Adaptive and Learning Control of port-Hamiltonian Systems: A Survey

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

Abstract

Port-Hamiltonian (PH) theory is a novel, but well established modeling framework for nonlinear physical systems. Due to the emphasis on the physical structure and modular framework, PH modeling has become a prime focus in system theory. This has led to a considerable research interest in the control of PH systems, resulting in numerous nonlinear control techniques. These methods can be broadly categorized into model-based control methods and learning or model-free methods. Although various articles and monographs provide a detailed overview of model-based control techniques for PH systems, no survey is specifically dedicated to the learning control methods applied in the PH framework. To this end, we provide a comprehensive review of the current learning methodologies for PH systems. After establishing the required theoretical background, we elaborate on various machine learning, iterative learning, and adaptive control techniques for PH systems. For each method we provide motivation for learning in the PH framework, followed by a detailed presentation of the respective control algorithm. In general, the advantages of combining learning control with the PH framework are: i) by using learning algorithms the complexity of the PH control design process can be reduced; ii) novel design problems can be solved which would otherwise be intractable, for example optimal control of a PH system; iii) prior knowledge of the system, in the form of a PH model, can improve the learning speed. In this paper, we survey both model-based and learning techniques in the PH framework, we discuss and elaborate on the advantages and disadvantages of these methods. We conclude the paper with notes on open research issues.

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

** Delft Institute of Applied Mathematics, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands e-mail: D.Jeltsema@tudelft.nl
I. INTRODUCTION

Port-Hamiltonian (PH) modeling of physical systems [1]–[3] has found a wide acceptance and recognition in the control community. Thanks to the underlying principle of system modularity and the emphasis on physical structures and interconnections, the PH formulation can be efficiently used to model complex multi-domain physical systems [4]. The main advantage of the PH approach is that the Hamiltonian can be used as a basis to construct a candidate Lyapunov function, thus providing insight into numerous system properties like passivity, stability, finite $L_2$ gain [1], etc. These features have led to a profound research focus on the control of port-Hamiltonian systems. There are numerous interrelated control methods which have been extended or developed specifically for PH systems, namely canonical transformation [5], control by interconnection (CbI) [4], [6], [7], energy-balancing [6], [8], interconnection and damping assignment passivity-based control (IDA-PBC) [9], [10] (for detailed list see Table I of Section VII), etc.

In this article we collectively denote them as model-based synthesis methods. For an in-depth review of prominent model-based control synthesis methods refer to Chapter 5 of [4] and the references therein. All these methods rely on the PH model of the physical system and generally the controller is obtained by solving a set of complex mathematical equations, either partial differential equations or algebraic-differential equations.

Using the stated model-based synthesis methods one can acquire a detailed insight of the closed-loop system. However, for various practical applications obtaining a precise system model is extremely difficult [4], thus making the model-based control synthesis problem hard to solve. Control synthesis methods have been developed that are less or not at all dependent on the system model, such as reinforcement learning (RL) [11]. These are called model-free or learning methods. For a given system, a learning method achieves the desired control objective by changing or adapting its control law depending on the interactions with the system. Thanks to this approach, design objectives such as robustness against model uncertainties and/or parameter variations can be achieved. However, learning methods suffer from several notable drawbacks, such as: slow and non-monotonous convergence and the non-interpretability of the learned control law, often arising from the ‘learning from scratch’ mindset.

The absolute reliance on the system model in model-based methods and the lack of system knowledge in learning algorithms, places these two approaches at two extremes of a spectrum.
Adaptive control and model-based learning lie close to the model-based and model-free methods, respectively. In this paper we review methods that explicitly combine learning with PH models, that we consider to lie in the middle of the spectrum between pure learning and pure model-based synthesis. The advantages of incorporating learning into the PH framework are:

- Learning can avoid the need for solving complex mathematical equations (PDE’s) analytically.
- Performance criteria can be incorporated via learning. For example, optimal control problems [12] have not been addressed by using solely model-based synthesis methods.

From the point of view of learning, the advantages of incorporating PH models are:

- Prior system information in the form of a PH model can significantly improve the rate of convergence.
- The resulting control laws can be interpreted in the context of physical systems.

The combination of learning with PH models opens new avenues for solving complex control problems which otherwise would be intractable using either method in isolation.

In this paper, we provide a comprehensive overview of various learning control methods that have been developed specifically for PH systems. When applicable, we present a simple algorithmic (pseudo-code) representation of the learning method.

The paper is organized as follows. In Section II, we review the fundamentals of PH systems and passivity-based control. The need for adaptive and learning control is explained in Section III. Sections IV through VII describe various learning methods that have been introduced for the control of PH systems. Starting with adaptive methods in Section IV, iterative and repetitive control methods are elaborated on in Section V. We discuss the use of evolutionary strategies and reinforcement learning for PH control in Sections VI and VII, respectively. Section VIII provides a discussion with notes on opportunities for future work and Section IX concludes the paper.

II. PORT-HAMILTONIAN MODELING AND CONTROL

In this section, we provide a theoretical background on port-Hamiltonian modeling, and highlight prominent control methods for PH systems.
A. Port-Hamiltonian Framework

Port-Hamiltonian\(^1\) systems are often considered as a generalization of Euler-Lagrangian or Hamiltonian systems. PH modeling stems from the port-based network modeling of multi-domain complex physical systems having distinct energy storage elements (e.g., electrical, mechanical, electro-mechanical, chemical, hydrodynamical and thermodynamical systems). A strong aspect of the port-Hamiltonian formalism is that it emphasizes the physics of the system by highlighting the relationship between the energy storage, dissipation, and interconnection structures. Additionally, finite-dimensional PH theory can be readily extended to infinite-dimensional (distributed-parameter) systems [13].

A time-invariant PH system in the standard input-state-output form is given as

\[
\dot{x} = (J(x) - R(x)) \frac{\partial H}{\partial x}(x) + g(x)u, \quad x \in \mathbb{R}^n,
\]

\[
y = g^T(x) \frac{\partial H}{\partial x}(x),
\]

where \(J(x) = -J^T(x) \in \mathbb{R}^{n \times n}\) is the skew-symmetric interconnection matrix, \(R(x) = R^T(x) \in \mathbb{R}^{n \times n}\) is the symmetric dissipation matrix, and \(g(x) \in \mathbb{R}^{n \times m}\) is the input matrix. The Hamiltonian \(H(x) \in \mathbb{R}\) is the system’s total stored energy, obtained by adding the energy stored in all the individual energy-storing elements. Signals \(u \in \mathbb{R}^m\) and \(y \in \mathbb{R}^m\) are called the port variables and their inner-product forms the supply rate which indicates the power supplied to the system.

**Example 1. PH modelling of a mechanical system:** Some systems have a natural PH representation, for example, a fully actuated mechanical system is described by:

\[
\begin{bmatrix}
\dot{q} \\
\dot{p}
\end{bmatrix} = \begin{bmatrix}
0 & I \\
-I & 0
\end{bmatrix} \begin{bmatrix}
0 & 0 \\
0 & D
\end{bmatrix} \begin{bmatrix}
\frac{\partial H}{\partial q}(x) \\
\frac{\partial H}{\partial p}(x)
\end{bmatrix} + \begin{bmatrix}
0 \\
I
\end{bmatrix} u,
\]

\[
y = \begin{bmatrix}
0 & I
\end{bmatrix} \begin{bmatrix}
\frac{\partial H}{\partial q}(x) \\
\frac{\partial H}{\partial p}(x)
\end{bmatrix},
\]

where the generalized position \(q \in \mathbb{R}^n\) and the momentum \(p \in \mathbb{R}^n\) form the system state \(x = (q, p)^T\). The matrix \(D \in \mathbb{R}^{n \times n}\) represents the dissipation. The Hamiltonian \(H(x)\) is the sum of

\(^1\)The terminology used in the literature also includes terms like port-controlled Hamiltonian systems (PCH), port-controlled Hamiltonian systems with dissipation (PCHD), generalized Hamiltonian systems, etc.
the kinetic and potential energy,

$$H(x) = \frac{1}{2} p^T M^{-1}(q) p + V(q), \quad (3)$$

with $M(q) \in \mathbb{R}^{n \times n}$ the mass-inertia matrix and $V(q) \in \mathbb{R}$ the potential energy.

**Example 2. PH modelling of an electro-mechanical system:** The magnetic levitation system of Figure 1 consists of two subsystems, namely i) a mechanical system — iron ball of mass $M$, ii) an electro-magnetic system — a coil of nominal inductance $L_0$ and resistance $Z$.

![Fig. 1. Magnetic levitation of an iron ball [3].](image)

The dynamics of the magnetic-levitation system using first principles are [9]:

$$\dot{\phi} = u - Zi,$$

$$M\ddot{q} = F_{\text{emf}} - Mg, \quad (4)$$

where $u$ and $i$ are the voltage across and the current through the coil, respectively, $q$ is the position of the ball and $F_{\text{emf}}$ is the magnetic force acting on the ball. The effective magnetic flux $\phi$ linking the coil is a function of the position $q$, and it can be approximated as $\phi = L(q)i$, with the varying inductance

$$L(q) = \frac{L_0}{1 - q}. \quad (5)$$
Using (5), the effective force $F_{\text{emf}}$ on the iron ball is

$$F_{\text{emf}} = \frac{1}{2} \frac{\partial L(q)}{\partial q} I^2,$$

resulting in the Hamiltonian

$$H(x) = Mgq + \frac{p^2}{2M} + \frac{1}{2L_0} (1-q) \phi^2,$$

where $p = M \dot{q}$ is the momentum of the iron ball. Substituting (4)-(7) in (1), one can represent the system’s dynamic equations in the PH form as

$$\begin{bmatrix}
\dot{q} \\
\dot{p} \\
\dot{\phi}
\end{bmatrix} = \begin{bmatrix}
0 & 1 & 0 \\
-1 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix} \begin{bmatrix}
\frac{\partial H}{\partial q} \\
\frac{\partial H}{\partial p} \\
\frac{\partial H}{\partial \phi}
\end{bmatrix} + \begin{bmatrix}
0 \\
0 \\
1
\end{bmatrix} u,$$

$$y = \begin{bmatrix}
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
\frac{\partial H}{\partial q} \\
\frac{\partial H}{\partial p} \\
\frac{\partial H}{\partial \phi}
\end{bmatrix},$$

where

$$\begin{bmatrix}
\frac{\partial H}{\partial q} \\
\frac{\partial H}{\partial p} \\
\frac{\partial H}{\partial \phi}
\end{bmatrix} = \begin{bmatrix}
Mg - \frac{\phi^2}{2L_0} \\
\frac{p}{M} \\
\frac{(1-q)\phi}{L_0}
\end{bmatrix}.$$
\[ H(x); \]
\[ \dot{H}(x) = \frac{\partial H^T}{\partial x} \dot{x} \]
\[ = \frac{\partial H^T}{\partial x} (J(x) - R(x)) \frac{\partial H}{\partial x} + \frac{\partial H^T}{\partial x} g(x) u \]
\[ = -\frac{\partial H^T}{\partial x} R(x) \frac{\partial H}{\partial x} + u^T y, \]  \hspace{1cm} (10)

where \( u^T y \) is the supply rate. It is defined as the product of conjugate variables which equals the power supplied to the system, for instance, Voltage \( \times \) Current, Force \( \times \) Velocity, etc. For a semi-positive definite dissipation matrix (i.e. \( R(x) \geq 0 \)) the equality (10) reduces to
\[ \dot{H}(x) \leq u^T y. \]  \hspace{1cm} (11)

Equation (11) is called the dissipation inequality and it implies that in the presence of dissipation the change in the system’s total stored energy is less than or equal to the supply rate with the difference being the dissipated energy [14].

B. Control of PH systems

Passivity-Based control (PBC) is a model-based nonlinear control methodology that exploits the passivity property of a system to achieve various control objectives. In this section we provide a brief overview of prominent static state-feedback PBC methods. We omit a dynamic feedback method called Control by Interconnection (CbI) due to space constraints. The interested reader may consult [6] and the references therein.

1) Stabilization via damping injection: Asymptotic stability of a given PH system can be achieved using (10). Consider a negative feedback to the system as
\[ u = -K(x)y, \]  \hspace{1cm} (12)
with \( K(x) = K^T(x) \in \mathbb{R}^{m \times m} \) a positive definite damping injection matrix, to be designed by the user. Then the dissipation inequality (10) becomes
\[ \dot{H}(x) \leq -y^T K(x)y. \]  \hspace{1cm} (13)

By assuming zero state detectability, the asymptotic stability of the PH system (1) at the origin can be inferred [1]. Stabilizing the system at the origin which corresponds to the open loop minimum energy is not an enticing control problem. Alternatively a wider practical interest is
to stabilize the system at a desired equilibrium state $x_*$. In the PH framework this set-point regulation can be achieved by standard PBC — a combination of energy-shaping (ES) and damping-injection (DI) — as elaborated below.

2) Energy-Shaping and Damping-Injection (ES-DI): For a given port-Hamiltonian system (1), the ES-DI objective is to obtain a target closed-loop system [8]:

$$\dot{x} = \left(J(x) - R_d(x)\right) \frac{\partial H_d}{\partial x},$$  \hspace{1cm} (14)

where $R_d(x)$ is the desired dissipation matrix given as

$$R_d(x) = R(x) + g(x)K(x)g^T(x)$$ \hspace{1cm} (15)

in terms of the damping injection matrix $K(x)$. The desired closed-loop Hamiltonian $H_d(x)$ is obtained by adding an external energy component $H_a(x)$ to the Hamiltonian $H(x)$

$$H_d(x) = H(x) + H_a(x),$$ \hspace{1cm} (16)

such that it has a global minimum at the desired equilibrium $x_*$, i.e.,

$$x_* = \text{arg min} H_d(x).$$ \hspace{1cm} (17)

The desired closed-loop form of (14) can be obtained by using the control input$^2$

$$u = u_{es}(x) + u_{di}(x)$$

$$= (g^T(x)g(x))^{-1}g^T(x)(J(x) - R(x)) \frac{\partial H_a}{\partial x}(x)$$

$$- K(x)g^T(x) \frac{\partial H_d}{\partial x}(x),$$ \hspace{1cm} (18)

where the added energy term $H_a(x)$ is a solution of the set of PDE’s:

$$\begin{bmatrix} g^\perp(x)(J(x) - R(x))^T \\ g^T(x) \end{bmatrix} \frac{\partial H_a}{\partial x}(x) = 0,$$ \hspace{1cm} (19)

with $g^\perp(x)$ the full-rank left annihilator matrix of the input matrix $g(x)$, i.e., $g^\perp(x)g(x) = 0$. Among the solutions of (19) the one satisfying (17) is chosen. If the second part of the matching

$^2$Note that $g(x)$ is assumed to be full rank such that the matrix $g^T(x)g(x)$ is always invertible [15].
condition (19) is satisfied, then \( g^T(x) \frac{\partial H_d}{\partial x}(x) \) in (18) can be rewritten as

\[
g^T(x) \frac{\partial H_d}{\partial x}(x) = g^T(x) \left( \frac{\partial H}{\partial x}(x) + \frac{\partial H_a}{\partial x}(x) \right),
\]

\[
= g^T(x) \frac{\partial H}{\partial x}(x),
\]

\[
= y.
\]

A major drawback of the ES-DI approach is the dissipation obstacle\(^3\). This often limits the applicability of the method, whereas interconnection and damping assignment (IDA) PBC, explained below, does not suffer from this drawback.

3) Interconnection and Damping Assignment (IDA)-PBC: For a PH system (1) the IDA-PBC design objective is to obtain a closed-loop system of the form [9]:

\[
\dot{x} = (J_d(x) - R_d(x)) \frac{\partial H_d}{\partial x},
\]

(20)

where the desired interconnection and the damping matrices satisfy skew-symmetry and symmetric positive definiteness respectively, i.e.,

\[
J_d(x) = -J_d^T(x)
\]

\[
R_d(x) = R_d^T(x), \quad R_d(x) \geq 0.
\]

(21)

The closed-loop objective (20) can be achieved by the control input:

\[
u = \left( g^T(x)g(x) \right)^{-1} g^T(x) \left( (J_d(x) - R_d(x)) \frac{\partial H_d}{\partial x} - (J(x) - R(x)) \frac{\partial H}{\partial x} \right),
\]

(22)

where the desired Hamiltonian \( H_d(x) \) and system matrices \( J_d(x), R_d(x) \) are obtained by solving the matching condition

\[
g^\perp(J(x) - R(x)) \frac{\partial H}{\partial x} = g^\perp(J_d(x) - R_d(x)) \frac{\partial H_d}{\partial x}
\]

(23)

such that \( H_d(x) \) satisfies the desired equilibrium condition (17).

Prior to solving the matching condition (23), some facts about the choice of the system matrices of (20) need to be highlighted [10]:

- The desired interconnection matrix \( J_d(x) \) and the dissipation matrix \( R_d(x) \) can be freely chosen provided they satisfy skew-symmetry and positive semi-definiteness, respectively.

\(^3\)For definition and in-depth analysis see [8].
• The left-annihilator matrix $g^\perp(x)$ can be considered as an additional degree of freedom. Hence for a particular problem it can be appropriately chosen to reduce the complexity of the matching condition (23).

• The desired Hamiltonian $H_d(x)$ can be partially or completely fixed to satisfy the desired equilibrium condition (17).

Using the combination of the stated options there are three major approaches to solve the PDE of (23) [10]:

• **Non-parameterized IDA-PBC** — In this general form, first introduced in [9], the desired interconnection matrix $J_d(x)$ and the dissipation matrix $R_d(x)$ are fixed and the PDE (23) is solved for the energy function $H_d(x)$. Among the admissible solutions the one satisfying (17) is chosen.

• **Algebraic IDA-PBC** [5] — The desired energy function $H_d(x)$ is fixed thus making (23) an algebraic equation in terms of the unknown matrices $J_d(x)$ and $R_d(x)$.

• **Parameterized IDA-PBC** — Here, the structure of the energy function $H_d(x)$ is fixed. This imposes constraints on the unknown matrices $J_d(x)$ and $R_d(x)$, which need to be satisfied by the PDE (23) [10].

### III. Limitations of Model-Based Approach

Using the stated model-based synthesis methods one can acquire a detailed insight of the closed-loop system. However, model and parameter uncertainties can result in stability and performance issues as illustrated by the following example.

**Example 3. (Continued from Example 1)** Consider a fully actuated mechanical system (2). The standard ES-DI control input (18) that achieves the desired Hamiltonian $H_d(x) = \frac{1}{2} p^T M^{-1}(q)p + V_d(q)$ for the final potential energy $V_d(q) = \frac{1}{2} \bar{q}^T K_p \bar{q} = \frac{1}{2} (q - q_*)^T K_p (q - q_*)$ is

$$u = \frac{\partial V}{\partial q} - \frac{\partial V_d}{\partial q} - K_d y,$$

(24)

where $K_p$ is a proportional gain matrix and $K_d = K_d^T$ is the damping-injection matrix. These are chosen by the user. The control input (24) is same as a standard PD + gravity compensator to stabilize a system at $(q_*, 0)$. Note that (24) depends on the gradient of the system’s potential energy $V(q)$. In case of model or parameter uncertainty the gradient term will be inaccurate, hence resulting in non-zero steady state error [16].
Dynamical systems typically exhibit parameter variation during their operational life span. This variation generally requires fine tuning of the control parameters so as to achieve the desired control objective [17]–[19]. Robustness against parameter variation is often addressed either by robust or adaptive control methods. In robust control, the required behavior is achieved by making the closed-loop insensitive to parameter uncertainties, resulting in varying, but acceptable performance. In adaptive, the control parameters are modified online, resulting in the desired performance being always ensured, except during the transients. This makes adaptive control approaches more appealing when performance is of high importance, since in robust control, the tradeoff between performance and robustness is always present [20]. In Section IV, we discuss adaptive methods for port-Hamiltonian systems.

In many industrial applications, the system often executes the same task multiple times. For this repetitive operation, the idea of using previous time-history to improve the performance of the system has a rich literature spanning over three decades [21]. The general notion in these methods is to reduce in each repetition the cost function related to the tracking error. This is achieved by using the error information of the previous iteration to generate the control action in the current iteration [22]. Iterative control methods adapt the control signal directly, whereas in adaptive control the parameters of the control law are modified. The main objective of the iterative approach is to generate the desired output trajectory without using a priori system information. In [21] six postulates have been given in order to characterize an approach as iterative. If a given method satisfies all the six postulates (see Chapter 1.1 of [21]) then it is classified as iterative learning control (ILC). If the initial condition is different in each iteration then the iterative method is called repetitive control (RC). Using a uniform mathematical framework, similarity and differences between ILC and RC have been described in [23]. In Section V, we discuss ILC and RC methods for port-Hamiltonian systems.

The need to modify the controller parameters so as to satisfy a given performance criterion arises in various control applications, for e.g., optimal control. Iterative Feedback Tuning (IFT) [24] is one such prominent approach. IFT learns the optimal parameters by using the measured data from the system. The application of IFT for PH systems is discussed in Section V. Additionally there exist data-driven machine-learning techniques that can learn optimal control parameters for a given cost-function. Some prominent examples are Evolutionary Algorithms (EA) [25] and Reinforcement Learning (RL) [11]. The use of these methods for PH systems
will be discussed in Section VI and VII, respectively.

IV. ADAPTIVE CONTROL METHODS

The need to adapt the control parameters arises in many control applications. For example, there can be large variations in shape, size and weight of an object that is to be carried by a robotic manipulator arm. Similarly, there can be high fluctuation in the environmental parameters in process control applications. This requires online tuning, and adaptive control is one such prominent parameter tuning method [19]. Standard adaptive control framework consists of two components: a control input (25) and a parameter update law (26)

\[ u = \beta(x, \hat{\theta}) \]  
\[ \dot{\hat{\theta}} = \eta(x, \hat{\theta}, e) \]  

where \( e = (x - x^*) \) is the error between the desired and the actual state, \( \theta \) is the unknown parameter vector and \( \hat{\theta} \) is its estimate. Adaptive control methods are broadly classified as indirect and direct control methods [17], [18]. In indirect adaptive control, the unknown plant parameter vector \( \theta \) is estimated as \( \hat{\theta} \) and an appropriate controller is then devised. In direct adaptive control, \( \hat{\theta} \) is the estimate of an unknown parameter vector of the control law. The principle behind adaptive methods is to treat the estimate \( \hat{\theta} \) as the true value \( \theta \); this is called certainty equivalence principle [18], [19].

Since many multi-domain systems can be represented in the port-Hamiltonian form, continuous adaptation of PBC’s is desired. Let us introduce the adaptive control framework for port-Hamiltonian systems using a simple example.

Example 4. (Continued from Example 1 and Example 3) In order to compensate for model and parameter uncertainty, we can use an adaptive approach. The uncertain gradient term \( \frac{\partial V}{\partial q} \) of (24) is approximated as a product of an unknown parameter vector \( \theta \) and a known basis function matrix \( \phi(q) \) (called the regressor matrix in the adaptive control terminology) as

\[ \frac{\partial V}{\partial q} = \phi(x)\theta. \]  

The unknown parameter vector \( \theta \) is estimated as \( \hat{\theta} \), resulting in an approximate gradient function

\[ \frac{\partial \hat{V}}{\partial q} = \phi(q)\hat{\theta}. \]

12
As the closed-loop is required to be in the PH form, this enforces the parameter update law to be

\[ \dot{\hat{\theta}} = -Q\phi^T(q)y, \]  

(29)

where \( y \) is the system output of (2) and \( Q = Q^T \) is the update rate matrix. Combining (24) — (29), the closed-loop for (2) is

\[
\begin{bmatrix}
\dot{\hat{q}} \\
\dot{\hat{p}} \\
\dot{\hat{\theta}}
\end{bmatrix} =
\begin{bmatrix}
0 & I & 0 \\
-I & -(D + K_d) & \phi(x)Q \\
0 & -Q\phi^T(x) & 0
\end{bmatrix}
\begin{bmatrix}
\frac{\partial \hat{H}}{\partial \hat{q}} \\
\frac{\partial \hat{H}}{\partial \hat{p}} \\
\frac{\partial \hat{H}}{\partial \hat{\theta}}
\end{bmatrix},
\]  

(30)

where

\[ \hat{H} = \frac{1}{2}p^T M^{-1}(q) p + \frac{1}{2} q^T K_p \tilde{q} + \frac{1}{2} \tilde{\theta}^T Q^{-1} \tilde{\theta} \]  

(31)

is the closed-loop Hamiltonian, with \( \tilde{\theta} = \hat{\theta} - \theta \).

Thanks to its straightforward implementation, adaptive control was one of the first learning methods used in the PH framework [26]. The schematic representation of adaptive control for PH systems is illustrated in Figure 2.

The adaptive control law depends on the design objective for e.g., stabilization, tracking, etc. The parameter update rule is devised so as to ensure closed-loop remains in the PH form. Normally, a candidate Lyapunov function (31) is constructed with a minimum where both the desired control goal and zero error between the real model parameters and their estimates (\( \theta = \hat{\theta} \)) are attained. Adaptive control for port-Hamiltonian system has several advantages:

- As the closed-loop (Figure2) retains the PH structure various system properties like passivity and finite \( L_2 \) gain can be readily inferred;
• As the closed-loop (see (30)) is in the PH form it satisfies inequality (11), from which an upper bound on the error signal can be derived [27];
• By using the closed-loop Hamiltonian (31) and Barbalat’s lemma, uniform asymptotic stability of the equilibrium point \( x_\ast \) can be demonstrated. For proof see [28];
• Unlike prominent adaptive methods [20], [29] redefinition of the error signal is not required for adaptive control of PH systems [27].

Because of the above advantages the adaptive framework for PH systems has been used in various applications. Notable examples include power system stabilization [26], adaptive tracking control and disturbance rejection [16], [27], [28], simultaneous stabilization of a set of uncertain PH systems [30], [31], and stabilization of time-varying PH systems [32].

V. ITERATIVE AND REPETITIVE CONTROL METHODS

Iterative methods are based on the notion that, for a repetitive task the performance of the system can be improved by using the time-history of previous executions [22]. The basic working principle of the iterative method is to find an input trajectory \( u_d(t) \) such that it results in the prescribed system output \( y_d(t) \). This objective is achieved by an iterative law

\[ u_{i+1}(t) = u_i(t) + \gamma(e_i(t)) = u_i(t) + \gamma(e_i(t)) \]

where \( e_i(t) \) is the tracking error in the \( i^{th} \) iteration and \( \gamma \) is an algorithm specific update function [21]. Most of the available iterative methods either use D-Type or its modifications like PD-Type or PID-Type update function. The main advantage of using these standard update rules is that the resulting iterative method can learn a required input without using a priori system information. However, the standard methods suffer from non-asymptotic convergence of the tracking error. Using the adjoint output of a system, a mechanism to ensure monotonic error convergence for repetitive tasks, was proposed in [33]. Unlike the standard iterative methods the adjoint based iterative method requires the complete system information, thus hindering its usability.

In [34] the self-adjoint property was shown for controlled-Hamiltonian systems [35]. This means that the algorithm-specific update function \( \gamma \) can be obtained without prior system information while guaranteeing asymptotically convergence. Using this feature, various iterative methods for controlled-Hamiltonian systems were proposed in [12], [36], [37]. Prior to elaborating on
iterative methods and their applications, we give an example to illustrate the transformation of a
given PH system to a controlled-Hamiltonian form and provide a short overview on variational
and adjoint operators for controlled-Hamiltonian systems.

A. From PH to controlled-Hamiltonian

A controlled-Hamiltonian system — following the terminology of [35] — is described as:

\[ \dot{x} = (J-R) \frac{\partial H}{\partial x}(x,u), \]
\[ y = -\frac{\partial H}{\partial u}(x,u). \]  
(33)

Note that (33) differs from the PH system representation of (1) in the input \( u \), output \( y \) and
constant system matrices \( J \) and \( R \). Although the two types of systems are structurally different,
there is a subclass of systems that can be written in both forms, as illustrated by the following
example.

Example 5. Continued from Example 2

For the sake of simplicity, by using the resistance of \( Z = 1 \Omega \) and inductance \( L_0 = 1 \text{H} \), we
can rewrite (7) as the controlled-Hamiltonian in terms of input \( u \) and state \( x = (q,p,\phi)^T \)
\[ H(x,u) = Mgq + \frac{p^2}{2M} + \frac{1}{2}(1-q)\phi^2 - \frac{\phi}{Z}u. \]  
(34)

Note that the product \( \frac{\phi}{Z}u \) has the unit of energy. Equation (8) can be transformed to a controlled-
Hamiltonian form (33) as

\[
\begin{bmatrix}
\dot{q} \\
\dot{p} \\
\dot{\phi}
\end{bmatrix}
= \begin{bmatrix}
0 & 1 & 0 \\
-1 & 0 & 0 \\
0 & 0 & -1
\end{bmatrix}
\begin{bmatrix}
\frac{\partial H}{\partial q}(x,u) \\
\frac{\partial H}{\partial p}(x,u) \\
\frac{\partial H}{\partial \phi}(x,u)
\end{bmatrix}
\begin{bmatrix}
(J-R)
\end{bmatrix}.
\]  
(35)

Due to the structural differences, the transformation from PH to a controlled-Hamiltonian
system is not always possible. Hence, iterative methods that rely on the controlled-Hamiltonian
formulation (33) – explained later in this section – are only applicable to a limited set of PH
systems, such as fully actuated mechanical systems.

Remark 1: The transformation to the controlled-Hamiltonian form looses certain properties
of the original PH system, for example, (35) is not passive w.r.t. the supply rate \( u^Ty \).
B. Variational system

Consider a general nonlinear system operator $\Sigma$, acting on an input signal $u$ and resulting in a system output $y$

$$\Sigma(u) : \begin{cases} \dot{x} = f(x,u), & x \in \mathbb{R}^n, u \in \mathbb{R}^m \\ y = h(x,u) \end{cases}$$  \hspace{1cm} (36)

where $f$ is a system function and $h$ is an output function. One can linearize (36) along an input and system trajectory, $u(t)$ and $x(t)$, respectively, resulting in a linear time-variant (LTV) system [38]:

$$d\Sigma(u_v) : \begin{cases} \dot{x}_v = \left( J - R \right) \frac{\partial f(x,u)}{\partial x} x_v(t) + \frac{\partial f(x,u)}{\partial u} u_v(t), \\ y_v = \frac{\partial h(x,u)}{\partial x} x_v(t) + \frac{\partial h(x,u)}{\partial u} u_v(t), \end{cases}$$  \hspace{1cm} (37)

where $(x_v, u_v, y_v)$ are the variational state, input and outputs, respectively. They represent the variation along the trajectories $(x, u, y)$. For any controlled-Hamiltonian system (33) (or the subset of PH systems (1)) one can obtain the variational system using (37) [34]

$$d\Sigma(u_v) : \begin{cases} \dot{x}_v = (J - R) \frac{\partial H_v}{\partial x_v} (x, u, x_v, u_v), \\ y_v = -\frac{\partial H_v}{\partial u_v} (x, u, x_v, u_v), \end{cases}$$  \hspace{1cm} (38)

where $H_v(x, u, x_v, u_v)$ is the new controlled-Hamiltonian

$$H_v(x, u, x_v, u_v) = \frac{1}{2} \begin{bmatrix} x_v \\ u_v \end{bmatrix}^T \frac{\partial^2 H(x,u)}{\partial (x,u)^2} \begin{bmatrix} x_v \\ u_v \end{bmatrix},$$  \hspace{1cm} (39)

provided there exists a transformation matrix $T \in \mathbb{R}^{n \times n}$ that satisfies

$$J = -TJT^{-1}$$

$$R = TRT^{-1}$$

$$\frac{\partial^2 H(x,u)}{\partial (x,u)^2} = \begin{pmatrix} T & 0 \\ 0 & I \end{pmatrix} \frac{\partial^2 H(x,u)}{\partial (x,u)^2} \begin{pmatrix} T^{-1} & 0 \\ 0 & I \end{pmatrix},$$  \hspace{1cm} (40)

Unfortunately obtaining a transformation matrix $T$ so as to satisfy (40) is rather difficult. For a fully actuated mechanical system, a simple trick to circumvent this problem has been demonstrated using an internally stabilizing PD controller in [12].
C. Adjoint system

For a given LTV system
\[
\Sigma(u) : \begin{cases}
\dot{x} = A(t)x(t) + B(t)u(t), \\
y = C(t)x(t) + D(t)u(t),
\end{cases}
\]
(41)
the adjoint operator is
\[
\Sigma^*(u^*) : \begin{cases}
\dot{x}^* = -A^T(t)x^*(t) - C^T(t)u^*(t), \\
y^* = B^T(t)x^*(t) + D^T(t)u^*(t),
\end{cases}
\]
(42)
and it is related to the original system (41) by the inner-product [39]
\[
\langle y, \Sigma(u) \rangle = \langle \Sigma^*(y), u \rangle.
\]
(43)
The adjoint operator of a given system possesses various interesting properties, namely it can be used for model-order reduction [40], adjoint-based optimal control [33], etc.

Since the variational systems (37) and (38) are in LTV form one can obtain their respective adjoint forms. In [34], assuming invertibility of \(J - R\), the adjoint of the controlled-Hamiltonian system (33) was obtained
\[
d\Sigma^*(u^*) : \begin{cases}
\dot{x}^* = -(J - R) \frac{\partial H^*}{\partial x^*}(x, u, x^*, u^*), \\
y^* = -\frac{\partial H^*}{\partial u^*}(x, u, x^*, u^*),
\end{cases}
\]
(44)
in terms of the new controlled-Hamiltonian \(H^*(x, u, x^*, u^*)\)
\[
H^*(x, u, x^*, u^*) = \frac{1}{2} \begin{bmatrix} x^* \\ u^* \end{bmatrix}^T \frac{\partial^2 H(x, u)}{\partial (x, u)^2} \begin{bmatrix} x^* \\ u^* \end{bmatrix}.
\]
(45)
From (38) and (44) it is evident that the variational and the adjoint of a Hamiltonian system have similar state-space realizations. In [34] it is shown that – under the assumption of nonsingularity of \(J - R\) or the time symmetry of the Hessian of \(H(x, u)\) – they are related by a time-reversal operator, i.e.,
\[
(d\Sigma(u_v))^* := d\Sigma^*(u_v) = \mathcal{R} \circ d\Sigma(u_v),
\]
(46)
where \(\mathcal{R}\) is the time-reversal operator
\[
\mathcal{R}(u(t)) = u(T - t) \forall t \in [0, T].
\]
(47)
This implies that the adjoint of a variational controlled-Hamiltonian system can be obtained from the variational system itself. Additionally the complexity involved in obtaining the variational system \( d\Sigma \) can be avoided by using the local linear approximation [38]

\[
d\Sigma(u_v) \approx \Sigma(u + u_v) - \Sigma(u).
\] (48)

Hence the adjoint output of a controlled-Hamiltonian system can be obtained from the actual system output without any a priori system information [34].

D. Iterative learning control (ILC)

To get an ILC law that ensures monotonic error convergence one needs to constrain the update function \( \gamma \) of equation (32) such that a quadratic cost function

\[
J(y_i) = \int_{t_0}^{t_1} (y_d(t) - y_i(t))^T Q (y_d(t) - y_i(t)) dt,
\] (49)

is reduced in every iteration, where \( Q \) is a positive definite weight matrix [12]. From the principles of functional analysis, \( \gamma \) needs to be the negative gradient of the cost function

\[
dJ(y_i) = -2\langle Q(y_d - y_i), dy_i \rangle \\
= -2\langle Q(y_d - y_i), d\Sigma (du_i) \rangle \\
= -2\langle d\Sigma^* (Q(y_d - y_i)), du_i \rangle,
\] (50)

where we have used \( dy_i = d\Sigma (du_i) \) for the second identity this refers to the small change in the system output \( dy_i \) due to small variation in the system input \( du_i \), and for the last identity we have used the equality (43). The resulting ILC law is

\[
u_{i+1}(t) = u_i(t) + K_i (d\Sigma^* (Q(y_d - y_i))) \gamma,
\] (51)

where \( K_i \) is a user-defined constant.

Using (46)–(48) in (51) the update law becomes

\[
u_{i+1}(t) = u_i(t) + K_i (\mathcal{R} \circ \Sigma (u_i + Q(y_d - y_i)) - \mathcal{R} \circ \Sigma (u_i)),
\] (52)

as this involves two output trajectories, the update law (52) can be split into a two-step iteration law [12] :

\[
u_{2i+1} = u_{2i} + \mathcal{R} (Q(y_d - y_{2i})),
\]
\[
u_{2i+2} = u_{2i} + K_i \mathcal{R} (y_{2i+1} - y_{2i}).
\] (53)
The working of the iterative control law is illustrated by the following example.

**Example 6.** Figure 3 shows a schematic of a two degree-of-freedom (DOF) manipulator arm. This system can be represented both in the PH form (1) and in the controlled-Hamiltonian form (33).

The 2-DOF manipulator arm is characterized by link length $l_i$, mass of link $m_i$, center of gravity $r_i$, and moment of inertia $I_i$ where $i \in \{1, 2\}$. The arm’s motion is confined to the horizontal plane hence it has no potential energy term. The generalized position of the arm $q = (q_1 \ q_2)^T$ along with the momentum $p = (p_1 \ p_2)^T = M(q)\dot{q}$ constitute the system state $x = (q, p)^T$. The mass-inertia matrix is

$$M(q) = \begin{bmatrix} C_1 + C_2 + 2C_3 \cos(q_2) & C_2 + C_3 \cos(q_2) \\ C_2 + C_3 \cos(q_2) & C_2 \end{bmatrix},$$

(54)

with the constants $C_1, C_2, C_3$ defined as

$$C_1 = m_1 r_1^2 + m_2 l_1^2 + I_1$$

$$C_2 = m_2 r_2^2 + I_2$$

$$C_3 = m_2 l_1 r_2.$$

For numerical values of the parameters see [12]. For a desired system output

$$y_d(t) = \begin{pmatrix} 0.5 - 0.5 \cos(\pi t) \\ 0.5 \cos(\pi t) - 0.5 \end{pmatrix}$$

(55)
the ILC law (53) is evaluated for 20 iteration, the resulting quadratic cost (49) for $Q=I$ is in Figure 4.

It must be noted that the ILC update law (53) does not depend on the system entities $J,R,H(x,u)$ and the transformation matrix $T$. They are required to demonstrate self-adjointness of a given Hamiltonian system, hence the only requirement for the update law (53) is that the input-output data samples are obtained from a controlled-Hamiltonian system. In general ILC for PH system has various advantages:

- The ILC algorithm (53) is less sensitive to measurement noise since it does not require higher order derivatives of the error signal;
- System convergence to a global minimum can be demonstrated, for proof see [12];
- Monotonic convergence of the error signal in the sense of $L_2$-norm can be shown [12].

Owing to its simplicity and advantages, the iterative law (53) has been extended to address different control problems, namely optimal gait generation [41], [42], optimal control [43], optimal control with input constraints [44], and control of nonholonomic Hamiltonian systems [45], [46].

E. Repetitive control

Using ILC one can readily achieve tracking or disturbance rejection of periodic signals. However, a major drawback of ILC is that, prior to every iteration it requires the same initial state of the system. For various applications this requirement can be extremely hard to satisfy.
This disadvantage can be overcome by using yet another time-periodic trajectory learning method called repetitive control (RC). Repetitive control does not rely on the initial conditions of the system as it uses a desired trajectory of an infinite time horizon [21].

Repetitive control for a general system relies on the internal model principle, which can be loosely stated as follows: in order to track or reject a periodic signal the model of this signal must be included in the closed-loop [23]. Almost all the RC algorithms use the internal model principle by having the model of T-periodic signals in the closed-loop [47]. However, repetitive control for PH systems, introduced in [36], does not require the signal model since it is based on the variational symmetry of Hamiltonian systems (46)–(48). Similar to ILC, repetitive control of a PH system achieves the local minimum for a given quadratic cost function (49). The RC framework of [36] is summarized in Algorithm 1.

Algorithm 1 Repetitive control

1: Given a controlled-Hamiltonian system (33) and a cost function (49)
2: repeat: for every iteration $i$
3: Using the iterative control law (53) obtain a suitable control input $u_i(t), t \in [t_0, t_1]$
4: Apply $u_i(t)$ to system (33) and observe the system states $x(t), t \in [t_0, t_1]$. For time $t > t_1$ set $u_i(t) = 0$
5: Wait until the system state $x(t)$ converges within a predefined error bound $b$, for $\tau_i$ the shortest time duration s.t. $\|x(\tau_i)\| < b$. Calculate the excess convergence time $\Delta \tau_i = \tau_i - t_1$
6: Evaluate the cost function (49)
7: until the cost function ceases to decrease

In [36] it is stated that $\Delta \tau_i \to 0$ as $i \to \infty$. Next, this is demonstrated for trajectory tracking of a 2-DOF manipulator arm. The same task as in (55) is repeated for a desired system output

$$y_d(t) = \begin{pmatrix} 0.5 + 0.5 \sin(\pi t) \\ -0.5 - 0.5 \sin(\pi t) \end{pmatrix}. \quad (56)$$

The resulting cost, using (49) with $Q = I$, is given in Figure 5.

F. Iterative feedback tuning

Using ILC or RC one can obtain a control input that achieves the tracking of a desired reference. An equally important control objective is online tuning of the feedback control
parameters so as to achieve a desired performance criterion irrespective of model variation or parameter uncertainty. For example, the parameters of a state-feedback controller can be adjusted online via experiments to reduce a quadratic type of cost. One such online tuning method for repetitive tasks is iterative feedback tuning (IFT). Introduced in [48], IFT has found a wide acceptance as a self-tuning control design method for nonlinear or model-uncertain systems [24], [49]. By using self-adjointness of the Hamiltonian systems, an IFT algorithm has been established for controlled-Hamiltonian systems in [37].

Consider, for instance, the Hamiltonian of system (33) that has been modified by using energy-shaping feedback as

\[ H(x,u) = \frac{1}{2} p^T M^{-1}(q)p + \frac{1}{2} q^T K_p q - u^T q \]  

where \( x = (q,p)^T \) is the system state vector. If the parameter \( K_p \) is tuneable, we can rewrite the Hamiltonian (57) as a function of unknown parameter matrix \( \Theta \)

\[ H(x,u,\Theta) = \frac{1}{2} p^T M^{-1}(q)p + \frac{1}{2} q^T \Theta q - u^T q. \]  

If the design objective is to stabilize the system at the origin by using minimum energy, then the cost function can be formulated as

\[ J(q,\Theta) = \sum_{i=1}^{n} \int_{0}^{T} \frac{1}{2} Q_1 q_i^2 \, dt + \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{1}{2} Q_2 \Theta_{i,j}^2 \]  

where \( Q_1, Q_2 \) are weighting constants. The optimal parameter values for \( \Theta_{i,j} \) that achieves the minimal cost can be obtained by using the gradient descent method. In [37], a gradient expression has been devised which is similar to (50). The gradient of (59) also depends on the adjoint of
the variational operator. By using the self-adjointness property of the controlled-Hamiltonian, (46)–(48), an iterative feedback tuning algorithm is given in [37]. The feasibility of the algorithm was demonstrated by stabilizing a 3 DOF mass-spring-damper system at the origin.

VI. EVOLUTIONARY STRATEGIES

Solving the matching conditions of the non-parameterized IDA-PBC (23) can be tedious and cumbersome. To circumvent this issue, the authors of [50] parameterized the added energy component $H_a(x)$ as:

$$H_a(\xi, x) = h^T(P_1, P_2, x)P_0h(P_1, P_2, x)$$ (60)

where $P_i$, for $i \in \{0, 1, 2\}$, are the unknown symmetric square matrices of dimension $n \times n$. Additionally, $P_0$ is constrained to be positive definite. A single vector $\xi$ is constructed by vectorizing the unknown elements of $P_i$. To satisfy the equilibrium condition (17), the sigmoid function $h(P_1, P_2, x)$ is chosen such that $h(P_1, P_2, x_*) = 0$. The matrices $P_i$ are assumed to be full rank. The unknown parameters of the added energy function are learnt using Evolutionary Algorithms (EA), which is an instance of a stochastic optimization method.

Evolutionary algorithms are inspired by biology. In EA a population of candidate solutions called parents are evolved by a variation operator and a subset is chosen using population selection to form new candidate solutions. The variation operator and the population selection together result in a new offspring generation. This process is repeated until a desired objective is achieved. EA can be partitioned in three comparable methods [51], namely, Evolutionary Programming (EP), Genetic Algorithm (GA), and Evolutionary Strategies (ES). The main difference among these methods are in:

- problem representation: real valued vectors, finite state machines, strings of data, etc.
- offspring generation: mutation, cross-over, recombination, elitism, self-adaptation, etc.

For example, in ES the optimization problem is represented by a real valued vector. The variation operator is often a combination of mutation and self-adaptation of the parents and the population selection mechanism is either described by $(\lambda, \mu)$-ES or $(\lambda + \mu)$-ES methods, where $\lambda$ represents the number of offsprings and $\mu$ the number of parents. In the first approach, $\lambda$ offsprings are created by variation of $\mu$ parents. Irrespective of the fitness of the new offsprings $\lambda$, the original $\mu$ parents are discarded prior to the next iteration. In the second approach, after
creating a set of $\lambda$ offsprings, the worst fit individuals are discarded from the total population of $(\lambda + \mu)$.

A special form of this method is (1+1)-ES where one offspring is created from one parent, both the individuals are compared and the fitter one is reselected as parent for the next iteration. A pseudocode of $(\lambda + \mu)$-ES is given in Algorithm 2 [25].

**Algorithm 2** $(\lambda + \mu)$-Evolutionary strategy

1: $k = 0$
2: Initialize $\mu$ parents: $\xi$
3: **Evaluate:** Fitness function $\Phi(\xi)$
4: **Repeat**
5: **Execute:** Obtain $\lambda$ offsprings $\xi'$ from $\xi$ via mutation and self-adaptation.
6: **Evaluate:** Fitness function $\Phi(\xi')$
7: **Update:** Parent $\xi \leftarrow$ select best fit among $(\xi \cup \xi')$
8: **Until** termination condition

In [50] the unknown parameters $P_i$ of the parent $\xi$ are learnt using a variation of Algorithm 2 called evolutionary strategy IDA-PBC (ES-IDA-PBC). To ensure the reliability of the learnt parameters $P_i$, the control objective is verified for a set of initial states called the basin of attraction. The problem specific fitness function $\Phi(\xi)$ is computed using the pseudocode provided in Algorithm 3, see [50] for the detailed implementation.

The authors demonstrate their results by stabilizing an underactuated Hamiltonian system (ball and beam) at a desired position.

**VII. REINFORCEMENT LEARNING (RL)**

It is well known that due to the dissipation obstacle in standard PBC (19) only a part of the system’s energy can be shaped [8]. In [52] an effective method to separate the shapable and non-shapable components of the system’s Hamiltonian is provided. For example, for mechanical systems only the potential energy can be shaped using energy shaping techniques whereas the kinetic energy remains unaltered [9]. Hence the desired Hamiltonian for a mechanical system always consists of the original kinetic energy $T(x)$ term and the shaped potential energy term
Algorithm 3 $\phi(\xi)$-Fitness function

1: Given a PH system (1), user defined performance index function $\rho(x)$, and set of initial conditions $Init$
2: for 1: every initial condition indexed by $j$
3: \hspace{0.5cm} $x(t_0) \leftarrow Init_j$
4: for 2: every time step indexed by $i$
5: \hspace{1cm} Calculate the added energy $H_a(\xi,x)$ of (60)
6: \hspace{1cm} Calculate $u(\xi,x)$ of (22)
7: \hspace{1cm} Integrate $\dot{x} = (J(x) - R(x)) \frac{\partial H}{\partial x} + g(x)u(\xi,x)$ to obtain the new system state $x(t_i)$
8: \hspace{1cm} Calculate the performance index: $\rho_i = \rho(x(t_i))$
9: end for 2
10: $f_j \leftarrow \text{Sum}(\rho_i)$ (other operators such as $\text{Max}$ can be used, depending on the design objective)
11: end for 1
12: $f \leftarrow \text{Sum}(f_j)$ (other operators can be used to obtain $f$)
13: Return fitness value $f$

$V_d(x)$, i.e.,

$$H_d(x) = T(x) + V_d(x).$$

(61)

For general physical systems this can be written in terms of non-shapable component of the original Hamiltonian $H^{ns}(x)$ and shapable component $H^s_d(x)$, i.e.,

$$H_d(x) = H^{ns}(x) + H^s_d(x).$$

(62)

It is common in the ES-DI PBC framework (see Section II-B2), that the shaped component of (62) is chosen to be quadratic in the increments [8]. Instead, in [52] the shaped component is chosen as a linearly parameterized function approximation:

$$\hat{H}_d(x) = H^{ns}(x) + \xi^T \phi_{es}(x),$$

(63)

where $\phi_{es}(x) \in \mathbb{R}^e$ is a known basis function vector and $\xi \in \mathbb{R}^e$ is the unknown parameter vector. The damping injection matrix $K(x) \in \mathbb{R}^{m \times m}$ of (15) is considered as an additional degree
of freedom. It is parameterized in terms of an unknown parameter matrix \( \psi \) and a known basis function vector \( \phi_{di}(x) \in \mathbb{R}^r \) as

\[
\left[ \hat{K}(x, \psi) \right]_{ij} = \sum_{l=1}^{r} [\psi]_{ijl} [\phi_{di}(x)]_{l}.
\]

(64)

Additionally, the parameter vector \( \psi \in \mathbb{R}^{m \times m \times r} \) is constrained as

\[
[\psi]_{ijl} = [\psi]_{jil}
\]

(65)

in order to satisfy the symmetry condition of the damping injection matrix.

**Example 7.** (Continued from Example 1) Consider a parameterized desired potential energy \( V_d(x) = \xi^T \phi_{es}(x) \) from (16). The required additional energy so as to satisfy the matching condition (19) for a mechanical system (2) is

\[
H_a(x) = H_d(x) - H(x) = \xi^T \phi_{es}(x) - V(q).
\]

(66)

Correspondingly, the energy-shaping input for (2) is

\[
u_{es}(q) = (g^T(x)g(x))^{-1}g^T(x)(J(x) - R(x)) \frac{\partial H_a}{\partial x}(x)
\]

\[-\xi^T \phi_{es}(q) + \frac{\partial V}{\partial q}(q).\]

(67)

Using the damping-injection component \( u_{di}(x) = \hat{K}(x, \psi)y \), the ES-DI control law for a fully actuated mechanical system is

\[
u(x; \xi, \psi) = -\xi^T \frac{\partial \phi_{es}}{\partial q}(q) + \frac{\partial V}{\partial q}(q) - \hat{K}(x, \psi)y.
\]

(68)

This results in the closed-loop system

\[
\begin{bmatrix}
\dot{q} \\
\dot{p}
\end{bmatrix} =
\begin{bmatrix}
0 & I \\
-I & -D - \hat{K}(x, \psi)
\end{bmatrix}
\begin{bmatrix}
\frac{\partial H_d}{\partial q}(x) \\
\frac{\partial H_d}{\partial p}(x)
\end{bmatrix}
\]

(69)

with the closed-loop Hamiltonian \( H_d(x) = \frac{1}{2} p^T M^{-1}(q)p + \xi^T \phi_{es}(q) \).

□

In [52], the unknown parameters of (68) are efficiently learnt using a novel RL approach, called energy balancing actor-critic (EBAC). Prior to elaborating on the algorithm, we provide a brief overview of RL and in particular the actor-critic method.
A. Background on reinforcement learning

Reinforcement learning is a model-free semi-supervised machine learning paradigm [11]. Using RL, optimal controllers can be learnt for a very general class of nonlinear systems [53], [54]. Unlike the classical control approaches that require extensive prior information about the system, RL works on the principle of ‘learning from scratch’. The controller (in RL termed as the agent) optimizes its behavior by 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. In RL the objective is to maximize the total accumulated reward called return. The expected value of the return is given by a value function. Maximizing the value-function results in a desired control law (policy).

Generally, RL algorithms are devised for systems with discrete-time, discrete-state, and discrete-action spaces. However, most physical systems have continuous state-space and also the control action needs to be continuous. This complexity is often addressed by using function approximation methods [55]. Based on whether the policy or the value function is explicitly parameterized, RL methods are broadly classified into three categories [56]:

- **Actor-only** methods use a parameterized family of policies;
- **Critic-only** methods rely on a parameterized value function from which an approximate optimal policy is derived;
- **Actor-Critic** methods use a parameterized actor that results in an explicit policy. The critic, a parameterized value function, provides an evaluation of the actor’s performance.

B. Actor-Critic Reinforcement Learning

The Actor-Critic (AC) method uses two independent function approximators [56] to obtain a continuous control action for continuous-state systems. The actor represents a parameterized control law or policy \( u = \hat{\pi}(x, \vartheta) \) in terms of an unknown parameter vector \( \vartheta \in \mathbb{R}^{n_a} \). The critic approximates the value function \( \hat{V}(x, \theta) \) in terms of the unknown parameter vector \( \theta \in \mathbb{R}^{n_c} \).

For the deterministic discounted reward setting, the true value function is

\[
V(x_k) = \sum_{i=0}^{\infty} \gamma^i \rho(x_{k+i+1}, u(x_{k+i})) = \sum_{i=0}^{\infty} \gamma^i r_{k+i+1},
\]

(70)

where \( \gamma \) is the discount factor and \( \rho \) is the problem specific reward function that provides an
instantaneous reward \( r_k \). System (70) is approximated by a value function
\[
\hat{V}(x, \theta) = \theta^T \phi_c(x),
\]
where \( \phi_c(x) \in \mathbb{R}^{n_c} \) is a known basis function vector. In order to achieve continuous control-actions the policy \( \hat{\pi}(x, \vartheta) \) is approximated by
\[
\hat{\pi}(x, \vartheta) = \vartheta^T \phi_a(x),
\]
where \( \phi_a(x) \in \mathbb{R}^{n_a} \) is a known basis function vector.

Using AC, the reinforcement learning objective can be stated as: find a policy \( \hat{\pi}(x, \vartheta) \) such that the approximate value function \( \hat{V}(x, \theta) \) is maximized.

The unknown critic parameters are updated using a gradient-ascent rule as
\[
\theta_{k+1} = \theta_k + \alpha_c \delta_{k+1} \nabla_{\theta} \hat{V}(x_k, \theta_k),
\]
where \( \alpha_c \) is the critic learning rate, and \( \delta_{k+1} \) is the temporal difference [11]
\[
\delta_{k+1} = r_{k+1} + \gamma \hat{V}(x_{k+1}, \theta_k) - \hat{V}(x_k, \theta_k).
\]
Using the eligibility trace \( e_k \in \mathbb{R}^l \) the parameter convergence rate can be increased [11]. The modified parameter update rule using eligibility traces is
\[
e_{k+1} = \gamma \lambda e_k + \nabla_{\theta} \hat{V}(x_k, \theta_k),
\]
\[
\theta_{k+1} = \theta_k + \alpha_c \delta_{k+1} e_{k+1},
\]
where \( \lambda \in [0, 1] \) is the trace decay rate.

Using a zero-mean Gaussian noise \( \Delta u_k \) as an exploration term the control input to the system is
\[
u = \hat{\pi}(x_k, \vartheta_k) + \Delta u_k.
\]
The policy parameter \( \vartheta \) is increased in the direction of the exploration term if the resulting temporal difference \( \delta_{k+1} \) (74) due to the control input of (76) is positive. Otherwise it is decreased. The parameter update rule in terms of the actor learning rate \( \alpha_a \) is
\[
\vartheta_{k+1} = \vartheta_k + \alpha_a \delta_{k+1} \Delta u_k \nabla_{\vartheta} \hat{\pi}(x_k, \vartheta_k).
\]

**Remark 2**: Similar to the critic, eligibility traces (75) can also be used for the actor parameter update.
A well-known drawback of RL methods is the slow convergence of the learning algorithm. Total absence of information about the system necessitates exploring a large part of the state-space prior to learning an optimal trajectory. However, by incorporating partial knowledge about the system, the learning speed can be significantly increased [52], [57]. To this end, a novel actor-critic RL paradigm called energy balancing actor-critic (EBAC) was developed in [52]. Using EBAC, the added energy term $H_a(x)$ (i.e., parameter $\xi$ of (63)) and the damping matrix $K(x)$ (i.e., parameter $\psi$ of (64)) are learnt, thus avoiding solving complex matching equations and simultaneously enhancing the rate of convergence.

C. Energy-balancing actor critic (EBAC)

Using an additive zero-mean Gaussian noise term as an exploration signal, the parameters are learned using the EBAC Algorithm 4. The block diagram representation of the EBAC algorithm is given in Figure 6. For details, see [52].

Fig. 6. Energy balancing actor-critic.

In general the use of data-driven methods like reinforcement learning for PH control has various advantages:

- One can avoid computing explicit solutions of the PDE’s in (19);
- Unlike the standard-PBC only the local specification of the stabilization objective is sufficient for the EBAC;
- The use of prior information, in the form PH model, enhances the rate of convergence (see [52]).
- Thanks to the online-learning feature of the EBAC method robustness against model and parameter uncertainty can be achieved (see [58]).
Algorithm 4 Energy-Balancing Actor-Critic

Input: System (1), $\lambda$, $\gamma$, $\alpha_{a,i}$ for each actor, $\alpha_c$.

1: $e_0(x) = 0 \quad \forall x$

2: Initialize $x_0$, $\theta_0$, $\xi_0$, $\psi_0$

3: $k = 1$

4: loop

5:   Execute: Apply action $u_k = \hat{\pi}(x_k, \xi_k, \psi_k) + \Delta u_k$, observe next state $x_{k+1}$ and reward $r_{k+1} = \rho(x_{k+1}, u_k)$

6:   Temporal Difference:

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

8:   Critic Update:

9:   for $i = 1, \ldots, n_c$ do

10:     $e_{i,k+1} = \gamma \lambda e_{i,k} + \nabla \theta_{i,k} \hat{V}(x_k, \theta_k)$

11:     $\theta_{i,k+1} = \theta_{i,k} + \alpha_c \delta_{k+1} e_{i,k+1}(x)$

12:   end for

13:   Actor update:

14:   for $i = 1, \ldots, e$ do

15:     $\xi_{i,k+1} = \xi_{i,k} + \alpha_a \Delta u_k \nabla \xi_{i,k}(\hat{\pi}(x_k, \xi_k, \psi_k))$

16:   end for

17:   for $i = 1, \ldots, m(m+1)r/2$ do

18:     $\psi_{i,k+1} = \psi_{i,k} + \alpha_a \Delta u_k \nabla \psi_{i,k}(\hat{\pi}(x_k, \xi_k, \psi_k))$

19:   end for

20: end loop

- Nonlinearities such as control saturation can be easily handled by RL.

The EBAC algorithm 4 is used to stabilize a pendulum with saturated inputs at the upright position in [52]. In [58] EBAC was extended to regulate multi-input multi-output systems, demonstrated by stabilizing a 2-DOF manipulator arm at the desired equilibrium. For simulation and experimental results see [58].
VIII. Discussion and Outlook

Model-based control methods for PH systems aim to render the closed-loop passive. This is achieved by shaping the overall energy to have a minimum at the desired equilibrium. Since energy-shaping characterizes the closed-loop Hamiltonian – a candidate Lyapunov function – this ensures stability of the equilibrium point. Asymptotic stability of the closed-loop system can be demonstrated using LaSalle’s invariance principle. For some physical systems, like fully actuated mechanical systems, the PH model-based controller synthesis is straightforward.

Although PH model-based control approaches have various attractive properties they also have their challenges. For example, the feedback controller may fail to achieve the desired objective in the presence of model uncertainties and/or disturbances. Additionally, it is extremely difficult to consider performance measures in the PH model-based design approach. Some of these challenges can be overcome by using learning control methods. For example, online learning methods like EBAC and IFT are insensitive to model uncertainties. Also in learning algorithms, performance criteria can be readily considered.

Additionally, because of the PH system properties, learning processes are often enhanced. For example, in ILC monotonic error convergence is considered as a desirable property. For LTI systems this has been shown using the system’s Markov parameters, while for a general nonlinear system it is still an active research area. However, for a PH system, due to its self-adjointness property, the monotonic convergence of the iterative algorithm is shown to be implicit [12], [36].

A major hurdle in reinforcement learning is slow convergence of the algorithm often arising due to the ‘learning from the scratch’ mindset. In EBAC, prior knowledge in the form of a PH system has been experimentally shown to enhance the learning rate compared to standard actor-critic methods [52].

Along with the added advantages, learning control in the PH framework introduces new challenges. For example, as in the case of many data-driven techniques – like reinforcement learning and evolutionary algorithms – EBAC and ES-IDA-PBC are affected by the ‘curse of dimensionality’. Exploration, an integral part of actor-critic techniques (hence also EBAC), may not be feasible when it is too dangerous to allow the controller to explore the system’s state-space, particularly in safety-critical applications. For EBAC and ES-IDA-PBC algorithms the proof of convergence is yet to be shown. The proof of convergence for iterative methods
has been demonstrated, but only for controlled-Hamiltonian systems. Parameter convergence in adaptive control methods for PH systems is available only for tracking control of a fully actuated mechanical system.

In Table I we list various control methods for port-Hamiltonian systems. Even though the choice of a particular control method is highly problem specific, the table provides an initial suggestion for selecting the feasible control method in the PH framework. Depending on the type of feedback, Table I classifies the model-based approaches into static or dynamic methods, respectively. During learning some control parameters are adjusted hence it serves no purpose to classify learning algorithms for PH systems as static or dynamic. For model-based control methods, it is shown that in general from the perspective of stabilization there is no additional advantage in considering dynamic controllers since an equivalent static state-feedback can achieve the design objective [6]. If the desired closed-loop Hamiltonian and system structures are known a priori, then static methods, particularly the passivity-based approaches like ES-DI, IDA-PBC and CT, can be used with relatively low effort. However, it must be noted that fixing a priori the desired closed-loop system structure for a multi-domain physical system is generally cumbersome. Although most of the PH model-based control methods are devised for stabilization, modifications of IDA-PBC and CT to reflect a time-varying Hamiltonian can been used for tracking applications [59], [60].

<table>
<thead>
<tr>
<th>Stabilization</th>
<th>Model-based methods</th>
<th>Learning methods</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Static feedback</td>
<td>Dynamic feedback</td>
</tr>
<tr>
<td>Energy shaping and damping injection (ES-DI) [6], Interconnection and damping assignment (IDA)-PBC [9], power based method [27], Canonical transformation (CT) [5]</td>
<td>Control by interconnection (CbI) or energy-casimir methods [4], [6], [7]</td>
<td>Energy balancing actor-critic (EBAC) [52], Iterative feedback tuning (IFT) [37], Evolutionary strategy (ES-IDA-PBC) [50], Adaptive control (AC) [26], [28], [30], [31]</td>
</tr>
<tr>
<td>Tracking</td>
<td>Modified IDA-PBC [59], Canonical transformation [60]</td>
<td>-</td>
</tr>
</tbody>
</table>
In Table II we list the requirement of prior system information for various learning methods that are discussed in this paper. With the exception of the evolutionary strategy (ES-IDA-PBC), all other learning methods of Table II can be used for online control tuning, hence they are capable of handling model and parameter uncertainties. For example, stabilization methods like EBAC and IFT, can learn an optimal feedback law even for an imprecise model. Iterative methods like ILC and RC are independent of system parameters, the only requirement being measured data arising from a controlled-Hamiltonian system. Their robustness against model and parameter uncertainty is implied. Additionally, ILC and RC can compensate for periodic disturbances in the system.

<table>
<thead>
<tr>
<th>No a prior model information</th>
<th>Uncertain but known model and known control structure</th>
<th>Precise model and known control structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterative learning control (ILC) [12], Repetitive control (RC) [45], Iterative feedback tuning (IFT) [37]</td>
<td>Adaptive control (AC) [26], [28], [30], [31]. Energy balancing actor-critic (EBAC) [52]</td>
<td>Evolutionary strategy (ES-IDA-PBC) [50]</td>
</tr>
</tbody>
</table>

Because of the capability to learn parameters for known control structures, we believe that the extension of the online methods like IFT and RL for PH control can provide an exciting research avenue. For example, similar to EBAC one can use the RL framework to obtain the unknown elements of the control law (22). The RL or IFT based learning framework can be extended for stabilization and tracking of multi-domain physical systems. Similarly, iterative methods like ILC and RC can be extended to solve optimal control problem for electro-mechanical systems.

**IX. Conclusion**

In this paper we have provided a comprehensive review of various state-of-the-art learning and adaptive control methods for PH systems. We have highlighted the fact that, even though passivity-based approaches for PH systems generally achieve desired results, their learning extensions have added new values. A few notable examples are: i) the use of learning algorithms (such as EBAC and ES-IDA-PBC) reduces the complexity of the PH control synthesis; ii) iterative and repetitive control for PH systems achieve monotonic error convergence; iii) global
asymptotic stability can be shown for adaptive control of PH systems. Thanks to learning, control design problems can be solved, which would otherwise be intractable for PH systems. However, introducing learning does involve a certain degree of compromise particularly in terms of computational cost, and memory.

REFERENCES


