IIR Adaptive Wavelet Network with Reinforcement Learning

Iván S. Razo-Zapata, Luis E. Ramos-Velasco, and Julio Waissman-Vilanova

Research Center in Information Technologies and Systems,
Autonomous University of the State of Hidalgo,
Carr. Pachuca-Tulancingo, Km. 4.5, C.P. 42084, Pachuca,
Hidalgo, Mexico. {ri096373,julio,lramos}@uaeh.reduaeh.mx
http://www.uaeh.edu.mx/investigacion/sistemas/

Abstract. This paper presents a novel approach of reinforcement learning for continuous systems. The scheme is based in wavelet networks to approximating the continuous space of states. The structure of the wavelet network is dynamically generated accord to the explored regions and trained with a modified Q-Learning algorithm. The wavelet network include a IIR filter in order to make smooth controllers. This novel approach is called adaptive wavelet reinforcement learning control (AWRLC). Simulations of applying the proposed method to underactuated systems are performed to demonstrate the properties of the adaptive wavelet network controller.

1 Introduction

Reinforcement learning (RL) is learning to perform sequential decision tasks without explicit instructions, only optimizing a criterion about how the task is perform. So, the learner doesn't know which actions to take, but instead must discover which actions yield the most reward by trying them. This method, is goal-directed, and seems better adapted to the solution of a kind of control problems [1, 2], which ones about searching a final goal, and the problem is to find a policy that reach this goal [3].

The basic RL algorithms use a look-up table scheme in order to represent the value function $Q(s, a)$. Unfortunately this representation is limited when working with continuous spaces like physical systems. Several approaches can be applied to deal with this problems, like function approximation techniques. Neural networks offers an interesting perspective due to their ability to approximate nonlinear functions [4].

In recent years, wavelets have attracted much attention in many scientific and engineering research areas. Wavelets possess two features that make them especially valuable for data analysis: they reveal local properties of the data and they allow multiscale analysis. The local property is useful for applications that requires online response to changes, such a controlling process. Wavelets and neural networks have been combined [5, 6], to form a class of networks, so called

Computer Engineering.
Research in Computing Science 30, 2007, pp. 129- 138

wavelet networks, which are capable of handling moderately high-dimensional problems [4].

Inspired by the theory of multi-resolution analysis of wavelet transform and suitable adaptive control laws, an adaptive wavelet network is proposed for approximating action-value functions [7]. In this paper, we propose an adaptive wavelet reinforcement learning control (AWRLC) whose design is based on the promising function approximation capability of wavelet networks. The goal of the paper is to propose a control scheme based on RL algorithms and an AWRLC to control underactuated systems. We proposed a IIR filter in order to avoid bang-bang controllers. In this work the the *Pendubot* was used like example to evaluate the advantages and disadvantages of AWRLC methods for control of underactuated systems.

The work is organized as follows. Section 2 presents the reinforcement learning approach. In Section 3 is summariezed the background about wavelets networks, while Section 4 shows the control scheme which is implemented in the system. Section 5 gives the results obtained by numerical simulation. Finally, in Section 6 conclusions from results and future work are presented.

2 Reinforcement Learning

Q-Learning is a reinforcement learning method where the learner builds incrementally a Q-function which attempts to estimate the discounted future rewards for taking actions given states. the system is assumed as a Markov Decision Process (MDP) [3]. So, in a common control task maximize the total return R_t expressed in (1) is the main objective.

$$R_t = \sum_{k=0}^{T} \gamma^k r_{t+k+1} \tag{1}$$

Where R_t is the total return at state s_t and r_t is the reward value (numerical) when the system reach the state s_t.In this way, the output of the Q-function for state s_t and action a_t is denoted by $Q(s_t, a_t)$. When action a_t has been chosen and applied, the system is moved to a new state, s_{t+1}, and a reinforcement signal, r_{t+1}, is received, $Q(s_t, a_t)$ is updated by [3]:

$$Q(s_{t+1}, a_{t+1}) \longleftarrow Q(s_t, a_t) + \alpha\delta \tag{2}$$

where

$$\delta = r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)$$

$0 \leq \alpha \leq 1$ is the *learning rate*, and $0 \leq \gamma \leq 1$ is called the *discount*, this parameter is used to decrease r_{t+1} in the total return (1).

3 Wavelet Networks

Wavelets are a class of functions which have some interesting and special properties. These properties are localization in scale and time, compact support, multiresolution analysis among others. The original objective of the theory of wavelets is to construct orthogonal bases of $L_2(\mathbb{R})$. These bases are constituted by translations and dilations of the same function ψ called "mother wavelet" [8].

The structure of a wavelet network is a type of building block similar to a RBF network [6]. This building block allows the approximation of unknown functions by the concept of the multi-resolution approximation. The building block is formed by shifting and dilating the basis function ψ, (the modified version is its "daughter wavelet") and a "father wavelet" ϕ. Most commonly, wavelet bases are derived using shift-invariance and dyadic dilation. In this way we use the dyadic series expansion

$$\psi_{jk}(x) = 2^{j/2}\psi(2^j x - k), \quad j,k \in \mathbb{Z} \tag{3}$$

which is integral power of 2 for frequency partitioning. The daughter wavelet (3) is obtained from a mother wavelet function ψ by a binary dilation (i.e. dilation by 2^j) and a dyadic translation (of $k/2^j$).

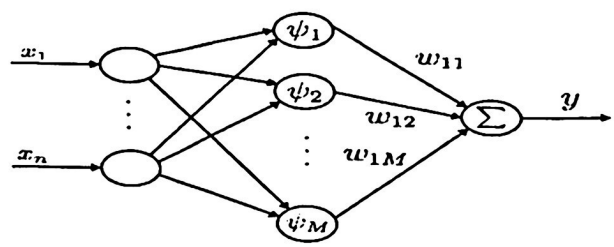

Fig. 1: Structure of wavelet network.

In this way combination of wavelet and neural networks can handle problems of large dimensions well and can make constructing network easily. The basic structure of a wavelet network is illustrated in Fig 1. The operation of each layer is summarized as follows [9]

- Using I_i^j and O_i^j to denote the input and output of the ith node in the jth layer, in first layer inputs are introduce into the network

$$O_i^1 = I_i^1 = x_i, \quad i = 1,2,\ldots,n$$

- Second layer consists of wavelet which one corresponding to pairs of (j,k) in (3), and the inputs and outputs of the wavelet nodes in this layer can be described as

$$\begin{cases} I_i^2 = [O_1^1,\ldots,O_n^1]^T \\ O_i^2 = 2^{j/2}\psi(2^j I_i^2 - k) \end{cases} \quad i = 1,2,\ldots,m$$

– Finally the input-output reltion in third layer is expressed with

$$y = O^3 = \sum_{i=1}^{m} w_{ij} O_i^2$$

4 Control Scheme

The control scheme is based in a wavelet network (W) trained with reinforcement learning. Accord with Fig. 1 the structure of the wavelet network is composed by three layers. First layer is the input layer, third layer is the output layer, and the second layer is the hidden layer with wavelet functions as activation functions. In order to deal with continuous actions our scheme presents an extra layer. This layer is an Infinite Impulse Response (IIR) synopsis network, which computes a continuos action accord to the outputs in the third layer (see Fig. 2a).

The main motivation for to implementing a wavelet network in this scheme is because artificial neural networks allow to approximate unknown functions, and with RL algorithms the idea is to approximate a Q-function. The wavelet network structure could be represent like a MISO system, with n input variables and only one output variable, like in Fig. 2a.

4.1 Building the Wavelet Network

The process of building the architecture of the wavelet network is performed on-line with the exploration through new states, due in part to the absence of data generated in past operations (training data). This growing of wavelet basis provide support to new states in order to approximate the Q-function with better accuracy.

New wavelet basis are generated with series expansion techniques, in this case was applied (3) with n-dimensions, with a tensor product [10–12].

$$\psi(X) = \psi(x_1, x_2, \ldots, x_n) = \prod_{j=1}^{n} \psi(x_j) \tag{4}$$

The mother wavelet implemented in this scheme is Mexican Hat defined as follows

$$\psi(x) = \frac{2}{\sqrt{3\sqrt{\pi}}}(1 - x^2)e^{\frac{-x^2}{2}} \tag{5}$$

The translation parameter in each one of the dimensions is determined by $k = x_i * 2^j$ where j is the scale of the wavelet basis and x_i is an element without support in the structure of W, when the learning process begins, the first neuron with support in W is created over the initial coordinates $x_1 = 0, x_2 = 0, ..., x_n = 0$.

In this way is possible initialize the learning with W empty, and W grows accord to the explored regions. The growing is controlled by the election of a threshold ξ which determines the minimum value in order to consider if a given state has support in the structure of the network. The diagram shown in Fig. 3 presents the algorithm for the construction of the wavelet network W.

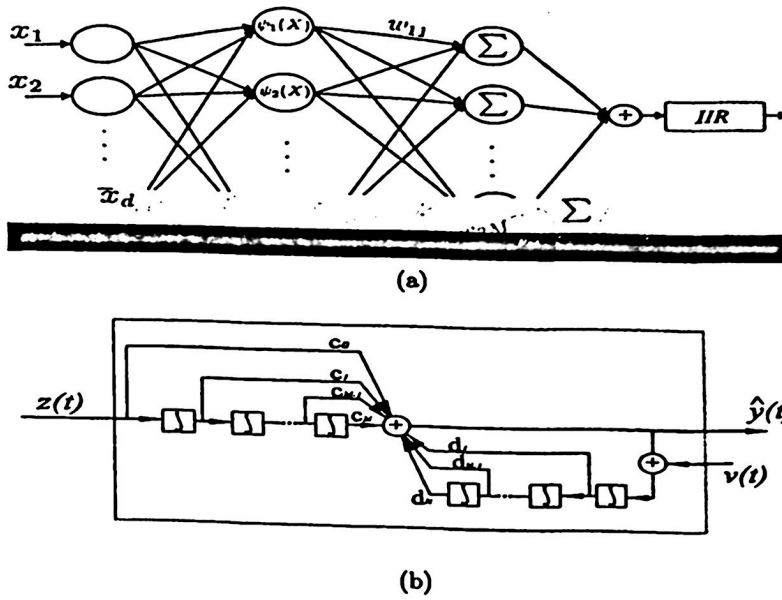

Fig. 2: *IIR Adaptive Wavelet Networks Structure: (a) Like a MISO system with n inputs and one output, in the second layer activation functions are wavelets; (b) IIR Model.*

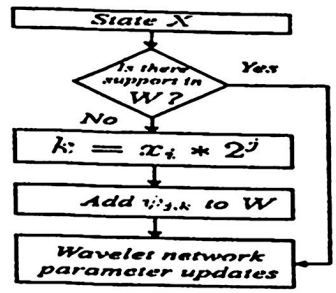

Fig. 3: Building the wavelet network.

4.2 Training the Wavelet Network

The training process in **W** consists in update the weights of layer 3. This training is performed on-line according to the interaction in the environment.

The updating rule is a combination between reinforcement learning and the gradient descent method. The reinforcement learning method implemented is *Q-Learning* due to its capacity of learning over the best action independently of the action selected. The updating rule applied in **W** is given by:

$$w_{i,j}(t+1) = w_{i,j}(t) + \Delta w, \quad j = 1, \ldots, m \tag{6}$$

$$\Delta w = \alpha \left[r_{t+1} + \gamma \max_{y_M} y_j(t+1) - y_j(t) \right] |\psi(X)|$$

where α is the learning rate, r_{t+1} is the reward at the time $t+1$, and γ is the discount with the same function that in (2).

The structure shown in Fig. 2a could be see like a MIMO system until layer 3 [10]. At the time t there is a vector $Y = [y_1, y_2, \ldots, y_m]^T$ with m outputs. So $y_i(t)$ is the output value of the neuron i in layer 3, where $i \in 1, 2, \ldots, m$. The value of i represents the selected action in the exploration process, treated in the next section.

In the training applied to neuronal networks, target values are required. And, in this scheme the target values is the maximum output given by **W** at the time $t+1$ in neurons of layer 3. In this way, the ap_ on of $Q(s,a)$ is achieved on-line without training data. In each iteration several neurons could be activated, but only the weights of neurons with value of activation bigger than the threshold $0 < \xi < 1$ are modified.

4.3 tion

Durin process, the network **W** allows to make elections between the se s). This selection can be applied with some kind of explo- ration)r $\epsilon-$ *greedy* [3]. The selection of actions is performed with the ou layer 3 of **W**, $(y_1, y_2, \ldots, y_m)$. Each output represents the appro: for the function $Q(s,a)$. And in a *greedy* exploration, the neuron with maximum value always is taken as control action.

4.4 Operation Process

After the learning process (*i.e* **W** is built and trained), the last layer in **W** computes continuos actions, which ones will be applied to the continuous system. This layer computes actions accord to the discrete actions learned for neurons in layer 3. In the last layer only one neuron is presented (Fig. 2a), and consist in a IIR filter.

5 Application

The control scheme AWRLC presented in Section 4 was applied to one under-actuated system (*Pendubot*) in order to control it in one equilibrium point. For the Pendubot the equilibrium position is the called UP-UP configuration (the first and the second link in vertical position like in Fig 4). The set of parameters applied in learning process with **W** is summarized in Table 1.

Table 1: Operation parameters

Parameter	Description	Value
n	Number of inputs.	4, $X = [x_1, x_2, x_3, x_4]^T$
m	Number of neurons (outputs) in second layer.	3, $Y = [y_1, y_2, y_3]^T$
$\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2$	Initial conditions	$X = [\pi, 0, \pi, 0]^T$
$A(s)$	Set of actions.	$-1, 0, +1$
α	Learning rate.	0.2
γ	Discount.	0.9
ϵ	Exploration $\epsilon-greedy$.	0.1
$\psi(x)$	Mother Wavelet.	Mexican Hat
j	Scale of wavelets.	3
ξ	Threshold.	0.2

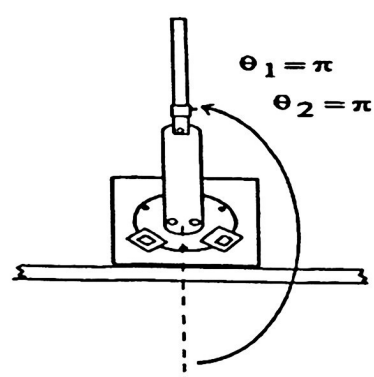

Fig. 4: Pendubot configuration UP-UP.

The control task in the system, is realized by **W** with a learning process based in RL. And the rewards for learning were assigned in continuous form,

according to $\theta_1, \dot{\theta_1}, \theta_2$ and $\dot{\theta_2}$. The desired value for θ_1 and θ_2 is around π (UP-UP Configuration). So rewards were given accord to the angular position of both links, this assignation was realized as follows:

- -3 When $\theta_1 \simeq \pi$, $\theta_2 \neq \pi$, $\dot{\theta_1} > 0$ and $\dot{\theta_2} > 0$
- -2 When $\theta_1 \simeq \pi$, $\theta_2 \simeq \pi$, $\dot{\theta_1} > 0$ and $\dot{\theta_2} > 0$
- -1 When $\theta_1 \simeq \pi$, $\theta_2 \simeq \pi$, $\dot{\theta_1} \simeq 0$ and $\dot{\theta_2} > 0$
- $+1$ When $\theta_1 \simeq \pi$, $\theta_2 \simeq \pi$, $\dot{\theta_1} \simeq 0$ and $\dot{\theta_2} \simeq 0$

The simulation was applied with a time step of 0.01s during 2000 episodes, with $j = 3$. A pruning process was applied in order to reduce the number of neurons. This pruning process is due to the growing of number of neurons in second layer related with the exploration property of RL algorithms, and some neurons doesn't have influence in the control task. So, 10 operation process were performed and the most important neurons were identified. Originally the network W has 2463 neurons, but with the pruning process the number was reduced to 131.

The reduced structure of AWRLC was applied to the system allowing to control the system in the desirable position, although to disturbances introduced to the torque of the first link.

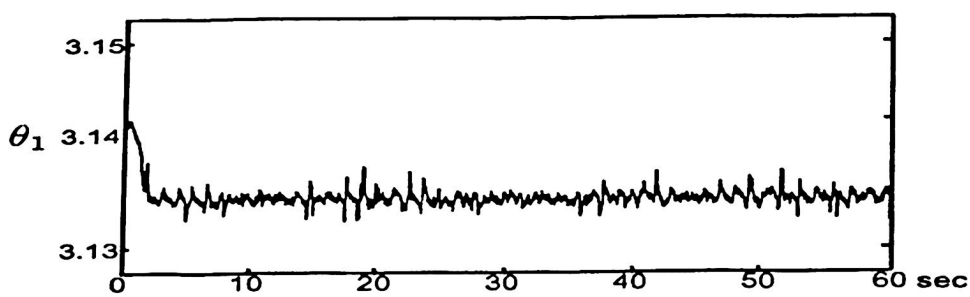

Fig. 5: θ_1, mass and large of both links like in training.

Fig. 6.a shows the behavior of the first link in 60 seconds of operation. Random disturbances were applied every 0.5 seconds of $+2$ and -2. In this plot the value of θ_1 is around to the desirable value in spite of the disturbances. In the same way Fig. 6.b presents the behavior of the second link during 60 seconds. It is possible to observe that the position is stable around an ideal value.

The main difference between the scheme presented in [10] and the actual AWRLC is the application of continuous actions. In [10] actions applied to the underactuated system are purely discrete actions. Fig. 7 presents the continuous actions applied to the system and disturbances introduced in 3 seconds of operation.

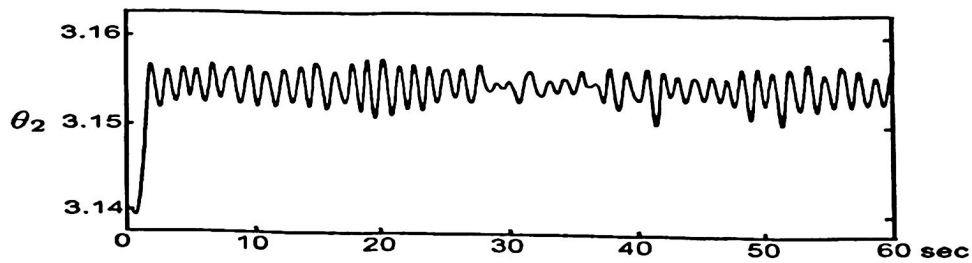

Fig. 6: θ_2, mass and large of both links like in training.

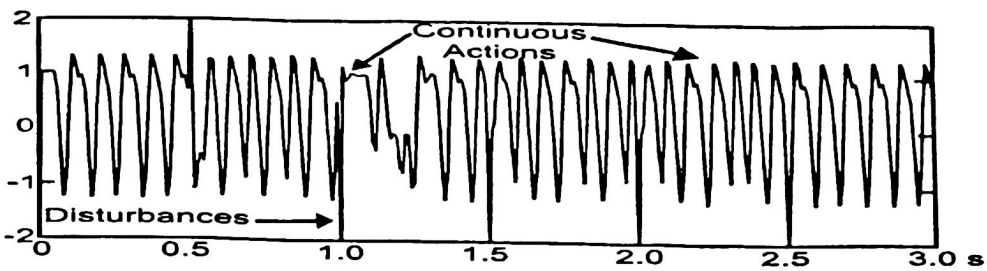

Fig. 7: Continuous actions and disturbances.

6 Conclusions and Future Work

6.1 Conclusions

In this work is presented the AWRLC scheme based in Reinforcement Learning algorithms, Wavelet Networks structure and IIR filter. AWRLC allow to deal with continuous spaces of states and actions. This represents an improvement with respect to the work presented in [10]. A simulation results of a *Pendubot* system was presented as illustrative example. The control of this system is specially difficult since is an underactuated mechanisms (two degrees of freedom and only one input).

Control scheme is based on learning methods, modifying one of the most popular RL algorithms: Q-Learning with adaptive wavelets networks. This approach uses a wavelet networks to approximate a Q-function, where the function gives the optimal control policy. Finally a IIR filter in order to avoid bang-bang controllers, which is applied to the underactuated system.

The simulation results show that this controller provides a good performance when keeping the systems in the unstable vertical positions. Results indicate that the AWRLC is a potentially attractive alternative for underactuated systems.

The algorithm for build wavelets networks in Fig. 3 represents an advantage working with physical systems with unknown limits of operation, because

this method of generating neurons create support in regions accord to the explorations. Multi-resolution is other attractive property handled by this kind of wavelet networks. The scales of resolution allow to approximate with good accuracy unknown functions, and in this case coarse approximations doesn't produces optimal control policies.

6.2 Future Work

The optimality over the policy built by AWRLC is one of the topics in further discussion. Besides, convergence and stability studies are in development actually.

References

1. I. S. Razo-Zapata, L.E. Ramos-Velasco and J. Waissman-Vilanova, Reinforcement Learning of Underactuated Systems, *in International Symposium on Robotics and Automation (ISRA 2006)*, San Miguel Regla, Hidalgo, México, 2006, pp. 420-424.
2. K. S. Fu, Learning Control Systems-Review and Outlook, *IEEE Transactions on Automatic Control*, vol. AC-15, 1970, pp. 210–221.
3. R. S. Sutton, A. G. Barto, *Reinforcement Learning An Introduction*, The MIT Press, 1998.
4. M. T. Hagan, H. B. Demuth and M. Beale, *Neural Network Design*, PWS Publishing Company, 1996.
5. Q. Zhang, and A. Beneveniste, Wavelet Networks, *IEEE Trans. Neural Networks*, vol. 3, 1992, pp. 889–898.
6. Q. Zhang, Using Wavelet Networks In Nonparametric Estimation, *IEEE Trans. Neural Networks*, vol. 8, 1997, pp. 227–236.
7. J. Xu and Y. Tan, *Nonlinear Adaptive Wavelet Control Using Constructive Wavelet Networks*, Proceedings of the American Control Conference, Arlington, VA, 2001.
8. S. Mallat, A Theory For Multiresolution Signal Decomposition: The Wavelet Representation, *IEEE Transactions Pattern Recognition and Machine Intelligence*, vol. 11, 1989, pp. 674–693.
9. W. Sun and Y. Wang and J. Mao, Using Wavelet Network for Identifying the Model of Robot Manipulator, *World Congress on Intelligent Control and Automation*, 2002, pp. 1634–1638.
10. I. S. Razo-Zapata, J. Waissman-Vilanova and L.E. Ramos-Velasco, Reinforcement Learning in Continuous Systems: Wavelet Networks Approach, Analysis and Desing of Intelligent Systems using Soft Computing Techniques, Series: Advances in Soft Computing , Vol. 41, Melin, P.; Castillo, O.; Ramrez, E.G.; Kacprzyk, J.; Pedrycz, W. (Eds.), 2007, XXI, 855 p. ISBN: 978-3-540-72431-5, Imprint: Springer-Verlang Brlin Heidelberg, 2007.
11. R. Wai and J. Chang, Intelligent Control of Induction Servo Motor Drive Via Wavelet Neural Network, *Electric Power Systems Research*, 2002, pp. 67–76.
12. J. Zhao, B. Chen and J. Shen, Multidimensional Non-Orthogonal Wavelet-Sigmoid Basis Function Neural Network for Dynamic Process Fault Diagnosis, *Computers and Chemical Engineering*, 1998, pp. 83-92.