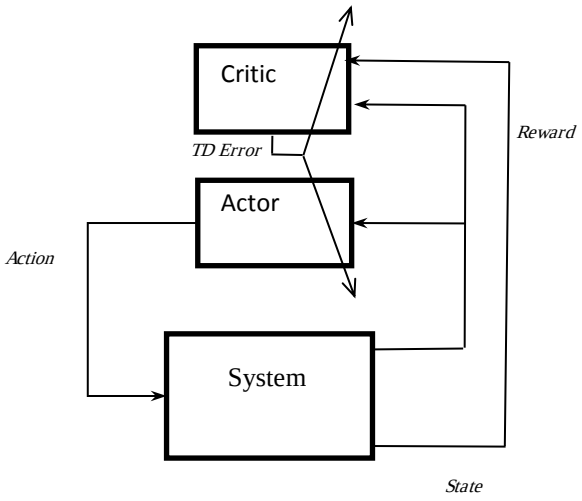

Figure3 Actor-critic framework

The system sends a reinforcement signal, which is essentially the same as the reward, to the critic indicating that the current policy is either correct or not in that particular state. The critic evaluates the policy using the temporal difference error "TD-error". This error is used to adjust both the critic and the actor. The update rule of the critic in an Actor-Critic is quite similar to the update rule of SARSA, although Actor-Critic uses the state value function instead of the action value function.

$$V_{t+1}(s_t) = V_t(s_t) + \alpha[r_t + \gamma V_{t+1}(s_t) - V_t(s_t)] \tag{3}$$

The update rule as presented in equation (3) is used to update the value function, which is found in the critic. A number of methods to update the actor exists. One method uses the output of the actor as input for the critic [9][9]. The total input of the critic consists of the state and the action. The update function of the critic is the same as the equation (3). The update function of the actor tries to minimize the error between the value function and the desired target U, as seen in equation (4).

$$\varepsilon_{ac} = V_t(s_{t+1}) - U \tag{4}$$

The expected reward to reach the target state is usually equal to 0. The update function of the actor, when function approximators are used, is shown in equation (5).

$$\Delta w = -\alpha(V_t(s_{t+1}) - U)\frac{\partial V_t(s_{t+1})}{\partial w} \tag{5}$$

The parameters of the networks are updated by using the derivative of the value function of the next state, whereas it is more common to use the derivative of the value function of the current state. For this paper, a Radial Basis Function (RBF) [7][4] neural network will be employed to model both the actor and the critic with one hidden layer. It has the characteristics of a simple structure, strong global approximation ability and a quick and easy training algorithm.

The structure of adaptive PID controller based on actor critic learning is shown in Figure (5). The design idea is an incremental PID controller [11] given by following equation,

$$u(t) = u(t-1) + \Delta u(t) = u(t-1) + K(t)X(t)$$
$$= u(t-1) + k_I(t)x_1(t) + k_P(t)x_2(t) + k_D(t)x_3(t) \quad (6)$$

where $X(t) = [x_1(t), x_2(t), x_3(t)] = [e(t), \Delta e(t), \Delta^2 e(t)]$

The error is defined as $e(t) = y_d(t) - y(t)$,

where $\Delta e(t) = e(t) - e(t-1)$ and $\Delta^2 e(t) = e(t) - 2e(t-1) + e(t-2)$

As shown In Fig. 4, $y_d(t)$ and $y(t)$ are the desired and the actual system outputs, respectively. The RBF configuration is shown in Figure (4). The Actor network is used to estimate a policy function and realizes the mapping from the current system state vector to the recommended PID parameters $K(t)$. The Critic network receives a system state vector and an external reinforcement signal (i.e., immediate reward) $r(t)$ from the environment and produces a TD error (i.e., internal reinforcement signal or Temporal different error) $\delta_{TD}(t)$ and an estimated value function $V(t)$.

The reward function has the following form,

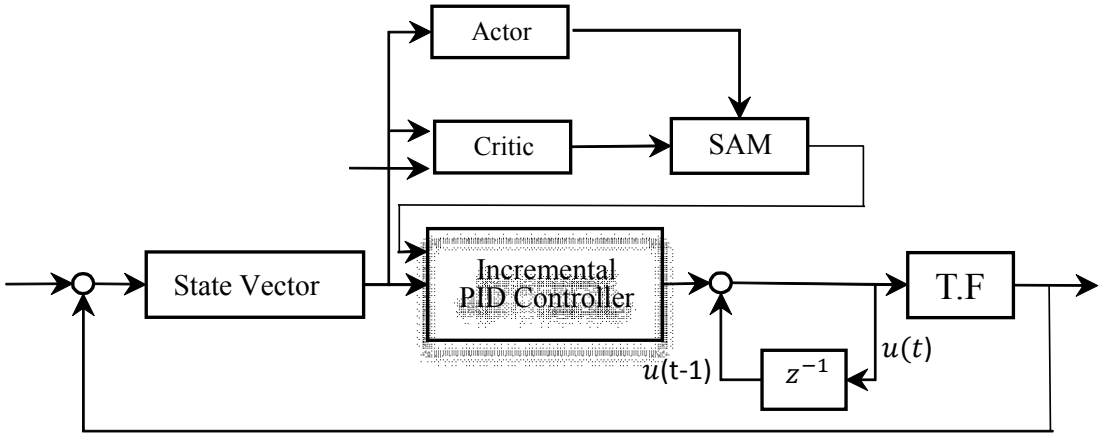

Figure 4. PID tuning based on RL

$$r(t) = \begin{cases} 0.5 \times |y - y_d| & |y - y_d| > 0.08 \\ 0.1 & 0.02 \le |y - y_d| \le 0.08 \\ 0 & |y - y_d| < 0.02 \end{cases} \quad (7)$$

As seen in the figure (5), we will be using only one network for both actor and critic, which means they will share the input layer and the hidden layer. The output layer, however, is different in both cases. The arrangement and meaning for each layer will be as follows,

Layer (1) input layer, in which there is one unit for each input. The input in this case is the $X(t)$ vector we discussed before. No process will be done in this layer, and the vector $X(t)$ will be transmitted directly to the next layer.

Layer (2) hidden layer, the function used to update the weight in this layer is a Gaussian function. The unit output of the hidden layer is,

$$\phi_j(t) = exp\left(-\frac{\|x(t) - \mu_j(t)\|^2}{2\sigma_j^2(t)}\right) \quad (8)$$

where $j = 1,2, \dots, h$, h is the number of hidden unit, μ and σ are the mean and standard deviation.

Layer 3, output layer, which has actor output and critic output.

The actor output $\bar{K}(t) = \left[\bar{k}_I, \bar{k}_P, \bar{k}_D\right]$ and can be calculated as

$$\bar{K}(t) = \sum_{j=1}^{h} w_{nj}(t)\phi_j(t)\, n = 1,2,3 \tag{9}$$

The critic output is the value function $V(t)$, which can be calculated as

$$V(t) = \sum_{j=1}^{h} v_j(t)\phi_j(t) \tag{10}$$

where $w_{nj}(t)$and $v_j(t)$ denote the weight between the hidden unit and actor and critic unit, respectively.

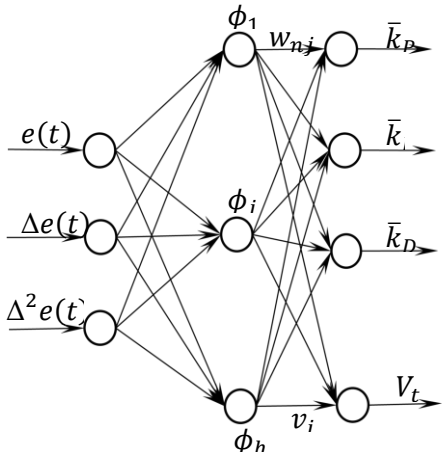

Figure 5. Actor-critic based On RBF network

To increase system efficiency, a stochastic action modifier (SAM) is used to increase exploitation [5][7]. The output from the Actor network is input to SAM and also the output of the value function for the critic network, the output of SAM, is used to modify the PID gain. SAM will be in the form of Gaussian noise function η_k, the output of which is

$$K(t) = \bar{K}(t) + \eta_k(0, \sigma_V(t)) \tag{11}$$

where $\sigma_V(t) = \frac{1}{1+\exp(2V(t))}$

As mentioned before, the output of the critic is defined by the equation (3). The output of the critic is used to calculate the prediction error (Temporal different error), which is defined as,

$$\delta_{TD} = r_t + \gamma V_{t+1}(s_t) - V_t(s_t) \tag{12}$$

The error of the critic, as defined in equation (12), is used for the objective function $\varepsilon_c(t)$, which is used to update the weights in the critic network,

$$\varepsilon_c(t) = \frac{1}{2}\delta_{TD}^2(t) \tag{13}$$

Since both actor and critic share the same network, the weights can be written as,

$$w_{nj}(t+1) = w_{nj} + \alpha_a \delta_{TD}(t)\phi_j(t)\frac{k_n(t) - \bar{k}_n(t)}{\sigma_V(t)} \tag{14}$$

$$v_j(t+1) = v_j(t) + \alpha_C \delta_{TD}(t)\phi_j(t) \tag{15}$$

where $\alpha_a$and $\alpha_C$are the learning rate for actor and critic

4. Results

In this section, we will present simulation results of the proposed adaptive critic autopilot applied to STT missiles. Figure 6 shows the step response for the different dynamics of the missile. The simulation show that online tuning capability shortens the learning time and is able to control the missile in different stage.

Table 2 Shows comparison of risetime and settling time for both classical PID and RL controller.

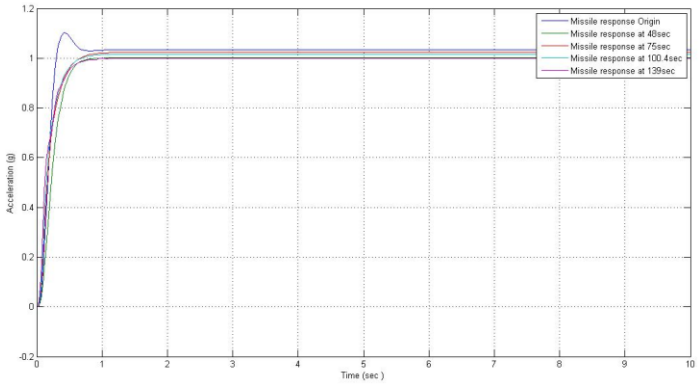

Figure 6 Step Response of STT missile with different dynamics

Table 2. Comparison between PID and RL controller

Case	t (sec)	Rise time		Settling time	
		Classical tech.	Modern tech.	Classical tech.	Modern tech.
1	3	0.109	0.416	6.490	0.742
2	5	0.394	0.444	2.200	1.140
3	12	0.238	0.441	1.240	0.673
4	20	0.030	0.143	0.313	1.280
5	26	0.348	0.429	2.290	0.903
6	27	0.428	0.442	2.930	0.865

Compare to response of the original gain schedule PID the response is acceptable. Moreover, the actuator needs less effort to achieve the required trajectory. The main advantage is to predict the dynamic change online and this increase the robustness of the proposed controller since it will depend on more accurate dynamics instead of simplified dynamics.

5. Conclusion

In this paper, a new adaptive critic autopilot has been proposed to control STT missiles. reinforcement learning approaches combined with robust adaptive control and Lyapunov theory, all parameters of PID can be online tuned with satisfactory tracking performance and guaranteed robust stability.

Unlike traditional control design, our autopilot for continuous flight scenario only requires a single proposed autopilot design as appose to many linear controllers in gain-scheduled autopilot design. Consequently, we can use only one autopilot through the entire flight process containing various flight conditions by adaptive control law and adaptive updating laws. Simulation results for the proposed autopilot applied to STT missiles demonstrate that the control objectives can be achieved effectively and successfully.

References

[1] A. G. Barto, R. S. Sutton, Anderson C W. "Neuronlike adaptive elements that can solve difficult learning control problems.",*IEEE Transactions on Systems*, *Man and Cybernetics*, 1983, 13(5): 834–846.
[2] J. H. Blakelock,"Automatic control of aircraft and missiles", Wiley, 1991.
[3] I. O. Bucak and M. Zohdy, "Reinforcement learning control of nonlinear multi-link system", Engineering Applications of Artificial Intelligence Journal, Vol. 14, 2001.
[4] I. O. Bucak M. A. Zohdy and M. Shillor, "Motion control of nonlinear spring by reinforcement learning," Control and Intelligent Systems, Vol.36, 2008.
[5] W. Cheng, J. YI and D. Zhao, "Application of actor-Critic Learning To Adaptive Sate space construction," Proceeding of the Third International Conference on Machine Learning and Cybernetics, Shanghai, 2004
[6] C.P.Mracek and J.R. Cloutier, "Full Envelope Missile Longitudinal Autopilot Design using the State-Dependent Riccati Equation Method", AIAA-97-3767, August 1997.
[7] D. Martinec and M. Bundzel, "Evolutionary Algorithms and Reinforcement Learning in Experiments with Slot Cars", International Conference on Process Control (PC), 2013, Slovakia.
[8] Jin-Sung Kim , JonghyunJeon , HoonHeo, "Design of adaptive PID for pitch control of large wind turbine generator", 10th International Conference on Environment and Electrical Engineering (EEEIC), 2011.
[9] R. S. Sutton, and A. G. Barto, "Reinforcement Learning: An Introduction", The MIT Press, 1998
[10] J. Si, and Y. T. Wang, "On-line learning by association and reinforcement," IEEE Transactions on Neural Networks", 2001, 12(2):264–276.
[11] X. WANG, Y. CHENG and W. SUN, "A Proposal of Adaptive PID Controller Based on Reinforcement" Learning J. China Univ. Mining & Technology 2007, 17(1): 0040–0044.