Control by Interconnection using Reinforcement Learning

Anshuman Bhattacharjee

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Control by Interconnection using Reinforcement Learning

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Anshuman Bhattacharjee

October 19, 2015

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Control by Interconnection using Reinforcement Learning

by

Anshuman Bhattacharjee

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: October 19, 2015

Supervisor(s):

Dr.ir. G. Delgado Lopes

S. P. Nageshrao

Reader(s):

Dr.ir. G. Delgado Lopes

S. P. Nageshrao

Dr.ir. A. Hegyi

Abstract

The dynamics of many physical processes can be described by port-Hamiltonian (PH) models where the importance of the energy function can be seen. In Control by Interconnection (CbI), the controller is another PH system that is connected to the plant through a power preserving interconnection to add up the energy functions. However, a major issue in this is that the choice of Casimir function and controller Hamiltonian is left to the discretion of the designer and requires experience to make a good choice. In this thesis, an attempt is made to eliminate this problem by using machine learning algorithms (in particular, reinforcement learning) to let the computer "learn" the best controller design.
Moreover, the assumption that both the plant and the controller must be passive leads to what is known as the dissipation obstacle, which means that dissipation is allowed only on those states/coordinates of the energy function which do not require shaping. This imposes restrictions on the applications. Here, it is attempted to try to go beyond this dissipation obstacle and achieve a dynamic feedback controller.

Table of Contents

List of Figures

List of Tables

Acknowledgements

During the course of this thesis, I have not only gained interesting insights in the field of control, but I have also incurred debts of gratitude to many, most of whom I will try to acknowledge here.

I would like to express my heartfelt thanks and gratitude to my supervisors Gabriel Lopes and Subramanya Nageshrao. Without their guidance, advice and encouragement this thesis would not have been possible.

I would like to thank my family for their unconditional love and support during the thesis, and for having the patience to listen to me ramble on about my work.

And finally, I would like to thank all my friends and fellow students at DCSC for the support and encouragement they have provided me with, as well as the fruitful discussions on various aspects of control. In particular, I would like to thank Cees Verdier and Tim de Bruin for all the interesting discussions about machine learning algorithms and artificial intelligence.

In case I have missed anyone, it is not my intention and I do apologise. Thank you all very much.

Delft, University of Technology — Anshuman Bhattacharjee
October 19, 2015

"Failure is central to engineering. Every single calculation that an engineer makes is a failure calculation. Successful engineering is all about understanding how things break or fail."

—*Henry Petroski*

To 1,3,7-Trimethylxanthine, for helping me get my brain into gear, so I could understand why things break.

Chapter 1

Introduction

The port-Hamiltonian (PH) framework [4] allows many physical systems to be described in terms of the interconnection structure of the elements and an energy function. This is particularly useful in the context of modelling of complex non-linear systems as simpler sub-systems that can be interconnected to model a larger more complex physical system. PH theory also provides control engineers with a number of tools to exploit this structure when designing controllers. A popular methodology to do so is Passivity Based Control (PBC) [5, 6], which exploits the property of passivity to achieve the desired control objective by rendering the closed loop passive with respect to a desired storage function. The interested reader is referred to [3, 4, 5, 6] for more details on PH systems and PBC. However, most PBC methods require state information, which is not always available in practice. One prominent output feedback method for PH systems is Control by Interconnection (CbI). In this method, the controller also modelled as a PH system. Interconnecting the two in a power preserving manner, the energy of the controller gets added to the energy of the plant. A relation between the plant and controller states is found by means of dynamical invariants (called Casimir functions). By using this, the energy of the closed loop system can be shaped as desired. Some advantages to using CbI are:

- CbI has a very intuitive way of presenting the control design objective in terms of shaping the energy of the system via energy exchange with the environment.
- CbI is an output feedback method which does not require state information, which is often not available in practice.
- Properties like stability and passivity can be guaranteed by CbI which makes them ideal for environments where multiple dynamical systems interact with each other.

However, there are some significant disadvantages as well:

- Deriving a control law for CbI involves solving Partial Differential Equations (PDEs) which have multiple solutions and choosing the best solution requires experience in control design. Moreover, solving PDEs numerically can be computationally expensive.

- CbI does not take into account input saturation, which is a problem often encountered in real life applications where actuators have a limited operating range and trying to use the actuators outside of this range may damage them.
- There is no standard way of incorporating performance criteria into CbI.

In the recent years, there has been a trend to incorporate machine learning into control. With the decline in the cost of computational power, machine learning algorithms like Reinforcement Learning (RL) are becoming more popular and viable. Particularly in the case of robotics, where frequently the robot must navigate unstructured or unknown terrain, or in the case of complex tasks which cannot always be clearly pre-defined, some sort of adaptive or learning control technique is required. RL is inspired from how animals (and humans) interact with and learn from their environment. Some examples showing the success of RL in robotics and control can be found in [7, 8, 9].

RL is useful when there is a lack of information about the dynamical system or the environment. However, there are disadvantages to this method as well. The speed of learning can be very slow, especially in high dimensional cases and it is often difficult to ensure the quality of the learned control policy.

Recently however, there have been some promising results incorporating RL into various state feedback PH controllers [1, 10, 11, 12]. Inspired by these results, this thesis seeks to incorporate RL with the CbI methodology to eliminate some of the drawbacks of both methods while retaining the advantages. Thus, the main motivation of this thesis is to embed RL into the CbI methodology in order to make it possible to easily find suitable Casimir functions and design an output feedback controller for PH systems.

However, the applicability of CbI is somewhat hindered by the dissipation obstacle, which requires that dissipation may not be present in the coordinates to be shaped. In this thesis, an attempt is made to formulate a dynamic controller that might allow one to circumvent the dissipation obstacle.

The rest of this thesis is organised as follows:

- **Chapter 2** introduces the concepts of the PH framework. It introduces PH systems theory and CbI and provides an example of using CbI to stabilise a model of a physical system.
- **Chapter 3** introduces the basic concepts of discrete time RL and sheds some light on previous work using RL with the PH framework, which provides the inspiration and motivation for this thesis.
- **Chapter 4** presents the developed methodology and algorithm.
- An attempt to further generalise CbI to go beyond the dissipation obstacle is made in **Chapter 5**.
- Finally, some conclusions and recommendations for future work are mentioned in **Chapter 6**.

Chapter 2

Port-Hamiltonian Systems

2-1 Introduction

Historically, mechanical and electrical engineering have taken slightly different approaches to physical systems. While most of the analysis of physical systems has been performed within the Lagrangian and Hamiltonian framework with its roots in analytical mechanics, the network point of view taken by electrical engineering is prevalent in the modelling and simulation of complex physical systems [3]. The port-Hamiltonian (PH) framework combines both these points of view - by associating the interconnection structure of the network model with a geometric structure given by a Dirac structure and the Hamiltonian dynamics are then defined with respect to this Dirac structure and the Hamiltonian given by the total stored energy of the system [3].

Apart from offering an intuitive and insightful framework for the modelling and analysis of complex physical systems, PH systems theory provides a natural starting point for control. Especially in the case for control of non-linear systems, it is widely recognised that the natural properties of the system should be exploited and PH systems theory provides a range of tools for doing so [4]. Furthermore, a very nice property of PH systems is that they are open dynamical systems and can interact with their environment through ports. In addition, the interconnection of two or more PH systems in a power preserving manner is again a PH system [3, 4].

2-2 Port Hamiltonian Systems

The general framework for PH systems was introduced in [13] and developed further in [14]. A review of PH systems and Passivity Based Control (PBC) can be found in [3, 4, 14, 5].

2-2-1 Input state output port-Hamiltonian Systems

An important class of PH systems in control engineering is the input state output PH system. A general input state output PH system is of the form[1] [4],

$$\begin{aligned} \dot{x} &= [J(x) - R(x)]\frac{\partial H(x)}{\partial x} + g(x)u, \\ y &= g^T(x)\frac{\partial H(x)}{\partial x}, \end{aligned} \tag{2-1}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$, $m \leq n$ is the control input, $J(x), R(x) : \mathbb{R}^n \mapsto \mathbb{R}^{n\times n}$ with $J(x) = -J(x)^T$ and $R(x) = R(x)^T \geq 0$ are the interconnection and dissipation (or damping) matrices respectively, $H(x) : \mathbb{R}^n \mapsto \mathbb{R}$ is the Hamiltonian, which is the total stored energy of the system, $u, y \in \mathbb{R}^m$ are the conjugated input output variables whose product has the units of power and $g(x) : \mathbb{R}^n \mapsto \mathbb{R}^{n\times m}$ is the input matrix (assumed to be full rank). For the remainder of this thesis, the matrix $F(x) : \mathbb{R}^n \mapsto \mathbb{R}^{n\times n}$ is denoted as,

$$F(x) := J(x) - R(x), \tag{2-2}$$

which satisfies $F(x) + F^T(x) = -2R(x) \leq 0$.
The power balance equation is then,

$$\dot{H}(x) = \frac{\partial^T H(x)}{\partial x}\dot{x} \tag{2-3}$$

$$= \frac{\partial^T H(x)}{\partial x}\left([J(x) - R(x)]\frac{\partial H}{\partial x}(x) + g(x)u\right) \tag{2-4}$$

$$= -\underbrace{\frac{\partial^T H(x)}{\partial x}R(x)\frac{\partial H(x)}{\partial x}}_{d(x)} + u^T y \tag{2-5}$$

where $d(x)$ is the natural dissipation of the system.
Since $R(x)$ is positive semi-definite, it follows that

$$\dot{H}(x) \leq u^T y. \tag{2-6}$$

If the Hamiltonian $H(x)$ is bounded from below and positive semi-definite, Eq. (2-6) is referred to as the *passivity inequality*. If the Hamiltonian is not bounded from below and nor is it positive semi-definite, then Eq. (2-6) is called the *cyclo-passivity inequality* [5].

2-2-2 Casimir Functions

An important property of PH systems is the existence of dynamical invariants independent of the Hamiltonian $H(x)$ of the system, known as *Casimir* functions [14]. If a function $C : \mathbb{R}^n \mapsto \mathbb{R}$ exists, satisfying,

$$\frac{\partial^T C(x)}{\partial x}[J(x) - R(x)] = 0, \quad x \in \mathbb{R}^n \tag{2-7}$$

[1] All vectors are column vectors. All gradients defined are also column vectors.

implying that the time derivative of C along the solutions of the PH system is zero for $u = 0$, such a function $C(x)$ is called a Casimir function. For arbitrary input functions, this holds if additionally [15],

$$\frac{\partial^T C(x)}{\partial x} g(x) = 0. \tag{2-8}$$

This is easily verified by

$$\dot{C} = \frac{\partial^T C(x)}{\partial x} \dot{x} \tag{2-9}$$

$$= \frac{\partial^T C(x)}{\partial x} [J(x) - R(x)] \frac{\partial H}{\partial x}(x) + \frac{\partial^T C(x)}{\partial x} g(x) u \tag{2-10}$$

$$= 0. \tag{2-11}$$

An important consequence of the existence of Casimir functions is that if $C_1(x), C_2(x), \cdots, C_r(x)$ are Casimir functions, then not only $\frac{dH}{dt} = 0$ for $u = 0$, but also:

$$\frac{d}{dt}(H + H_a(C_1, C_2, ..., C_r))(x(t)) = 0 \tag{2-12}$$

for any function $H_a : \mathbb{R}^r \mapsto \mathbb{R}$. This means that even though $H(x)$ is not positive definite at an equilibrium x^*, the function $H(x) + H_a(C_1, C_2, \cdots, C_r)(x)$ could possibly be positive definite at the equilibrium point by appropriately choosing H_a and thus may serve as a candidate Lyapunov function for stability analysis. This method is called the Energy-Casimir method and it has various applications in the control of PH systems, most notably in Control by Interconnection (CbI).

2-2-3 Passivity

The notion of *passivity* is an important concept in PH systems. Passivity is a fundamental property of dynamical systems and can be directly inferred from the (cyclo) passivity inequality (Eq. (2-6)). Integrating Eq. (2-5), we get the following energy balancing equation,

$$\underbrace{H[x(t)] - H[x(0)]}_{\text{stored energy}} = \underbrace{\int_0^t u^T(s) y(s)\, \mathrm{d}s}_{\text{supplied energy}} - \underbrace{\int_0^t \left[\frac{\partial H}{\partial x}[x(s)]\right]^T R(x(s)) \left[\frac{\partial H}{\partial x}[x(s)]\right] \mathrm{d}s}_{\text{dissipated energy}}. \tag{2-13}$$

Simply put, a passive system cannot generate energy on its own [4]. It is important to distinguish here between passive and cyclo-passive systems. In other words, a system is cyclo passive if it cannot create energy over closed paths in the state space. It might however, produce energy along some initial portion of its trajectory and in such a case, it will not be a passive system. Every passive system is a cyclo-passive system but the converse does not always hold [15]. This notion of passivity is often exploited in the control of PH systems.

2-3 Passivity Based Control

Passivity Based Control (PBC) is a generic name that refers to a controller design methodology which renders the system passive with respect to a desired storage function and injects damping, thus achieving stabilisation [6].

2-3-1 Standard Passivity Based Control

In the standard formulation of PBC, it is desired to design a control law $u = \beta(x) + v$ such that the closed loop satisfies the new power balancing equation [16],

$$\dot{H}_d(x) = v^T(s)z(s) - d_d(x) \tag{2-14}$$

where $H_d(x)$ is the desired energy function, the new passive output is given by z (may be equal to y) and the natural dissipation $d(x)$ has been replaced with some function $d_d(x) \geq 0$ to ensure a faster convergence rate. The desired energy function $H_d(x)$ is chosen such that it has a strict minimum at the desired x^* and is known as Energy Shaping (ES) whereas the modification of the dissipation function is referred to as Damping Injection (DI) [17].

2-3-2 Stabilisation via Energy Balancing

Defining the added energy function as

$$H_a(x) = H_d(x) - H(x), \tag{2-15}$$

a state feedback law is said to be *energy balancing* if this added energy is equal to the energy supplied by the environment,

$$\dot{H}_a(x) = -\beta^T(x)y. \tag{2-16}$$

The closed loop energy is equal to the difference between the stored and supplied energy and hence this class of PBC is known as Energy Balancing PBC.

2-3-3 Energy Shaping and Damping Injection

The process described in sub-section 2-3-2 of augmenting the plant Hamiltonian with some added energy $H_a(x)$ such that $H_d(x) = H(x) + H_a(x)$ has a minimum at the desired equilibrium, i.e,

$$x^* = \arg\min H_d(x) \tag{2-17}$$

is known as Energy Shaping (ES) [3].

In Damping Injection (DI), the closed loop asymptotic stability is achieved by injecting further damping into the system such that a target closed loop system is obtained,

$$\dot{x} = [J(x) - R_d(x)]\frac{\partial H_d(x)}{\partial x} \tag{2-18}$$

with $R_d(x) = R(x) + g(x)K_d(x)g^T(x)$ the desired dissipation matrix in terms of a damping injection matrix $K_d(x)$.

Combining the two, desired closed loop dynamics can be achieved by a control law,

$$u(x) = u_{ES}(x) + u_{DI}(x), \tag{2-19}$$

where,

$$u_{ES}(x) = (g^T(x)g(x))^{-1}g^T(x)[J(x) - R(x)]\frac{\partial H_a(x)}{\partial x}, \tag{2-20}$$

$$u_{DI}(x) = -K_d(x)g^T(x)\frac{\partial H_d(x)}{\partial x}. \tag{2-21}$$

The added energy $H_a(x)$ is found by finding the solutions to the Partial Differential Equation (PDE)

$$\begin{bmatrix} g^{\perp}[J(x) - R(x)] \\ g^T(x) \end{bmatrix} \frac{\partial H_a(x)}{\partial x} = 0, \tag{2-22}$$

where $g^{\perp}$ is the full rank left annihilator of $g(x)$, i.e. $g^{\perp}(x)g(x) = 0$. Amongst all the solutions, the one satisfying Eq. (2-17) is chosen [3].

2-3-4 Interconnection and Damping Assignment Passivity Based Control

As the name implies, in Interconnection and Damping Assignment Passivity Based Control (IDA-PBC) the system is controlled by assigning it a desirable damping and interconnection matrix such that the new energy function has the minimum at the desired point. The objective is again, to find a static control law $u = \beta(x)$ such that the closed loop dynamics are of the form,

$$\dot{x} = [J_d(x) - R_d(x)]\frac{\partial H_d}{\partial x} \tag{2-23}$$

where the new energy function $H_d(x)$ has a strict local minimum at the desired equilibrium point x_*, and $J_d(x)$ and $R_d(x)$ are the desired interconnection and damping matrices respectively [18, 6].

Given $J(x), R(x), H(x), g(x)$ and desired equilibrium $x_* \in \mathbb{R}^n$, assume that functions $\beta(x)$, $J_a(x)$, $R_a(x)$ and a vector function $K(x)$ can be found satisfying [18]

$$[J(x) + J_a(x) - (R(x) + R_a(x))]\, K(x) = -[J_a(x) - R_a(x)]\frac{\partial H}{\partial x}(x) + g(x) \tag{2-24}$$

such that the following holds:

1. Structure preservation:

$$J_d(x) := J(x) + J_a(x) = -[J(x) + J_a(x)]^T$$
$$R_d(x) := R(x) + R_a(x) = [R(x) + R_a(x)]^T \geq 0$$

2. Integrability: $K(x)$ is a gradient of a scalar function, i.e,

$$\frac{\partial K}{\partial x}(x) = \left[\frac{\partial K}{\partial x}(x)\right]^T \tag{2-25}$$

3. Equilibrium assignment: $K(x)$ at x_* verifies

$$K(x_*) = -\frac{\partial H}{\partial x}(x_*) \tag{2-26}$$

4. Lyapunov stability: The Jacobian of $K(x)$ at x_* satisfies the bound

$$\frac{\partial H}{\partial x}(x_*) \geq \frac{\partial^2 H}{\partial x^2}(x_*) \tag{2-27}$$

Under these conditions, the closed loop system $u = \beta(x)$ is a PH system with dissipation of the form given in Eq. (2-23) with

$$H_d(x) = H(x) + H_a(x) \tag{2-28}$$

and

$$\frac{\partial H_a}{\partial x}(x) = K(x). \tag{2-29}$$

Further, x_* will be a locally stable equilibrium of the closed loop system. If in addition, it can be ascertained that x_* is the largest invariant set under the closed loop dynamics contained in

$$\left\{ x \in \mathbb{R}^n | \left[\frac{\partial H_d}{\partial x}(x)\right]^T R_d(x) \frac{\partial H_d}{\partial x}(x) = 0 \right\} \tag{2-30}$$

then, x_* will be asymptotically stable [18, 6].

In practice, for systems of the form (2-1), $J_a(x)$ and $R_a(x)$ can be fixed and then solutions are found to the PDE

$$g^\perp[J(x) + J_a(x) - (R(x) + R_a(x))]\frac{\partial H_a}{\partial x}(x) = -g^\perp[J_a(x) - R_a(x)]\frac{\partial H}{\partial x}(x) \tag{2-31}$$

in terms of $H_a(x)$ where $g^\perp$ is the left annihilator matrix of $g(x)$, i.e, $g^\perp(x)g(x) = 0$. The control can then be calculated as [18]

$$\beta(x) = [g^T(x)g(x)]^{-1}g^T(x)\left\{[J(x) + J_a(x) - (R(x) + R_a(x))]\frac{\partial H_a}{\partial x}(x) + [J_a(x) - R_a(x)]\frac{\partial H}{\partial x}(x)\right\}. \tag{2-32}$$

2-4 Control by Interconnection (CbI)

The controllers mentioned in Section 2-3 are all state feedback controllers. However, in practice, state information is not always available. CbI takes an output feedback approach to control, wherein the energy shaping results from the interconnection of the plant system with a suitable controller system [15, 19].

Consider a PH system of the form given in Eq. (2-1) interconnected with a PH controller

$$\begin{aligned} \dot{\zeta} &= [J_c(\zeta) - R_c(\zeta)]\frac{\partial H_c(\zeta)}{\partial \zeta} + g_c(\zeta)u_c, \\ y_c &= g_c^T(\zeta)\frac{\partial H_c(\zeta)}{\partial \zeta} \end{aligned} \tag{2-33}$$

with state $\zeta \in \mathbb{R}^m$, input u_c, output y_c, and $H_c(\zeta)$ is the controller Hamiltonian. Using a standard power preserving negative feedback interconnection as,

$$\begin{bmatrix} u \\ u_c \end{bmatrix} = \begin{bmatrix} 0 & -I_m \\ I_m & 0 \end{bmatrix} \begin{bmatrix} y \\ y_c \end{bmatrix}, \tag{2-34}$$

the composed system is again a PH system and can be written as

$$\begin{bmatrix} \dot{x} \\ \dot{\zeta} \end{bmatrix} = \begin{bmatrix} J(x) - R(x) & -g(x)g_c^T(\zeta) \\ g_c(\zeta)g^T(x) & J_c(\zeta) - R_c(\zeta) \end{bmatrix} \begin{bmatrix} \frac{\partial H_{cl}}{\partial x}(x,\zeta) \\ \frac{\partial H_{cl}}{\partial \zeta}(x,\zeta) \end{bmatrix} \tag{2-35}$$

with H_{cl} the closed loop energy function given by

$$H_{cl}(x,\zeta) = H(x) + H_c(\zeta). \tag{2-36}$$

Now, although $H_c(\zeta)$ can be freely assigned, the energy of the plant $H(x)$ is given and thus it is not immediately clear how to effectively shape the closed loop energy of the system [15]. The idea now is to investigate the Casimir functions of the closed loop system as they are dynamical invariants that relate the plant states to the controller states [14, 20, 19].

In practice, we usually restrict ourselves (without much loss of generality) to Casimir functions of the form

$$C(x,\zeta) = \zeta - S(x) = 0 \tag{2-37}$$

where S is some function of x. The invariance condition $\dot{C}(x,\zeta) = 0$ results in the PDE

$$\begin{bmatrix} -\frac{\partial^T S(x)}{\partial x} & I_c \end{bmatrix} \begin{bmatrix} J(x) - R(x) & -g(x)g_c^T(\zeta) \\ g_c(\zeta)g^T(x) & J_c(\zeta) - R_c(\zeta) \end{bmatrix} = 0 \tag{2-38}$$

which can be expressed as the following chain of equalities [15, 21, 17]

$$\frac{\partial^T S(x)}{\partial x} J(x) \frac{\partial S(x)}{\partial x} = J_c(\zeta), \tag{2-39}$$

$$R(x)\frac{\partial S(x)}{\partial x} = 0, \tag{2-40}$$

$$R_c(\zeta) = 0, \tag{2-41}$$

$$J(x)\frac{\partial S(x)}{\partial x} = -g(x)g_c^T(\zeta). \tag{2-42}$$

Now, the x-dynamics of the system given in Eq. (2-35) is given as,

$$\dot{x} = [J(x) - R(x)]\frac{\partial H(x)}{\partial x} - g(x)g_c^T(\zeta)\frac{\partial H_c(\zeta)}{\partial \zeta} \tag{2-43}$$

Using Eq. (2-40) and Eq. (2-42), this can be written as,

$$\dot{x} = [J(x) - R(x)]\left(\frac{\partial H(x)}{\partial x} + \frac{\partial S(x)}{\partial x}\frac{\partial H_c(\zeta)}{\partial \zeta}\right) \tag{2-44}$$

Substituting $\zeta = S(x) + \kappa$ and using the chain rule of differentiation gives

$$\dot{x} = [J(x) - R(x)]\frac{\partial H_s(x)}{\partial x} \tag{2-45}$$

with $H_s(x) = H(x) + H_c(S(x) + \kappa)$. Thus the interconnection has resulted in a closed loop system with the same interconnection structure but with shaped energy.

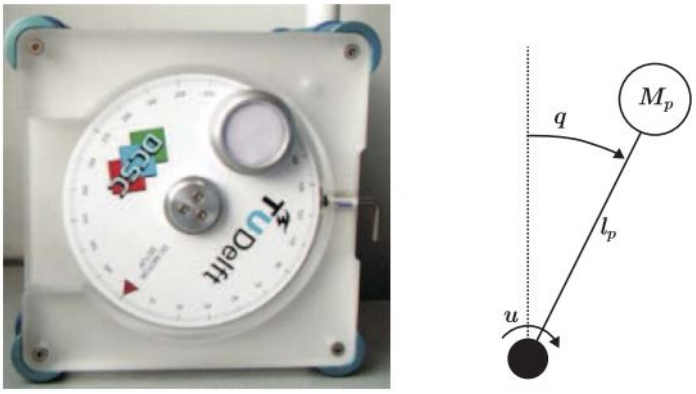


Figure 2-1: Inverted Pendulum [1]

2-4-1 Dissipation Obstacle

The condition given by Eq. (2-40) somewhat hinders the applicability of CbI. In essence, it states that the Casimir functions cannot depend on the coordinates that are subject to dissipation [21] i.e, dissipation is admissible only in those coordinates that do not require shaping of the energy. This is known as the *dissipation obstacle* [17]. This stymies the use of CbI for applications other than mechanical systems where the coordinates to be shaped are usually positions (which are unaffected by friction) [5].

2-5 Example of CbI - Inverted Pendulum

To illustrate CbI, the example of an inverted pendulum is taken which is a well known non-linear system. The model of an actual inverted pendulum setup available at the robotics lab at Delft Center for Systems and Control (DCSC) is used to perform some simulations.

Consider a pendulum as shown in Figure 2-1, actuated by applying a voltage u to the motor. The PH model of the pendulum is given by[2],

$$\begin{aligned} \underbrace{\begin{bmatrix}\dot{q}\\ \dot{p}\end{bmatrix}}_{\dot{x}} &= \left(\underbrace{\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}}_{J(x)} - \underbrace{\begin{bmatrix}0 & 0\\ 0 & c_p\end{bmatrix}}_{R(x)} \right) \underbrace{\begin{bmatrix}\nabla_q H(q,p)\\ \nabla_p H(q,p)\end{bmatrix}}_{\nabla_x H(x)} + \underbrace{\begin{bmatrix}0\\ \frac{K_p}{R_p}\end{bmatrix}}_{g(x)} u \\ y &= \underbrace{\begin{bmatrix}0 & \frac{K_p}{R_p}\end{bmatrix}}_{g^T(x)} \underbrace{\begin{bmatrix}\nabla_q H(q,p)\\ \nabla_p H(q,p)\end{bmatrix}}_{\nabla_x H(x)} \end{aligned} \tag{2-46}$$

where c_p is the damping caused due to the friction and the Hamiltonian of the system is given by,

[2] $\nabla_x = \partial/\partial x$

Table 2-1: Inverted Pendulum Model Parameters

Model Parameters	Symbol	Value	Units
Pendulum inertia	J_p	1.9×10^{-4}	kgm^2
Pendulum mass	M_p	5.2×10^{-2}	kg
Gravity	g_p	9.81	m/s^2
Pendulum length	l_p	4.2×10^{-2}	m
Friction	c_p	1×10^{-3}	Nms
Torque constant	K_p	5.6×10^{-2}	Nm/A
Rotor resistance	R_p	9.92	Ω

$$H(q,p) = \frac{1}{2J_p}p^2 + M_p g_p l_p(1+\cos q), \tag{2-47}$$

where J_p is the rotational moment of inertia, M_p is the mass of the pendulum, l_p is the length of the pendulum, and g_p is the gravitational constant. The states q, p, are the position (angle with respect to the normal) and the momentum respectively. The model parameters are given in Table 2-1 [22].

The natural equilibria of this system are found to be given by,

$$(q^*, p^*) = (k\pi, 0), \quad k \in \mathbb{Z} \tag{2-48}$$

as can also be seen from the system energy (or Hamiltonian) as shown in Figure 2-2. It can also be seen that the system has stable equilibria when k is odd and unstable equilibria when k is even. The desired objective is to shape the energy such that the pendulum stabilises at $(q^*, p^*) = (0, 0)$ corresponding to the upright position.

Let the controller also be a PH controller,

$$\begin{aligned} \dot{\zeta} &= g(\zeta)u_c, \\ y_c &= g^T(\zeta)\frac{\partial H_c(\zeta)}{\partial \zeta} \end{aligned} \tag{2-49}$$

interconnected with

$$\begin{bmatrix} u \\ u_c \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y \\ y_c \end{bmatrix}. \tag{2-50}$$

Choose $g_c(\zeta) = \frac{R_p}{K_p}$ for convenience and proceed as follows.

The interconnected system is now

$$\begin{bmatrix} \dot{q} \\ \dot{p} \\ \dot{\zeta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & -c_p & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \nabla_q H_{cl} \\ \nabla_p H_{cl} \\ \nabla_\zeta H_{cl} \end{bmatrix}. \tag{2-51}$$

A suitable Casimir function $C(q,p,\zeta)$ can now be found by solving

$$\begin{bmatrix} \frac{\partial C}{\partial q} & \frac{\partial C}{\partial p} & \frac{\partial C}{\partial \zeta} \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & -c_p & -1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \tag{2-52}$$

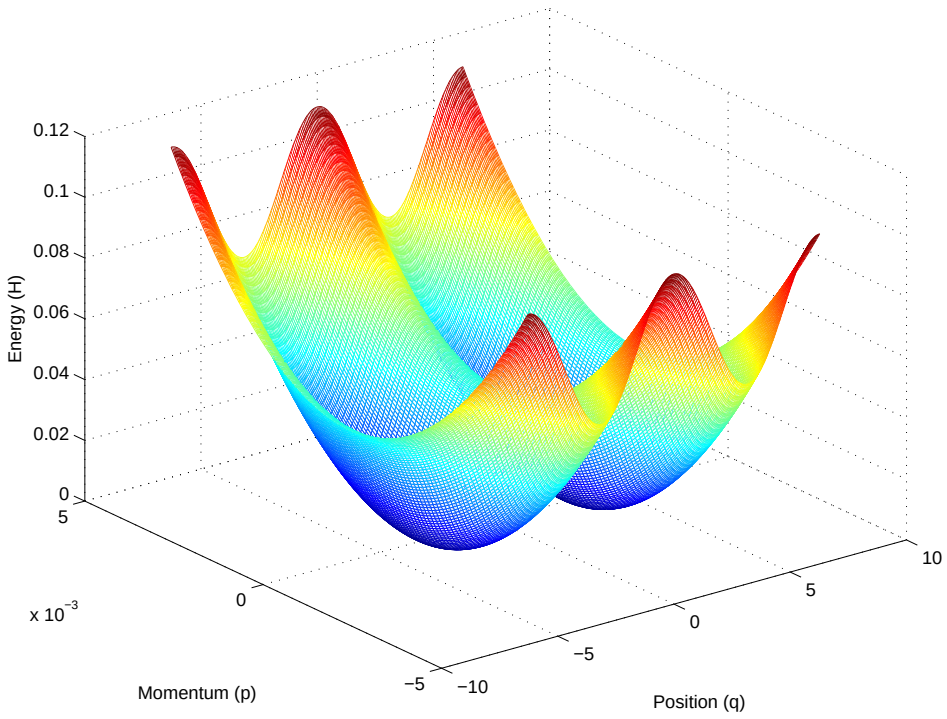


Figure 2-2: Inverted Pendulum: System Hamiltonian

which leads to $C(q,p,\zeta) = K(q-\zeta)$ and candidate Lyapunov functions

$$V(q,p,\zeta) = \frac{1}{2J_p}p^2 + M_p g_p l_p(1+\cos q) + H_c(\zeta) + K(q-\zeta) \tag{2-53}$$

with the functions $H_c(\zeta)$ and $K(q-\zeta)$ still to be determined suitably so as to ensure the desired local minimum condition $(q^*, 0, \zeta^*)$.

The equilibrium assignment gives

$$\frac{\partial V}{\partial q}(q^*,0,\zeta^*) = M_p g_p l_p(-\sin q^*) + \frac{\partial K}{\partial q}(q^*-\zeta^*) = 0 \tag{2-54}$$

$$\frac{\partial V}{\partial p}(q^*,0,\zeta^*) = 0 \tag{2-55}$$

$$\frac{\partial V}{\partial \zeta}(q^*,0,\zeta^*) = \frac{\partial H_c}{\partial \zeta}(\zeta^*) - \frac{\partial K}{\partial \zeta}(q^*-\zeta^*) = 0 \tag{2-56}$$

The condition for ensuring the minimum condition gives

$$\begin{bmatrix} M_p g_p l_p \cos q^* + \frac{\partial^2 K}{\partial q^2}(q^*-\zeta^*) & 0 & -\frac{\partial^2 K}{\partial q \partial \zeta}(q^*-\zeta^*) \\ 0 & 1 & 0 \\ -\frac{\partial^2 K}{\partial \zeta \partial q}(q^*-\zeta^*) & 0 & \frac{\partial^2 K}{\partial \zeta^2}(q^*-\zeta^*) + \frac{\partial^2 H_c}{\partial \zeta^2}(\zeta^*) \end{bmatrix} > 0, \tag{2-57}$$

allowing for many possible solutions.

For contrast, proceed again find a suitable Casimir function for this system via the approach given in Section 2-4.

Using Eq. (2-39) to Eq. (2-42), the following holds. Eq. (2-39) and Eq. (2-41) are already satisfied since $J_c = 0$ and $R_c = 0$. Eq. (2-40) is the dissipation obstacle which means in this case that only the coordinates of position, q, can be shaped.

Solve Eq. (2-42) which gives,

$$S(q) = q + \kappa \tag{2-58}$$

as the family of suitable Casimir functions.

It is now possible to shape the controller Hamiltonian $H_c(S(q))$ such that the closed loop energy has a minimum at the desired q_*.

Choosing,

$$H_c(q) = k_1\frac{q^2}{2} - M_p g_p l_p(1+\cos q), \tag{2-59}$$

with $k_1 = 5$, it is possible to achieve stabilisation at the desired $q_* = 0$.

The system energy, desired energy and controller Hamiltonian are shown in Figure 2-3.

The simulation results for control by interconnection for the given system is shown in Figure 2-4.

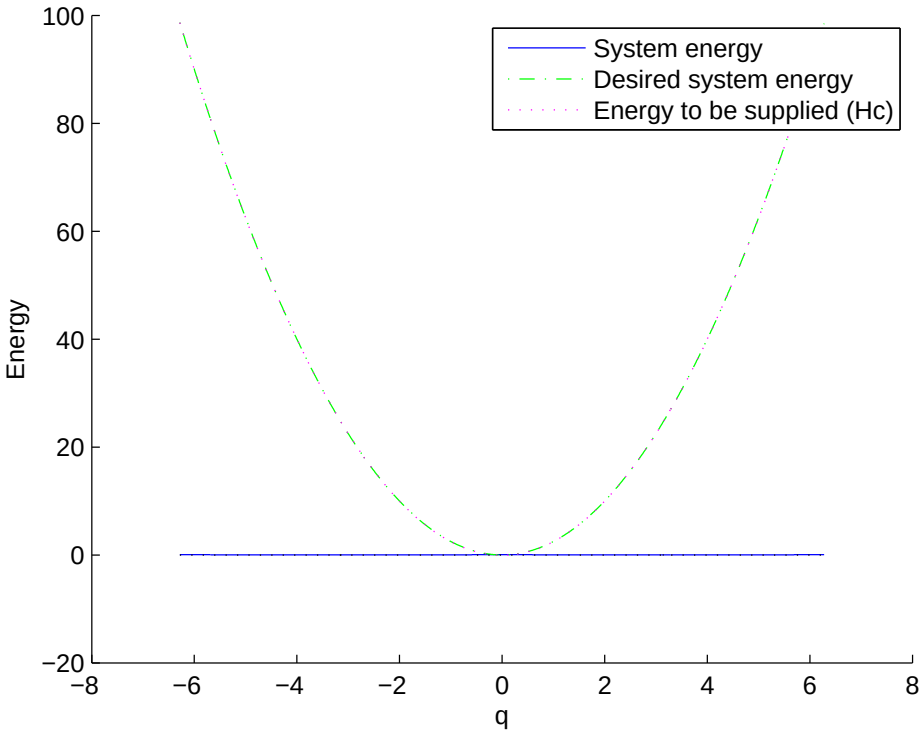


Figure 2-3: Inverted Pendulum: System, Controller and Closed Loop energy in q

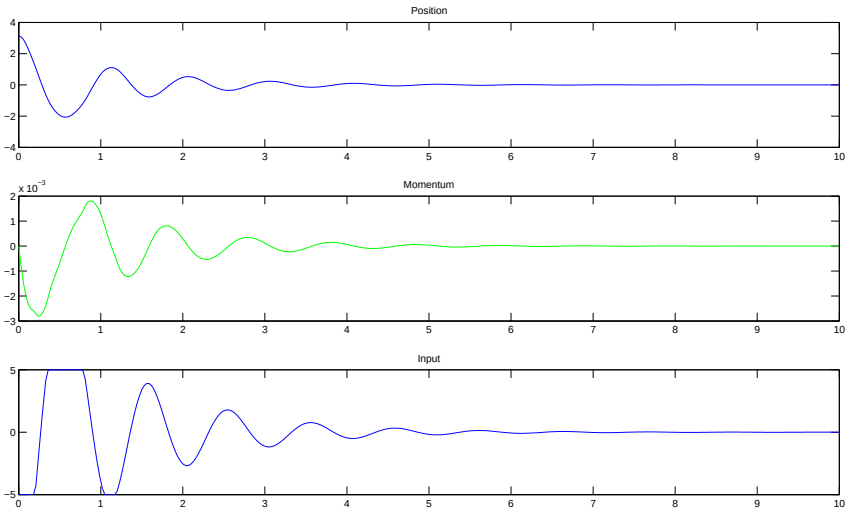


Figure 2-4: Cbl of Inverted Pendulum: Final Trajectory

2-6 Summary

In this chapter, the PH framework has been presented along with a number of control methods for PH systems. It is seen that most of these methods rely on the availability of state information. The CbI methodology has also been presented and with the aid of an example, it has been shown how CbI can be used to shape the energy of the system in the absence of state information. It is also seen that CbI requires solving a set of Partial Differential Equations (PDEs) which allow for many solutions for the choice of Casimir function and controller Hamiltonian and there does not appear to be an intuitive way to make the best choice.

Chapter 3

Reinforcement Learning

In a Reinforcement Learning (RL) problem, an agent learns the optimal strategy to complete some task based on its interactions with the environment. The agent takes some action (e.g. control strategy) and adapts its strategy based on the feedback from the environment in such a manner as to maximise a numerical reward [23].

Figure 3-1 shows the schematic of RL. At some point in time, the environment is in a state s_t. The agent takes an action a_t. As a result, the environment changes its state to s_{t+1} and gives the agent a reward r_{t+1}. Based on this reward, the agent understands whether the action it took was favourable or not and accordingly takes the next action.

3-1 Elements of Reinforcement Learning

Apart from the agent and the environment, a reinforcement learning system has the following main elements [23].

1. **Policy**: A policy governs the actions to be taken by the agent. Simply put, one can think of a policy as a mapping from the states of the environment to the actions to be taken when in those states. The policy defines the behaviour of the agent and may be stochastic in nature. Depending on the nature of the problem, the policy may be

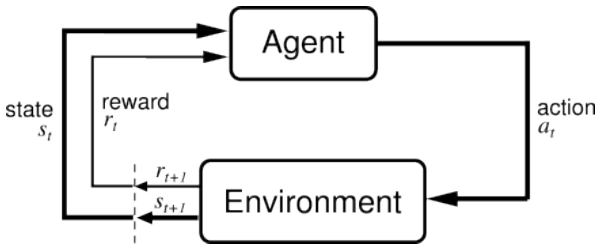


Figure 3-1: Schematic of Reinforcement Learning

a simple lookup table or even involve a complex search problem [23]. Thus, the policy essentially dictates what action to take in a certain state. In terms of control theory, the policy can be the control law to be followed.

2. **Reward function**: The reward function defines the goal in the reinforcement learning problem. The agent always tries to maximise the cumulative reward it can receive over time. So the reward function can be thought of as a measure of how desirable a certain state or a certain action is. Reward functions may not be altered by the agent [23].
3. **Value Function**: The reward function indicates how profitable or how desirable it is to take an action in the immediate sense. Contrastingly, the value function specifies what is the best course of action in the long run. The value of the state can be thought of as the total amount of reward an agent can expect to accumulate over time, starting from that state [23].
4. **Environment Model**: The model of the environment is something which mimics the behaviour of the environment. Given the current state and the action, it can predict the resultant next state and reward. It is used for planning purposes by the agent.

An important thing to note here is the difference between the reward and value. A state may have a high reward but a low value since it does not lead to any states which are beneficial in the long run. The opposite is also possible. A state may have a low reward but a high value because it leads to next states which offer a very high reward. Values are estimated from observing and interacting with the environment. Nevertheless, when making and evaluating decisions, the values are more important than the rewards [23].

3-2 Markov Decision Process

In this sub-section, the concepts of discrete time RL are introduced. An RL algorithm can be used to solve problems modelled as a Markov Decision Process (MDP). An MDP is denoted using $\langle X, U, f, \rho \rangle$, where X denotes the state space, U denotes the action space, $f : X \times U \times X \mapsto [0, \infty)$ is the state transition probability density function and $\rho : X \times U \times X \mapsto \mathbb{R}$ is the reward function.

The agent takes an action u_k from a state x_k to transition to a state x_{k+1} and for this, it receives an immediate reward r_{k+1},

$$r_{k+1} = \rho(x_k, u_k, x_{k+1}).$$

The reward function ρ is assumed to be bounded and the rewards depends on the previous state, current state and action taken. The goal of the RL agent is to find a policy which, when followed, allows it to accumulate the maximum reward in the long term. Thus, the agent wants to find a policy π, which maximises the expected value of some function g of the reward. This motivates us to define a cost function for the policy as

$$J(\pi) = E\{g(r_1, r_2, ...)|\pi\}. \tag{3-1}$$

In practice, usually the function g is either the discounted sum of rewards or the average reward [2]. In this thesis, only the discounted reward setting is considered[1].

[1]Interested readers may refer to [23, 2] for details on the average reward setting.

3-2-1 Discounted Reward and Value Functions

For the discounted reward setting, the cost function J is

$$J(\pi) = E\left\{\sum_{k=0}^{\infty} \gamma^k r_{k+1} | x_0, \pi\right\} \tag{3-2}$$

where $\gamma \in [0, 1)$ is reward discount factor and the initial state is $x_0 \in X$. During the learning process, the agent evaluates the cost J for a policy π, this is called policy evaluation. The resulting estimate of the cost J is called the value function. If the value function only depends on the state x, it is called the state value function. From the initial state the policy is followed and the state value function is

$$V^{\pi}(x) = E\left\{\sum_{k=0}^{\infty} \gamma^k r_{k+1} | x_0 = x, \pi\right\}. \tag{3-3}$$

Similarly, the state action value function defines the expected return starting from an initial state, applying some action, and then following the policy. The state action value function is

$$Q^{\pi}(x, u) = E\left\{\sum_{k=0}^{\infty} \gamma^k r_{k+1} | x_0 = x, u_0 = u, \pi\right\}. \tag{3-4}$$

The optimal policy is the policy that maximises these functions.

$$V^*(x) = \max_{\pi} V^{\pi}(x), \tag{3-5}$$

$$Q^*(x) = \max_{\pi} Q^{\pi}(x, u), \tag{3-6}$$

with $V^*(x)$ and $Q^*(x)$ the optimal state value function and state action value function respectively.

Eq. (3-3) and Eq. (3-4) can also be put in the recursive form [24]. The state value function is then given by

$$V^{\pi}(x) = E\left\{\rho(x, u, x') + \gamma V^{\pi}(x')\right\}, \tag{3-7}$$

and the state action value function by

$$Q^{\pi}(x, u) = E\left\{\rho(x, u, x') + \gamma Q^{\pi}(x', u')\right\}, \tag{3-8}$$

where x' is drawn from the probability distribution function $f(x, u, \cdot)$ and u' is drawn from $\pi(x', \cdot)$.

Eq. (3-7) and Eq. (3-8) are known as the Bellman Equations and optimality conditions for them are given by the Bellman optimality equations (Eq. (3-9) and Eq. (3-10)) [23]

$$V^*(x) = \max_{u} E\left\{\rho(x, u, x') + \gamma V^*(x')\right\}, \tag{3-9}$$

$$Q^*(x, u) = \max_{u} E\left\{\rho(x, u, x') + \gamma \max_{u'} Q^*(x', u')\right\}. \tag{3-10}$$

3-3 Types of Reinforcement Learning Algorithms

RL has several types of algorithms and these can broadly be classified into three categories:

- Actor only
- Critic only
- Actor-Critic

where actor refers to the policy function (or controller) and critic refers to the value function.

3-3-1 Actor only

Actor only algorithms typically work with a parametrised family of policies over which an optimisation procedure can be directly applied to select the best action. This has the advantage that a spectrum of continuous actions can be generated, but the optimisation procedures used (usually policy gradient methods) suffer from a high variance in the estimates of the gradient, thus slowing down the learning process [2].

Typically, policy gradient methods are actor only and do not use any stored value function. The policy (π) is parametrised by a vector $\vartheta \in \mathbb{R}^p$. Since the cost function is a function of the policy, it follows that the cost function $J(\pi)$ is a function of ϑ. Assuming that the parametrisation is differentiable, the gradient of the cost function with respect to ϑ is given by [2]

$$\nabla_\vartheta J = \frac{\partial J}{\partial \pi_\vartheta} \frac{\partial \pi_\vartheta}{\partial \vartheta}. \tag{3-11}$$

Standard optimisation techniques exist to find a local optimum of the cost function J. For example, a simple gradient ascent method would provide the following update equation,

$$\vartheta_{k+1} = \vartheta_k + \alpha_{a,k} \nabla_\vartheta J_k \tag{3-12}$$

where $\alpha_{a,k} > 0$ is a small enough learning rate that ensures that[2] $J(\vartheta_{k+1}) > J(\vartheta_k)$.

Such methods converge strongly to a local optimum and thus, a big advantage of using actor-only methods is their strong convergence property.

3-3-2 Critic only

Critic only methods use a state action value function and no explicit function for a policy [2]. A simple and intuitive way of deriving a policy for critic-only methods is to select "greedy actions" [23] i.e, the action for which the value function gives the highest expected return. The disadvantage of these methods is that to find out the optimal greedy action, one must perform an optimisation procedure in every state and this can be computationally

[2]The cost function here has been defined as the expected reward so we wish to maximise it. In case the cost function is defined in such a way that it should be minimised, the plus sign in (3-12) is replaced with a minus sign, resulting in $J(\vartheta_{k+1}) < J(\vartheta_k)$.

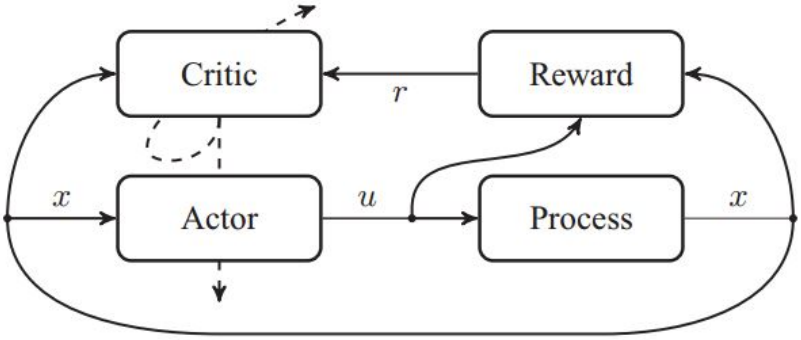


Figure 3-2: Schematic of actor-critic algorithm [2]

intensive. Critic only methods usually either work with discrete action spaces or if the action space is continuous, it is discretised using suitable function approximation. This approach however, undermines the ability to use continuous action spaces and to find the true optimum. Examples of critic-only methods include Q-learning and SARSA [2]. These methods first use function approximation for continuous action spaces to discretize it and then learn the optimal value function by finding online an approximate solution to Eq. (3-5) or Eq. (3-6) (as the case may be). Finally, the policy is calculated by

$$\pi(x) = \arg\max_u Q(x, u). \tag{3-13}$$

The downside to using critic only algorithms is that there is no guarantee on the optimality or near optimality of the resulting policy if used with just any approximated value function [2] as has been shown in [25] and [26].

3-3-3 Actor-Critic

Actor-critic methods, as the name indicates seek to combine the advantages of both the actor only and critic only methods. The parametrised policy provides the advantage of computing continuous actions without the need for optimisation procedures on a value function whereas the critic provides the actor with low-variance knowledge of the performance. The critic's estimate of the expected return allows the actor to update itself with gradients that have lower variance, thus speeding up the learning process considerably.

As shown in Figure 3-2, the learning agent is split into the actor (or policy function) and the critic (or value function). The actor generates a control input u, based on the current state x. The critic processes the reward r it receives from the environment and uses that to evaluate the quality of the current policy. Based on this, the critic updates the actor and itself. Let the value function be parametrised by $\theta \in \mathbb{R}^q$. We shall denote this with $V_\theta(x)$ or $Q_\theta(x, u)$. Assuming a linear parametrisation, we denote the features with ϕ and thus,

$$V_\theta(x) = \theta^T \phi(x) \tag{3-14}$$

or

$$Q_\theta(x, u) = \theta^T \phi(x, u) \tag{3-15}$$

The policy is parametrised by $\vartheta \in \mathbb{R}^p$ and will be denoted with $\pi_\vartheta(x, u)$.

The goal of the RL algorithm is to find the best policy possible for the MDP and for this, the critic must be able to accurately evaluate a given policy. Thus, the critic must find an approximate solution to the Bellman equation for that policy and the difference between the left and right hand sides of the Bellman equation is called the TD (Temporal Difference) error and is used to update the critic. The TD error is estimated as

$$\delta_{k+1} = r_{k+1} + \gamma V_{\theta_k}(x_{k+1}) - V_{\theta_k}(x_k) \tag{3-16}$$

where $\gamma \in (0, 1]$ is the reward discount factor. Using this in the critic update,

$$\theta_{k+1} = \theta_k + \alpha_{c,k}\delta_{k+1}\nabla_\theta V_{\theta_k}(x_k) \tag{3-17}$$

where $\alpha_{c,k} > 0$ is the learning rate of the critic. Using Eq. (3-14), this can be reduced to

$$\theta_{k+1} = \theta_k + \alpha_{c,k}\delta_{k+1}\phi(x_k) \tag{3-18}$$

This method is known as TD(0) learning. If we use eligibility traces, then let us have the eligibility trace vector for all q features at time k denoted by $z_k \in \mathbb{R}^q$ and update as

$$z_{k+1} = \lambda\gamma z_k + \nabla_\theta V_{\theta_k}(x_k) \tag{3-19}$$

This decays with time by a factor of $\lambda\gamma$ with $\lambda \in [0, 1)$ as the trace decay rate. This makes the equation give greater credit to the more recent features and significantly speeds up the learning process. Using this eligibility trace z_k, the critic update now becomes,

$$\theta_{k+1} = \theta_k + \alpha_{c,k}\delta_{k+1}z_{k+1} \tag{3-20}$$

The actor critic template for discounted setting can now be given as

$$\delta_k = r_{k+1} + \gamma V_{\theta_k}(x_{k+1}) - V_{\theta_k}(x_k) \tag{3-21}$$
$$z_{k+1} = \lambda\gamma z_k + \nabla_\theta V_{\theta_k}(x_k) \tag{3-22}$$
$$\theta_{k+1} = \theta_k + \alpha_{c,k}\delta_{k+1}z_{k+1} \tag{3-23}$$
$$\vartheta_{k+1} = \vartheta_k + \alpha_{a,k}\nabla_\vartheta J_k \tag{3-24}$$

3-4 Past work on port-Hamiltonian systems and Reinforcement Learning

There are a number of challenges that hinder the widespread use of Passivity Based Control (PBC) and Control by Interconnection (CbI). The most notable challenges are as follows [1, 22, 12, 11]

- It involves the solution of a complex set of Partial Differential Equation (PDE)s. This can be particularly difficult in the case of multi-domain physical systems.
- Solutions are founded on stability considerations and often overlook performance considerations.

- Model uncertainties can severely affect the performance of the designed PBC law.

Control laws have also been developed using RL and applied to a variety of systems, especially in the field of robotics (e.g, in [27] and [7]) but the drawback is that RL may have an extremely slow learning rate. As a result it may take a large number of trials to learn a near optimal policy.

To overcome these challenges, [1] presents a novel way of combining RL with port-Hamiltonian (PH) systems. The advantages of combining these two are that [1, 10],

- The control goal can be specified in a "local" fashion without considering the global system behaviour, say by defining the reward to be 0 in a small neighbourhood of the desired goal and a negative reward at all other points. This is especially useful for complex systems.
- Performance criteria can be included in the learning algorithm in addition to the stability properties provided by PBC and CbI.
- The introduction of learning provides a robustness to model uncertainty.

This approach of combining RL with PBC has produced promising results, as collected in [10].

3-5 Reinforcement Learning for port-Hamiltonian systems

In [1], the authors have applied RL to Energy Shaping (ES)-Damping Injection (DI). Consider a PH system of the form given in Eq. (2-1). Defining the added energy function as

$$H_a(x) = H_d(x) - H(x), \tag{3-25}$$

the feedback control law

$$u = g^{\dagger}(x)F(x)\frac{\partial H_a}{\partial x}(x) - K(x)y \tag{3-26}$$

satisfies the energy shaping and damping injection if the condition

$$\begin{bmatrix} g^{\perp}(x)F^T(x) \\ g^T(x) \end{bmatrix} \frac{\partial H_a}{\partial x}(x) = 0 \tag{3-27}$$

is satisfied where $F(x) = J(x) - R(x)$, $g^{\dagger} = (g^T(x)g(x))^{-1}g^T(x)$ and $K(x) = K^T(x) \geq 0$ is a positive semi-definite damping matrix that is used for damping injection [1].

The PDE (Eq. (3-26)) can now be reformulated in terms of desired closed loop energy $H_d(x)$ by using Eq. (3-25) as

$$\begin{bmatrix} g^{\perp}(x)F^T(x) \\ g^T(x) \end{bmatrix} (\nabla_x H_d(x) - \nabla_x H(x)) = 0 \tag{3-28}$$

and denoting the kernel of $A(x) = \begin{bmatrix} g^{\perp}(x)F^T(x) \\ g^T(x) \end{bmatrix}$ as

$$\ker(A(x)) = \left\{ N(x) \in \mathbb{R}^{n \times b} \colon A(x)N(x) = 0 \right\}. \tag{3-29}$$

This equation now reduces to

$$\nabla_x H_d(x) - \nabla_x H(x) = N(x)a \tag{3-30}$$

with $a \in \mathbb{R}^b$. Now if the state vector can be written as $x = \begin{bmatrix} w^T & z^T \end{bmatrix}$, where $z \in \mathbb{R}^c$ and $w \in \mathbb{R}^d$, $c + d = n$ corresponding to the zero and non-zero elements of $N(x)$ such that

$$\begin{bmatrix} \nabla_w H_d(x) \\ \nabla_z H_d(x) \end{bmatrix} - \begin{bmatrix} \nabla_w H(x) \\ \nabla_z H(x) \end{bmatrix} = \begin{bmatrix} N_w(x) \\ 0 \end{bmatrix} a \tag{3-31}$$

then it is clear that $\nabla_z H_d(x) = \nabla_z H(x)$ (also called the matching condition) and hence $\nabla_z H_d(x)$ cannot be freely chosen and only $\nabla_w H_d(x)$ can be chosen freely [1]. This stems from the *dissipation obstacle* (discussed in Section 2-4-1).

Parametrise this desired energy as

$$\hat{H}_d(x, \xi) := H(x) + \xi^T \phi_H(w) + \bar{H}_d(w) + C \tag{3-32}$$

where $\xi^T \phi_H(w)$ represents the linear in parameters function approximator with $\xi \in \mathbb{R}^e$ a parameter vector and $\phi_H(w)$ an appropriately chosen basis function (with e large enough to represent the desired closed loop energy), $\bar{H}_d(w)$ is an arbitrary function of w, and C chosen to render $\hat{H}_d(x, \xi)$ non-negative. Similarly, the desired damping matrix $K(x)$ can be parametrised as

$$[\hat{K}(x, \Psi)]_{ij} = \sum_{l=1}^{f} [\Psi]_{ijl} [\phi_K(x)]_l \tag{3-33}$$

with $\Psi \in \mathbb{R}^{m \times m \times f}$ and $[\Psi]_{ijl} = [\Psi]_{jil}$. The control law (Eq. (3-26)) now becomes

$$u(x, \xi, \Psi) = g^{\dagger} F \begin{bmatrix} \nabla_w \hat{H}_d(x, \xi) - \nabla_w H(x) \\ 0 \end{bmatrix} - \hat{K}(x, \Psi) g^T(x) \nabla_x \hat{H}_d(x, \xi) \tag{3-34}$$

A Temporal Difference (TD) actor-critic algorithm is chosen for the reinforcement learning algorithm because of the advantages mentioned in Section 3-3. The policy is chosen to be equal to the control law. At any time step k,

$$\hat{\pi}(x_k, \xi_k, \Psi_k) = u(x_k, \xi_k, \Psi_k). \tag{3-35}$$

Assuming some saturation function $\varrho : \mathbb{R}^m \to S$, $S \subset \mathbb{R}^m$, such that $\varrho(u(x)) \in S \ \forall u$ where S is the set of all valid control inputs. An exploration term is added keeping in mind the saturation constraint as

$$\Delta \hat{u}_k = u_k - \hat{\pi}(x_k, \xi_k, \Psi_k). \tag{3-36}$$

The gradients of the saturated policy can be calculated as [1]

$$\nabla_{\xi} \varrho(\hat{\pi}) = \nabla_{\hat{\pi}} \varrho(\hat{\pi}) \nabla_{\xi} \hat{\pi} \tag{3-37}$$

$$\nabla_{[\Psi]_{ij}} \varrho(\hat{\pi}) = \nabla_{\hat{\pi}} \varrho(\hat{\pi}) \nabla_{[\Psi]_{ij}} \hat{\pi} \tag{3-38}$$

It should be noted that the gradient of the saturation function needs to be calculated and this can usually be done analytically. Using the update equations given in Section 3-3, the update for the parameters of the desired Hamiltonian are [1]

$$\xi_{k+1} = \xi_k + \alpha_{a,\xi}\delta_{k+1}\Delta\bar{u}_k\nabla_\xi\varrho(\hat{\pi}(x_k,\xi_k,\Psi_k) \tag{3-39}$$

$$[\Psi_{k+1}]_{ij} = [\Psi_k]_{ij} + \alpha_{a,[\Psi]_{ij}}\delta_{k+1}\Delta\bar{u}_k\nabla_{[\Psi]_{ij}}\varrho(\hat{\pi}(x_k,\xi_k,\Psi_k) \tag{3-40}$$

And the control law is given by

$$u = \bar{\phi}_1(x) + \sum_i \xi_i\bar{\phi}_{2,i}(x) + \sum_j \bar{\Psi}_j\bar{\phi}_{3,j}(x) \tag{3-41}$$

with $\bar{\Psi}$ representing the stacked version of Ψ which is possible if the policy is parametrised in an affine way [1].

3-6 Solving algebraic IDA-PBC using Reinforcement Learning

Consider the Interconnection and Damping Assignment Passivity Based Control (IDA-PBC) control law given in Eq. (2-32) which can be re-written as [12]

$$u(x) = \beta_(x) = ((g^T(x)g(x))^{-1}g^T(x)(F_d(x)\nabla_x H_d(x) - F(x)) \tag{3-42}$$

with the well known matching condition

$$g^\perp(x)(F_d(x)\nabla_x H_d(x) - F(x)\nabla_x H(x)) = 0 \tag{3-43}$$

where $F(x) = [J(x) - R(x)]$ and similarly $F_d(x) = [J_d(x) - R_d(x)]$. In algebraic IDA-PBC, the desired energy function $H_d(x)$ is fixed and is typically quadratic in increments. As a result, Eq. (3-43) becomes an algebraic equation in unknown elements of $F_d(x)$. Parametrise the unknown matrix $F_d(x)$ as $F_d(x,\xi)$ to obtain

$$u(x,\xi) = ((g^T(x)g(x))^{-1}g^T(x)(\underbrace{\xi^T\phi(x)}_{F_d(x,\xi)}\nabla_x H_d(x) - F(x)) \tag{3-44}$$

where ξ is the unknown parameter matrix. The parameter vector ξ can be calculated using reinforcement learning in a similar fashion to the approach used in section 3-5. With the policy (actor) update equation as [12]

$$\xi_{i,k+1} = \xi_{i,k} + \alpha_{a,\xi}\delta_{k+1}\Delta u_k\nabla_{\xi_{i,k}}u_k(x,\xi) \tag{3-45}$$

where Δu_k is a Gaussian noise exploration term. Thus, the authors of [12] have solved algebraic IDA-PBC using RL and have implemented in for the swing up and stabilisation of a simple pendulum and for the stabilisation of a magnetic levitation system.

3-7 Summary

This chapter introduced the RL framework and presented the different types of RL, along with the advantages and disadvantages of the different methods. Some past work that has been done on PH systems and RL has also been presented and it is these successes that are the motivation for the development of the Control by Interconnection - Actor Critic (CBI-AC) algorithm presented in the next chapter.

Chapter 4

Control by Interconnection using Reinforcement Learning

As can be seen from the example given in Section 2-5, there are many possible solutions to finding a suitable Casimir function and controller Hamiltonian. However, particularly for complex systems, this task can be tedious as it involves solving multiple Partial Differential Equations (PDEs). Moreover, although there exist strong methods to analyse stability for port-Hamiltonian (PH) systems, it is a difficult task to incorporate performance criteria into the system. Reinforcement Learning (RL) on the other hand, is a suitable method to incorporate such criteria. Thus, motivated by the past work mentioned in Section 3-4, this thesis attempts to use RL to learn a suitable Casimir function for Control by Interconnection (CbI). The added advantage is that through the reward function in RL, it is possible to learn a controller that meets certain desired performance criteria as well.

4-1 Formulation as a Reinforcement Learning problem

Recall the input-state-output PH system as introduced in Eq. (2-1) of Section 2-2:

$$\begin{aligned} \dot{x} &= [J(x) - R(x)]\frac{\partial H(x)}{\partial x} + g(x)u, \\ y &= g^T(x)\frac{\partial H(x)}{\partial x}, \end{aligned} \tag{4-1}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$, $m \leq n$ is the control input, $J(x), R(x) : \mathbb{R}^n \mapsto \mathbb{R}^{n \times n}$, $J(x)$ and $R(x)$ are the interconnection and damping matrices respectively, $H(x) : \mathbb{R}^n \mapsto \mathbb{R}$ is the Hamiltonian, and $u, y \in \mathbb{R}^m$ are the input and output variables and $g(x) : \mathbb{R}^n \mapsto \mathbb{R}^{n \times m}$ is the input matrix.

Let us choose the controller as a possibly non-linear integrator as in Section 2-5.

$$\begin{aligned} \dot{\zeta} &= u_c, \\ y_c &= \frac{\partial H_c(\zeta)}{\partial \zeta}, \end{aligned} \tag{4-2}$$

where $\zeta \in \mathbb{R}^m$, $u_c, y_c \in \mathbb{R}^m$ and $H_c = \frac{1}{2}\zeta^T\zeta$. This is interconnected with the plant system using the standard power preserving interconnection,

$$\begin{bmatrix} u \\ u_c \end{bmatrix} = \begin{bmatrix} 0 & -I_m \\ I_m & 0 \end{bmatrix} \begin{bmatrix} y \\ y_c \end{bmatrix} \tag{4-3}$$

where I_m is the identity matrix of dimension m.

Depending on whether dissipation is present in the plant and on what states, the plant states can be split into the shapeable and non-shapeable components using the approach given in [1] (introduced briefly in this thesis in Section 3-5). The plant states can thus be written as $x = \begin{bmatrix} x_s \\ x_{ns} \end{bmatrix}$ where x_s denotes the shapeable components and x_{ns} denotes the non-shapeable components. The need for this is dictated by the dissipation obstacle (introduced in Section 2-4-1) as the Casimir function is function of only the shapeable components.

Applying the invariance condition, the Casimir function is $K(\zeta - x_s)$ which can be re-written in the form $\zeta = S(x_s)$.

The control law for the system is thus, $u = -y_c = -\frac{\partial H_c(\zeta)}{\partial \zeta} = -S(x_s)$, where $S(x_s)$ is as yet some unknown function. We can approximate this function using some differentiable linear-in-parameters basis function as

$$S(x_s) = \vartheta^T \varphi(x_s) \tag{4-4}$$

where $\varphi(x_s)$ is some suitable basis function.

The policy for the Actor-Critic Reinforcement Learning (ACRL) algorithm is chosen to be the approximated Casimir function,

$$\hat{\pi}(\vartheta, x_s) = S(x_s) = \vartheta^T \varphi(x_s). \tag{4-5}$$

so the control law is now

$$u = -S(x_s) = -\hat{\pi}(\vartheta, x_s). \tag{4-6}$$

We can now introduce the parameter update equations. Using a sampling time of T_s, we use the subscript k to denote the values at a time instant $k \cdot T_s$. Thus, we use x_{s_k}, ϑ_k, φ_k to denote the value of the parameters at a discrete time step k. Thus,

$$\hat{\pi}_k(\vartheta_k, \varphi_k) := \vartheta_k^T \varphi(x_{s_k}). \tag{4-7}$$

An exploration term Δu is then added to the control input in the direction of the policy. This exploration term is drawn from a zero mean normal distribution with a variance of σ^2.

To account for the control saturation, a saturation function is defined with

$$u_{sat}(x) = \varsigma(u(x)) \tag{4-8}$$

with $\varsigma : \mathbb{R}^m \mapsto S$, $S \subset \mathbb{R}^m$ such that

$$\varsigma(u(x)) \in S \quad \forall u. \tag{4-9}$$

Taking the exploration and control saturation into account, the control action becomes,

$$u_k = \varsigma(-(\hat{\pi}(\vartheta_k, x_{s_k}) + \Delta u_k)). \tag{4-10}$$

We can easily find the gradient of the policy as,

$$\nabla_\vartheta \hat{\pi}(\vartheta, \varphi, x_s) = \varphi(x_s). \tag{4-11}$$

The reward is comprised of some negative penalty on the states and is of the form,

$$r_{k+1} = \rho(x) \tag{4-12}$$

A simple example of a suitable reward function would be one that penalises the states quadratically depending on how far away they are from the desired states.

The value function can then be similarly parametrised as,

$$V(\theta, \phi, x) = \theta^T \phi(x), \tag{4-13}$$

where θ is a parameter vector and $\phi(x)$ is some suitable basis function which allows us to easily find the gradient of the value function as

$$\nabla_\theta V(\theta, \phi, x) = \phi(x). \tag{4-14}$$

Using δ_k to denote the Temporal Difference (TD) and z_k to denote the eligibility traces (as given in Section 3-3-3), we now get the critic update equations as,

$$\delta_{k+1} = r_{k+1} + \gamma V_{\theta_k}(x_k) \tag{4-15}$$
$$z_{k+1} = \lambda\gamma z_k + \nabla_\theta V_{\theta_k}(x_k) \tag{4-16}$$
$$\theta_{k+1} = \theta_k + \alpha_{c,k}\delta_k z_k \tag{4-17}$$

A positive TD means that the state the system moved towards has a higher value and it is more desirable to move in this direction. Thus, the actor is updated in the direction of the policy if the TD (Eq. (4-15)) is positive and away from the policy if the TD is negative. Thus, the actor update is

$$\vartheta_{k+1} = \vartheta_k + \alpha_a \delta_{k+1} \Delta u_k \nabla_\vartheta \hat{\pi}(x_{s_k}, \vartheta_k, \varphi_k). \tag{4-18}$$

The Hamiltonian of the interconnected system is

$$H_{cl} = H(x) + H_c(\zeta) \tag{4-19}$$
$$= H(x) + H_c(S(x_s)) \tag{4-20}$$

and the stability of the closed loop system can then be analysed numerically. The full CbI Actor-Critic (AC) algorithm is given in Algorithm 1.

Algorithm 1 Control by Interconnection - Actor Critic

Input: System (4-1), γ, λ, α_a, α_c

$z_0(x) = 0 \quad \forall x$
Initialise x_0, θ_0, ϑ_0
$k \leftarrow 1$
loop
  Execute:
  Draw $\Delta u_k \sim \mathcal{N}(0, \sigma^2)$, calculate control action $u_k = \varsigma(-(\hat{\pi}(\vartheta_k, x_{s_k}) + \Delta u_k))$
  Observe next state x_{k+1} and calculate reward $r_{k+1} = \rho(x_{k+1}, u_k)$
  Critic:
  TD: $\delta_{k+1} = r_{k+1} + \gamma V(\theta_k, x_{k+1}) - V(\theta_k, x_k)$
  Eligibility Trace: $z_{k+1} = \gamma\lambda z_k + \nabla_\theta V(\theta_k, x_k)$
  Critic Update: $\theta_{k+1} = \theta_k + \alpha_c \delta_{k+1} z_{k+1}$
  Actor:
  Actor Update:
  $\vartheta_{k+1} = \vartheta_k + \alpha_a \delta_{k+1} \Delta u_k \nabla_\vartheta \hat{\pi}(x_{s_k}, \vartheta_k)$
end loop

4-2 Mechanical Systems

To illustrate, consider a fully actuated mechanical system of the form

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -I & -\bar{R} \end{bmatrix} \begin{bmatrix} \nabla_q H(q,p) \\ \nabla_p H(q,p) \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u \tag{4-21}$$

$$y = \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} \nabla_q H(q,p) \\ \nabla_p H(q,p) \end{bmatrix} \tag{4-22}$$

with $q \in \mathbb{R}^{\bar{n}}$, $p \in \mathbb{R}^{\bar{n}}$ ($\bar{n} = (n/2), n$ even) the generalised position and momenta respectively, and $\bar{R} \in \mathbb{R}^{\bar{n} \times \bar{n}}$ the damping matrix. The system is of the form given in (4-1) with $\bar{R} > 0$ and the Hamiltonian for the system is given by

$$H(q,p) = \frac{1}{2} p^T M^{-1}(q) p + P(q) \tag{4-23}$$

with $M(q) = M^T(q) > 0$ the inertia matrix and $P(q)$ the potential energy of the system. The state vector can be split into $x_s = [q_1, q_2, \cdots, q_{\bar{n}}]^T$ and $x_{ns} = [p_1, p_2, \cdots, p_{\bar{n}}]^T$.

Using the controller structure (Eq. (4-2)) with $m = \bar{n}$, the Casimir function is known to be some function of the shapeable components and is thus parametrised as

$$S(x_s) = S(q) = \vartheta^T \varphi(x_s) = \vartheta^T \varphi(q) \tag{4-24}$$

where $q = x_s$.

The rest of the formulation follows that as given in Section 4-1. The validity of this method is demonstrated in the following sections using two examples of well known mechanical systems.

4-3 Example - Spring Mass Damper

The method is first tested using the example of a spring mass damper. A spring mass damper is a linear mechanical system with states $x = [q, p]^T$. The system matrices and Hamiltonian are respectively,

$$J(x) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \tag{4-25}$$

$$R(x) = \begin{bmatrix} 0 & 0 \\ 0 & c \end{bmatrix}, \tag{4-26}$$

$$g(x) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{4-27}$$

$$H(x) = \frac{1}{2m}p^2 + \frac{1}{2}kq^2. \tag{4-28}$$

The controller is the non-linear integrator of the form

$$\dot{\zeta} = u_c \tag{4-29}$$

$$y_c = \nabla H_c(\zeta). \tag{4-30}$$

Then $J(\zeta) = 0$, $R_c(\zeta) = 0$, $g_c = I$ (of appropriate dimensions). And we choose the controller Hamiltonian as

$$H_c(\zeta) = \frac{1}{2}\zeta^2 \tag{4-31}$$

for convenience.

Parametrising the actor as in Eq. (4-24), we define the reward function to be zero in the close vicinity of the desired set point and penalise the system at all other states.

The model parameters and simulation parameters are given in Table 4-1 and Table 4-2 respectively.

The results of our simulation are as follows. The final system trajectory is shown in Figure 4-1. It can be seen that the learned Casimir successfully stabilises the system at the desired set-point. The Hamiltonian of the closed loop system is shown in Figure 4-2. It is interesting to note that the Hamiltonian does not have a single minima at the desired point but rather also has other local minima. However in the region of interest along the trajectory from the initial condition to the set point, the system has only one local minima at the desired set point. The presence of the other local minima can be attributed to the fact that the algorithm has not sufficiently explored the rest of the state space. This is also reflected in the heat map of the value function as shown in Figure 4-3. The value function inaccurately estimates high values at the extremities of the state space, which can again be attributed to the fact that the algorithm has not explored those regions of the state space and thus has not been able to make an accurate estimate of the value function for those regions.

Table 4-1: Spring Mass Damper Model Parameters

Model Parameter	Symbol	Value	Units
Mass	m	1	kg
Friction	c	1	Nms
Spring Constant	k	2.5	N/m

Table 4-2: Spring Mass Damper Simulation Parameters

Simulation Parameter	Symbol	Value	Units
No. of trials	-	50	-
Time per trial	T_t	15	s
Sample time	T_s	0.01	s
Initial condition	x_0	$(0, 0)$	-
Desired set-point	x_{des}	$(-0.5, 0)$	-
Exploration variance	σ^2	0.15	-
Decay rate	γ	0.87	
Eligibility trace	λ	0.57	
Max input	u_{max}	2	N
Critic learning rate	α_c	6×10^{-4}	
Actor learning rate	α_a	7×10^{-4}	

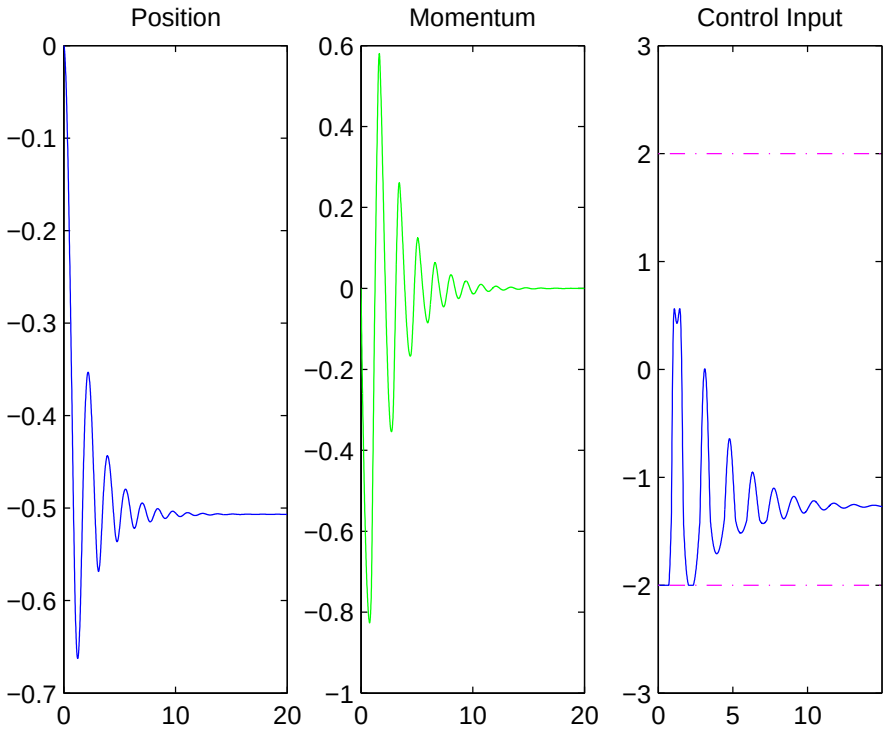


Figure 4-1: Spring Mass Damper: Final System Trajectory

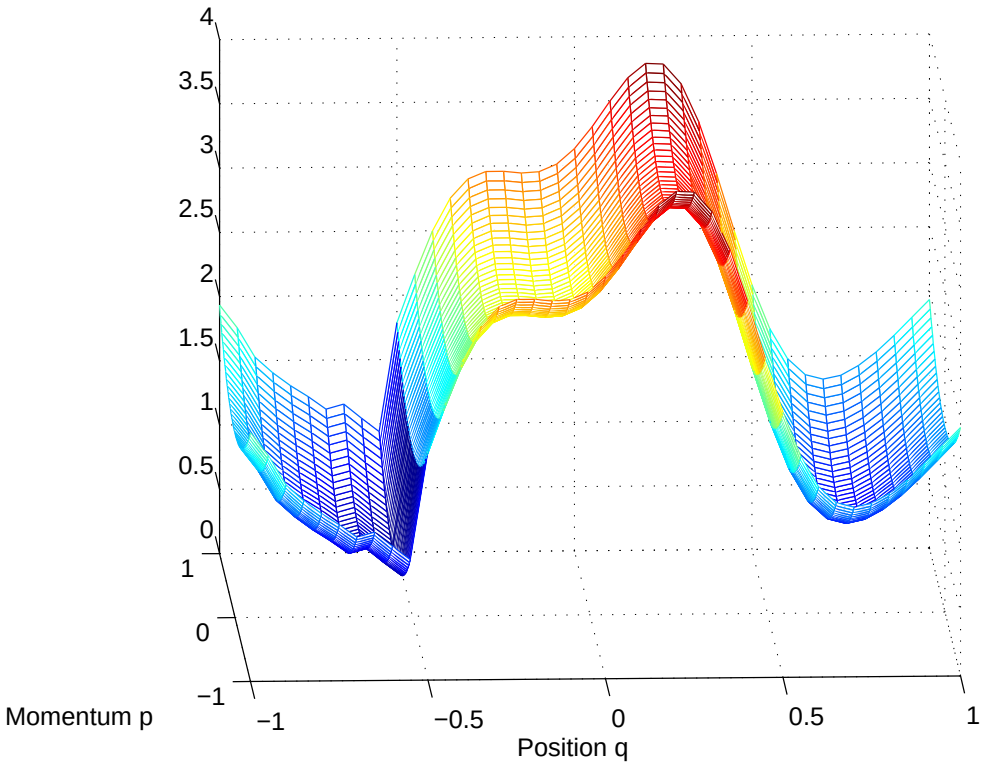


Figure 4-2: Spring Mass Damper: Closed Loop System Hamiltonian
The system follows a trajectory from $q = 0$ to $q = -0.5$ and within this range, the closed loop Hamiltonian has only one minima at $q = -0.5$.

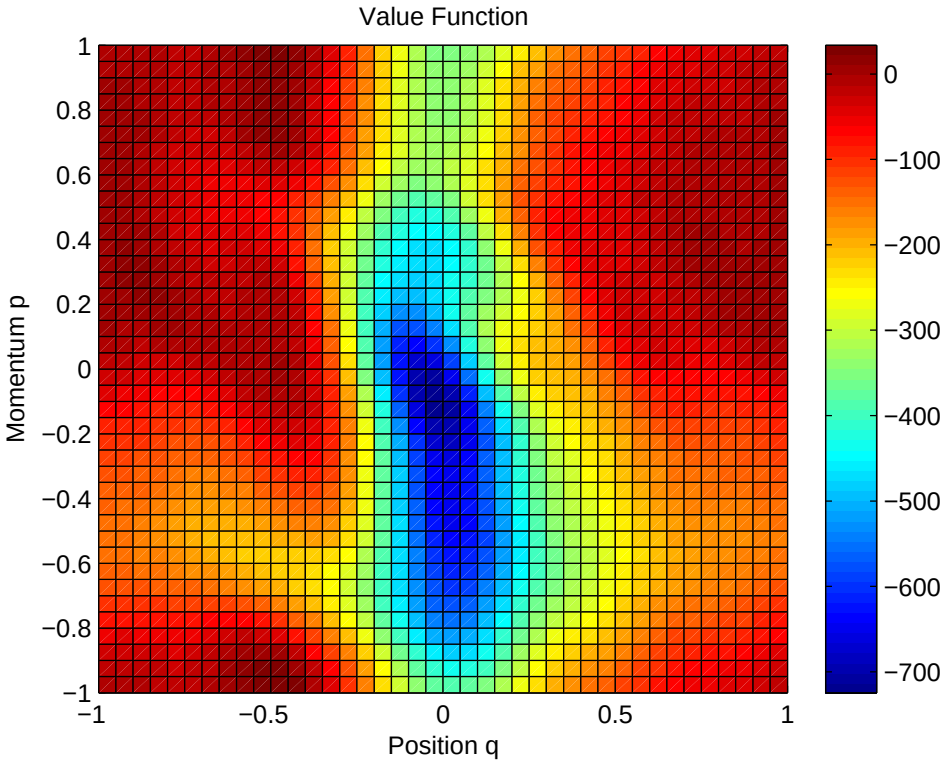


Figure 4-3: Spring Mass Damper: Value Function
The value function has a maxima at the desired set point $(-0.5, 0)$. However, it can be seen that the critic has incorrectly estimated high values at the extremities. This can be attributed to the fact that the algorithm did not explore that portion of the state space and hence has been unable to make a good estimate of the value function at those states.

Table 4-3: Inverted Pendulum Simulation Parameters

Simulation Parameter	Symbol	Value	Units
No. of trials	-	200	-
Time per trial	T_t	6	s
Sample time	T_s	0.03	s
Initial condition	x_0	$(pi, 0)$	-
Desired set-point	x_{des}	$(0, 0)$	-
Exploration variance	σ^2	0.1	-
Decay rate	γ	0.97	
Eligibility trace	λ	0.67	
Max input	u_{max}	5	N
Critic learning rate	α_c	1×10^{-2}	
Actor learning rate	α_a	1×10^{-8}	

4-4 Example - Inverted Pendulum

For the second example, an inverted pendulum is chosen as the plant system. The inverted pendulum is non-linear system that is often used as a benchmark problem in control. The same model of the system is used as in Section 2-5 and the model parameters are the same as given in Table 2-1. It is again desired that the system be stabilised in the upright position corresponding to $q = 0$.

The formulation as a reinforcement learning problem remains the same as in Section 4-2. The simulation parameters used are given in Table 4-3. From the final system trajectory using the learned controller shown in Figure 4-4, it can be seen that the system is successfully stabilised in the upright position with $(q, p) = (0, 0)$. However, it is interesting to note that in contrast to the case with the spring mass damper system, the closed loop system Hamiltonian for the inverted pendulum has a single minima at the desired point as can be seen from Figure 4-5. This is due to the fact that during the learning process, the pendulum swings through a variety of states and as a result the algorithm explores a larger portion of the total state space. This also results in a more accurate value function as can be seen from Figure 4-6. However, it can be seen that the value function still has some peaks at undesirable locations. It is possible to eliminate these peaks by reducing the critic learning rate α_c so that a better value function can be learned. However, this has the negative consequence of reducing the learning speed and since it can be seen that the policy learned is quite good, it is not necessary to do so.

The learned policy is shown in Figure 4-7. Since the Casimir is only a function of the shapeable state q, the policy reflects this and does not depend on the momentum. However, it is interesting to note that the value function takes into account all the states and thus, the policy learned is one that maximises the value function, thus indirectly taking into account the momentum of the system. The rewards earned by the algorithm over the course of the trials are shown in Figure 4-9. Figure 4-8 shows the sum of rewards per trial. It can be seen that the algorithm converges in 200 trials.

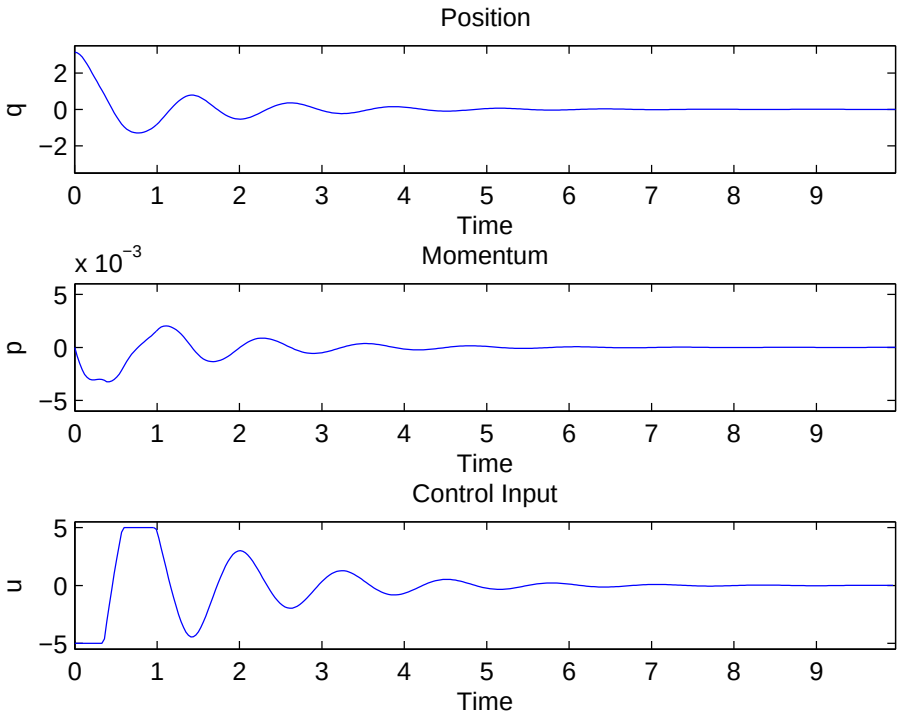


Figure 4-4: Inverted Pendulum: Final System Trajectory

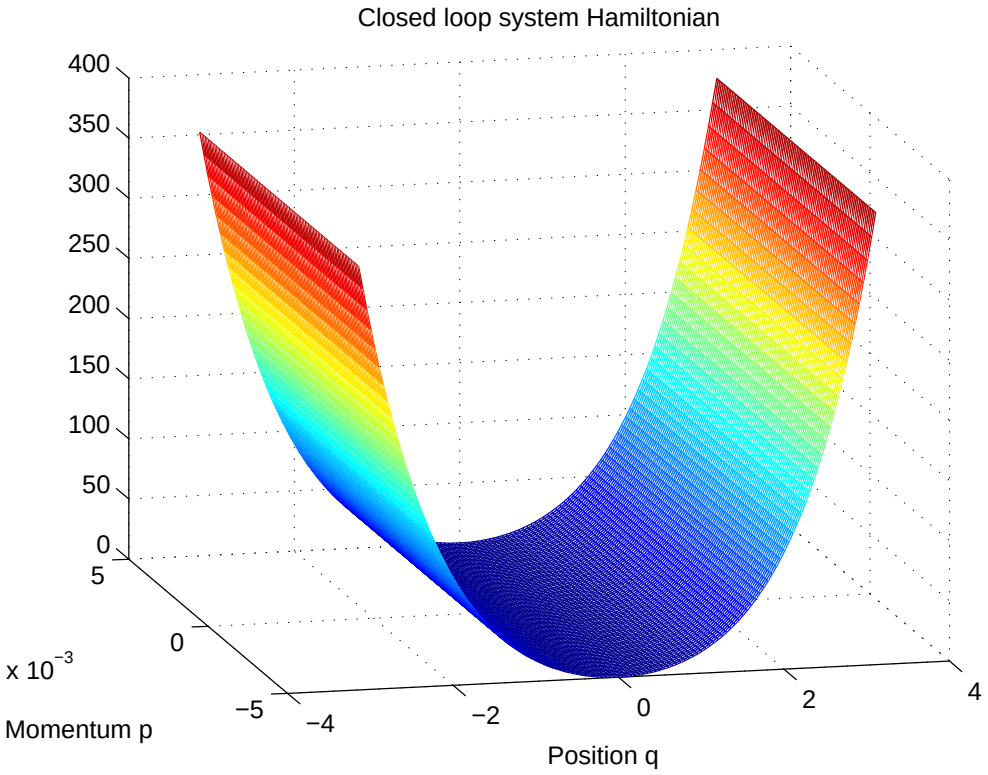


Figure 4-5: Inverted Pendulum: Final System Hamiltonian

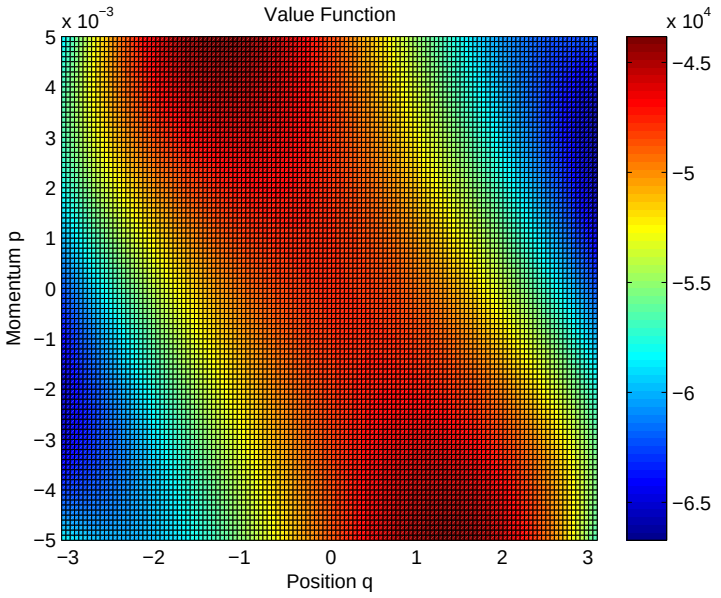


Figure 4-6: Inverted Pendulum: Value Function

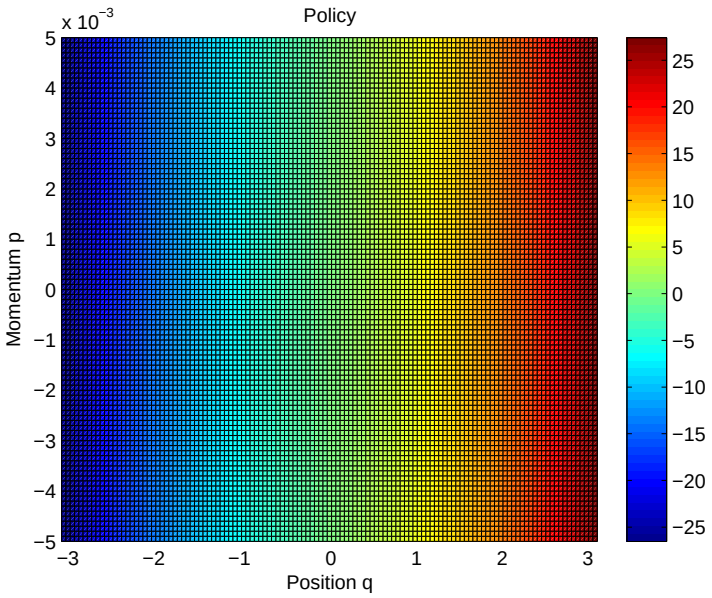


Figure 4-7: Inverted Pendulum: Policy
Since the Casimir is only a function of the shapeable component q, the policy reflects this and does not depend upon the momentum p.

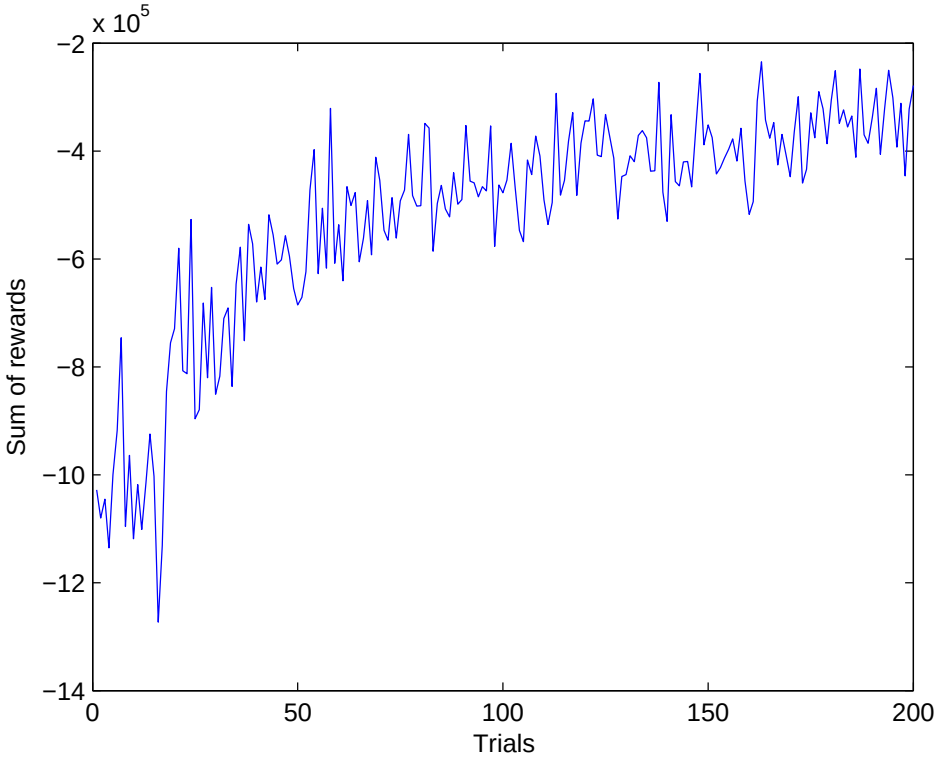


Figure 4-8: Inverted Pendulum: Sum of rewards per trial

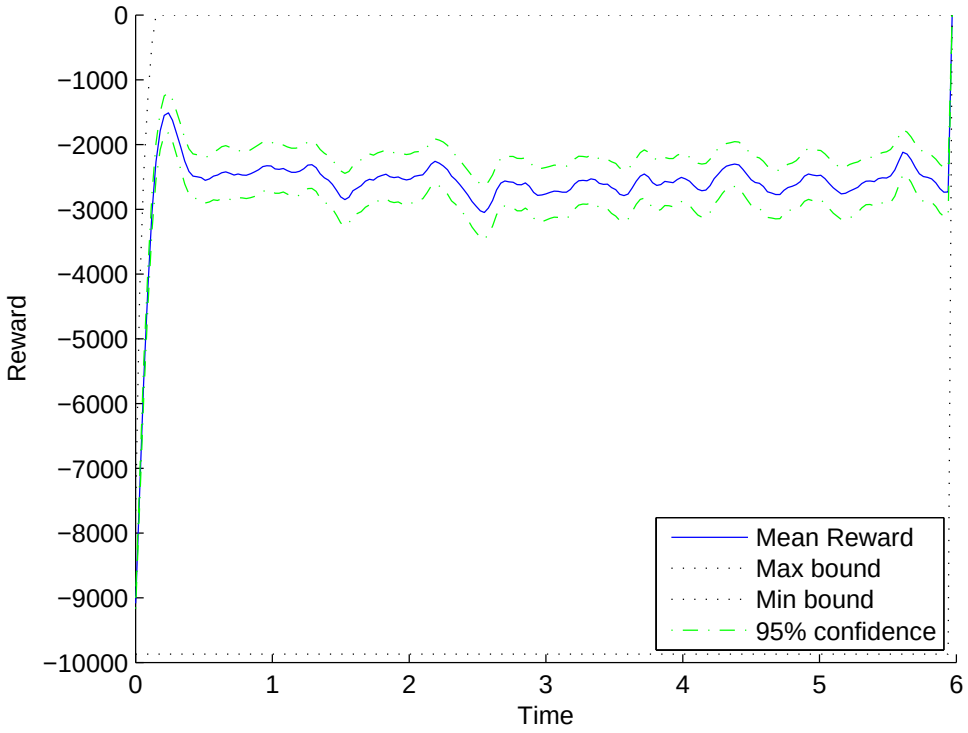


Figure 4-9: Inverted Pendulum: Rewards earned by the controller

4-5 Discussion

The feasability of using RL to find a suitable Casimir function for CbI has been shown in the preceding sections. In this section, some discussion on the method follows.

4-5-1 Choice of Reward Function

Choosing a suitable reward function is arguably one of the most important criteria for successful learning. In this thesis, the reward functions used were primarily quadratic reward functions of the form

$$r_A(x) = (x - x_{des})^T Q (x - x_{des}) \tag{4-32}$$

where Q is a suitable diagonal matrix with $Q_{(i,i)} \leq 0$. Thus, the system incurs a negative penalty at every state except the desired state when the reward is the maximum with $r_A(x_{des}) = 0$. However, although this works well, it is possible that other types of reward functions may be more suitable for certain types of systems. For the inverted pendulum system for example, the topology of the system is such that the system wraps around 2π. For such a system, using a reward function based on the cosine of the position would also be a very good choice. One such reward function might be

$$r_B(x) = Q_{(1,1)}(1 - \cos(q)) + Q_{(2,2)}p^2 \tag{4-33}$$

Additionally, if it is desired to incorporate performance criteria into the system, this can also be done using the reward function. For example, a reward function of the form

$$r_C(x) = -(x - x_{des})^T \bar{Q}^k (x - x_{des}) \tag{4-34}$$

where $\bar{Q}$ is a suitable diagonal matrix with $\bar{Q}_{(i,i)} \geq 1$ ensures that the penalty on the states increases as time progresses and the learning algorithm will thus, try to stabilise the system in the minimum possible time.

4-5-2 Saturation Function

In our algorithm, although the control input applied includes the saturation, we have not explicitly taken into account the saturation function when updating the actor. However, this can be easily corrected for by defining [22]

$$\Delta\bar{u} = -u_k - \hat{\pi}(x_{s_k}, \vartheta_k, \varphi_k). \tag{4-35}$$

Using this in the actor update, Eq. (4-18) thus becomes,

$$\vartheta_{k+1} = \vartheta_k + \alpha_a \delta_{k+1} \Delta\bar{u}_k \nabla_\vartheta \hat{\pi}(x_{s_k}, \vartheta_k, \varphi_k). \tag{4-36}$$

This has the effect that when the policy is such that the control input gets saturated, the policy will not be updated further in that direction, thus keeping the control input within the saturation bounds.

4-5-3 Learning Rates

In any RL algorithm, it is imperative to find good values for the learning rates. An improper learning rate may result in extremely poor or even no learning. However, this can be a tricky task. One of the methods used in the RL community to find suitable values of the learning rate is gridding, which is the approach that has been taken in this thesis. The possible values of the learning rates α's are gridded over a suitable range and for each point on the grid, the learning process is repeated until a satisfactory result is obtained and suitable set of values is found. Moreover, the simulation experiments can be automated to a large extent. The downside of this method, however, is that if the learning process is computationally intensive, then this may require significant time and/or processing power.

4-5-4 Function Approximation

To approximate the actor and the critic, function approximators are necessary. In this thesis, two types of function approximation have been used - the Fourier Basis function and Polynomial Basis function. Control by Interconnection - Actor Critic (CBI-AC) for the spring mass damper system was implemented using the Fourier Basis and inverted pendulum using the polynomial basis functions. It is found that the choice of basis function does not play a significant role in the implementation of the CBI-AC as long as the parameters are rich enough to describe the solution.

4-5-5 Feature Scaling

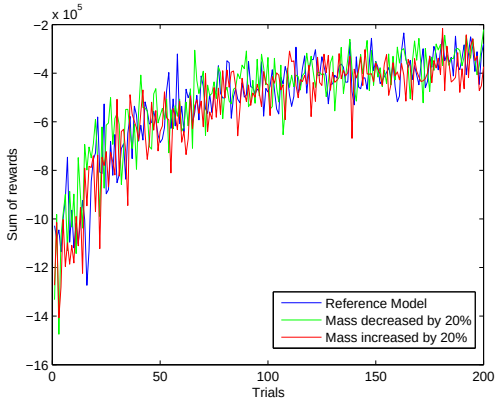
In this thesis, the states have been scaled according to

$$\bar{x}_i = \frac{x_i - x_{i,min}}{x_{i,max} - x_{i,min}}(\bar{x}_{i,max} - \bar{x}_{i,min}) + \bar{x}_{i,min} \tag{4-37}$$

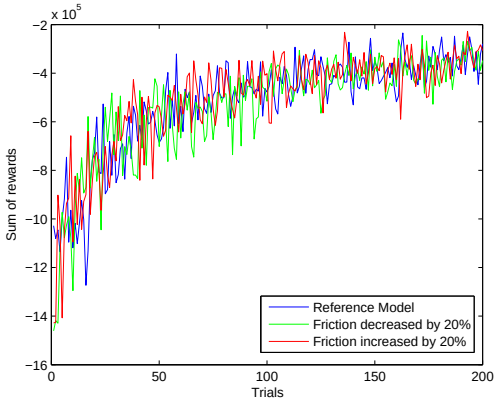
for $i = 1, \cdots, n$, with $(\bar{x}_{i,min}, \bar{x}_{i,max}) = (-1, 1)$. This is important as it ensures that each feature of the basis function has an equal contribution to the total policy and value function.

4-5-6 Robustness of the algorithm to model uncertainty

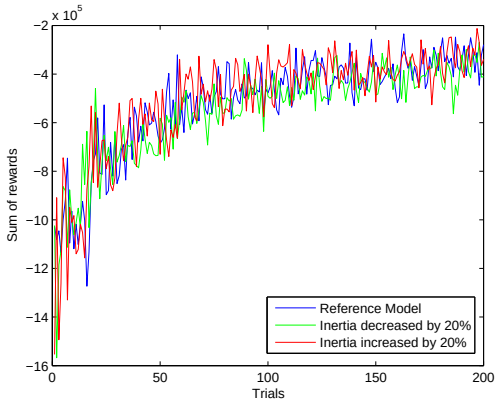
The robustness of the algorithm to variation in model parameters is a point of interest here. The learning rates of the algorithm, in particular, should not be too specific to the model otherwise a suitable gridding experiment (or some other method to find suitable learning rates) needs to performed every time to find the correct learning parameters. To check the robustness of the algorithm to uncertainties in the model, the parameters the mass, friction and inertia of the inverted pendulum were varied to $\pm 20\%$ and the CBI-AC algorithm was used to learn a controller. For all three cases, the algorithm was found to converge to a similar value and the performance of the algorithm is unaffected. Figure 4-10 shows the sum of rewards per trial earned by the algorithm in each case.



(a) Sum of rewards per trial for uncertainty in mass of the pendulum



(b) Sum of rewards per trial for uncertainty in friction of the pendulum



(c) Sum of rewards per trial for uncertainty in inertia of the pendulum

Figure 4-10: Sum of rewards per trial for $\pm 20\%$ variation in model parameters of the inverted pendulum

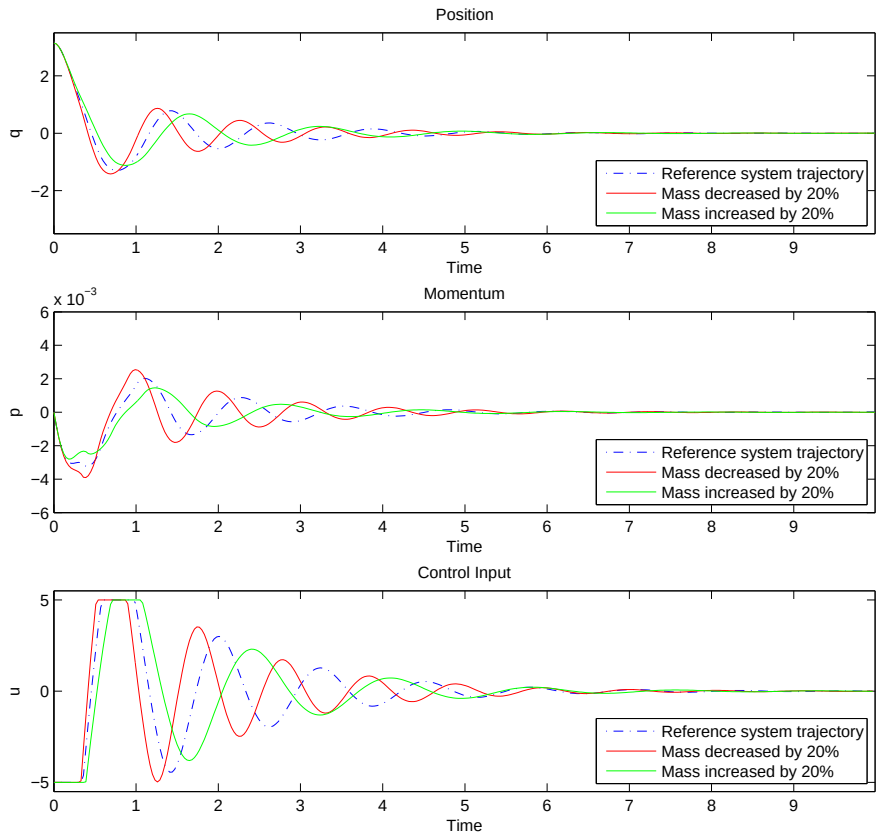


Figure 4-11: Robustness to uncertainty in the mass of the pendulum

4-5-7 Robustness of the learned controller to model uncertainty

It is also interesting to investigate the robustness of the learned controller to model uncertainty. This is investigated on the inverted pendulum model. The model parameters are varied to $\pm 20\%$ and the system is simulated with the previously learned controller.

- **Mass**: With a $\pm 20\%$ variation in the mass M_p, the learned controller is still able to stabilise the system. Figure 4-11 shows the variations in the system trajectory. It can be seen that when the mass is increased by 20%, the amount of oscillation in the system around the desired equilibrium point is less. However, the controller takes almost the same time to stabilise the system in all three cases. The learned controller is thus, robust to model uncertainty in the mass of the pendulum.

- **Friction**: When the friction is varied to within $\pm 20\%$ of the value, the learned controller is still able to stabilise the system as can be seen from Figure 4-12. However, variation in the friction parameter has a significant impact on the performance of the learned controller. When the friction is decreased by 20%, the controller takes significantly longer to stabilise the system. Increasing the friction results in the controller stabilising

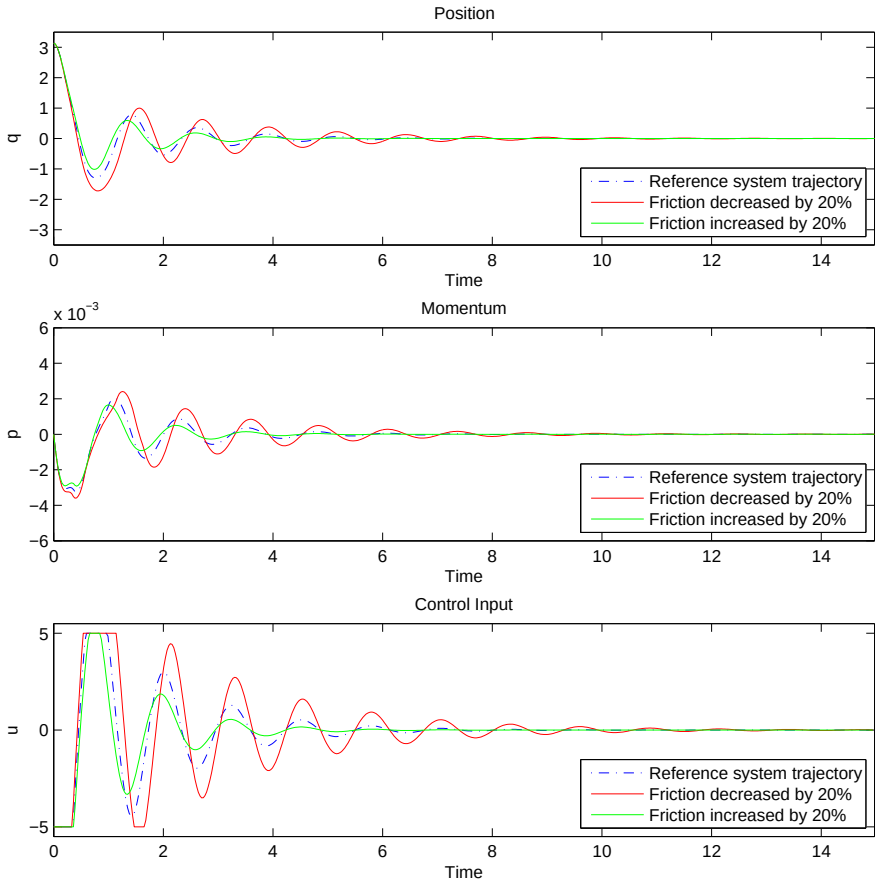


Figure 4-12: Robustness to uncertainty in the friction present in the inverted pendulum system

the system much faster. This is due to the fact that CbI is unable to change the interconnection or damping structure of the system and is only able to shape the energy. A system with a higher damping converges faster and this can be seen from the results. However, the system with higher friction requires a higher control input to stabilise. This is expected as the dissipation in the system has increased and thus, a greater amount of energy has to be injected into the system.

- **Inertia**: As can be seen from Figure 4-13, uncertainty in the inertia of the pendulum does not significantly impact the time taken to stabilise the system. However, it does affect the oscillations present in the system. Decreasing the inertia has the effect of causing the system to stabilise marginally faster and increasing the inertia has the opposite effect, causing the system to take marginally more time to stabilise. However, neither of these are significant changes to the performance of the controller.

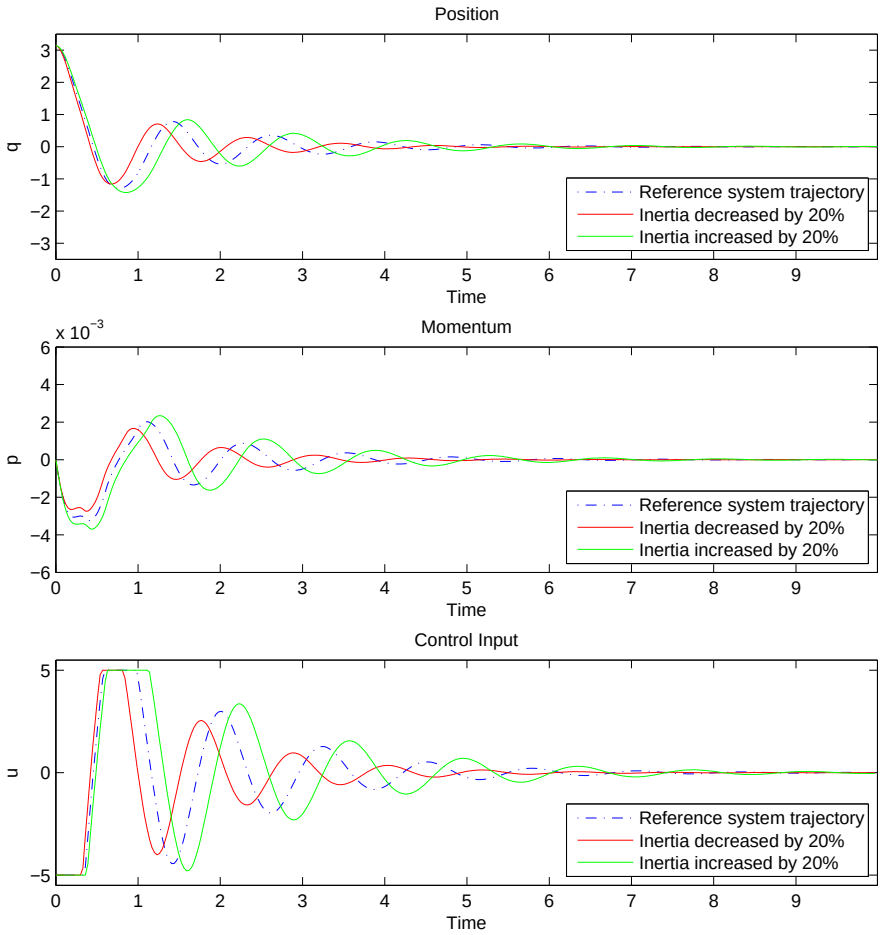


Figure 4-13: Robustness to uncertainty in the inertia of the pendulum

4-6 Summary

This chapter has developed and presented the CBI-AC algorithm that seeks to embed RL into CbI so as to retain the advantages of both these methods. The algorithm has been tested in simulation on two different mechanical systems - the spring mass damper and the inverted pendulum and has been found to successfully stabilise both the systems. It has been seen that the stability of the systems can be investigated and verified numerically. Moreover, both the algorithm and the learned controller have been found robust to variations in model parameters. However, it is seen that this method still suffers from the dissipation obstacle, which is an inherent drawback of CbI.

Chapter 5

Towards a dynamic controller

In the preceding chapters, it is shown that Control by Interconnection (CbI) is limited in its applicability due to the dissipation obstacle. This leads to the motivation to attempt to go beyond the dissipation obstacle in order to formulate a controller that might be able to stabilise even systems with pervasive dissipation, thus extending the applicability of CbI. Previously, it was thought that the limitations stem from the passivity constraint on the controller and in [21], the authors have tried to circumvent the dissipation obstacle by relaxing the passivity constraint on the controller. However, in [28], the authors have shown that dissipation obstacle effectively hampers the CbI methodology. This leads to the research goal of the second part of this thesis, which is to formulate a dynamic controller with enough freedom to go beyond the dissipation obstacle while still using CbI methodology.

To this end, first consider the input state output port-Hamiltonian (PH) system that has been the subject of study of this thesis so far,

$$\begin{aligned} \dot{x} &= \underbrace{[J(x) - R(x)]}_{F(x)} \nabla_x H(x) + g(x)u, \\ y &= g^T(x) \nabla_x H(x). \end{aligned} \tag{5-1}$$

Consider a PH controller

$$\begin{aligned} \dot{\xi} &= \underbrace{[J_c(\xi) - R_c(\xi)]}_{F_c(\xi)} \nabla_\xi H_c(\xi) + g_c(\xi)u_c, \\ y_c &= g_c^T(\xi) \nabla_\xi H_c(\xi), \end{aligned} \tag{5-2}$$

interconnected with the plant system using the standard power preserving interconnection

$$\begin{bmatrix} u \\ u_c \end{bmatrix} = \begin{bmatrix} 0 & -I_m \\ I_m & 0 \end{bmatrix} \begin{bmatrix} y \\ y_c \end{bmatrix}. \tag{5-3}$$

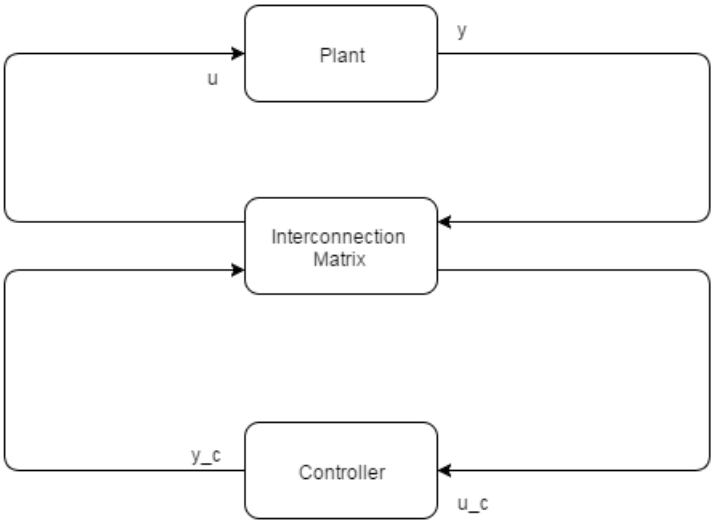


Figure 5-1: Interconnection of the plant and controller systems

The interconnected system can be written as

$$\begin{bmatrix} \dot{x} \\ \dot{\xi} \end{bmatrix} = \begin{bmatrix} J(x) - R(x) & -g(x)g_c^T(\xi) \\ g_c(\xi)g^T(x) & J_c(\xi) - R_c(\xi) \end{bmatrix} \begin{bmatrix} \nabla_x H_{cl}(x,\xi) \\ \nabla_\xi H_{cl}(x,\xi) \end{bmatrix}. \tag{5-4}$$

with $H_{cl} = H(x) + H_c(\xi)$.

However, while formulating both the plant and controller as PH systems is convenient, it has been seen that such a controller is limited in its applicability and it unable to shape the coordinates in which pervasive dissipation is present.

This leads to the question of how much can the structure of the controller be relaxed to allow more freedom in controller design while still being able to make strong statements about the stability of the interconnected system.

One possibility to extend the use of CbI is to use state modulated control by interconnection (as introduced in [5]), however authors of [5] have gone on to show that although this extends the set of plants for which CbI is applicable, this method still suffers from the dissipation obstacle.

However, motivated by this, this thesis starts investigation with the possibilities if both the plant and controller states are available.

The interconnection of the plant and the controller is explained via Figure 5-1. If the interconnection is a power preserving interconnection, then the PH structure is preserved and the interconnected system is also a PH system [3, 4].

Instead of the standard negative feedback power preserving interconnection, let the systems be interconnected with some state modulation as

$$\begin{bmatrix} u \\ u_c \end{bmatrix} = \begin{bmatrix} f_1(x,\xi) & f_2(x,\xi) \\ f_3(x,\xi) & f_4(x,\xi) \end{bmatrix} \begin{bmatrix} y \\ y_c \end{bmatrix}. \tag{5-5}$$

It should be noted however, that this is not a power preserving interconnection unless the matrices $f_1(x,\xi)$ and $f_4(x,\xi)$ are skew symmetric and $f_2^T(x,\xi) = -f_3(x,\xi)$ [29].

This leads to a closed loop system of the form

$$\begin{bmatrix}\dot{x}\\ \dot{\xi}\end{bmatrix} = \begin{bmatrix}F(x)+g(x)f_1(x,\xi)g^T(x) & g(x)f_2(x,\xi)g_c^T(\xi)\\ g_c(\xi)f_3(x,\xi)g^T(x) & F_c(\xi)+g_c(\xi)f_4(x,\xi)g_c^T(\xi)\end{bmatrix}\begin{bmatrix}\nabla_x H(x)\\ \nabla_\xi H_c(\xi)\end{bmatrix}. \tag{5-6}$$

However, as a result of this (non power preserving) state modulated interconnection, the PH structure that could be exploited to prove stability has been lost. There is thus, a trade-off between freedom in controller design and maintaining the PH structure when trying to use CbI. Imposing the restriction that the interconnection be power preserving leads to the following interconnection structure

$$\begin{bmatrix}u\\ u_c\end{bmatrix} = \begin{bmatrix}f_1(x,\xi) & f_2(x,\xi)\\ -f_2^T(x,\xi) & f_4(x,\xi)\end{bmatrix}\begin{bmatrix}y\\ y_c\end{bmatrix}. \tag{5-7}$$

where $f_1(x,\xi)$ and $f_4(x,\xi)$ are also skew symmetric matrices.

This allows the PH structure to be preserved and the closed loop system can be written as

$$\begin{bmatrix}\dot{x}\\ \dot{\xi}\end{bmatrix} = \begin{bmatrix}F(x)+g(x)f_1(x,\xi)g^T(x) & g(x)f_2(x,\xi)g_c^T(\xi)\\ -g_c(\xi)f_2^T(x,\xi)g^T(x) & F_c(\xi)+g_c(\xi)f_4(x,\xi)g_c^T(\xi)\end{bmatrix}\begin{bmatrix}\nabla_x H(x)\\ \nabla_\xi H_c(\xi)\end{bmatrix}, \tag{5-8}$$

with the closed loop Hamiltonian $H_{cl} = H(x) + H_c(\xi)$.

This state modulated interconnection has allowed some new freedom in the controller design. Since the matrix $f_1(x,\xi)$ is skew symmetric, $g(x)f_1(x,\xi)g^T(x)$ is also skew symmetric. This allows the modification of the interconnection structure of the plant system via the term $f_1(x,\xi)$. Since $F(x) = J(x) - R(x)$, where $J(x)$ is a skew symmetric matrix, if it is possible to find a skew symmetric matrix $f_1(x,\xi)$ such that

$$J_d(x) = J(x) + g(x)f_1(x,\xi)g^T(x), \tag{5-9}$$

where $J_d(x)$ is the desired interconnection structure, it is possible to change the interconnection structure as desired using this state modulated interconnection. This was not possible using the standard CbI methodology as shown in Eq. (2-45). However, it is still not possible to overcome the dissipation obstacle using this controller as can be shown. Restricting (without much loss of generality) to Casimir functions of the form $C(x,\xi) = \xi - S(x)$, the invariance condition is found to be

$$\begin{bmatrix}-\nabla_x S(x) & I\end{bmatrix}\begin{bmatrix}F(x)+g(x)f_1(x,\xi)g^T(x) & g(x)f_2(x,\xi)g_c^T(\xi)\\ -g_c(\xi)f_2^T g^T(x) & F_c(\xi)+g_c(\xi)f_4(x,\xi)g_c^T(\xi)\end{bmatrix}\begin{bmatrix}\nabla_x H\\ \nabla_\xi H_c\end{bmatrix} = 0. \tag{5-10}$$

Attempting to solve Eq. (5-10) again leads to the dissipation obstacle stymieing the use of CbI for shaping coordinates which have pervasive dissipation present in them. Moreover, a strong motivation for the use of CbI over state feedback control laws is that state information is in practice not always readily available and observers for PH systems can only be designed for a limited class of PH systems (as elaborated on in [30]).

This leads to the motivation to introduce dummy states in the controller model to introduce more freedom into the controller design, while trying to maintain the PH structure.

Let the controller structure be PH with

$$\begin{bmatrix}\dot{\xi}_1\\ \dot{\xi}_2\end{bmatrix} = \begin{bmatrix} Z_{11}(\xi_1,\xi_2) & Z_{12}(\xi_1,\xi_2)\\ -Z_{12}^T(\xi_1,\xi_2) & Z_{22}(\xi_1,\xi_2)\end{bmatrix}\begin{bmatrix}\nabla_{\xi_1}H_c(\xi_1)\\ \nabla_{\xi_2}H_c(\xi_2)\end{bmatrix} + \begin{bmatrix}g_{c_1}(\xi_1)\\ g_{c_2}(\xi_2)\end{bmatrix}u_c \tag{5-11}$$

$$y_c = \begin{bmatrix}g_{c_1}^T(\xi_1) & g_{c_2}^T(\xi_2)\end{bmatrix}\begin{bmatrix}\nabla_{\xi_1}H_c(\xi_1)\\ \nabla_{\xi_2}H_c(\xi_2)\end{bmatrix} \tag{5-12}$$

Interconnecting this with the plant system (Eq. (5-1)) using the standard negative feedback interconnection (Eq. (5-3)), the closed loop dynamics take the form

$$\begin{bmatrix}\dot{x}\\ \dot{\xi}_1\\ \dot{\xi}_2\end{bmatrix} = \begin{bmatrix} F(x) & -g(x)g_{c_1}^T(\xi_1) & -g(x)g_{c_2}^T(\xi_2)\\ g_{c_1}(\xi_1)g^T(x) & Z_{11}(\xi_1,\xi_2) & Z_{12}(\xi_1,\xi_2)\\ g_{c_2}^T(\xi_2)g^T(x) & -Z_{12}^T(\xi_1,\xi_2) & Z_{22}(\xi_1,\xi_2)\end{bmatrix}\begin{bmatrix}\nabla_x H(x)\\ \nabla_{\xi_1}H_c(\xi_1,\xi_2)\\ \nabla_{\xi_2}H_c(\xi_1,\xi_2)\end{bmatrix} \tag{5-13}$$

which is again a PH system with the interconnected system Hamiltonian

$$H_{cl}(x,\xi_1,\xi_2) = H(x) + H_c(\xi_1,\xi_2). \tag{5-14}$$

The idea behind this formulation is that the controller states ξ_1 can be directly related to the plant states by finding a suitable Casimir function $C(x,\xi_1)$, which still allows some more freedom in controller design via the dummy states ξ_2. Suppose a suitable Casimir function is found,

$$C(x,\xi_1) = \xi_1 - S(x_s) = 0, \tag{5-15}$$

where x_s denotes the shape-able components of the plant, then the closed loop Hamiltonian of the system becomes,

$$H_{cl}(x,\xi_1,\xi_2) = H(x) + H_c(\xi_1,\xi_2) \tag{5-16}$$

$$= H(x) + H_c(S(x_s),\xi_2). \tag{5-17}$$

With a suitable choice of the controller system matrices $Z(\xi_1,\xi_2)$, it is hoped that this freedom in controller design may allow one to go beyond the dissipation obstacle. The states ξ_1 can be used to shape the energy of the coordinates not affected by dissipation (i.e. the shape-able coordinates x_s) and with a suitable choice of controller dynamics, the dummy states ξ_2 may be made to evolve in such a manner that they can shape the coordinates with dissipation.

It should be noted here that systems with pervasive dissipation extract an infinite amount of energy from the controller at equilibrium. However, the goal here is to try to formulate a dynamic controller that renders the closed loop system passive with respect to the desired equilibrium point.

Thus, the goal is to find a controller system as given in Eq. (5-11) such that at the desired equilibrium x^*,

$$[J(x^*) - R(x^*)]\nabla_x H(x^*) + g(x^*)u^* = 0 \tag{5-18}$$

and correspondingly

$$u^* = -y_c^* = -\begin{bmatrix} g_{c_1}^T(\xi_1^*) & g_{c_2}^T(\xi_2^*) \end{bmatrix} \begin{bmatrix} \nabla_{\xi_1} H_c(\xi_1^*) \\ \nabla_{\xi_2} H_c(\xi_2^*) \end{bmatrix} \tag{5-19}$$

It is however, not an easy task to find suitable controller matrices and there does not appear to be an intuitive way of deriving a suitable controller. Motivated by the earlier success with Reinforcement Learning (RL), it is interesting to now investigate if this problem can be formulated as an RL problem and whether a suitable controller can be learned.

5-1 Formulation as a Reinforcement Learning Problem

The first method attempted to formulate this as an RL problem is as follows.

It is desired that the algorithm learn the matrices Z_{11}, Z_{12} and Z_{22} such that the closed loop system is passive with respect to a desired equilibrium point.

To this end, choosing for convenience

$$\begin{bmatrix} g_{c_1}(\xi_1) \\ g_{c_2}(\xi_2) \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix} \tag{5-20}$$

and fixing the controller Hamiltonian as $H_c(\xi_1, \xi_2) = \frac{1}{2}\xi^T\xi$, the controller system is now

$$\begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \end{bmatrix} = \begin{bmatrix} Z_{11}(\xi_1, \xi_2) & Z_{12}(\xi_1, \xi_2) \\ -Z_{12}^T(\xi_1, \xi_2) & Z_{22}(\xi_1, \xi_2) \end{bmatrix} \begin{bmatrix} \nabla_{\xi_1} H_c(\xi_1) \\ \nabla_{\xi_2} H_c(\xi_2) \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u_c \tag{5-21}$$

$$y_c = \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} \nabla_{\xi_1} H_c(\xi_1) \\ \nabla_{\xi_2} H_c(\xi_2) \end{bmatrix}. \tag{5-22}$$

The matrices Z_{11}, Z_{12} and Z_{22}, can be parametrised using function approximation as

$$Z_{11} = \vartheta_{11}\varphi(\xi) \tag{5-23}$$

$$Z_{11} = \vartheta_{12}\varphi(\xi) \tag{5-24}$$

$$Z_{11} = \vartheta_{22}\varphi(\xi) \tag{5-25}$$

where ϑ_{11}, ϑ_{12} and ϑ_{22} are some parameter vectors rich enough to capture the solution and $\varphi(\xi)$ is a suitable linear in parameters basis function.

The controller is now,

$$\begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \end{bmatrix} = \begin{bmatrix} \vartheta_{11}\varphi(\xi) & \vartheta_{12}\varphi(\xi) \\ -(\vartheta_{12}\varphi(\xi))^T & \vartheta_{22}\varphi(\xi) \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u_c \tag{5-26}$$

$$y_c = \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}. \tag{5-27}$$

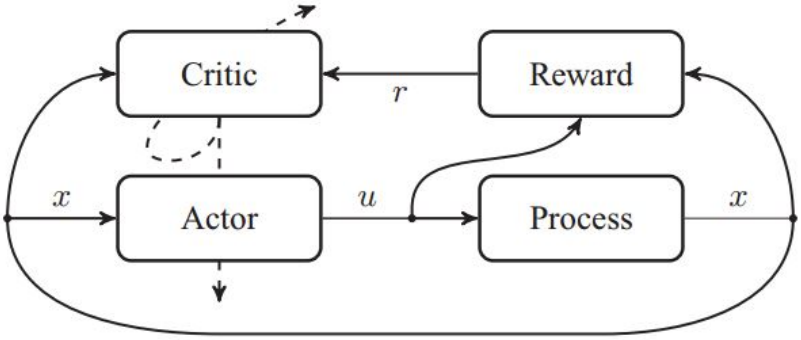


Figure 5-2: Schematic of actor-critic algorithm [2]

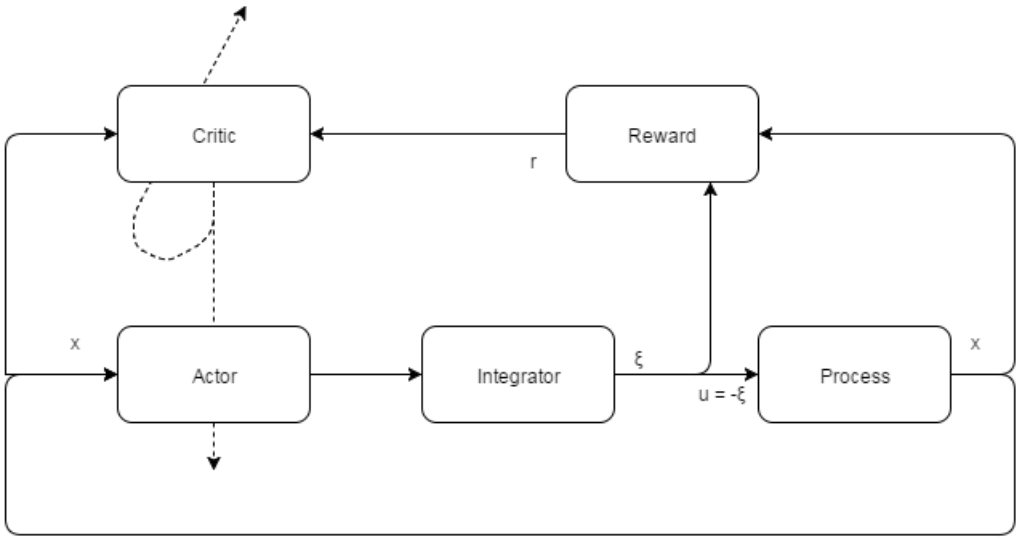


Figure 5-3: Problem with dynamic controller ACRL formulation

The problem that arises now is what to choose as a suitable policy and how to define a suitable reward function. An intuitive choice would be to choose the dynamics of the controller as the policies to be optimised but this leads to some problems with implementation in the RL framework.

Usually in the context of Actor-Critic Reinforcement Learning (ACRL), a control action is chosen according to the policy and based on the reward received and the approximated value function learned, the policy is updated using gradient ascent methods. The schematic for ACRL is shown again in Figure 5-2. And the current formulation is shown in Figure 5-3.

In the current formulation of our controller however, the control input $u = -yc = -\xi_2$ is not directly influenced by the policy but rather indirectly through the evolution of the controller states ξ. So it is not directly clear how to choose a suitable policy.

However, this leads to an interesting thought. If it were possible to formulate the problem such that the actor directly supplied the control action, this issue could possibly be overcome.

The dynamics of the controller states are given by

$$\dot{\xi}_1 = Z_{11}(\xi_1, \xi_2)\xi_1 + Z_{12}(\xi_1, \xi_2)\xi_2 \tag{5-28}$$

$$\dot{\xi}_2 = -Z_{12}^T(\xi_1, \xi_2)\xi_1 + Z_{22}(\xi_1, \xi_2)\xi_2 + y. \tag{5-29}$$

Define $s_1 = \dot{\xi}_1$, $s_2 = \dot{\xi}_2$. This leads us to

$$s_1 = Z_{11}(\xi_1, \xi_2)\xi_1 + Z_{12}(\xi_1, \xi_2)\xi_2 \tag{5-30}$$

$$s_2 = -Z_{12}^T(\xi_1, \xi_2)\xi_1 + Z_{22}(\xi_1, \xi_2)\xi_2 + y \tag{5-31}$$

With some abuse of notation, if the matrices Z_{11} and Z_{22} are invertible[1], then it follows that

$$\xi_1 = Z_{11}^{-1}s_1 - Z_{11}^{-1}Z_{12}\xi_2 \tag{5-32}$$

$$\xi_2 = Z_{22}^{-1}s_2 + Z_{22}^{-1}Z_{12}^T\xi_1 + Z_{22}^{-1}y \tag{5-33}$$

and if it is possible to get the values for s_1, s_2 then it may be possible to circumvent the problem. However, this imposes further an unnecessary restriction on the controller. Therefore, this approach was abandoned and instead, the following approach is tried.

Pulling out the integrator from the controller leaves the rest of the dynamics as a static map as shown in Figure 5-4. The control input required from the controller at equilibrium can be easily computed from knowledge of the plant dynamics. The algorithm should now try to learn a policy such that at the equilibrium, the controller output is the required u^*. Using the reward function, it is possible to let the RL agent know that it should try to formulate dynamics that allows it to stabilise the plant at the desired state. The block diagram of this controller scheme is shown in Figure 5-5.

Following this, the reward function has to be defined in such a manner that there is a negative reward incurred for the plant states not being close to the desired set point and also a negative penalty incurred on the controller output being away from the output to maintain the desired equilibrium. Thus, the reward function is of the form

$$r_{k+1} = \rho(x, \xi_2). \tag{5-34}$$

[1]By Jacobi's theorem, skew symmetric matrices of odd dimensions are singular and hence, not invertible

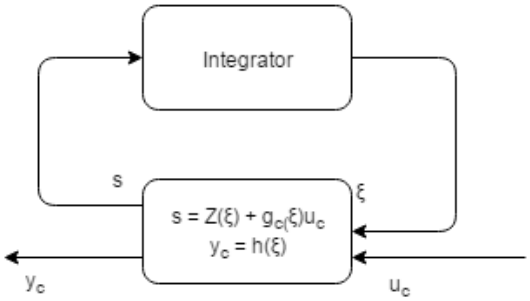


Figure 5-4: Pulling out the integrator leaving the rest of the controller dynamics

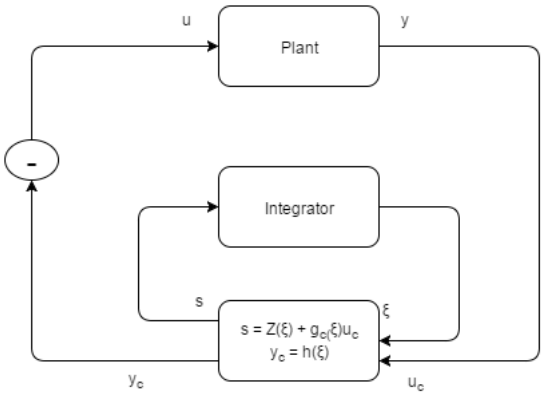


Figure 5-5: Dynamic controller system

Since the state ξ_1 does not influence the output of the controller and by extension does not influence the input to the plant, but only the controller dynamics, there is no penalty on the controller states ξ_1. When implementing this however, care must be taken to ensure some suitable bounds on the controller state ξ_1 to prevent the controller states from becoming unstable. This can be done either via the reward function or by defining saturation bounds on the controller states.

The policy are now taken as the parametrised sub-matrices of the controller dynamics. The policies are a function of ξ and the critic is parametrised as a function of both the plant and controller state spaces.

$$V(\theta, x, \xi) = \theta^T \phi(x, \xi) \tag{5-35}$$

5-2 Update equations

The dynamic controller finally has been parametrised as an RL problem. All that remains now, is to introduce the update equations. Using the subscript k to denote the value at a time step k, the gradient of the value function can be easily found

$$\nabla_\theta V_{\theta_k}(x_k, \xi_k) = \phi(x, \xi). \tag{5-36}$$

Using again the ACRL scheme given in Section 3-3-3, the critic can now be updated as follows.

$$\delta_{k+1} = r_{k+1} + \gamma V_{\theta_k}(x_k, \xi_k) \tag{5-37}$$
$$z_{k+1} = \lambda\gamma z_k + \nabla_\theta V_{\theta_k}(x_k, \xi_k) \tag{5-38}$$
$$\theta_{k+1} = \theta_k + \alpha_{c,k}\delta_k z_k \tag{5-39}$$

where δ is the Temporal Difference (TD), z_k is the eligibility trace and α_c is a suitable critic learning rate.

Updating the policy is a more difficult task. The policy depends on only the states ξ. A positive TD means that the controller dynamics are such that the closed loop system is moving towards a more desirable state and so the policy should be updated in that direction. Drawing the exploration term from a zero mean standard normal distribution with a variance of σ^2,

$$\Delta Z \sim (0, \sigma^2). \tag{5-40}$$

The policy is now updated with

$$\vartheta_{11_{k+1}} = \vartheta_{11_k} + \alpha_a\delta_{k+1}\Delta Z_k\nabla_\vartheta\hat{\pi}(\xi_k, \vartheta_k, \varphi_k), \tag{5-41}$$
$$\vartheta_{12_{k+1}} = \vartheta_{12_k} + \alpha_a\delta_{k+1}\Delta Z_k\nabla_\vartheta\hat{\pi}(\xi_k, \vartheta_k, \varphi_k), \tag{5-42}$$
$$\vartheta_{22_{k+1}} = \vartheta_{22_k} + \alpha_a\delta_{k+1}\Delta Z_k\nabla_\vartheta\hat{\pi}(\xi_k, \vartheta_k, \varphi_k). \tag{5-43}$$

Unfortunately, so far simulations based on this formulation have failed to achieve the intended results. In the following section, some discussion follows on this methodology and why this might be the case.

5-3 Discussion

Although the research towards learning a dynamic seemed at first glance to be extremely promising, unfortunately, so far the desired results have been elusive. Some discussion on the possibilities of what might have caused the negative results follows.

5-3-1 Verifying that the problem satisfies the Markov property

Due to the nature of the problem formulation, it is not immediately clear how RL can be applied to this scenario. As mentioned previously in Section 3-2, RL can be applied to solve problems modelled as a Markov Decision Process (MDP).

In words, a discrete time system satisfies the Markov property if the current state of the system x_k depends only on the immediate past x_{k-1}. It is known that the plant system satisfies the Markov property. Verifying that the controller formulation also satisfies this is simple and intuitive.

Recall Eq. (5-32) and Eq. (5-33). This is equivalent to saying that at any time step k, the system is of the form

$$\xi_{k+1} = \bar{Z}(s_k, \xi_k, y_k). \tag{5-44}$$

Thus, the evolution of the state depends only on the state directly preceding it and the system forms a Markov chain and thus, RL can be used to solve this problem.

5-3-2 Possible causes for negative results

1. **Learning rates**: It is possible that finding correct learning rates was the problem. During simulations, both the controller and plant systems were found to converge to their natural equilibrium and the algorithm did not learn anything. Finding good learning rates in RL can be a tricky problem. In this thesis, the gridding approach was used to try to find suitable learning rates but it is possible that a more refined search to find better learning rates might deliver better results.

2. **Exploration**: The exploration term in RL is responsible for the agent trying out new actions that may not be the currently optimal policy in order to explore the environment. In this thesis, the exploration term has been drawn from a zero mean normal distribution with a chosen variance σ^2. It is however possible that due to the nature of the problem, the amount of exploration used was not enough to move towards a more optimal policy and hence learning did not occur. As has been shown in the example using the spring mass damper in Section 4-3, the value function cannot make accurate estimates about the region of the state space that it has not explored. Perhaps using a higher level of exploration or an exploration term of a different type (for example, a multisine wave) might yield better results.

5-4 Summary

It has been seen in the preceding chapter that CbI suffers from the dissipation obstacle. This chapter attempted to increase the freedom in controller design when using CbI. It was seen that there is a trade-off between maintaining structure and flexibility in controller design. Finally, it was proposed to augment the controller states with dummy states in order to increase the freedom in controller design while still maintaining the PH structure and use RL to learn such a controller. However, as of the time of writing this thesis, the author has been unable to produce positive results with this method.

Chapter 6

Conclusions and Recommendations

6-1 Conclusions

The primary goal of this thesis was to design a methodology that combined the advantages of Control by Interconnection (CbI) for port-Hamiltonian (PH) systems and Reinforcement Learning (RL). Thus, CbI (Section 2-4) was combined with Actor-Critic Reinforcement Learning (ACRL) (Section 3-3-3) to yield the Control by Interconnection - Actor Critic (CBI-AC) algorithm (Chapter 4). The plant states are partitioned into the shape-able and non shape-able components and using CBI-AC, the energy of the closed loop system can be easily shaped. The developed methodology was tested in simulation and the following conclusions can be drawn:

- CBI-AC is an output feedback method that does not require state information. The closed loop energy of the system can be effectively shaped and the added advantage is that the learned controller can be interpreted in terms of energy exchange. Stability can easily be analysed numerically from the closed loop energy of the system.
- There is no longer a need to explicitly solve a set of Partial Differential Equations (PDEs). Moreover, the choice of a suitable Casimir function and controller Hamiltonian is one that often requires experience to make a good choice. The CBI-AC algorithm eliminates the need for this.
- Control saturation can be incorporated into the learning methodology.
- The performance of the CBI-AC learning algorithm is found to be robust to changes in model parameters. However, it is possible that for a significantly large change in the model parameters ($> 20\%$), the learning rates may need to be tuned again.
- The learned controller is found to be robust to changes in model uncertainty up to $\pm 20\%$ as well. The learned controller is able to stabilise the system even with a mismatch in model parameters, as simulations have shown. Thus, for implementation on physical

systems, it is possible that the controller initially be learned on a model of the system in simulation, and then if required, finally tuned on the physical set-up.

- Finding the correct learning rates can sometimes be quite challenging. In this thesis, the gridding approach was used successfully. However, for systems which require a long time to simulate, some other method to find suitable learning parameters may need to be used.

However, the dissipation obstacle still limits the use of this controller methodology as it is still not possible to shape the coordinates that have pervasive dissipation in them. It was seen that this is an inherent limitation of CbI and as a result the secondary goal of this thesis was to try to formulate a controller methodology that allows enough freedom to go beyond the dissipation obstacle, while still maintaining the CbI (Chapter 5). The following can be concluded:

- There is an inherent trade-off between freedom in controller design and structure of the system. Preserving the PH structure of the system allows us to exploit this to prove stability but comes at the cost of freedom in controller design and interconnection.
- Assuming availability of state information, a state modulated interconnection allows a controller design using which the interconnection structure of the plant can be modified. However, the damping matrix cannot be modified using CbI.
- It may be possible to use dummy states to increase the available freedom in controller design. However, the problem then arises as to how to design the controller dynamics. It was hoped that RL would be able to provide a solution to this but so far there have been no positive results using the framework specified in Chapter 5.

6-2 Recommendations and Future Work

- It would be interesting to see to what extent performance criteria can be incorporated into the CBI-AC algorithm. It is hoped that with CBI-AC, it might be possible to design controllers that can in addition to guaranteeing local stability, also guarantee some desired performance constraints. This can be done by defining suitable reward functions.
- The effect of the initialisation of the value function and policy is also of interest. In this thesis, the parameter vector for both the actor and the critic were initialised with zeros. Learning can be considerably sped up by using the knowledge of the model to initialise the parameters closer to the optimal value.
- One of the major drawbacks of CbI is that it is still hampered by the dissipation obstacle. However, using the knowledge of the model may allow for controller design that allows us to go beyond the dissipation obstacle. In particular RL algorithms like Model Learning Actor Critic (MLAC) [31] could be used to learn a suitable controller that could go beyond the dissipation obstacle without having explicit access to state information.

Finally, it is hoped that this thesis will be another step into bringing the fields of RL and PH systems closer together, with the ultimate goal of seeing the two used in everyday life.

Bibliography

[1] O. Sprangers, R. Babuška, S. P. Nageshrao, and G. A. D. Lopes, "Reinforcement learning for port-hamiltonian systems," *Cybernetics, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.

[2] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuška, "A survey of actor-critic reinforcement learning: Standard and natural policy gradients," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 42, no. 6, pp. 1291–1307, 2012.

[3] A. v. d. Schaft, "Port-hamiltonian systems: an introductory survey," 2006.

[4] A. van der Schaft and D. Jeltsema, "Port-hamiltonian systems theory: An introductory overview," *Foundations and Trends in Systems and Control*, vol. 1, no. 2-3, pp. 173–378, 2014.

[5] R. Ortega, A. van der Schaft, F. Castanos, and A. Astolfi, "Control by interconnection and standard passivity-based control of port-hamiltonian systems," *Automatic Control, IEEE Transactions on*, vol. 53, no. 11, pp. 2527–2542, 2008.

[6] R. Ortega and E. García-Canseco, "Interconnection and damping assignment passivity-based control: A survey," *European Journal of Control*, vol. 10, no. 5, pp. 432–450, 2004.

[7] S. K. Chalup, C. L. Murch, and M. J. Quinlan, "Machine learning with aibo robots in the four-legged league of robocup," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 3, pp. 297–310, 2007.

[8] J. E. Domenech, C. V. Regueiro, C. Gamallo, and P. Quintia, "Learning wall following behaviour in robotics through reinforcement and image-based states," in *Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on*, pp. 2101–2106.

[9] A. El-Fakdi and M. Carreras, "Two-step gradient-based reinforcement learning for underwater robotics behavior learning," *Robotics and Autonomous Systems*, vol. 61, no. 3, pp. 271–282, 2013.

[10] S. P. Nageshrao, G. A. D. Lopes, D. Jeltsema, and R. Babuska, "Port-hamiltonian systems in adaptive and learning control: A survey," *Automatic Control, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[11] S. P. Nageshrao, G. A. D. Lopes, D. Jeltsema, and R. Babuška, "Passivity-based reinforcement learning control of a 2-dof manipulator arm," *Mechatronics*, vol. 24, no. 8, pp. 1001–1007, 2014.

[12] S. P. Nageshrao, G. Lopes, D. Jeltsema, and R. Babuška, "Interconnection and damping assignment control via reinforcement learning," in *World Congress*, vol. 19, pp. 1760–1765.

[13] B. Maschke and A. Van der Schaft, "Port-controlled hamiltonian systems: modelling origins and systemtheoretic properties," 1991.

[14] A. Van der Schaft, *L2-gain and passivity techniques in nonlinear control*. Springer Science & Business Media, 2012.

[15] R. Ortega, A. van der Schaft, B. Maschke, and G. Escobar, "Energy-shaping of port-controlled hamiltonian systems by interconnection," in *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, vol. 2, pp. 1646–1651 vol.2.

[16] D. Jeltsema, R. Ortega, and J. M.A. Scherpen, "An energy-balancing perspective of interconnection and damping assignment control of nonlinear systems," *Automatica*, vol. 40, no. 9, pp. 1643–1646, 2004.

[17] R. Ortega, A. J. Van der Schaft, I. Mareels, and B. Maschke, "Putting energy back in control," *Control Systems, IEEE*, vol. 21, no. 2, pp. 18–33, 2001.

[18] R. Ortega, A. van der Schaft, B. Maschke, and G. Escobar, "Interconnection and damping assignment passivity-based control of port-controlled hamiltonian systems," *Automatica*, vol. 38, no. 4, pp. 585–596, 2002.

[19] A. J. van der Schaft, *Port-Hamiltonian Systems: Network Modeling and Control of Nonlinear Physical Systems*, vol. 444 of *International Centre for Mechanical Sciences*, book section 9, pp. 127–167. Springer Vienna, 2004.

[20] V. Duindam, A. Macchelli, S. Stramigioli, and H. Bruyninckx, *Modeling and control of complex physical systems*. Springer, 2009.

[21] J. Koopman and D. Jeltsema, "Casimir-based control beyond the dissipation obstacle," *ArXiv e-prints*, 2012.

[22] O. Sprangers, *Embedding machine learning into passivity theory: a port-Hamiltonian approach*. Thesis, 2012.

[23] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*. MIT Press, 1998.

[24] D. P. Bertsekas, *Dynamic programming and optimal control*, vol. 1. Athena Scientific Belmont, MA, 1995.

[25] L. Baird, "Residual algorithms: Reinforcement learning with function approximation," in *ICML*, pp. 30–37.

[26] G. J. Gordon, "Stable function approximation in dynamic programming," report, DTIC Document, 1995.

[27] A. Cherubini, F. Giannone, L. Iocchi, M. Lombardo, and G. Oriolo, "Policy gradient learning for a humanoid soccer robot," *Robotics and Autonomous Systems*, vol. 57, no. 8, pp. 808–818, 2009.

[28] M. Zhang, R. Ortega, D. Jeltsema, and H. Su, "Further deleterious effects of the dissipation obstacle in control-by-interconnection of port-hamiltonian systems," *Automatica*, vol. 61, pp. 227–231, 2015.

[29] C. Secchi, S. Stramigioli, and C. Fantuzzi, *Control of interactive robotic interfaces: A port-Hamiltonian approach*, vol. 29. Springer Science & Business Media, 2007.

[30] A. Venkatraman and A. J. van der Schaft, "Full-order observer design for a class of port-hamiltonian systems," *Automatica*, vol. 46, no. 3, pp. 555–561, 2010.

[31] I. Grondman, M. Vaandrager, L. Busoniu, R. Babuska, and E. Schuitema, "Efficient model learning methods for actor-critic control," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, no. 3, pp. 591–602, 2012.

Glossary

List of Acronyms

DCSC	Delft Center for Systems and Control
CbI	Control by Interconnection
PH	port-Hamiltonian
RL	Reinforcement Learning
PBC	Passivity Based Control
PDE	Partial Differential Equation
PDEs	Partial Differential Equations
ES	Energy Shaping
DI	Damping Injection
IDA-PBC	Interconnection and Damping Assignment Passivity Based Control
MDP	Markov Decision Process
TD	Temporal Difference
AC	Actor-Critic
ACRL	Actor-Critic Reinforcement Learning
CBI-AC	Control by Interconnection - Actor Critic