Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Interconnection and Damping Assignment Control via Reinforcement Learning

S.P. Nageshrao * **G.A.D. Lopes** * **D. Jeltsema** ** **R. Babuška** *

* *Delft Center for Systems and Control (DCSC), Mekelweg 2*
** *Delft Institute of Applied Mathematics, Mekelweg 4,*
Delft University of Technology,2628 CD Delft, The Netherlands
e-mail: {S.P.Nageshrao, G.A.DelgadoLopes, D.Jeltsema, R.Babuska}@tudelft.nl

Abstract: Interconnection and Damping Assignment Passivity Based Control (IDA-PBC) is a nonlinear state-feedback control approach which can be used for stabilization and tracking control of a wide range of physical systems. Among many variants of this technique, algebraic IDA-PBC is by far the simplest method as it involves differential-algebraic equations rather than partial-differential equations. However, the equations are generally under-determined and the possible solutions for the unknown elements are non trivial. This issue becomes more evident for tasks on nonlinear systems, for example, swing-up and stabilization of a pendulum or stabilization of electro-mechanical systems. In this paper we mitigate some of the difficulties of algebraic IDA-PBC by using Reinforcement learning. Thanks to the robustness properties of learning the resulting learning based algorithm is insensitive to model uncertainties. We demonstrate the usefulness of the proposed learning algorithm both by simulations and through experimental validation.

Keywords: Passivity-based control, interconnection and damping assignment, nonlinear control, reinforcement learning, actor-critic scheme.

1. INTRODUCTION

Port-Hamiltonian (PH) theory is a well developed modeling framework to represent the dynamics of complex physical systems. As PH theory is derived from network based modelling of physical systems, it can be used to model multi-domain systems. Notable examples include mechanical, electrical, electro-mechanical, thermal systems, and their combinations. For in-depth analysis and examples, see (Duindam et al., 2009). A major advantage of the PH representation is that it highlights the relationship between various system characteristics, such as energy storage, dissipation, and interconnection. This emphasizes the suitability of energy-based methods for controlling PH systems (Ortega et al., 2001).

Passivity-based control (PBC), introduced by Ortega and Spong (1989), is a prominent energy-based control method for PH systems. PBC achieves the control objective, for example stabilization, by making the closed-loop passive in relation to a desired Hamiltonian. PBC for PH systems can be classified into three main categories: stabilization by damping-injection, energy-shaping and damping-injection (ES-DI), and interconnection and damping assignment IDA-PBC. The first, damping-injection, is the simplest approach but it has only limited applicability. Energy shaping and damping injection is widely used for stabilizing mechanical systems, but due to dissipation constraints ES-DI cannot be used to control all multi-domain system, for example electromechanical systems (Ortega et al., 2001). Additionally due to its inherent limitations ES-DI cannot be used for tracking control. The third approach, IDA-PBC, can be used to solve various control problems and on a wide range of physical systems. For example, by using IDA-PBC one can achieve regulation and tracking of multi-domain complex systems like magnetic levitation (Ortega et al., 2001).

In all the stated PBC methods, the control law that achieves the desired control objective is obtained by solving a set of partial differential equations. Most of the available literature on PBC for PH systems deals with simplification of the complex partial differential equations either for a particular example or for a restricted set of systems (Ortega and Garcia-Canseco, 2004). In general, solving the partial differential equations can be extremely difficult, thus hindering the use of PBC methods. Another disadvantage of model-based synthesis methods like PBC is their strong dependency on the system model (Duindam et al., 2009).

Fortunately, some alternative control methods exists that are (quasi-)independent of the system model. One such example is Reinforcement Learning (RL), a semi-supervised learning-based optimal-control method (Sutton and Barto, 1998). In RL, the controller (alternatively termed as 'pol-

icy' or 'agent' in RL literature) improves its behavior by continuously interacting with the system. For each interaction, the controller receives a numerical reward which is a function of the system's state transition and the control effort. The RL algorithm's objective is to maximize the total cumulative reward (Sutton and Barto, 1998; Busoniu et al., 2010). However, RL algorithms suffer from a few notable drawbacks, such as the black-box nature of the learned control law and slow convergence of the algorithm. Additionally, prior system knowledge that is often available for physical systems cannot be used in standard RL algorithms.

To address some of the issues of energy-shaping and damping-injection PBC and RL, a novel learning algorithm called *energy-balancing actor-critic* was developed by Sprangers et al. (2012). This method was extended by Nageshrao et al. (2014) to multi-input multi-output systems. However, the energy-balancing actor-critic algorithm can only be applied to a subset of physical systems, for example, fully-actuated mechanical systems. Additionally, tracking control problems cannot be solved by the algorithm due to the inherent limitations of energy-shaping and damping-injection PBC. The goal of this paper is to address some of the stated issues. We introduce a novel learning algorithm for algebraic IDA-PBC. The algorithm is evaluated in simulations and real-time experiments of the pendulum swing-up task. As demonstrated by Åström et al. (2008), the synthesis of a single smooth control law that achieves both swing-up and stabilization is rather involved. In this work we have considerably reduced the complexity of the design approach by using RL. One major advantage of IDA-PBC is its usability for controlling multi-domain systems. This is demonstrated in our second example: stabilization for the magnetic-levitation system.

This paper is organized as follows: Section 2 gives the theoretical background on PH systems and IDA-PBC. In Section 3, we provide a brief overview of reinforcement learning and introduce the RL based algorithm that solves the algebraic IDA-PBC synthesis problem. In Section 4, we evaluate the learning algorithm on the pendulum swing-up and stabilization task, and on the stabilization of a magnetic-levitation system. Finally, Section 5 concludes the paper.

2. IDA-PBC

2.1 Theoretical background

The dynamics of a time-invariant, input-affine, nonlinear system, can be represented as

$$\dot{x} = f(x) + g(x)u, \tag{1}$$

where $x \in \mathbb{R}^n$ is the state vector, function $f(x)\colon \mathbb{R}^n \to \mathbb{R}^n$ describes the system dynamics, and $g(x)\colon \mathbb{R}^m \to \mathbb{R}^n$ is the input function.

The objective of IDA-PBC is to find a state-feedback law $u = \beta(x)$ such that the resulting closed-loop is of the form (Ortega and Garcia-Canseco, 2004).

$$\dot{x} = F_\mathrm{d}(x)\nabla_x H_\mathrm{d}(x), \tag{2}$$

where ∇ denotes the gradient of a scalar function, $H_\mathrm{d}(x) \in \mathbb{R}$ is the desired closed-loop Hamiltonian, and $F_\mathrm{d}(x) \in \mathbb{R}^{n\times n}$ is the desired system matrix. This matrix can be separated into a skew-symmetric interconnection matrix $J_\mathrm{d}(x) \in \mathbb{R}^{n\times n}$ and a symmetric dissipation matrix $R_\mathrm{d}(x) \in \mathbb{R}^{n\times n}$. They satisfy the relation

$$F_\mathrm{d}(x) = J_\mathrm{d}(x) - R_\mathrm{d}(x). \tag{3}$$

The desired closed-loop Hamiltonian $H_\mathrm{d}(x)$ has a local minimum at the desired equilibrium $x_* \in \mathbb{R}^n$:

$$x_* = \arg\min H_\mathrm{d}(x). \tag{4}$$

Using the Moore-Penrose inverse of the input matrix $g(x)$, the control law $\beta(x)$ that achieves the desired closed-loop (2) is

$$\beta(x) = \big(g^T(x)g(x)\big)^{-1} g^T(x)\Big(F_\mathrm{d}(x)\nabla_x H_\mathrm{d}(x) - f(x)\Big). \tag{5}$$

The unknown elements of $F_\mathrm{d}(x)$ and $H_\mathrm{d}(x)$ can be obtained using the matching condition

$$g^\perp(x)\big(F_\mathrm{d}(x)\nabla_x H_\mathrm{d}(x) - f(x)\big) = 0, \tag{6}$$

where $g^\perp(x) \in \mathbb{R}^{(n-m)\times n}$ is the full-rank left annihilator matrix of $g(x)$, i.e., $g^\perp(x)g(x) = 0$.

Using the matching condition (6), one can obtain a maximum of $n - m$ free elements. However, the total number of free elements in $F_\mathrm{d}(x)$ and $H_\mathrm{d}(x)$ is larger. This issue is generally addressed either by constraining $F_\mathrm{d}(x)$, $H_\mathrm{d}(x)$ or both. Depending on the design choice, there are three main variants of IDA-PBC (Ortega and Garcia-Canseco, 2004):

- *Non-parameterized IDA-PBC* (Ortega et al., 2002). In this general form, the desired system matrix $F_\mathrm{d}(x)$ is fixed and the partial differential equation (6) is solved for the desired closed-loop Hamiltonian $H_\mathrm{d}(x)$. Among the admissible solutions, the one satisfying (4) is chosen.
- *Algebraic IDA-PBC* (Fujimoto and Sugie, 2001). The desired energy function $H_\mathrm{d}(x)$ is fixed, typically quadratic in increments (i.e., $H_\mathrm{d}((x - x_*)^2)$). This makes (6) an algebraic equation in unknown elements of $F_\mathrm{d}(x)$.
- *Parameterized IDA-PBC* (Ortega and Garcia-Canseco, 2004). Here the partial structure of the energy function $H_\mathrm{d}(x)$ is fixed. This imposes constraints on the unknown matrix $F_\mathrm{d}(x)$, which needs to be satisfied by the partial differential equation (6).

2.2 Algebraic IDA-PBC

In this section we explain algebraic IDA-PBC method using a fully actuated mechanical system as an example. We also highlight some of the difficulties encountered in using algebraic IDA-PBC. Consider a fully actuated mechanical system

$$\begin{bmatrix}\dot{q}\\ \dot{p}\end{bmatrix} = \begin{bmatrix}0 & I\\ -I & 0\end{bmatrix}\begin{bmatrix}\dfrac{\partial H}{\partial q}(x)\\ \dfrac{\partial H}{\partial p}(x)\end{bmatrix} + \begin{bmatrix}0\\ I\end{bmatrix}u, \tag{7}$$

where the state vector $x = [q^T\, p^T]^T$ consists of the generalized position $q \in \mathbb{R}^{\bar{n}}$ and generalized momentum $p \in \mathbb{R}^{\bar{n}}$, with $2\bar{n} = n$. The total energy or the system Hamiltonian $H(x)$ is given by the sum of the kinetic and potential energy:

$$H(x) = \frac{1}{2}p^T M^{-1}(q)p + V(q), \tag{8}$$

where the mass-inertia matrix $M(q) \in \mathbb{R}^{\bar{n}\times\bar{n}}$ is positive-definite. The potential energy term $V(q) \in \mathbb{R}$ is bounded from below.

In algebraic IDA-PBC, one can choose the desired closed-loop Hamiltonian to be quadratic in increments. Condition (4) at $x_* = [q_*^T\ 0]^T$ can be satisfied by choosing $H_\mathrm{d}(x)$ as

$$H_\mathrm{d}(x) = \frac{1}{2}p^T M^{-1}(q)p + \frac{1}{2}(q-q_*)^T\Lambda(q-q_*) \quad (9)$$

where $\Lambda \in \mathbb{R}^{\bar{n}\times\bar{n}}$ is a positive-definite scaling matrix.

For a generic system matrix $F_\mathrm{d}(x)$

$$F_\mathrm{d}(x) = \begin{bmatrix} F_{11}(x) & F_{12}(x) \\ F_{21}(x) & F_{22}(x) \end{bmatrix} \quad (10)$$

by using (7)–(10) in (6) we obtain the algebraic equation

$$F_{11}(x)\Lambda(q-q_*) + F_{12}(x)M^{-1}(q)p - M^{-1}(q)p = 0, \quad (11)$$

which can be trivially solved by choosing $F_{11}(x) = 0$ and $F_{12}(x) = I$. Similarly by substituting (7)–(10) in (5) the control law will be

$$u = \beta(x) = F_{21}(x)\Lambda(q-q_*) + F_{22}(x)M^{-1}(q)p + \frac{\partial H}{\partial q}, \quad (12)$$

where the unknown entries F_{21} and F_{22} need to chosen appropriately. For simple control problems, like stabilization of the mass-spring-damper, the choice of F_{21} and F_{22} is straightforward (Ortega et al., 2001). However, for more challenging control tasks such as the pendulum swing-up and stabilization, finding these parameters can be difficult.

In this work, rather than choosing the unknown elements F_{21} and F_{22}, we parameterize them by using linear-in-parameters function approximators

$$\beta(x) = \xi_1^T\phi(x)\Lambda(q-q_*) + \xi_2^T\phi(x)M^{-1}(q)p + \frac{\partial H}{\partial q}, \quad (13)$$

where $\xi = [\xi_1^T\ \xi_2^T]^T$ is the unknown parameter vector and $\phi(x)$ is a user-defined matrix of basis functions. In this work, we use fourier basis functions (Konidaris et al., 2008). The parameter vector ξ of (13) is learned using actor-critic RL. Prior to introducing the algorithm, we provide a brief overview of RL actor-critic methods.

3. REINFORCEMENT LEARNING

Reinforcement learning is a semi-supervised, stochastic, model-free optimal control method. The controller (in RL termed the learning agent) achieves the required optimal behaviour by constantly interacting with the system. In each interaction, the agent applies a control action $u_k = \pi(x_k)$, which is a general nonlinear state-feedback law. The control input u_k results in a system transition to a new state x_{k+1}. Along with the state transition, a numerical reward $r_k = \rho(x_k, u_k)$ is provided, which is often based on the error between the new state and the desired final goal x_*. In RL, the control objective is to maximize the long-term cumulative reward, called the return. The expected value of the return is represented by a value-function (Sutton and Barto, 1998)

$$V^\pi(x_k) = \sum_{i=0}^{\infty}\gamma^i\rho(x_{k+i+1}, \pi(x_{k+i})) = \sum_{i=0}^{\infty}\gamma^i r_{k+i+1}, \quad (14)$$

where k and i are time indices, $0 < \gamma < 1$ is the discount factor, and ρ is a user-defined, problem-specific reward function providing an instantaneous reward r.

Depending on whether the RL algorithm searches for a value function, for a control law or both, RL methods are broadly classified into three subcategories (Grondman et al., 2012):

- Actor-only: these methods directly search for an optimal control law.
- Critic-only: these methods learn an optimal value function. The control law is then obtained from the value function by one-step optimization.
- Actor-Critic (AC): these methods explicitly search for an optimal control law – the actor. Additionally, the critic learns a value-function and provides an evaluation of the controller's performance.

In the following section we provide a brief review of actor-critic methods.

3.1 Actor-Critic

Generally, RL methods are used for systems having discrete state-spaces and finite action-spaces. However, most physical systems have continuous state-spaces and the control law also needs to be continuous. This problem is often solved by using function approximation – for methods and examples, see Chapter 8 in (Sutton and Barto, 1998) and (Busoniu et al., 2010).

The AC method consists of two independent function-approximators (Grondman et al., 2012). The critic approximates the value-function (14) using the parameter vector $\theta \in \mathbb{R}^{n_\mathrm{c}}$ and a user defined basis function vector $\phi_\mathrm{c}(x) \in \mathbb{R}^{n_\mathrm{c}}$

$$\hat{V}^{\hat{\pi}}(x,\theta) = \theta^T\phi_\mathrm{c}(x). \quad (15)$$

Similarly, the actor approximates the policy by using the parameter vector $\vartheta \in \mathbb{R}^{n_\mathrm{a}}$

$$\hat{\pi}(x,\vartheta) = \vartheta^T\phi_\mathrm{a}(x), \quad (16)$$

where $\phi_\mathrm{a}(x) \in \mathbb{R}^{n_\mathrm{a}}$ is a vector of user-defined basis functions.

The reinforcement learning objective can be restated as follows: *Find an optimal policy $\hat{\pi}(x,\vartheta)$, such that for each state x, the discounted cumulative reward $\hat{V}^{\hat{\pi}}(x,\theta)$ is maximized.*

The unknown critic parameters are updated using the gradient-ascent rule

$$\theta_{k+1} = \theta_k + \alpha_\mathrm{c}\delta_{k+1}\nabla_\theta\hat{V}(x_k,\theta_k), \quad (17)$$

where α_c is the critic update rate, and δ_{k+1} is the temporal difference, obtained as (Sutton and Barto, 1998)

$$\delta_{k+1} = r_{k+1} + \gamma\hat{V}(x_{k+1},\theta_k) - \hat{V}(x_k,\theta_k). \quad (18)$$

The rate of parameter convergence can be increased by using the eligibility trace $e_k \in \mathbb{R}^{n_\mathrm{c}}$, yielding the following parameter update rule:

$$\begin{aligned} e_{k+1} &= \gamma\lambda e_k + \nabla_\theta\hat{V}(x_k,\theta_k), \\ \theta_{k+1} &= \theta_k + \alpha_\mathrm{c}\delta_{k+1}e_{k+1}, \end{aligned} \quad (19)$$

where $\lambda \in [0,1]$ is the trace decay rate.

Using a zero-mean white Gaussian noise Δu_k as an exploration term, the control input to the system is

$$u_k = \hat{\pi}(x_k,\vartheta_k) + \Delta u_k. \quad (20)$$

The policy parameters ϑ of (16) are increased in the direction of the exploration term Δu_k, if the resulting

temporal difference δ_{k+1} of (18) due to control input (20) is positive. Otherwise they are decreased. The actor parameter update rule in terms of the update rate α_{a} is

$$\vartheta_{k+1} = \vartheta_k + \alpha_{\mathrm{a}}\delta_{k+1}\Delta u_k \nabla_\vartheta \hat{\pi}(x_k, \vartheta_k). \tag{21}$$

Remark: Similarly to the critic, eligibility traces can also be used for the actor parameter update. For the sake of simplicity, this is not used in the current work.

3.2 AIDA-AC algorithm

The algebraic interconnection and damping assignment actor-critic algorithm (AIDA-AC) is constructed as follows. Consider the generic algebraic IDA control law in (5), parameterize the matrix F_{d} as $F_{\mathrm{d}}(x,\xi)$ to obtain

$$\hat{\pi}(x,\xi) = \left(g^T(x)g(x)\right)^{-1} g^T(x)\Big(\underbrace{\xi^T\phi(x)}_{F_{\mathrm{d}}(x,\xi)}\nabla_x H_{\mathrm{d}}(x) - f(x)\Big), \tag{22}$$

where ξ is the unknown parameter matrix. These parameters are updated by using the actor-critic scheme in Algorithm 1. A block diagram representation of the algebraic IDA learning algorithm is given in Fig. 1.

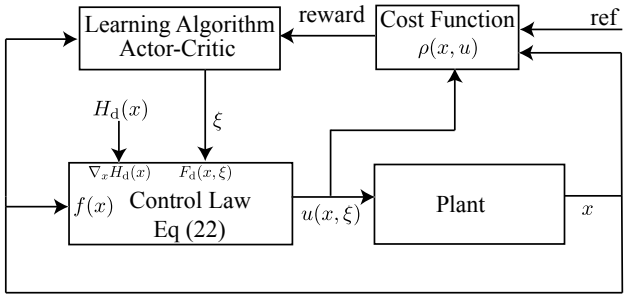

Fig. 1. Block diagram representation of AC algorithm for Algebraic IDA-PBC.

4. EXAMPLES

The AIDA-AC algorithm is evaluated for the pendulum swing-up and stabilization task and the stabilization of a magnetic-levitation system.

4.1 Pendulum swing-up and stabilization

Devising a single smooth energy-based control-law that can achieve both swing-up and stabilization of a pendulum is an arduous task, as explained by Åström et al. (2008). Here, we show that by using the learning method of Section 3.2 we are able to achieve swing-up and stabilization of a pendulum with low controller complexity. We use a laboratory setup shown in Fig. 2 along with a schematic drawing of the pendulum.

The system dynamics are

$$J_{\mathrm{p}}\ddot{q} = M_{\mathrm{p}} g l_{\mathrm{p}} \sin(q) - b\dot{q} + \frac{K_{\mathrm{p}}}{R_{\mathrm{p}}}u, \tag{23}$$

where q is the angular position. System (23) can be written in state-space form in terms of state vector $x = (q, p)$ where $p = J_p\dot{q}$ is the momentum, see (Sprangers et al., 2012). We adopt the same parameter values as in Table 1 of (Sprangers et al., 2012). The objective is to find a feedback control law $u = \beta(x)$ resulting in a closed-loop

$$\begin{bmatrix}\dot{q}\\ \dot{p}\end{bmatrix} = \begin{bmatrix}0 & 1\\ -F_{21}(x) & -b\end{bmatrix}\begin{bmatrix}\nabla_q H_{\mathrm{d}}(x)\\ \nabla_p H_{\mathrm{d}}(x)\end{bmatrix}. \tag{24}$$

The full system state is given by $x = [q\ p]^T$. The desired Hamiltonian is chosen to be quadratic in increments

$$H_{\mathrm{d}}(x) = \frac{1}{2}\gamma_{\mathrm{q}}(q - q_*)^2 + \frac{p^2}{2J_{\mathrm{p}}}, \tag{25}$$

where γ_{q} is a unit conversion factor and $H_{\mathrm{d}}(x)$ satisfies the desired equilibrium condition (4) at $x_* = (q_*, p) = (0, 0)$. Using (23)–(25) in (5) we get the control law as

$$\begin{aligned}\beta(x) &= -F_{21}(x)\gamma_{\mathrm{q}}(q - q_*) - M_{\mathrm{p}} g l_{\mathrm{p}} \sin(q)\\ &= -\xi^T\phi(x)\gamma_{\mathrm{q}}(q - q_*) - M_{\mathrm{p}} g l_{\mathrm{p}} \sin(q).\end{aligned} \tag{26}$$

The unknown vector ξ is learned using Algorithm 1 with the actor and critic learning rates given in Table 1. For other simulation parameters, see Table 2 of (Sprangers et al., 2012).

Algorithm 1 Algebraic IDA-PBC actor-critic algorithm

Input: System (1), λ, γ, $\alpha_{\mathrm{a}\xi}$ for actor, α_{c} for critic.
$e_0(x) = 0 \quad \forall x$
Initialize x_0, θ_0, ξ_0
for number of trails **do**
  $k \leftarrow 1$
  loop until number of samples
    Execute: Draw action using (22), apply the control input $u_k(x,\xi) = sat\left(\hat{\pi}(x_k, \xi_k) + \Delta u_k\right)$ to (1), observe next state x_{k+1} and reward $r_{k+1} = \rho(x_{k+1})$
    Temporal Difference:
      $\delta_{k+1} = r_{k+1} + \gamma\hat{V}(x_{k+1}, \theta_k) - \hat{V}(x_k, \theta_k)$
    Critic Update:
      for $i = 1, \ldots, n_{\mathrm{c}}$ **do**
        $e_{i,k+1} = \gamma\lambda e_{i,k} + \nabla_{\theta_{i,k}}\hat{V}(x_k, \theta_k)$
        $\theta_{i,k+1} = \theta_{i,k} + \alpha_{\mathrm{c}}\delta_{k+1}e_{i,k+1}(x)$
      end for
    Actor update:
      for $i = 1, \ldots, n_{\mathrm{a}}$ **do**
        $\xi_{i,k+1} = \xi_{i,k} + \alpha_{\mathrm{a}\xi}\delta_{k+1}\Delta u_k \nabla_{\xi_{i,k}} u_k(x,\xi)$
      end for
  end loop
end for

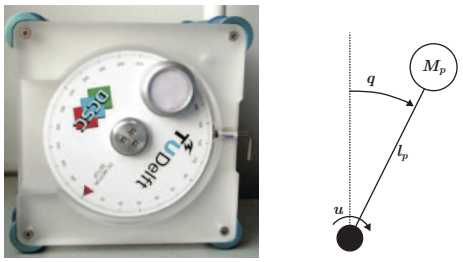

Fig. 2. Pendulum setup and schematic.

Table 1: Learning rates for the pendulum swing-up task

Learning rate	Symbol	Value [Units]
Learning rate critic	α_{c}	0.01 [-]
Learning rate $F_{21}(x)$	$\alpha_{\mathrm{a}\xi}$	1×10^{-8} [-]

Two controllers are learnt: one in simulation and the other on the physical system. Figure 3 illustrates the evaluation of the learned control laws in simulation and experiment.

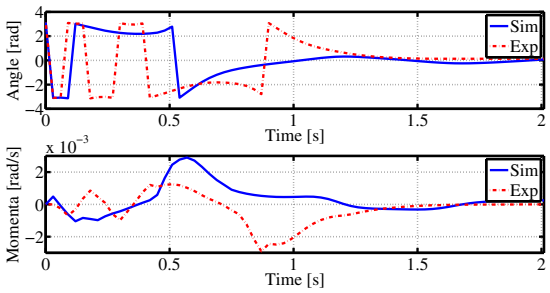

Fig. 3. Simulation and experimental result for pendulum.

Due to the limited actuation, the pendulum first builds up the required momentum by swinging back and forth. After sufficient energy is achieved, the controller is able to swing-up and stabilize the pendulum at the desired upright position.

4.2 Stabilization of magnetic-levitation system

The dynamics of the magnetic-levitation system (Hafner and Riedmiller, 2011), illustrated in Fig. 4, are

$$M\ddot{q} = Mg - \frac{e^2 C_1}{2\big(C_1 + L_0(C_2+q)\big)^2},$$

$$\dot{e} = -R\frac{e(C_2+q)}{C_1 + L_0(C_2+q)} + u, \tag{27}$$

where q is the position of the steel ball, and $e = L(q)i$ is the magnetic flux, a function of the current i through the coil and the varying-inductance $L(q)$ given by

$$L(q) = \frac{C_1}{C_2+q} + L_0. \tag{28}$$

The actuating signal is the voltage u across the coil. The system parameters are given in Table 2.

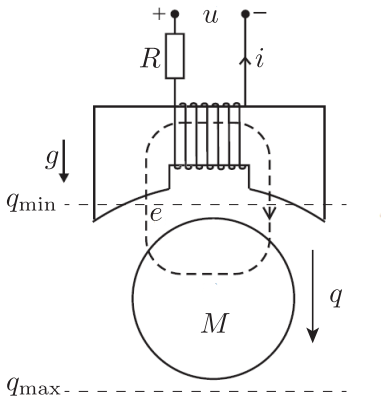

Fig. 4. Schematic of the magnetic-levitation system. Adopted from (Schaft, 2006).

Table 2: System parameters for magnetic levitation

Model parameters	Symbol	Value	Units
Mass of steel ball	M	0.8	kg
Electrical resistance	R	11.68	Ω
Coil parameter 1	C_1	1.6×10^{-3}	Hm
Coil parameter 2	C_2	7×10^{-3}	m
Nominal inductance	L_0	0.8052	H
Gravity	g	9.81	m/s^2

For this system we obtain a control law $u = \beta(x)$ with the resulting closed loop:

$$\begin{bmatrix}\dot{q}\\ \dot{p}\\ \dot{e}\end{bmatrix} = \begin{bmatrix}0 & 1 & 0\\ 1 & -F_{22}(x) & F_{23}(x)\\ 0 & -F_{23}(x) & -R\end{bmatrix}\begin{bmatrix}\frac{\partial H_\mathrm{d}}{\partial q}(x)\\ \frac{\partial H_\mathrm{d}}{\partial p}(x)\\ \frac{\partial H_\mathrm{d}}{\partial e}(x)\end{bmatrix}. \tag{29}$$

The system state is $x = [q\, p\, e]^T$, where $p = M\dot{q}$ is the momentum. The desired Hamiltonian is again chosen to be quadratic in increments:

$$H_\mathrm{d}(x) = \frac{1}{2}\gamma_q(q-q_*)^2 + \frac{p^2}{2M} + \frac{1}{2L_0}(e-e_*)^2, \tag{30}$$

with $e_* = \sqrt{2Mg/C_1}\big(C_1 + L_0(C_2+q_*)\big)$. The desired Hamiltonian $H_\mathrm{d}(x)$ satisfies the equilibrium condition (4) at $x_* = (q_*, 0, e_*)^T$. The control law using (5) is

$$\begin{aligned}\beta(x) &= -F_{23}(x)\frac{p}{M} - R\frac{(e-e_*)}{L_0} + R\frac{e(C_2+q)}{\big(C_1+L_0(C_2+q)\big)}\\ &= -\xi^T\phi(x)\frac{p}{M} - R\frac{(e-e_*)}{L_0} + R\frac{e(C_2+q)}{\big(C_1+L_0(C_2+q)\big)}\end{aligned} \tag{31}$$

The unknown parameter vector ξ of (31) is learnt using the AIDA-AC algorithm 1. It must be noted that we did not explicitly considered the matching condition so as to have a higher freedom in learning. The simulation parameters are given in Table 3.

Table 3: Learning parameters for magnetic levitation

Parameter	Symbol	Value	Units
Trials	–	100	-
Time per trial	T_t	2	s
Sample time	T_s	0.004	s
Decay rate	γ	0.95	-
Eligibility trace	λ	0.65	-
Learning rate critic	α_c	0.01	-
Learning rate $F_{23}(x)$	$\alpha_{\mathrm{a}\xi}$	1×10^{-7}	-

Due to physical constraints, the control input and system states are bounded, and their respective ranges are given in Table 4. A stabilizing control law without premagnetization was learned in simulation. The resulting learning curve and a sample simulation of the learnt control law are illustrated in Figures 5 and 6, respectively.

Table 4: Bounds on system states and input

System state	Symbol	Value	Units
Input voltage	$u_{\max}$	60	V
	$u_{\min}$	-60	V
Position	$q_{\max}$	13	mm
	$q_{\min}$	0	mm
Momentum	$p_{\max}$	3×10^{-1}	kg m/s
	$p_{\min}$	-3×10^{-1}	kg m/s
Magnetic flux	$e_{\max}$	3	Wb
	$e_{\min}$	-3	Wb

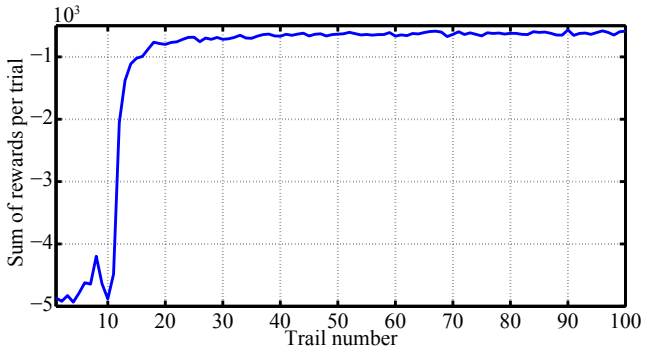

Fig. 5. Magnetic levitation learning curve.

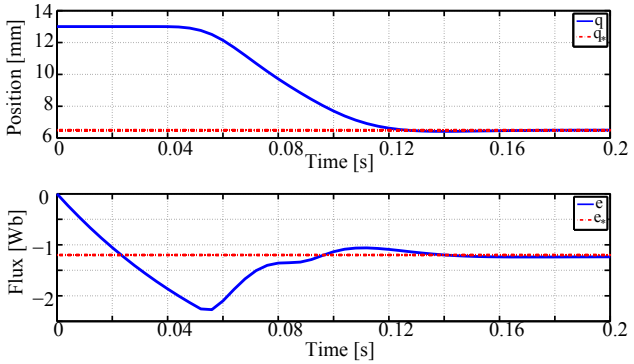

Fig. 6. Evaluation of learned control law for magnetic levitation.

Although there is an input, the steel ball stays in the rest position (i.e. $13mm$) till 0.05 seconds since this is the time required to magnetize the coil.

5. CONCLUSIONS

In this work we have presented a novel approach to obtain the parameters of the algebraic IDA-PBC control law. Actor-critic reinforcement learning algorithm is used to learn the parameters. Thanks to the PBC control law structure, the RL algorithm is augmented with prior information which improves the learning performance. We have observed that the learning algorithm proposed here is less sensitive to model and parameter uncertainties, and experimental results have shown faster learning times than the standard actor-critic.

There are numerous open points that are yet to be addressed. For example, the negative semi-definiteness of the derivative of the desired Hamiltonian is relaxed. The proof of convergence of the proposed learning algorithm is yet to be shown. Potential future work includes extending the RL based algorithm to parameterized and non-parameterized IDA-PBC, and extending the learning algorithms to address tracking control.

REFERENCES

Karl J Åström, Javier Aracil, and Francisco Gordillo. A family of smooth controllers for swinging up a pendulum. *Automatica*, 44(7):1841–1848, 2008.

Lucian Busoniu, Robert Babuska, Bart De Schutter, and Damien Ernst. *Reinforcement learning and dynamic programming using function approximators*. CRC Press, 2010.

Vincent Duindam, Alessandro Macchelli, and Stefano Stramigioli. *Modeling and control of complex physical systems*. Springer, Berlin Heidelberg, Germany, 2009. ISBN 978-3-642-03196-0.

Kenji Fujimoto and Toshiharu Sugie. Canonical transformation and stabilization of generalized Hamiltonian systems. *Systems & Control Letters*, 42(3):217–227, 2001.

Ivo Grondman, Lucian Busoniu, Gabriel AD Lopes, and Robert Babuska. A Survey of Actor-Critic Reinforcement Learning: Standard and Natural Policy Gradients. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(6):1291–1307, 2012.

Roland Hafner and Martin Riedmiller. Reinforcement learning in feedback control. *Machine learning*, 84(1-2):137–169, 2011.

George Konidaris, Sarah Osentoski, and PS Thomas. Value function approximation in reinforcement learning using the Fourier basis. *Computer Science Department Faculty Publication Series*, page 101, 2008.

Subramanya P. Nageshrao, Gabriel A. D. Lopes, Dimitri Jeltsema, and Robert Babuska. Passivity-based reinforcement learning control of a 2-dof manipulator arm. *Mechatronics*, 2014.

Romeo Ortega and Eloisa Garcia-Canseco. Interconnection and damping assignment passivity-based control: A survey. *European Journal of Control*, 10(5):432–450, 2004.

Romeo Ortega and Mark W Spong. Adaptive motion control of rigid robots: A tutorial. *Automatica*, 25(6): 877–888, 1989.

Romeo Ortega, Arjan J Van Der Schaft, Iven Mareels, and Bernhard Maschke. Putting energy back in control. *Control Systems, IEEE*, 21(2):18–33, 2001.

Romeo Ortega, Arjan Van Der Schaft, Bernhard Maschke, and Gerardo Escobar. Interconnection and damping assignment passivity-based control of port-controlled Hamiltonian systems. *Automatica*, 38(4):585–596, 2002.

Arjan Schaft. Port-Hamiltonian systems: an introductory survey. In *Proceedings of the international congress of mathematicians*, pages 1339–1365, Madrid, Spain, 2006. EMS Publishing House.

Olivier Sprangers, Gabriel A. D. Lopes, Subramanya P. Nageshrao, and Robert Babuska. Energy-balancing passivity-based control through reinforcement learning. *Submitted*, 2012.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. Cambridge Univ Press, 1998.