PID Control as a Process of Active Inference Applied to a Refrigeration System

Master's Thesis

Adrián Rocandio Moreno

Aalborg University
Control and Automation

June 2021

Department of Electronic Systems
Fredrik Bajers Vej 7B
9220 Aalborg
www.es.aau.dk

AALBORG UNIVERSITY

Title:

PID Control as a Process of Active Inference Applied to a Refrigeration System

Project:

Master's Thesis

Semester:

Fourth

Project Period:

01/02/2021 - 03/06/2021

Project Group:

1034

Group Members:

Adrián Rocandio

Supervisors:

Henrik Schiøler
Roozbeh Izadi-Zamanabadi
Basil M. Al-Hadithi

Pages:

50

Submission:

03/06/2021

This report is made with LaTeX

Abstract:

Classical PID control is a widely used technique in many industrial applications due to its good performance and relatively low complexity. Nevertheless, these regulators are not sufficient in some cases. This project investigates a novel probabilistic interpretation of PID control. Under this framework, it is assumed that only sensed variables are accessible. That is, no prior information of the process is available (i.e., plant model). Thus, the controller is furnished with a simple generative model that tries to deduce the measurement causes. This model, which is refined with every new measurement, permits designing the PID regulator. The innovation with respect to the classical approach is that here the controller gains encode measurement noise properties that can be inferred. The model enhancement and the applied control law obey a biological principle known as *free energy.*

The thesis proposes to implement this PID regulator in a refrigeration process. Specifically, it is aimed to control the evaporator outlet temperature. Simulation results prove good performance when dealing with changes in the set-point. The robustness test, however, shows poor outcomes as the system's response is not able to recover from a small input disturbance. Furthermore, the controller is sensitive to subtle changes in certain parameters when tuning, thus leading to instability.

Preface

This master's thesis is written by Adrián Rocandio during the 4^{th} semester of the Control and Automation MSc at Aalborg University.

The thesis is based on a probabilistic interpretation of the PID controller, where the controller gains encode measurement noise properties. This control framework is founded on some principles taken from the biological field. The goal of the project is to determine whether this new type of controller can be implemented in an industrial application, namely in a refrigeration system. This project was carried out in collaboration with Danfoss, being Roozbeh Izadi-Zamanabadi their representative at Aalborg University.

Throughout the semester, access to the laboratory was limited due to the COVID-19 outbreak. Furthermore, technical issues were encountered in the refrigeration system whenever the entrance to the facilities was possible. Hence, this thesis is purely based on theory and simulations.

I would like to thank my supervisors Henrik Schiøler, Roozbeh Izadi-Zamanabadi and Basil M. Al-Hadithi for the comments, feedback and guidance during the semester.

I would also like to thank my mother for her unconditional support and help during difficult times. To my father, for showing me that there is always an opportunity for learning and improving. To my sister, for her honesty and strong principles. Hope you can find motivation in this report to overcome your academic doubts.

"The first principle is that you must not fool yourself,
and you are the easiest person to fool."

— Richard Feynmann

Reading guide

This report is intended to be read in numerical order. References to figures are shown as, e.g., Fig. 3.1 denoting the first figure of the third chapter. References to equations follow the same procedure but the term Eq. is employed instead. Additionally, the equation number is presented between parenthesis, e.g., Eq. (3.1). The variables presented in this thesis, e.g., state x, are time-dependent, i.e., $x(t)$. However, the latter notation is disregarded for the sake of clarity. Moreover, time derivatives are expressed with a dot over the corresponding variable. For instance, considering the variable x, its derivative is denoted by $\dot{x}$ and its double derivative by $\ddot{x}$.

Nomenclature

Symbols and parameters for the refrigeration system

Symbol/Parameter	Description	Unit
k_1	Nonlinearity gain	•
k_2	Nonlinearity gain	•
f_c	Condenser fan frequency	Hz
f_{cp}	Compressor frequency	Hz
OD	Opening degree of the expansion valve (%)	•
OD^*	Nonlinearity input offset (%)	•
P_c	Condensing pressure	bar
P_e	Evaporating pressure	bar
$\dot{Q}_c$	Heat transfer rate in condenser	W
$\dot{Q}_e$	Heat transfer rate in evaporator	W
T_c	Condensing temperature	°C
$T_{c,i}$	Condenser inlet temperature	°C
$T_{c,o}$	Condenser outlet temperature	°C
T_{cr}	Cold reservoir temperature	°C
T_e	Evaporating temperature	°C
$T_{e,o}$	Evaporator outlet temperature	°C
$T^*_{e,o}$	Nonlinearity output offset	°C
T_{hr}	Hot reservoir temperature	°C
T_{sat}	Saturation temperature	°C
T_{sc}	Subcooling temperature	°C
T_{sh}	Superheating temperature	°C
$\dot{W}_{cp}$	Compressor power consumption	W

Symbols and parameters for the controller design

Symbol/Parameter	Description
$E(y,x)$	Laplace-encoded energy: $-\ln p(y,x)$
F	Variational free energy
$f(x,v)$	Agent's dynamical model of hidden states/inputs
$g(x,v)$	Agent's mapping from hidden states/inputs to observations
$p(y,x)$	Generative density
$q(x)$	R-density, agent's approximate posterior
v	Exogenous inputs: references, disturbances or noise
x	Hidden causes of sensory input (hidden states)
y	Sensory data (measurements)
z	Measurement noise
α	Agent's dynamical model parameter
γ	Hyperparameters: measurement and process precisions

γ_z	Log sensory precisions
ε_z	Measurement error, i.e., $y - x$
ε_ω	Error dynamics, i.e., $\dot{x} - f(x, v)$
η	Perception learning rate
θ	Parameters: α
κ	Action learning rate
λ	Damping term for hyperparameters update
μ	Agent's states. Encodes beliefs about hidden states
π_ω	Process noise precision (inverse of the variance)
π_z	Measurement noise precision
ρ	Hyperparameters learning rate
σ^2_{opt}	R-density's optimal variance which minimises F
ω	Process noise

Abbreviations

EEV	Electronic expansion valve
FEP	Free energy principle
FOPDT	First order plus dead time
KF	Kalman filter
KL	Kullback-Leibler
PID	Proportional-integral-derivative controller
VFE	Variational free energy

Contents

1 | Introduction

Control engineering emerged from the necessity of analysing and designing regulatory mechanisms. Due to its conceptual similarities with biological systems, over the last decades control engineering has become more relevant in the biological field. In 1948, the mathematician Norbert Wiener, through his pioneering work on cybernetics, saw a set of problems, common to both the *machine* and the *living tissue*, centered around questions of communication and control. For instance, regulation issues like maintenance of cellular behaviours and the appropriate response to environmental signals can only be achieved by systems that are robust to certain perturbations and sensitive to others. Since these behaviours demand the use of feedback, tools from control theory provide a good framework to examine and devise self-regulating systems [Iglesias and Ingalls 2009].

In 2006, Friston proposed that the laws underlying self-organisation or self-regulation in biological agents like cells, plants or brains obeyed a principle denoted as *free energy*. This principle, inspired by Helmholtz' thermodynamic free energy [C. L. Buckley et al. 2017], shows that biological systems model their surroundings and act on it so as to reach a certain stability [Karl Friston 2012]. On the one hand, deviations from these stable states are measured by an information-theoretic concept known as *surprisal*. On the other hand, the biological agent models the environment via Bayesian inference, a statistical procedure that describes the optimal way to update agent's beliefs when making new observations (i.e., receiving new sensory input) [Smith, Karl Friston, and Whyte 2021].

The aforementioned concepts are employed to develop a probabilistic interpretation of a proportional-integral-derivative (PID) controller for biological systems and whose gains can be attained optimally [Baltieri and C. Buckley 2019]. Despite its focus on neuronal activities of the brain, this thesis proposes a nonbiological application, namely an industrial refrigeration system. The novelty with respect to classical controllers is that here the agent, apart from furnishing a control law, tries to infer the hidden causes of sensory input by means of a generative model. This model consists of a set of stochastic differential equations that best describe the environment according to the agent. This scheme, known as *Variational filtering*, estimates the conditional density of hidden states in a similar manner to the extended Kalman and Particle filters [Karl Friston, Stephan, et al. 2010]. However, the former optimises this density based on generalised coordinates to arbitrarily high order which eventually allow the construct of the regulator. As opposed to traditional PID controllers, these gains encode random measurement noise. Therefore, estimating the PID gains implies inferring the random fluctuations present in the system.

2 | Problem Analysis

PID control is one of the most employed control algorithms in industrial applications, with more than 90% of total controllers implementing PID or PI regulation [Baltieri and C. Buckley 2019]. Its easy implementation, together with a design based on some rules of thumb makes the PID an appealing regulator for practical purposes (e.g., cruise control on a car or temperature control in a room). PID control provides a good response in processes whose dominant dynamics are of second order. Nevertheless, due to its limited complexity, classical PID regulators are not sufficient in some scenarios. For instance, in processes of higher-order dynamics, systems with long dead time, oscillatory modes or non-periodic disturbances [Åström and Hägglund 1995].

This thesis investigates a new approach to PID controller design, namely from a probabilistic perspective, and tests it on a refrigeration system. The current chapter is divided into three sections and attempts to present the target of the project. For this purpose, a brief description of the classical PID control is presented in the first section. The second section details what the new probabilistic interpretation of PID control consists of. Finally, the last section analyses what is intended to achieve with this project and ends up establishing the problem formulation.

2.1 Classical PID Control

PID control is based on a closed-loop strategy with a negative feedback scheme. Negative feedback methods are founded on the difference between the measured value of the variable to be controlled (e.g., temperature) and its desired value, known as reference or set-point. This produces an error which the controller tries to minimise. Eq. (2.1) expresses the aforementioned in a mathematical way:

$$e(t) = r(t) - y(t) \tag{2.1}$$

where $e(t)$ is the error, $r(t)$ the reference and $y(t)$ the value of the measured variable at time t. The aim of the regulator is to apply a signal to the system to be controlled such that this difference is minimised. There are numerous controllers which utilise this scheme. The most simple ones are, for instance, on-off or proportional controllers. The former applies the maximum/minimum control signal if the error is greater/smaller than zero, whereas the latter uses a constant to multiply the error. On the one hand, on-off control increases the oscillations in the output as the system overreacts due to a small change in the error signal (controllers with hysteresis or dead zone are used instead). On the other hand, proportional control solves this issue but leads to static or steady state error [Åström and Hägglund 1995]. PID controllers elegantly deal with both of these problems by adding to the standard negative feedback with proportional control, P, an integral, I, and a derivative, D, term. While the integral term accumulates the error over time in order to cancel out steady state errors, the derivative term helps to the transient response by decreasing the amplitude of the oscillations of the controlled signal [Baltieri and C. Buckley 2019]. The control signal, $u(t)$, generated by a PID controller is usually expressed in the following form:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, dt + K_d \frac{de(t)}{dt} \tag{2.2}$$

where K_p, K_i and K_d are the proportional, integral and derivative gains, respectively, which are used to tune the effect of the P, I and D terms on the system response. Fig. 2.1 depicts the PID controller in a block diagram.

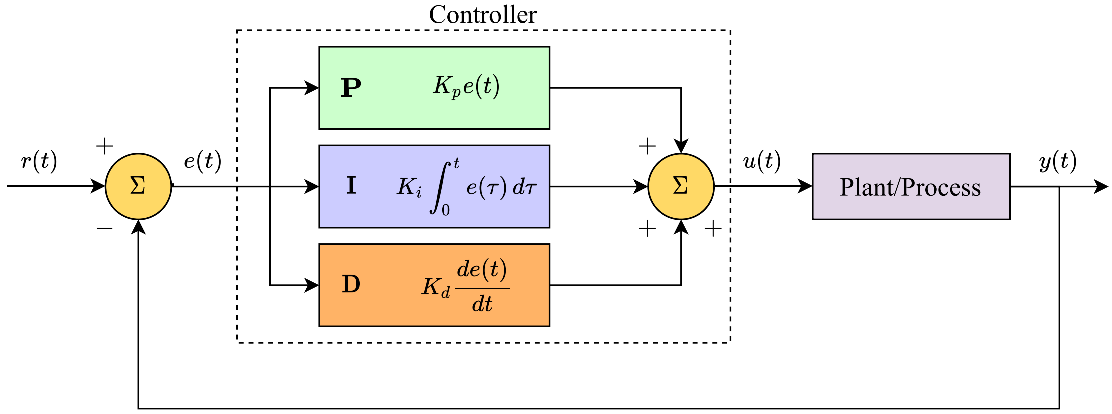


Figure 2.1: Block diagram of a PID controlled process.

Despite PID regulators can be designed analytically, their usefulness emerges when the mathematical model of the process is unknown[1]. Nonetheless, as mentioned earlier, there are some cases where PID is not sufficient. Situations where the dynamics of the process to be controlled become complicated, a more sophisticated controller may be needed.

This project examines the PID controller from a different perspective. Here, the regulator not only provides a control law but tries to infer the measurements' hidden causes by means of some information-theoretic and probabilistic concepts. Moreover, the PID gains encode random measurement noise that can be deduced within this framework. Next section develops these notions so as to provide an overall image of the PID regulator as a process of *active inference*.

2.2 PID Control as a Process of Active Inference

The regulator under study rests on the free energy principle. This concept, taken from the biological field, provides an explanation of how a system behaves in its surroundings.

[1]In 1942, Ziegler and Nichols proposed a method to tune the PID gains empirically.

This conduct is such that minimises an information theory concept known as surprisal which measures preferred observations deviation. Thus, the controller will act in its surroundings to modify the outcomes in a way that minimises surprisal. Surprisal is attained by computing the so-called *model evidence* or *marginal likelihood* in Bayesian inference, see [Smith, Karl Friston, and Whyte 2021]. Since this is often intractable, an upper bound denoted as *variational free energy* is employed instead. Hence, minimising the latter minimises surprisal, see Section 4.2.

In order to choose appropriate actions, the controller needs to know what caused the observed variables (i.e., measurements). These causes are unknown by the controller and are commonly referred to as *hidden states* or observations' *hidden causes*. The controller tries to infer these hidden states by means of a *generative model*. This model provides a mathematical description of how the observations were generated according to the controller. The controller's beliefs about measurements' hidden causes are refined with every new observation in a process known as *perception*. Furthermore, the controller chooses actions that change measurements in a way that minimises the aforesaid variational free energy. This scheme is known as active inference and is depicted in Fig. 2.2.

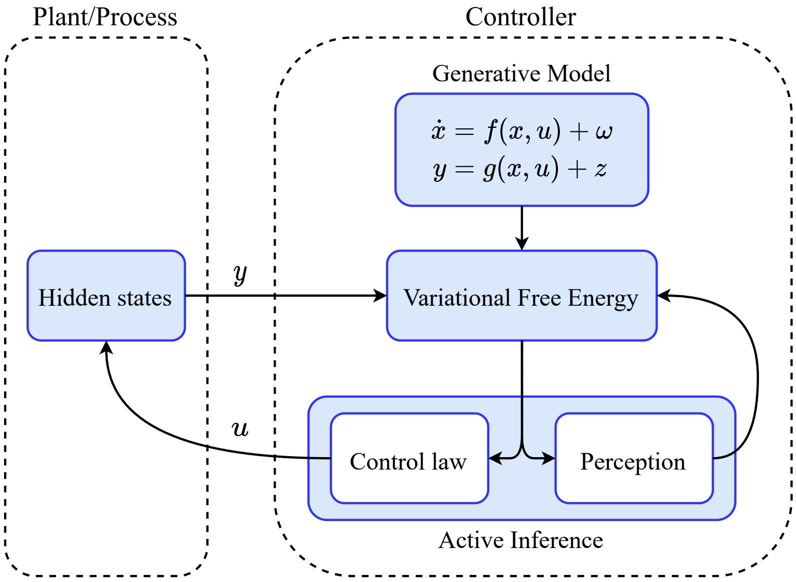


Figure 2.2: Schematic of the active inference framework.

As can be seen, hidden states generate some observations denoted with y. The regulator receives these measurements and, together with its beliefs about hidden states (i.e., perception) and the generative model, the variational free energy is computed. The control law and perception are updated in a manner that minimises this variational free energy. The control signal is then applied to the surroundings (i.e., plant or process) which will generate a new observation. The generative model presented in Fig. 2.2 expresses the hidden state dynamics model through the function $f(\cdot)$, whereas $g(\cdot)$ is a function that maps the hidden states into observations. Note that these functions may depend on the control signal as well. In this report, $f(\cdot)$ and $g(\cdot)$ are considered linear functions (i.e., linear generative model). The hidden state dynamics and measurements can be affected by random fluctuations denoted with ω and z, respectively.

Sections 4.3 and 4.4 show that, by choosing an appropriate linear generative model one can achieve a PID-like control law that minimises variational free energy. Moreover, these gains encode measurement noise properties. Therefore, deducing the noise present in the measurements permits inferring the PID gains and, as a result, attaining an optimal tuning.

2.3 Problem Formulation

This thesis examines the new PID control approach applied to a nonbiological system, namely to a refrigeration process. In general, refrigeration systems use PID regulation to control, for instance, the evaporator outlet temperature. As it is described in Section 3.1, a proper outlet temperature is crucial for the right functioning of the compressor as it ensures that no liquid exits the evaporator. Thus, the target of the project is to determine if this PID interpretation is suitable for industrial applications.

The aforementioned could be summed up in the following problem formulation:

Can PID control as a process of active inference be useful to regulate the evaporator outlet temperature in a refrigeration process?

3 Refrigeration System

Refrigeration systems cover a wide range of industrial applications, from food cooling to air conditioning [Dincer 2017]. All these follow the same principle: they utilise a vapour-compression cycle to transfer heat. This chapter outlines the basic concepts that take part in a refrigeration process. Nonetheless, more detailed information can be found in [Larsen 2006] and [Kasper Vinther 2014]. The first section of this chapter explains the different elements that compose the system as well as the phases involved in the thermodynamic cycle. The second section provides a mathematical model to determine the outlet temperature of the evaporator given an opening degree of the expansion valve. This model is based on the test bench provided by Danfoss A/S to Aalborg University.

3.1 The Vapour-Compression Cycle

The goal of a vapour-compression cycle is to remove heat from a cold reservoir (e.g., a cold storage room) and transfer it to a hot reservoir, usually the surroundings [Larsen 2006]. For this purpose, a refrigerant circulates between two heat exchangers, namely an evaporator and a condenser, as shown in Fig. 3.1.

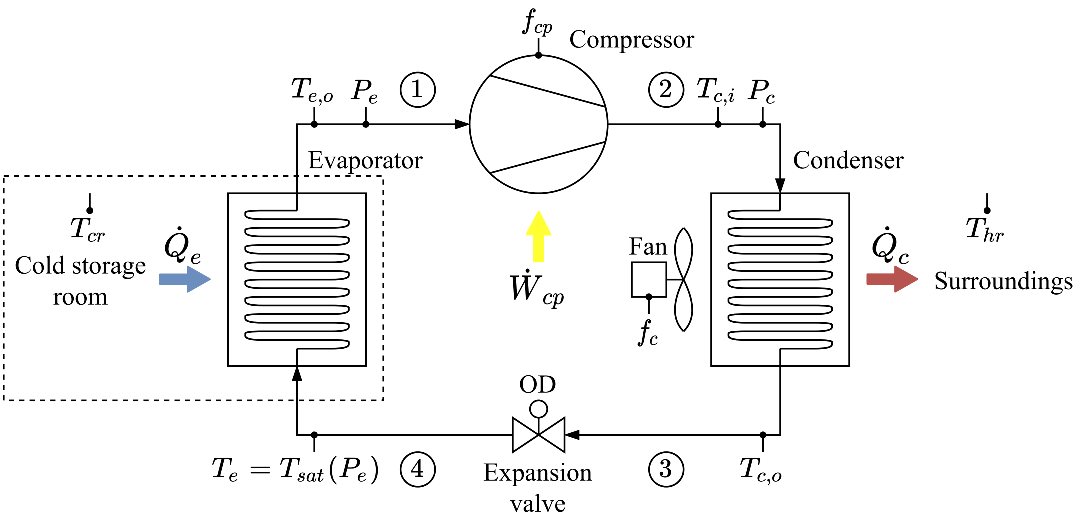


Figure 3.1: Schematic of a basic refrigeration system.

In order to establish the required heat transfer, the temperature in the cold/hot reservoir needs to be higher/lower than the evaporation/condensation temperature. This is achieved by the usage of an expansion valve and a compressor, respectively. While the former leads to a pressure drop of the refrigerant, the latter circulates the refrigerant from the evaporator to the condenser by increasing its pressure. Thus, the vapour-compression cycle consists of four connected processes: compression, condensation, expansion and evaporation. Fig. 3.2 depicts the entire cycle in a pressure-enthalpy diagram.

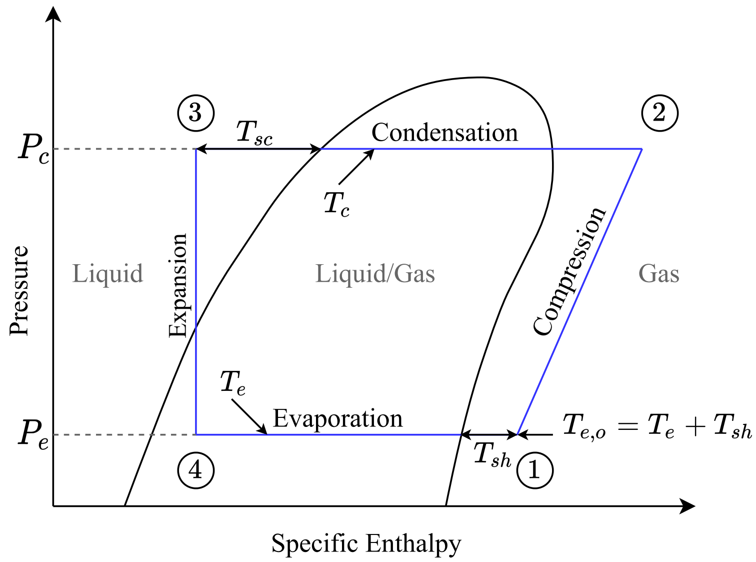


Figure 3.2: Pressure-enthalpy diagram of a refrigeration cycle.

As can be seen, it is assumed an isobaric condensation and evaporation. Next, these four processes are detailed by following Fig. 3.1 and Fig. 3.2.

1 → 2 **Compression**: at the compressor inlet, the refrigerant is in gas phase with low pressure, P_e, and temperature, $T_{e,o}$. By compressing the refrigerant, the pressure as well as the temperature increase. In Fig. 3.1, $\dot{W}_{cp}$ indicates the compressor power consumption (f_{cp} denotes the frequency at which the compressor operates). It is assumed that no heat is transmitted to the surroundings during the compression phase (i.e., adiabatic process).

2 → 3 **Condensation**: the refrigerant is then directed to a condenser unit. Here, the refrigerant starts to condense at a constant pressure, P_c, changing its phase from gas to liquid. Usually, a fan blowing air across the condenser facilitates the heat transfer, $\dot{Q}_e$. Through the last part of the condenser, the refrigerant temperature is lowered below its condensing temperature so as to ensure that the refrigerant enters the expansion valve in liquid phase. This step is known as subcooling and is represented by T_{sc} in Fig. 3.2.

3 → 4 **Expansion**: the expansion valve separates the high-pressure side, P_c, from the low-pressure side, P_e. Thereby, the liquid is exposed to a large pressure decrease, provoking a fraction of the refrigerant to evaporate. This partial phase change causes the temperature to drop down to the evaporation temperature, T_e, determined by the low pressure P_e[1]. Since no work is done during the expansion, and considering that the valve is properly insulated, the enthalpy remains unchanged. From the expansion valve, the refrigerant flows to the evaporator.

[1]Refrigerants have the property, along with other fluids and gases, that the saturation temperature, T_{sat}, uniquely depends on the pressure [Larsen 2006].

④ → ① **Evaporation**: the low temperature in the evaporator inlet, T_e, enables a heat transfer from the cold reservoir to the refrigerant. Hence, the remaining refrigerant in liquid phase evaporates at a constant temperature. At the evaporator outlet, all of the refrigerant has evaporated and the temperature, $T_{e,o}$, has increased slightly above the evaporation temperature, T_e. This temperature increment is known as superheat, T_{sh}, and ensures that no liquid gets into the compressor. Superheat is controlled by regulating the mass flow into the evaporator through the expansion valve and suitable levels are between 6 and 12 degrees [Kasper Vinther 2014]. The refrigerant has now completed the vapour-compression cycle and returns to the compressor inlet.

The following section furnishes a mathematical model to describe the behaviour of the evaporator outlet temperature, $T_{e,o}$, as a function of the electronic expansion valve (EEV) opening degree, OD.

3.2 Evaporator Model

The behaviour of the evaporator outlet temperature as a function of the EEV opening degree is nonlinear. This relationship is illustrated in Fig. 3.3, where $T_{c,r}$ is the cold reservoir temperature and T_e the evaporation temperature. The evaporator input/output relationship is almost flat for low opening degrees and then suddenly drops close to the minimum stable superheat and then flattens again when the evaporator is flooded (i.e., high opening degrees) [Kasper Vinther 2014].

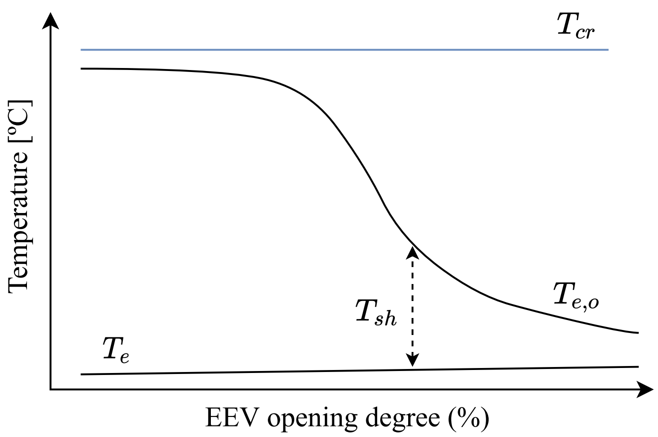


Figure 3.3: Nonlinear relationship between EEV opening degree, OD, and evaporator outlet temperature, $T_{e,o}$.

To model this nonlinearity and the evaporator dynamics, [Kasper Vinther 2014] proposes a Wiener-Hammerstein model structure given as:

$$y(t) = (f_o \circledast f(f_i \circledast u))\,(t) \tag{3.1}$$

where u is the input to the system, y the output, f_i the input dynamics function (whose Laplace domain transfer function is $F_i(s) = \mathcal{L}\{f_o(t)\}$), $f(\cdot)$ the static nonlinear function, f_o is the function describing the output dynamics (having the Laplace domain transfer function $F_o(s) = \mathcal{L}\{f_o(t)\}$) and $\circledast$ the convolution operator. Fig. 3.4 depicts the Wiener-Hammerstein model structure.

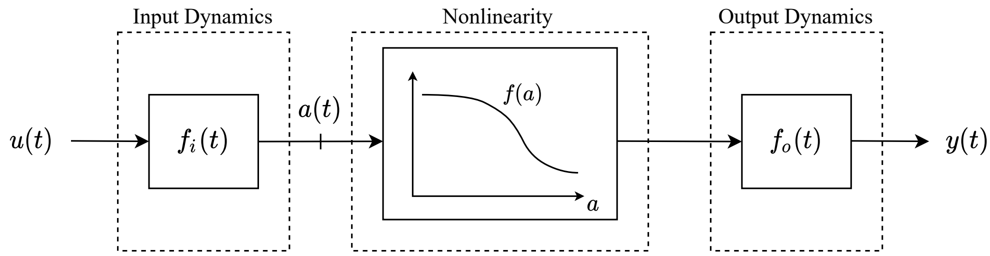


Figure 3.4: Wiener-Hammerstein model structure.

It is assumed that $f(\cdot)$ is either monotonically decreasing or increasing, time-invariant, smooth and bounded within the input set $a \in A \subset \mathbb{R}$. Moreover, the first three derivatives with respect to a are also bounded and continuous, and $f'(\cdot)$ is either non-negative or non-positive bell shaped with a unique extremum. Therefore, the second derivative, $f''(\cdot)$ has a unique zero-crossing. That said, the Wiener-Hammerstein model in Laplace domain is parametrised by a first-order transfer function:

$$F_i(s) = \frac{1}{T_{sysi}s + 1} \tag{3.2}$$

a static nonlinearity:

$$T_{e,o} = -k_1 \arctan\left(k_2(OD - OD^*)\right) + T^*_{e,o} \tag{3.3}$$

and a first-order plus dead time dynamics (FOPDT):

$$F_o(s) = \frac{1}{T_{sys}s + 1}\exp(-sT_d) \tag{3.4}$$

where the gains k_1, k_2, the time constant T_{sys}, the delay T_d and the maximum slope point $(OD^*, T^*_{e,o})$ of the nonlinear function can be identified using a simple ramp test and a biased relay feedback test, as shown in [Kasper Vinther 2014]. Furthermore, input dynamics are assumed to be fast, hence negligible compared to the output dynamics (time constant, T_{sysi}, is set to 2) [Vinther et al. 2013]. These tests were conducted in the Water chiller system located at Aalborg University which operates with the R134a refrigerant. The obtained parameters can be found in Table 3.1.

Parameter	Water chiller
k_1	4.89
k_2	1.31
T_{sys}	31.51
T_d	26
OD^*	50.78
$T^*_{e,o}$	12.41

Table 3.1: Identified evaporator model parameters for the water chiller system.

4 | Controller Design

This chapter aims to describe and develop the mathematical foundation of the PID controller as a process of active inference employing linear generative models. Due to the unfamiliarity of some of these concepts, the chapter is divided into different sections, where each of them tries to shed light in every term separately, following a divide-and-conquer strategy.

The first section starts defining some widely used terms in the machine learning field as *agent* or *environment* and what do they really imply in a control process. The following section deals with the explanation of the free energy principle. This concept plays a key role in the interpretation of the controller design since it provides a statement of how a biological system (an agent) behaves so as to thrive in its medium (environment). The next section establishes a set of linear equations that best describe the environment according to the agent (linear generative model). The fourth section handles how the agent can update its perception of the world by minimising the free energy. The fifth section furnishes a framework for active inference. That is, how an agent can act on its surroundings in order to reach some desired states. Finally, the last section puts all these concepts together to construct the PID controller.

4.1 The Agent-Environment Interface

This terminology is commonly utilised in the reinforcement learning literature. Reinforcement learning is a class of algorithms in the field of machine learning that aims learning from interaction to achieve a goal [Sutton and Barto 2018]. The learner and decision maker is the agent. It interacts, through actions, with the environment which comprises everything that the agent cannot control [Wiering and Otterlo 2012]. These interact continuously: the agent selecting actions and the environment responding to these actions and presenting new situations to the agent. The environment may also give a reward as feedback. Therefore, the goal of the agent is to perform actions that maximize the reward over time.

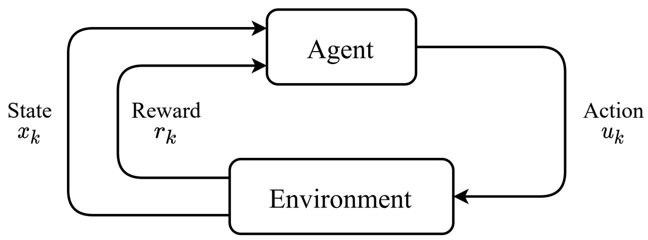


Figure 4.1: The agent-environment interaction.

Fig. 4.1 depicts a block diagram of the above described elements. At every time step, $k = 0, 1, 2, ...$, the agent receives some representation of the environment's state, x_k, and, based on this information, selects an action u_k. One time step later, the agent receives a reward r_{k+1} and finds the environment in a new state, x_{k+1}. This process is iterated until the system reaches the target. As it will be seen later, some of these states cannot be known directly by the agent. They are usually termed as environmental hidden states or hidden causes. The measurable states are typically referred to as observations or outcomes.

From a control theory perspective, the agent and environment would correspond to the controller and plant (controlled system), respectively. Specifically, in the case under study, the agent would correspond to the signal governing the opening degree of the valve while the environment would be composed of the refrigeration system elements (e.g., valve, evaporator, etc.). From a biological point of view, the agent could be the brain, whereas the rest of the body and its surroundings the environment. Hence, the agent-environment interface is used to describe this interaction so as to reach a desired environment's state.

4.2 The Free Energy Principle

The free energy principle (henceforth, FEP) provides an explanation to the behaviour of (biological) agents when interacting with its environment. Furthermore, it supplies a mathematical framework which, ultimately, permits the design of a controller. The controller regarded in this report rests upon this principle.

The free energy principle first emerged to provide a unified theory for the brain. However, FEP soon attempted to extend beyond the brain sciences to account for other biological processes [C. L. Buckley et al. 2017]. Due to its origins in neuroscience and biology, some vocabulary from these fields will be adopted throughout this section.

FEP starts with the premise that biological agents resist the dispersion of their sensory input (i.e., observations) despite fluctuations in the environment. This is characterised by open[1] organisms that exhibit *homeostasis*[2]. In other words, biological creatures must maintain their states within certain bounds [Karl Friston 2012]. Nevertheless, this statement might lead to the following question: how a biological system, exposed to random and unpredictable fluctuations in its milieu, can restrict itself to occupying a limited number of states? The answer proposed is based on free energy minimisation which asserts that biological systems model their environment, predict what will happen next and encounter violations of those predictions. By changing their perception of the environment and acting on it they can limit the number of states they can find themselves in. Thereby, agents must avoid the occurrence of events which are atypical in their habitable environment (e.g., a fish out of water). [C. L. Buckley et al. 2017]. The atypicality of an event can be quantified by the negative natural logarithm of the

[1] In the sense that they exchange energy and entropy with the environment.

[2] *Hómios*, similar and *stásis*, standing still. Is the ability to maintain a relatively stable internal state that persists despite changes in the environment (e.g., body temperature within limits).

probability of its sensory data:

$$-\ln p(y) \tag{4.1}$$

Where y corresponds to the observations perceived by the agent (sensory data). This term is commonly known in information theory as self-information or surprisal and measures the deviation from preferred outcomes. Thus, it can be seen that lower probability events will generate a higher surprisal (e.g., $-\ln 0.5 = 0.69$ while $-\ln 0.9 = 0.1$). This implies that maximising the model evidence of the environment, $p(y)$, minimises the surprisal. For instance, consider the body temperature. Humans can only survive if body temperatures continue to be observed within the range of $36.5 - 37.5$ degrees Celsius. Hence, the human body implicitly entails a higher probability of making such observations. If a human agent perceives that body temperature differs from the expected temperature, it will act on the environment so as to minimise this deviation (e.g., seek shelter when it is cold). In this sense, body temperature within survivable ranges are the least surprising [Smith, Karl Friston, and Whyte 2021]. A different perspective for understanding why an agent must minimise surprisal is based on the concept of *entropy*. Consider the mean of the surprisal:

$$\mathbb{E}[-\ln p(y)] = \int -\ln p(y)p(y)\, dy \tag{4.2}$$

In information theory, Eq. (4.2) is known as entropy. This means that minimising surprisal also minimises the entropy or dispersion of sensory outcomes.

As stated in Section 4.1, there are some environmental states that cannot be perceived by the agent. These correspond to the hidden causes, x, of the sensory input, y. Thus, to maximise preferred outcomes (minimise surprisal), the agent must maintain a generative model, $p(y, x)$, of these hidden states that is sufficiently accurate to anticipate and avoid observations that imply high surprisal. In other words, the agent must have a model of how these observations were generated. Mathematically, the probability of sensory input based on a generative model can be expressed, in the continuous case, as:

$$p(y) = \int p(y, x)\, dx = \int p(y \mid x)p(x)\, dx \tag{4.3}$$

Moreover, the joint or generative density (G-density) can be factorised into the likelihood and the prior probabilities of the hidden states, as shown in the right-hand side of Eq. (4.3). Assuming that this equation is feasible, the agent can now update its beliefs about the hidden states by means of Bayesian inference:

$$p(x \mid y) = \frac{p(y, x)p(x)}{p(y)} = \frac{p(y \mid x)p(x)}{\int p(y \mid x)p(x)} \tag{4.4}$$

Hence, given some prior beliefs of the hidden states, $p(x)$, and the generative model, the agent can infer the posterior probabilities, $p(x \mid y)$, of the hidden causes that generated the observations. That is to say, the agent updates its own model of the world (environment) with every new observation. This step is known as perception.

Unfortunately, Eq. (4.3) is often analytically intractable. In the discrete case, when this integral reduces to a sum, the number of calculations may grow exponentially with the number of states. To deal with this, an auxiliary probability density which represents the current best guess of the sensory input causes is introduced. This auxiliary density has been defined as recognition density (R-density), $q(x)$, and tries to approximate the true posterior, $p(x \mid y)$ [C. L. Buckley et al. 2017].

To measure the difference between the R-density and the true posterior, the Kullback-Leibler (KL) divergence is employed:

$$D_{\mathrm{KL}}\left(q(x) \mid\mid p(x \mid y)\right) := \int q(x) \ln\left(\frac{q(x)}{p(x \mid y)}\right) dx \tag{4.5}$$

When $q(x)$ matches $p(x \mid y)$, the divergence becomes zero. Thereby, a recognition density that minimises the divergence will provide a good approximation to the true posterior. Nevertheless, since the true posterior cannot be known, the KL divergence cannot be evaluated either. However, utilising Bayes' theorem, Eq. (4.5) can be rewritten as:

$$\begin{aligned} D_{\mathrm{KL}}\left(q(x) \mid\mid p(x \mid y)\right) &= \int q(x) \ln\left(\frac{q(x)p(y)}{p(y,x)}\right) dx \\ &= \int q(x) \ln\left(\frac{q(x)}{p(y,x)}\right) dx \; + \; \ln p(y) \int q(x)\, dx \end{aligned} \tag{4.6}$$

Assuming that the R-density is normalised, that is:

$$\int q(x)\, dx = 1 \tag{4.7}$$

The Kullback-Leibler divergence can be reformulated as:

$$D_{\mathrm{KL}}\left(q(x) \mid\mid p(x \mid y)\right) = \underbrace{\int q(x) \ln\left(\frac{q(x)}{p(y,x)}\right) dx}_{F} \; + \; \ln p(y) \tag{4.8}$$

The first term on the right-hand side is known as free energy or variational free energy (VFE) and is denoted by F. Thereby, minimising VFE with respect to the R-density will minimise the KL divergence. Furthermore, the KL divergence is always greater or equal than zero, see Appendix A.1 for the proof. This leads to the following inequality:

$$F \geq -\ln p(y) \tag{4.9}$$

where the right-hand side of the inequality corresponds to the surprisal. As stated previously, the surprisal cannot be known directly. Nonetheless, as it will be seen later, VFE can be evaluated directly since $q(x)$ and $p(y, x)$ can be freely specified. Therefore, VFE provides an upper bound on the surprisal, thus minimising VFE will minimise the surprisal.

In conclusion, minimising VFE with respect to the recognition density given an appropriate model for the generative density permits approximating the true posterior. The following subsection establishes the R-density.

4.2.1 R-density

In order to minimise the VFE with respect to the R-density, the latter must be specified. It is assumed that this density takes Gaussian form. In this scenario, the sufficient statistics of this Gaussian form (i.e., mean and variance) become parameters which can be optimised numerically to minimise VFE [C. L. Buckley et al. 2017]. Thereby, the R-density takes the following form in the univariate case:

$$q(x) \equiv \mathcal{N}(x\,;\,\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right) \tag{4.10}$$

where μ and σ^2 correspond to the mean and variance, respectively, of a single environmental variable, x.

Splitting the fraction term in the VFE expression leads to:

$$F = \int q(x)E(y,x)\,dx + \int q(x)\ln q(x)\,dx \tag{4.11}$$

where:

$$E(y,x) = -\ln p(y,x) \tag{4.12}$$

By analogy with Helmholtz' thermodynamic free energy, the first term in Eq. (4.11) is called *average energy* and the second term is known as the negative entropy [Adkins 1975]. Substituting Eq. (4.10) into Eq. (4.11) yields:

$$\begin{aligned} F &= -\ln\sqrt{2\pi\sigma^2}\int q(x)\,dx - \frac{1}{2\sigma^2}\int q(x)(x-\mu)^2\,dx + \int q(x)E(y,x)\,dx \\ &= -\frac{1}{2}\ln 2\pi\sigma^2 - \frac{1}{2} + \int q(x)E(y,x)\,dx \end{aligned} \tag{4.13}$$

where the first two terms on the right-hand side of the second step have been simplified by means of the normalisation property, see Eq. (4.7), and the definition of variance in the continuous case, respectively. The third term, however, demands further considerations. It will be assumed that the energy, $E(y,x)$, is a smooth function of x and that the R-density is sharply peaked at its mean value, μ. In other words, the Gaussian density is squeezed towards a delta function [C. L. Buckley et al. 2017]. Under these assumptions, it is noticed that the integration is appreciably nonzero only at the peak. Thus, $E(y,x)$ can be evaluated around $x=\mu$ using Taylor expansion with respect to a small increment, δx:

$$\int q(x)E(y,x)\,dx \approx \int q(x)\left(E(y,\mu) + \left.\frac{\partial E(y,x)}{\partial x}\right|_{\mu}\delta x + \frac{1}{2}\left.\frac{\partial^2 E(y,x)}{\partial x^2}\right|_{\mu}\delta x^2\right)dx \tag{4.14}$$

Applying the normalisation property in the first term and substituting $\delta x = x-\mu$ results in:

$$\int q(x)E(y,x)\,dx \approx$$

$$E(y,\mu) + \left.\frac{\partial E(y,x)}{\partial x}\right|_{\mu}\int q(x)(x-\mu)\,dx + \frac{1}{2}\left.\frac{\partial^2 E(y,x)}{\partial x^2}\right|_{\mu}\int q(x)(x-\mu)^2\,dx \tag{4.15}$$

The integral in the second term becomes zero since it corresponds to the expectation of a random variable:

$$\mathbb{E}[X] = \int xq(x)\,dx = \mu \tag{4.16}$$

Furthermore, identifying the expression for the variance in the third term allows approximating the average energy as:

$$\int q(x)E(y,x)\,dx \approx E(y,\mu) + \frac{1}{2}\left.\frac{\partial^2 E(y,x)}{\partial x^2}\right|_{\mu} \sigma^2 \tag{4.17}$$

Substituting all the terms derived so far in Eq. (4.13) furnishes an appropriate expression for the VFE:

$$F = E(y,\mu) + \frac{1}{2}\left(\left.\frac{\partial^2 E(y,x)}{\partial x^2}\right|_{\mu} \sigma^2 - \ln 2\pi\sigma^2 - 1\right) \tag{4.18}$$

which is now a function of the the sensory input and the Gaussian mean and variance i.e., $F(y,\mu,\sigma^2)$. To simplify further, the variance dependency in Eq. (4.18) is removed by taking the derivative with respect to σ^2 as follows:

$$\frac{\partial F}{\partial \sigma^2} = \frac{1}{2}\left(\left.\frac{\partial^2 E(y,x)}{\partial x^2}\right|_{\mu} - \frac{1}{\sigma^2}\right) \tag{4.19}$$

Setting Eq. (4.19) equal to zero and solving for the variance leads to:

$$\sigma^2_{\text{opt.}} = \left(\left.\frac{\partial^2 E(y,x)}{\partial x^2}\right|_{\mu}\right)^{-1} \tag{4.20}$$

where $\sigma^2_{\text{opt.}}$ stands for the optimal variance i.e., the variance that minimises the VFE. Substituting Eq. (4.20) into Eq. (4.18) one gets the final simplified form of the VFE:

$$F = E(y,\mu) - \frac{1}{2}\ln 2\pi\sigma^2_{\text{opt.}} \tag{4.21}$$

Now, the VFE only depends on the first-order Gaussian statistics (i.e., the mean) of the environmental variable, x, and the sensory input, y. In other words, the VFE has been rearranged in terms of a generative density (G-density), $p(y,\mu)$, and the R-density's mean. The following section tackles how this joint density can be obtained by means of a dynamic generative model.

4.2.2 G-density

Once the recognition density has been defined, one needs to specify the generative density so as to evaluate the VFE. For this purpose, the agent needs a model of the environmental causes of sensory data. In other words, a model of the hidden states and their relation to the observations perceived by the agent. It is assumed a dynamical generative model that follows a Langevin-type equation:

$$
\begin{aligned}
y &= g(x,v) + z & \dot{x} &= f(x,v) + \omega \\
\dot{y} &= \frac{\partial g(x,v)}{\partial x}\dot{x} + \frac{\partial g(x,v)}{\partial v}\dot{v} + \dot{z} & \ddot{x} &= \frac{\partial f(x,v)}{\partial x}\dot{x} + \frac{\partial f(x,v)}{\partial v}\dot{v} + \dot{\omega} \\
&\vdots & &\vdots
\end{aligned}
\tag{4.22}
$$

with x as the hidden states, y as observations and v representing the exogenous inputs[3]. Functions $g(\cdot)$ and $f(\cdot)$ map hidden states/inputs to observations and the dynamics of hidden states/inputs, respectively. Random fluctuations in the dynamics (i.e., process noise) are denoted by ω, whereas z represents the measurement noise. The dot symbols denote time derivatives, that is, the higher orders of a variable.

For practical purposes, the nonlinear terms (e.g., $\dot{x}^2$, $\dot{x}\ddot{x}$, etc.) are neglected under a local linearity assumption, see [Karl Friston, Mattout, et al. 2007]. Then, Eq. (4.22) can be rewritten in a more compact form:

$$
\tilde{y} = g(\tilde{x}, \tilde{v}) + \tilde{z} \qquad \dot{\tilde{x}} = f(\tilde{x}, \tilde{v}) + \tilde{\omega} \tag{4.23}
$$

where the tilde sign comprises a variable and its higher orders (e.g., $\tilde{y} = \{y, \dot{y}, \ddot{y}, ...\}$). The stochastic model in Eq. (4.23) can then be described in terms of a generative density:

$$
p(\tilde{y}, \tilde{x}, \tilde{v}\,;\, \theta, \gamma) = p(\tilde{y} \mid \tilde{x}, \tilde{v}\,;\, \theta, \gamma) p(\tilde{x}, \tilde{v}\,;\, \theta, \gamma) \tag{4.24}
$$

which can be factorised, respectively, into the likelihood and the prior, as shown in the right-hand side of the equation. Here, θ represents the parameters of the generative model functions, that is, $f(\cdot)$ and $g(\cdot)$, while the hyperparameters, γ, encode the properties of the random fluctuations, $\tilde{\omega}$ and $\tilde{z}$. Bear in mind that Eq. (4.24) has been defined previously as $p(x, y)$, but the exogenous inputs, as well as the parameters and hyperparameters are now included in the joint distribution.

In Subsection 4.2.1, the VFE was determined in terms of the average energy, $E(y, \mu)$, and the R-density's sufficient statistics (i.e., mean, μ, and the optimal variance, σ^2_{opt}), see Eq. (4.21). Since free energy needs to be minimised, one can neglect the constant variance term as it does not affect the optimisation problem. Thus, VFE can be approximated as:

$$
F \approx E(\tilde{y}, \tilde{\mu}, \tilde{v}\,;\, \theta, \gamma) = -\ln p(\tilde{y}, \tilde{\mu}, \tilde{v}\,;\, \theta, \gamma) = -\ln p(\tilde{y} \mid \tilde{\mu}, \tilde{v}\,;\, \theta, \gamma) - \ln p(\tilde{\mu}, \tilde{v}\,;\, \theta, \gamma) \tag{4.25}
$$

Conceptually, this expression suggests that the agent represents only the most likely environmental causes of sensory data, $\tilde{\mu}$, and not the details of their distribution per se [C. L. Buckley et al. 2017]. Furthermore, it is assumed that $\tilde{z}$ and $\tilde{\omega}$ are modelled as white Gaussian noise:

$$
\tilde{z} \sim \mathcal{N}(0, \sigma^2_{\tilde{z}}) \qquad \tilde{\omega} \sim \mathcal{N}(0, \sigma^2_{\tilde{\omega}}) \tag{4.26}
$$

Therefore, the likelihood and prior in Eq. (4.24) will also follow a Gaussian distribution with variance $\sigma^2_{\tilde{z}}$ and $\sigma^2_{\tilde{\omega}}$, respectively. Hence, the likelihood can be expressed as:

$$
p(\tilde{y} \mid \tilde{\mu}, \tilde{v}\,;\, \theta, \gamma) = \frac{1}{\sqrt{2\pi\sigma^2_{\tilde{z}}}} \exp\left(-\frac{(\tilde{y} - g(\tilde{\mu}, \tilde{v}))^2}{2\sigma^2_{\tilde{z}}} \right) \tag{4.27}
$$

[3] References, disturbances and noise.

while the prior is described by:

$$p(\tilde{\mu}, \tilde{v}\,;\,\theta, \gamma) = p(\tilde{\mu} \mid \tilde{v}\,;\,\theta, \gamma)p(\tilde{v}\,;\,\theta, \gamma) \tag{4.28}$$

To simplify the formulation, it will be supposed that there is no uncertainty on the priors for v, so that $p(\tilde{v}\,;\,\theta, \gamma)$ in Eq. (4.28) becomes a delta function and inputs, $\tilde{v}$, is reduced to their prior expectations [Baltieri and C. Buckley 2019]. In other words, it is considered that the exogenous inputs are known. Rewriting Eq. (4.28) leads to:

$$\begin{aligned} p(\tilde{\mu}, \tilde{v}\,;\,\theta, \gamma) &= p(\mu, \dot{\mu}, \ddot{\mu}, ... \mid \tilde{v}\,;\,\theta, \gamma) = p(\dot{\tilde{\mu}} \mid \tilde{\mu}, v\,;\,\theta, \gamma) \\ &= \frac{1}{\sqrt{2\pi\sigma^2_{\tilde{\omega}}}} \exp\left(-\frac{\left(\dot{\tilde{\mu}} - f(\tilde{\mu}, \tilde{v})\right)^2}{2\sigma^2_{\tilde{\omega}}}\right) \end{aligned} \tag{4.29}$$

Substituting Eq. (4.27) and Eq. (4.29) into Eq. (4.25) yields the final expression for the VFE:

$$F \approx \frac{1}{2}\left(\pi_{\tilde{z}}\left(\tilde{y} - g(\tilde{\mu}, \tilde{v})\right)^2 + \pi_{\tilde{\omega}}\left(\dot{\tilde{\mu}} - f(\tilde{\mu}, \tilde{v})\right)^2 - \ln(\pi_{\tilde{z}}\pi_{\tilde{\omega}})\right) \tag{4.30}$$

where $\pi_{\tilde{z}}$ and $\pi_{\tilde{\omega}}$ correspond to the precisions, that is, the inverse of the variances (e.g., $\pi_z = 1/\sigma^2_z$). These precisions form the set of hyperparameters, i.e., $\gamma = \{\pi_{\tilde{z}}, \pi_{\tilde{\omega}}\}$, and can be optimised utilising a gradient-descent technique, see Subsection 4.6.1. In addition, the constant term that appears when developing Eq. (4.25) has been neglected since it does not contribute to the VFE minimisation.

Having defined the variational free energy by means of the recognition and generative densities, the following step is to determine the $f(\cdot)$ and $g(\cdot)$ functions. Next section provides a mathematical description of these functions which will ultimately permit the minimisation of the variational free energy and the design of the PID controller.

4.3 Linear Generative Model

To implement the PID control, the agent's generative model described in Eq. (4.23) is established as a generalised linear state-space model of second order (i.e., two higher orders, anything beyond that is zero mean Gaussian noise) [Baltieri and C. Buckley 2019]:

$$\begin{aligned} y &= \mu + z & \dot{\mu} &= -\alpha(\mu - v) + \omega \\ \dot{y} &= \dot{\mu} + \dot{z} & \ddot{\mu} &= -\alpha(\dot{\mu} - \dot{v}) + \dot{\omega} \\ \ddot{y} &= \ddot{\mu} + \ddot{z} & \dddot{\mu} &= -\alpha(\ddot{\mu} - \ddot{v}) + \ddot{\omega} \end{aligned} \tag{4.31}$$

where $\alpha \in \theta$ is a parameter. As stated previously, the agent models the environmental causes of sensory data, x, considering only their expected value, μ (R-density's first-order Gaussian statistics). Moreover, each dynamical/measurement order receives independent white Gaussian noise denoted by $z, \omega, \dot{z}, \dot{\omega}, ...$. Thus, the likelihood characterised in Eq. (4.27) can be redefined as:

$$\begin{aligned} p(\tilde{y} \mid \tilde{\mu}, \tilde{v}\,;\,\theta, \gamma) &= p(y \mid \mu\,;\,\gamma)p(\dot{y} \mid \dot{\mu}\,;\,\gamma)p(\ddot{y} \mid \ddot{\mu}\,;\,\gamma) \\ &= \frac{1}{\sqrt{2\pi\sigma^2_z}}\frac{1}{\sqrt{2\pi\sigma^2_{\dot{z}}}}\frac{1}{\sqrt{2\pi\sigma^2_{\ddot{z}}}} \exp\left(-\frac{\varepsilon^2_z}{2\sigma^2_z} - \frac{\varepsilon^2_{\dot{z}}}{2\sigma^2_{\dot{z}}} - \frac{\varepsilon^2_{\ddot{z}}}{2\sigma^2_{\ddot{z}}}\right) \end{aligned} \tag{4.32}$$

For notational simplicity, the terms ε_z, $\varepsilon_{\dot{z}}$, $\varepsilon_{\ddot{z}}$ are employed to represent the error in each measurement order:

$$\begin{aligned}
\varepsilon_z &= y - \mu \\
\varepsilon_{\dot{z}} &= \dot{y} - \dot{\mu} \\
\varepsilon_{\ddot{z}} &= \ddot{y} - \ddot{\mu}
\end{aligned} \tag{4.33}$$

The prior in Eq. (4.29) can be restated in the following form:

$$\begin{aligned}
p(\tilde{\mu}, \tilde{v}\,;\,\theta,\gamma) =&\ p(\dot{\mu} \mid \mu, v\,;\,\theta,\gamma)p(\ddot{\mu} \mid \dot{\mu}, \dot{v}\,;\,\theta,\gamma)p(\dddot{\mu} \mid \ddot{\mu}, \ddot{v}\,;\,\theta,\gamma) \\
=&\ \frac{1}{\sqrt{2\pi\sigma_\omega^2}}\frac{1}{\sqrt{2\pi\sigma_{\dot{\omega}}^2}}\frac{1}{\sqrt{2\pi\sigma_{\ddot{\omega}}^2}} \exp\left(-\frac{\varepsilon_\omega^2}{2\sigma_\omega^2} - \frac{\varepsilon_{\dot{\omega}}^2}{2\sigma_{\dot{\omega}}^2} - \frac{\varepsilon_{\ddot{\omega}}^2}{2\sigma_{\ddot{\omega}}^2}\right)
\end{aligned} \tag{4.34}$$

In the same manner, ε_ω, $\varepsilon_{\dot{\omega}}$, $\varepsilon_{\ddot{\omega}}$ symbolise the error terms in each dynamical order:

$$\begin{aligned}
\varepsilon_\omega &= \dot{\mu} + \alpha\,(\mu - v) \\
\varepsilon_{\dot{\omega}} &= \ddot{\mu} + \alpha\,(\dot{\mu} - \dot{v}) \\
\varepsilon_{\ddot{\omega}} &= \dddot{\mu} + \alpha\,(\ddot{\mu} - \ddot{v})
\end{aligned} \tag{4.35}$$

Substituting Eq. (4.32) and Eq. (4.34) into Eq. (4.25), the variational free energy in Eq. (4.30) then becomes:

$$\begin{aligned}
F \approx \frac{1}{2}\Bigg(&\pi_z\,(y-\mu)^2 + \pi_{\dot{z}}\,(\dot{y}-\dot{\mu})^2 + \pi_{\ddot{z}}\,(\ddot{y}-\ddot{\mu})^2 + \pi_\omega\,(\dot{\mu} + \alpha\,(\mu - v))^2 \\
&+ \pi_{\dot{\omega}}\,(\ddot{\mu} + \alpha\,(\dot{\mu} - \dot{v}))^2 + \pi_{\ddot{\omega}}\,(\dddot{\mu} + \alpha\,(\ddot{\mu} - \ddot{v}))^2 - \ln\left(\pi_z\pi_{\dot{z}}\pi_{\ddot{z}}\pi_\omega\pi_{\dot{\omega}}\pi_{\ddot{\omega}}\right)\Bigg)
\end{aligned} \tag{4.36}$$

So far, the recognition and generative densities together with the linear generative model have been used to provide a tractable expression of the VFE. The upcoming sections furnish a gradient-descent scheme to minimise VFE, thereby diminishing the surprisal, see Eq. (4.9).

4.4 Variational Free Energy Minimisation

The agent has two ways of minimising VFE and therefore surprisal: by changing its beliefs or hypothesis (perception) or change the world (action). For instance, a robot may believe that its limb is raised, but observes it is not, then it can change its belief or raise the limb [Pio-Lopez et al. 2016]. The current section copes with the VFE minimisation through perception.

4.4.1 Perception

The linear generative model defined in the previous section describes how the environment behaves according to the agent. That is, what are the hidden causes of sensory input.

Since the hidden states cannot be known directly, their expected value, $\tilde{\mu}$, is employed instead. Thus, $\tilde{\mu}$ encodes the agent's beliefs about the hidden states. With every new observation, these beliefs are refined in a way that minimise VFE, thus making the R-density a good approximation of the true posterior, see Eq. (4.8). This process where the internal model (i.e., agent's model) is updated is known as perception.

It is suggested that agent's beliefs change in such a way that they implement a gradient-descent scheme on VFE [C. L. Buckley et al. 2017]:

$$\tilde{\mu}^{\tau+1} = \dot{\tilde{\mu}}^{\tau} - \eta \nabla_{\tilde{\mu}} F \tag{4.37}$$

where τ represents each time point when the agent's states are updated and η stands for the learning rate. The $\nabla_{\tilde{\mu}}$ operator denotes the gradient of the VFE with respect to the agent's states, that is:

$$\nabla_{\tilde{\mu}} F = \frac{\partial F}{\partial \tilde{\mu}^{\tau}} = \left[\frac{\partial F}{\partial \mu^{\tau}}, \frac{\partial F}{\partial \dot{\mu}^{\tau}}, \frac{\partial F}{\partial \ddot{\mu}^{\tau}} \right]^{\top} \tag{4.38}$$

Note that Eq. (4.37) is slightly different from the traditional gradient-descent expression. This is due to the minimisation of the generalised state components that represent a trajectory rather than a static state. As a result, this minimisation is achieved when the temporal dynamics of the gradient descent match the hidden states estimates, so that $\tilde{\mu}^{(\tau+1)} = \dot{\tilde{\mu}}$ rather than $\tilde{\mu}^{(\tau+1)} = 0$ [Baltieri and C. Buckley 2019].

Applying Eq. (4.37) to each agent's state (estimates) yields:

$$\begin{aligned}
\mu^{\tau+1} &= \dot{\mu}^{\tau} - \eta\Big(-\pi_z \big(y - \mu^{\tau}\big) + \pi_\omega \alpha \big(\dot{\mu}^{\tau} + \alpha\big(\mu^{\tau} - v\big)\big)\Big) \\
\dot{\mu}^{\tau+1} &= \ddot{\mu}^{\tau} - \eta\Big(-\pi_{\dot{z}} \big(\dot{y} - \dot{\mu}^{\tau}\big) + \pi_{\dot{\omega}} \alpha \big(\ddot{\mu}^{\tau} + \alpha\big(\dot{\mu}^{\tau} - \dot{v}\big)\big) + \pi_\omega \big(\dot{\mu}^{\tau} + \alpha\big(\mu^{\tau} - v\big)\big)\Big) \\
\ddot{\mu}^{\tau+1} &= \dddot{\mu}^{\tau} - \eta\Big(-\pi_{\ddot{z}} (\ddot{y} - \ddot{\mu}^{\tau}) + \pi_{\ddot{\omega}} \alpha \big(\dddot{\mu}^{\tau} + \alpha\big(\ddot{\mu}^{\tau} - \ddot{v}\big)\big) + \pi_{\dot{\omega}} \big(\ddot{\mu}^{\tau} + \alpha\big(\dot{\mu}^{\tau} - \dot{v}\big)\big)\Big)
\end{aligned} \tag{4.39}$$

Computing these equations with every new observation updates the agent's beliefs about the hidden states and, as a consequence, VFE is minimised. Nevertheless, as mentioned earlier, the agent can also minimise VFE by acting on the environment, thus changing sensory input. This is known as active inference and is dealt in the following section.

4.5 Active Inference

Perceptual inference, described in previous section, minimises VFE by changing the agent's states to better predict sensory data (i.e., perception). Active inference, however, modifies observations through action so as to fit better sensory predictions. In this section, it is proposed a gradient-descent scheme analogous to the aforementioned, but for action.

4.5.1 Action

To minimise VFE via active inference, the agent needs an inverse model of how observations change with action, since VFE does not explicitly depend on action, see

Eq. (4.36). For instance, for a single agent's state, μ, the inverse model could be written as $y = y(u)$, where u represents the action and y is a single sensory channel. Thus, given an inverse model, one can evaluate how the VFE changes with respect to action by means of the chain rule:

$$\frac{\partial F}{\partial u} \equiv \frac{\partial F}{\partial \tilde{y}} \frac{\partial \tilde{y}}{\partial u} \tag{4.40}$$

The assumption that agents have innate knowledge about the mapping between actions and observations seems problematic. Nonetheless, under the free energy principle, it is considered that the execution of motor control in biological agents (e.g., brain) depends only on proprioceptors[4] (i.e., internal sensors) which can be satisfied by reflex arcs [C. L. Buckley et al. 2017].

Thereby, applying the gradient-descent technique to calculate the action which minimise VFE yields:

$$u^{\tau+1} = -\kappa \sum \frac{\partial \tilde{y}}{\partial u^{\tau}} \nabla_{\tilde{y}} F \tag{4.41}$$

where κ represents the learning rate for action. In this case, $\nabla_{\tilde{y}}$ computes the gradient of the VFE with respect to the observations, $\tilde{y}$:

$$\nabla_{\tilde{y}} F = \frac{\partial F}{\partial \tilde{y}} = \left[\frac{\partial F}{\partial y}, \frac{\partial F}{\partial \dot{y}}, \frac{\partial F}{\partial \ddot{y}} \right]^{\top} \tag{4.42}$$

As a result, developing Eq. (4.41) leads to the following control law:

$$u^{\tau+1} = -\kappa \Big(\pi_z (y - \mu) \frac{\partial y}{\partial u^{\tau}} + \pi_{\dot{z}} (\dot{y} - \dot{\mu}) \frac{\partial \dot{y}}{\partial u^{\tau}} + \pi_{\ddot{z}} (\ddot{y} - \ddot{\mu}) \frac{\partial \ddot{y}}{\partial u^{\tau}} \Big) \tag{4.43}$$

Nevertheless, partial derivatives $\partial \tilde{y} / \partial u$ are still unspecified. To achieve PID-like control, it is assumed that the agent adopts the simplest (i.e., linear) relationship between the action and its effect on sensory input across all orders [Baltieri and C. Buckley 2019]:

$$\frac{\partial y}{\partial u} = \frac{\partial \dot{y}}{\partial u} = \frac{\partial \ddot{y}}{\partial u} = 1 \tag{4.44}$$

Hence, Eq. (4.43) can be rewritten as:

$$u^{\tau+1} = \kappa \Big(\pi_z (\mu - y) + \pi_{\dot{z}} (\dot{\mu} - \dot{y}) + \pi_{\ddot{z}} (\ddot{\mu} - \ddot{y}) \Big) \tag{4.45}$$

which is consistent with the expression of a traditional PID controller. Here, $\pi_z, \pi_{\dot{z}}, \pi_{\ddot{z}}$ would correspond to the integral, proportional and derivative gains, respectively.

Eq. (4.44) provides a linear relation between sensed inputs and action. Specifically, positive actions increase the observations linearly, whereas negative actions decrease them. However, an action may not be able to change the observation y and its higher orders in the same manner (e.g. action cannot change position, velocity and acceleration identically). [Baltieri and C. Buckley 2019] argues that these derivatives only encode

[4] Any receptor (as in the gut, blood vessels, muscles, etc) that supplies information about the state of the body.

sensorimotor[5] dependencies that allow for sub-optimal control, in the same way as classical PID controllers are only approximate solutions.

Next section deals with the optimal tuning of the PID gains (i.e., measurement precisions) and provides an algorithm to implement the PID controller as a process of active inference.

4.6 Controller Implementation

Thus far, it has been explained how a linear generative model furnishes a model of the environment to the agent which ultimately, along with the recognition density, permits attaining the VFE expression. Minimising VFE through perception and action approximates the R-density to the true posterior probability density and, as a result, surprisal decreases. While perception implies modifying the agent's beliefs about the world, action interacts with the environment so as to seek preferred observations. In this section, the algorithm to implement perceptual and active inference is detailed. Nonetheless, action and perception include some hyperparameters (i.e., measurement and process precisions) that need to be specified first. These precisions can be adjusted manually by trial and error or can be inferred by employing a gradient-descent approach. The following subsection describes how the hyperparameters, $\gamma = \{\pi_{\tilde{z}}, \pi_{\tilde{\omega}}\}$, can be tuned optimally. This would correspond to deducing the process and measurement noise of the system.

4.6.1 Optimal Tuning of Hyperparameters

In statistics, a hyperparameter is a parameter from a prior distribution; it captures the prior belief before data is observed. In this report, it is considered to be the set containing all the measurement and process precisions. In this subsection, hyperparameters are estimated by means of a gradient-descent approach. In order to infer these hyperparameters, some assumptions must be made. First, it is supposed that hyperparameters, γ, change in a much slower time-scale with respect to agent's states, μ. Second, changes in γ with respect to a small time interval have a much smaller effect on VFE than μ, that is:

$$\frac{\partial F}{\partial \gamma}\frac{d\gamma}{dt} \ll \frac{\partial F}{\partial \mu}\frac{d\mu}{dt} \tag{4.46}$$

The latter entails that, from the gradient-descent perspective, what is relevant for γ is not the VFE, but the accumulation (i.e., the integration over time) [C. L. Buckley et al. 2017]:

$$S = \int F(\tilde{y}, \tilde{\mu}, \tilde{v}\,;\,\theta, \gamma)dt \tag{4.47}$$

Hence, applying the gradient-descent method leads to:

$$\gamma^{\tau+1} = -\rho\nabla_\gamma S \tag{4.48}$$

[5] Of or relating to both the sensory and motor functions of an organism or to the nerves controlling them.

where ρ is the learning rate for the hyperparameters update and $\nabla_\gamma S$ is the gradient of S with respect to γ. However, the implementation of Eq. (4.48) requires the explicit integration of VFE over time. This can be solved by applying temporal differentiation on both sides:

$$\dot{\gamma}^{\tau+1} = -\rho \nabla_\gamma F \tag{4.49}$$

which leads to a second-order online update scheme. For practical purposes, Eq. (4.49) is reduced to a simpler set of first-order differential equations (with $\rho = 1$) [Baltieri and C. Buckley 2019]:

$$\begin{aligned} \gamma^{\tau+1} &= \dot{\gamma}^{\tau} \\ \dot{\gamma}^{\tau+1} &= -\nabla_\gamma F - \lambda \dot{\gamma}^{\tau} \end{aligned} \tag{4.50}$$

where $\dot{\gamma}$ is a prior on the motion of hyperparameters which includes a damping term, λ. Since the derivative of the free energy with respect to hyperparameters is strictly positive (i.e. $\partial F/\partial\gamma > 0$), it does not provide a steady-state solution for the gradient-descent. This damping term stabilises the solution by reducing oscillations around the real equilibrium of the system [Baltieri and C. Buckley 2019].

Although Eq. (4.50) can be applied to infer process noise precisions, $\pi_{\tilde{\omega}}$, only updates on sensory precisions are regarded in this report. Henceforth, $\gamma_{\tilde{z}}$, is utilised to denote the measurement precisions hyperparameters. Since precisions need to be positive (i.e., variances can only be positive), the following constraint is included:

$$\pi_{\tilde{z}} = \exp(\gamma_{\tilde{z}}) \tag{4.51}$$

This exponential mapping makes measurement precisions strictly positive. Substituting Eq. (4.51) into the VFE expression, Eq. (4.36), and applying the gradient-descent algorithm described in Eq. (4.50) for each precision order yields:

$$\begin{aligned} \gamma_z^{\tau+1} &= \dot{\gamma}_z^{\tau} \\ \dot{\gamma}_z^{\tau+1} &= -\frac{\partial F}{\partial \gamma_z} - \lambda \dot{\gamma}_z^{\tau} = -\frac{1}{2}\Big(\exp(\gamma_z)\,(y-\mu)^2 - 1\Big) - \lambda \dot{\gamma}_z^{\tau} \\ \gamma_{\dot{z}}^{\tau+1} &= \dot{\gamma}_{\dot{z}}^{\tau} \\ \dot{\gamma}_{\dot{z}}^{\tau+1} &= -\frac{\partial F}{\partial \gamma_{\dot{z}}} - \lambda \dot{\gamma}_{\dot{z}}^{\tau} = -\frac{1}{2}\Big(\exp(\gamma_{\dot{z}})\,(\dot{y}-\dot{\mu})^2 - 1\Big) - \lambda \dot{\gamma}_{\dot{z}}^{\tau} \\ \gamma_{\ddot{z}}^{\tau+1} &= \dot{\gamma}_{\ddot{z}}^{\tau} \\ \dot{\gamma}_{\ddot{z}}^{\tau+1} &= -\frac{\partial F}{\partial \gamma_{\ddot{z}}} - \lambda \dot{\gamma}_{\ddot{z}}^{\tau} = -\frac{1}{2}\Big(\exp(\gamma_{\ddot{z}})\,(\ddot{y}-\ddot{\mu})^2 - 1\Big) - \lambda \dot{\gamma}_{\ddot{z}}^{\tau} \end{aligned} \tag{4.52}$$

These hyperparameters are updated online until convergence. Then, the inferred sensory precisions can be obtained by applying the mapping shown in Eq. (4.51):

$$\pi_{\tilde{z}} = \Big[\exp(\gamma_z) \quad \exp(\gamma_{\dot{z}}) \quad \exp(\gamma_{\ddot{z}})\Big]^\top \tag{4.53}$$

Once it is established how the PID gains can be tuned optimally, one has the necessary tools to implement the PID controller under the free energy principle.

4.6.2 Algorithm

Until now, it has been proposed a gradient-descent scheme to update the agent's states (i.e., beliefs), action and sensory precisions. However, it has not been explained how these three relate to each other. This subsection puts these concepts together so as to construct the algorithm employed for simulation.

Fig. 4.2 provides an schematic of how perception and action work together to minimise VFE and, as a consequence, surprisal. Firstly, the agent receives some sensory input, $\tilde{y}$, generated by some (unknown) environmental hidden states, x. These measurements are utilised to update agent's beliefs about the world (i.e., perception, $\tilde{\mu}$). Then, observations and perception are employed to infer sensory precisions and upgrade the control law (i.e., action, u). Action is applied to the environment which will change the observations in the following time instant. This process is iterated until the agent's states and the observations follow the desired set point introduced through the exogenous inputs, $\tilde{v}$ [6].

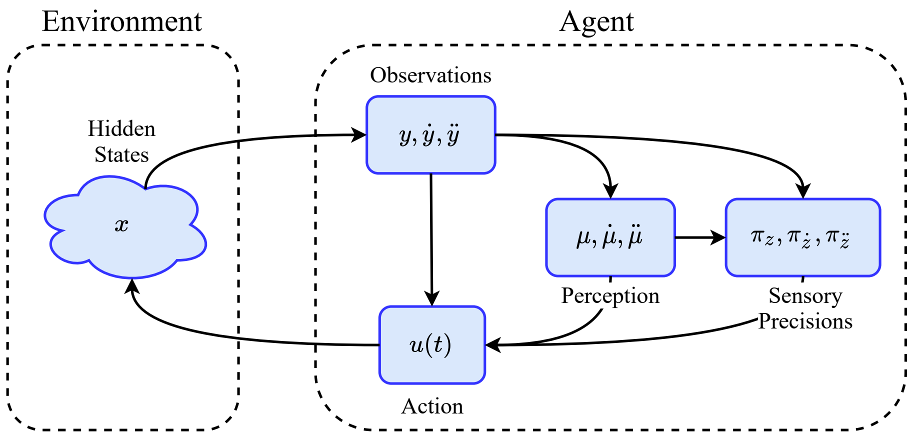


Figure 4.2: Connection between sensory input, $\tilde{y}$, perceptual inference, $\tilde{\mu}$, active inference, u, and tuning of hyperparameters, $\pi_{\tilde{z}}$.

In Section 4.3, the agent's states dynamics in the linear generative model included a parameter denoted by α. One could also have applied a gradient-descent approach to infer this parameter, see [C. L. Buckley et al. 2017]. Nevertheless, in this report, the decay parameter, α, is treated as a large constant (theoretically, $\alpha \to \infty$) [Baltieri and C. Buckley 2019].

The controller can be implemented as depicted in Algorithm 1. Firstly, one needs to initialise the agent's states, the parameters and hyperparameters. Furthermore, the exogenous inputs, $\tilde{v}$, which here correspond to references, must be set. Then, the agent receives sensory input at every sample. These observations are utilised to update agent's states (see line 4), as shown in Eq. (4.39). Perception latest update along with sensory input are employed to compute action which will be applied to the environment (lines 11 and 12), as established in Eq. (4.45). When the agent's states are settled and follow,

[6] Although exogenous inputs can include disturbances and noisy terms, in this report only references are regarded.

approximately, the desired set points, hyperparameters can start to be updated (lines 5 and 6). These updates are then mapped to sensory precisions (line 7) through Eq. (4.51) and introduced in action and perception.

Algorithm 1: Active Inference PID Controller

1: **initialization**: $\tilde{\mu}$, $\tilde{v}$, θ, γ
2: **repeat**
3: Receive observations, $\tilde{y}$
4: Perception: update agent's states, $\tilde{\mu}$
5: **if** $\tilde{\mu} \approx \tilde{v}$ **then**
6: Update sensory hyperparameters, $\gamma_{\tilde{z}}$
7: Modify agent's sensory precisions, $\pi_{\tilde{z}} = \exp(\gamma_{\tilde{z}})$
8: **end if**
9: Action: update agent's control law, u
10: Apply action to the environment
11: **until** task done

To construct the PID regulator, however, one needs to have access up to second-order derivatives of the sensory input (i.e., $\dot{y}$, $\ddot{y}$). In most cases, these higher-order derivatives are not measurable as they require adding extra sensors to the physical setup. In the evaporator case, the derivatives must be estimated since the only measured variable is the evaporator outlet temperature. Moreover, the evaporator model developed in [Kasper Vinther 2014] includes input and output quantisations and therefore discrete differentiation will lead to noisy estimates. In this report, the Kalman filter algorithm is considered instead. Fig. 4.3 depicts how the Kalman filter is utilised in order to apply the PID control.

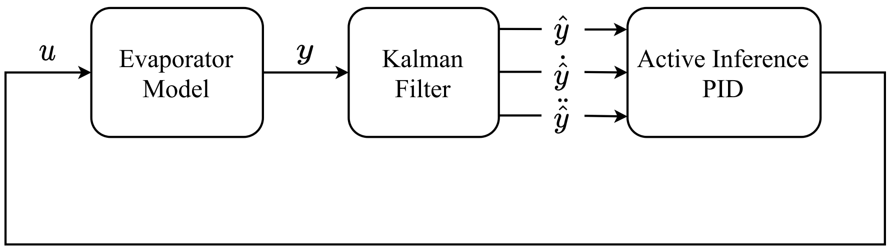


Figure 4.3: Block diagram representing the role of the Kalman filter in the active inference framework.

Kalman filter outputs, $\hat{y}$, $\dot{\hat{y}}$ and $\ddot{\hat{y}}$ denote the measurement estimate and its two higher-order derivatives, respectively. Evaporator outlet temperature is also estimated so as to remove the errors due to quantisation. Appendix A.2 introduces the theory related to the Kalman filter and describes how this algorithm is implemented.

Next chapter describes how simulations are conducted and presents the attained results.

5 | Simulation

The current chapter deals with the simulation of the PID controller applied to the evaporator model developed in [Kasper Vinther 2014]. This model provides the evaporator outlet temperature given an opening degree of the expansion valve. The chapter is arranged as follows. First, the simulation is run considering fixed PID gains. That is, measurement precisions are determined by trial and error and without updating the hyperparameters. Second, the PID gains are tuned optimally following the procedure shown in Subsection 4.6.1. The simulations are carried out in the MATLAB and Simulink environment. Details regarding the implementation and the generated code can be found in Appendix A.3.

5.1 Fixed PID Gains

This subsection analyses the attained results for fixed measurement precisions, $\pi_{\tilde{z}}$ (i.e., the PID gains). As in the classical control approach, these remain fixed over the simulation and are determined by the designer. However, some parameters need to be specified first. Table 5.1 shows the chosen values for the simulation.

Description	Value
Sampling time	$T_s = 1$ s
Opening degree	$OD = 51$ %
Outlet temperature reference	$T_{ref} = 11$ ºC
Process precisions	$\pi_{\tilde{\omega}} = 10^{-9}[1\ \ 1\ \ 1]$
Measurement noise standard deviation	$\sigma_{\tilde{z}} = [0.1\ \ 0.08\ \ 0.005]$
Linear generative model parameter	$\alpha = 10^5$
Perception learning rate	$\eta = 0.01$
Action learning rate	$\kappa = 0.01$
Hyperparameters learning rate	$\rho = 0.1$
Hyperparameters damping term	$\lambda = 0.9$

Table 5.1: Simulation parameters.

The chosen opening degree of the EEV is close to the maximum slope point so that the outlet temperature can be clearly determined. It is assumed that no process noise is added to the system. Measurement noise, with standard deviation $\sigma_{\tilde{z}}$ is added to the KF estimates.

The system is simulated with the following measurement precisions:

$$\pi_{\tilde{z}} = \begin{bmatrix} \exp(-2) & \exp(-3) & \exp(-4) \end{bmatrix} \tag{5.1}$$

where each element of the vector corresponds to the integral, proportional and derivative gains, respectively. Fig. 5.1 depicts the simulation results for the measured and the perceived temperature and the opening degree of the expansion valve. Higher-order terms are not plotted as they oscillate around zero.

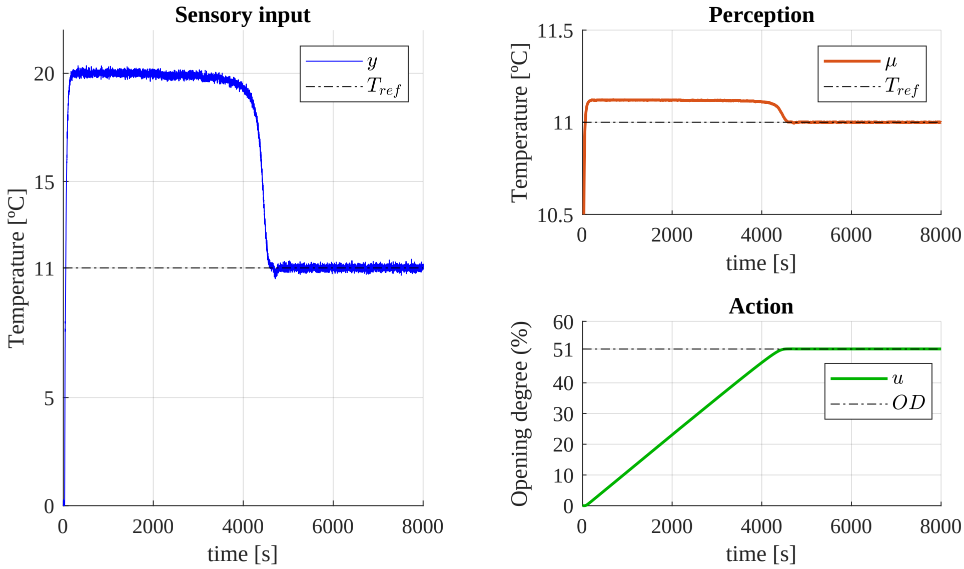


Figure 5.1: Left: measured evaporator outlet temperature, $T_{e,o}$. Top right: temperature perception, μ. Bottom right: opening degree of the EEV (action).

As can be seen, the outlet temperature flattens for low opening degree values of the EEV and starts dropping after 40%, approximately. Moreover, agent's belief, μ, is able to follow the desired reference. Nevertheless, there is a small error that is removed through action. This is the essence of active inference: by acting on the environment and bringing sensory input to the desired value (i.e., $T_{ref} = 11\,°\mathrm{C}$), the agent state becomes equal to its desired state. The obtained control signal, however, does not seem suitable as requires more than 4000 seconds to drive the outlet temperature to the established set point.

The employed learning rates are determined by trial and error, considering that higher learning rate values provide more relevance to the updates. That is, larger higher rates entail a faster response but at the cost of potentially generating oscillations.

The next step is to determine the effect of the PID gains on the system. For this purpose, three different measurement precisions are applied where each order (i.e., π_z, $\pi_{\dot{z}}$, $\pi_{\ddot{z}}$) is changed separately. The utilised gains are as follows:

$$
\begin{aligned}
\pi_{\tilde{z}}^{(1)} &= \begin{bmatrix} \exp(-1) & \exp(-3) & \exp(-4) \end{bmatrix} \\
\pi_{\tilde{z}}^{(2)} &= \begin{bmatrix} \exp(-2) & \exp(0) & \exp(-4) \end{bmatrix} \\
\pi_{\tilde{z}}^{(3)} &= \begin{bmatrix} \exp(-2) & \exp(-3) & \exp(0) \end{bmatrix}
\end{aligned} \tag{5.2}
$$

Fig. 5.2 shows the attained results, where superscripts denote the gains employed in each case, see Eq. (5.2).

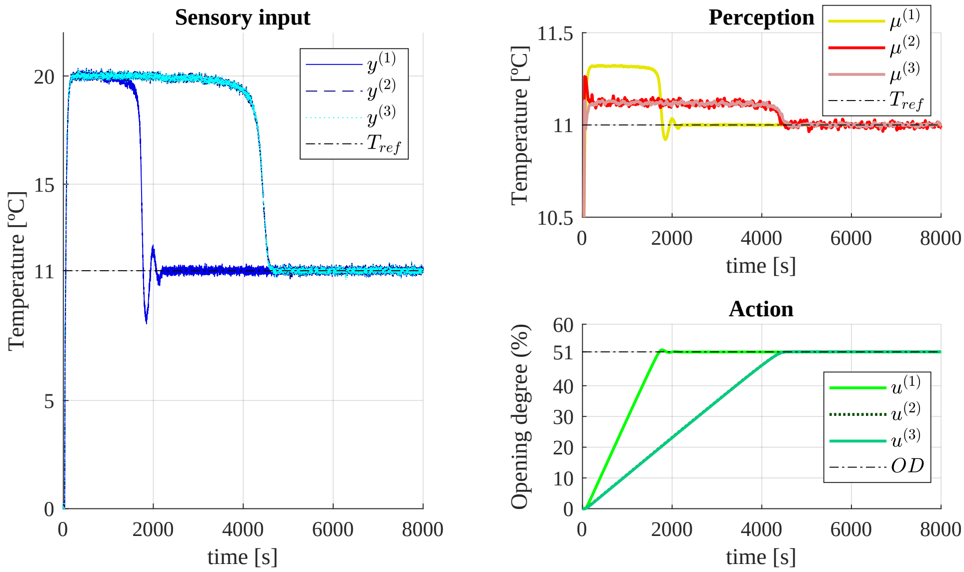


Figure 5.2: Comparison between different measurement precisions.

Increasing the measurement precision associated with temperature (i.e., π_z) leads to a faster response, see $y^{(1)}$ in the figure. The response presents a small overshooting due to the appearance of small fluctuations in the control signal around the operating point ($u^{(1)}$ in Fig. 5.2). Since the chosen reference, T_{ref}, is close to the maximum slope point, small variations in the opening degree of the EEV alter the temperature considerably (see Fig. 3.3). In this situation, agent's perception, $\mu^{(1)}$, tries to follow the measurements.

When the measurement precisions related to the temperature derivatives increase, no significant differences can be perceived in the control signal or outlet temperature response. Nonetheless, agent's state μ exhibits more overshooting and noise when increasing the first-order derivative gain, $\pi_{\dot{z}}$. This is related to augmenting the influence of the first-order derivative error in perception (i.e., $\dot{y} - \dot{\mu}$), thus implying a more aggressive response. Furthermore, more noise is coupled since the first-order derivative of the temperature contains a higher standard deviation, see $\mu^{(2)}$ in Fig. 5.2. The aforementioned derivative error also appears in the control law expression but has barely any effect as the action's learning rate is small enough to compensate for it.

The simulations regarded so far, assumed larger measurement than process precisions. This entails that the agent has a higher confidence in sensory input than in its internal model (i.e., linear generative model). Now, it is analysed how changing process precisions affect the system response. Fig. 5.3 illustrates the outlet temperature and its corresponding agent's state for three distinct values of the process precision π_ω.

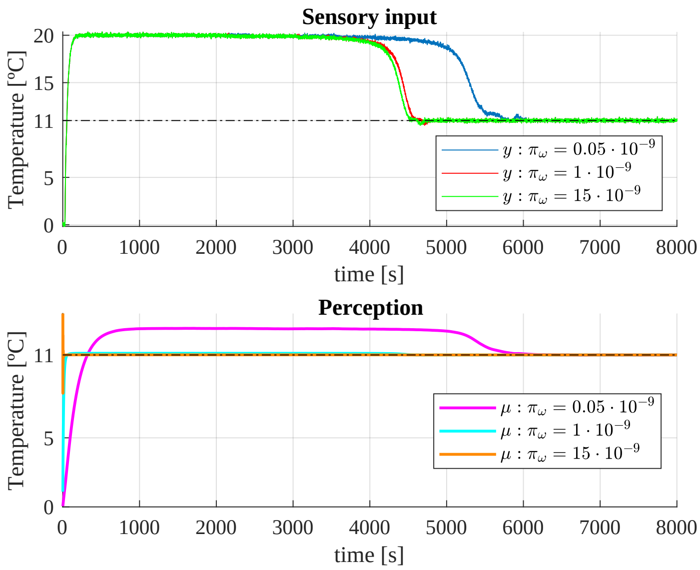


Figure 5.3: Comparison between different process precisions.

As can be observed, larger process precision entails an expected low uncertainty on the dynamics leading to a faster response of the outlet temperature. What is more, allows the agent's state to settle faster. Smaller process precisions, however, account for higher variance/uncertainty and thus changes in the agent's states dynamics are to be expected, making the transitions to reference values slower [Baltieri and C. Buckley 2019].

5.2 Optimal Tuning of PID Gains

So far, it has been shown the effect that different parameters have on the system's response. In this scenario, the PID gains remained fixed throughout the entire simulation. The current section displays how these PID gains (i.e., measurement precisions, $\pi_{\tilde{z}}$) can be tuned online in an automated and optimal manner. Hence, the reasoning exposed in [Baltieri and C. Buckley 2019] for the optimal tuning of PID gains is followed. Thus, PID gains are maintained fixed until the agent settles around the desired state as described in Algorithm 1. Then, sensory precisions are updated online until convergence.

The simulation parameters chosen in this case are presented in Table 5.2. The selected initial measurement precisions are:

$$\pi_{\tilde{z}} = \begin{bmatrix} \exp(-1) & \exp(-3) & \exp(-4) \end{bmatrix} \tag{5.3}$$

The hyperparameters start to be updated after $4 \cdot 10^3$ samples and stop at $12 \cdot 10^3$.

Description	Value
Sampling time	$T_s = 1$ s
Opening degree	$OD = 51$ %
Outlet temperature reference	$T_{ref} = 11$ ºC
Process precisions	$\pi_{\tilde{\omega}} = 10^{-9}[0.05\ \ 1\ \ 1]$
Measurement noise standard deviation	$\sigma_{\tilde{z}} = [0.1\ \ 0.08\ \ 0.005]$
Linear generative model parameter	$\alpha = 10^5$
Perception learning rate	$\eta = 0.1$
Action learning rate	$\kappa = 0.01$
Hyperparameters learning rate	$\rho = 0.005$
Hyperparameters damping term	$\lambda = 0.9$

Table 5.2: Simulation parameters.

Simulation results are illustrated in Fig. 5.4.

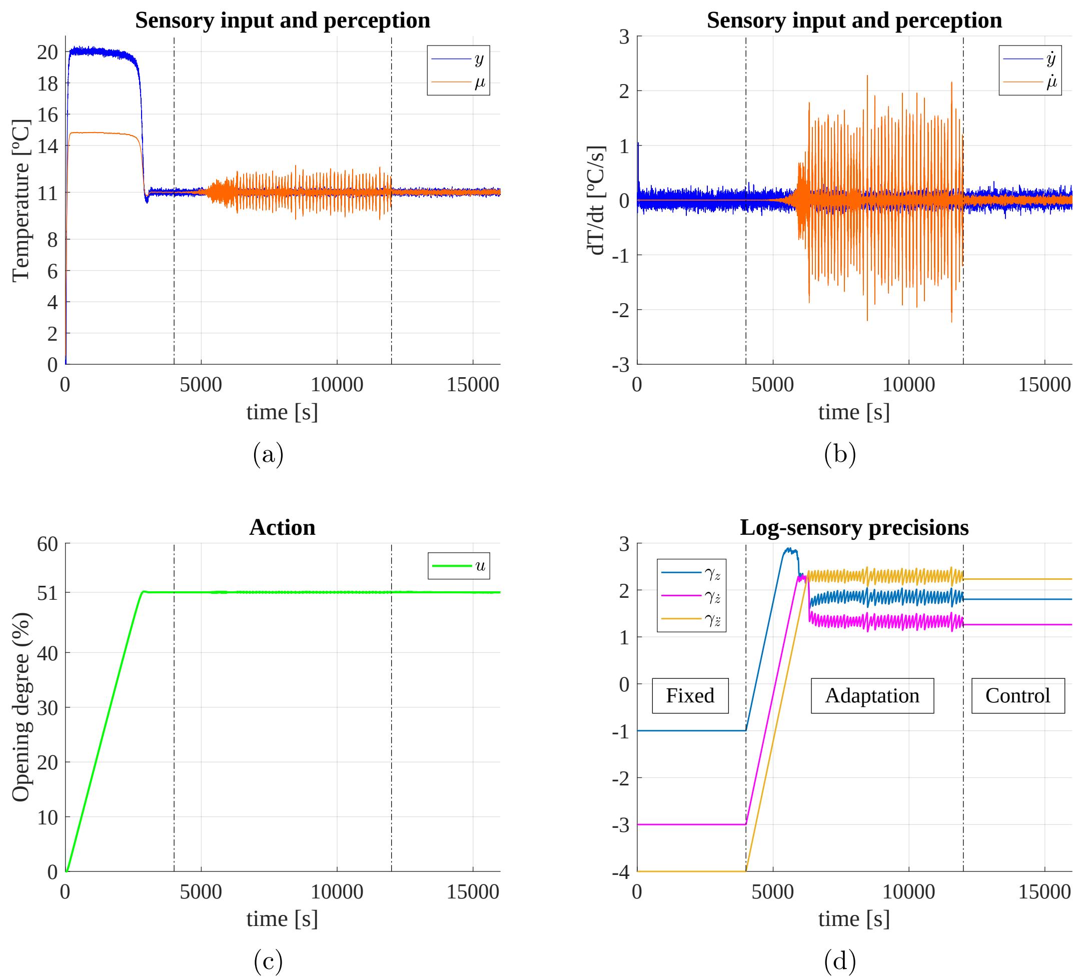


Figure 5.4: Hyperparameters optimisation. This simulation depicts the control of the evaporator outlet temperature during the optimisation of log-sensory precisions, $\gamma_{\tilde{z}}$. (a) Measured temperature and its corresponding agent's state. (b) Measured temperature derivative and its corresponding agent's state. (c) Action. (d) Log-sensory precisions (i.e., hyperparameters) optimisation.

Fig. 5.4a and 5.4b show how agent's states (i.e., μ and $\dot{\mu}$) are modified and become noisy signals when updating the hyperparameters online. When hyperparameters are settled, agent's states encode the noise that have been inferred, see Fig. 5.4d. That is, now perception includes the precision that the controller thinks is present in the sensory input. The attained hyperparameters in closed-loop are:

$$\gamma_{\tilde{z}} = \begin{bmatrix} 1.80 & 1.26 & 2.23 \end{bmatrix} \tag{5.4}$$

which corresponds to the log-PID gains, see Eq. (4.51). Closed-loop here refers to the fact of updating the hyperparameters and modifying the controller gains while running the simulation. Converting the added measurement noise, $\sigma_{\tilde{z}}$, into log-sensory precisions so as to compare with Eq. (5.4), one gets:

$$\ln\left(1/\sigma_{\tilde{z}}^2\right) = \begin{bmatrix} 4.60 & 5.05 & 10.59 \end{bmatrix} \tag{5.5}$$

Agent's perception of temperature is relatively accurate as can be seen in the last simulated samples of Fig. 5.4a. Higher order derivatives, however, are not that precise (see Fig. 5.4b). In any case, the goal of this simulation is to find the optimal gains that furnishes the best system's response in closed-loop. If one wants to deduce the noise encoded in sensory input, the hyperparameters update should be inferred in open-loop. That is to say, hyperparameters are updated but PID gains remain unchanged. This is depicted in Fig. 5.5, where measurement precisions are updated in open-loop employing the gains in Eq. (5.3) and the parameters shown in Table 5.2.

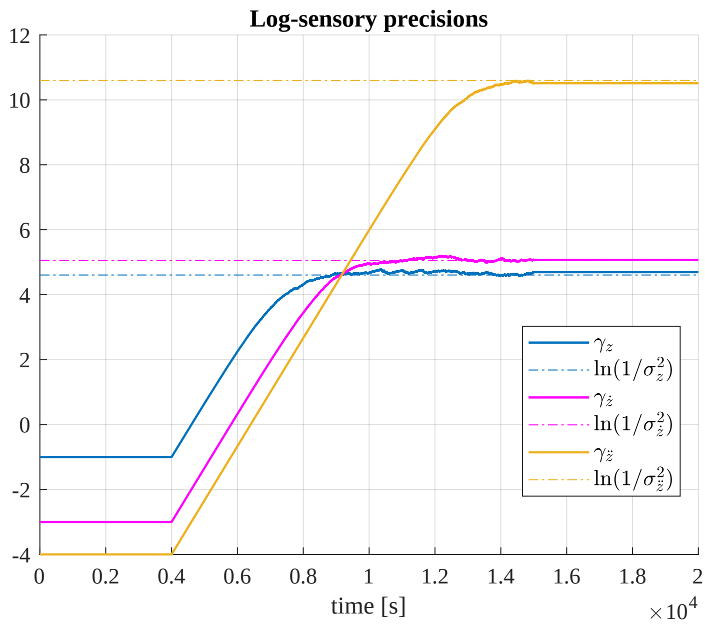


Figure 5.5: Hyperparameters update without modifying the PID gains.

The horizontal dashed lines belong to noise log-precisions[1] added to each measurement order. As can be noticed, the gradient-descent scheme proposed in 4.6.1 can also be

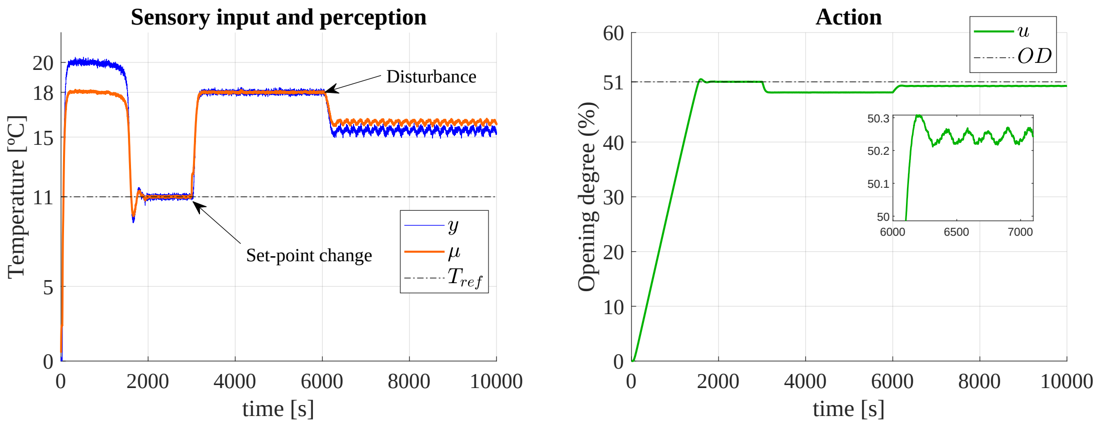
[1]Each measurement noise variance is transformed to its corresponding precision and the natural logarithm is applied next.

utilised to infer the white noise included in the system. Nevertheless, this open-loop inference entails high PID gains that lead to unstable responses.

Now, the obtained optimal gains in closed-loop are used to evaluate robustness and performance. That is, the behaviour when disturbances and set-point changes are added to the system. Thus, converting the hyperparameters in Eq. (5.4) into PID gains yields:

$$\pi_{\tilde{z}} = \begin{bmatrix} \exp(1.80) & \exp(1.26) & \exp(2.23) \end{bmatrix} \tag{5.6}$$

A new reference of 18 ºC and an input disturbance of 0.01% are applied at $3 \cdot 10^3$ at $6 \cdot 10^3$ seconds, respectively. Fig. 5.6 show the simulation results for this scenario.



(a) (b)

Figure 5.6: System's response to set-point change and disturbance. (a) Measured temperature, y, and agent's state, μ. (b) Action.

On the one hand, the system reacts properly to the reference variation: the control signal diminishes to increase the outlet temperature while the agent's state changes fast enough to follow the sensory input. On the other hand, however, the control signal cannot handle small disturbances. As a result, y and μ are not capable of returning to the desired set-point (i.e., 18 ºC). Furthermore, small oscillations can be observed in Fig. 5.6a. This effect is caused by the waving produced in the control signal, see Fig. 5.6b. As aforementioned, small changes in the control signal provoke big changes in the outlet temperature since the system is operating close to the maximum slope point. Action oscillations arise because the control law update tries to compensate the disturbance. Nonetheless, this update does not have enough influence to modify the control law significantly. This could be due to a small action learning rate, κ, or to a small difference between agent's state μ and sensory input y, see Eq. (4.45). This could be solved by rather increasing the action learning rate or the measurement precisions, so as to amplify the difference amongst sensory input and perception in the control signal and oblige the system to track the set-point. Nevertheless, these modifications could not be simulated as the system is very sensitive to subtle changes in parameters which results in instability.

6 Assessment

This chapter pretends to evaluate the entire project. Section 6.1 examines some assumptions regarded in the controller design as well as reflects on simulation results and the unsuccessful tests performed in the laboratory. In addition, some conclusions regarding the PID controller performance and robustness are revealed in Section 6.2. Finally, Section 6.3 contemplates what could be the next steps to take in the project.

6.1 Discussion

The current section delves into the assumptions regarded for the controller design and analyses the attained simulation results. Furthermore, it is discussed what could have gone wrong in the laboratory when tests were carried out.

6.1.1 Controller Design

In Subsection 4.2.1, the R-density means to approximate the true posterior probability and furnishes a first approach to the VFE, see Eq. (4.21). Nevertheless, to reach this expression it is considered that the energy, $E(y, x)$, is a smooth function of x and that the R-density, assumed to be Gaussian, is sharply peaked at its mean value. The latter supposition entails that the R-density's variance is small which may appear troublesome since it suggests that no uncertainty is present in the environmental variables (i.e., hidden causes of sensory input). However, this does not imply that process uncertainty is neglected as it is included in the G-density via expected process precisions, $\pi_{\tilde{\omega}}$, see Eq. (4.30). Intuitively, this means that the agent encodes uncertainties about its model of how hidden states relate to each other and to sensory signals [C. L. Buckley et al. 2017]. The principal benefit of adopting the Gaussian form assumption, known as *Laplace approximation*, is that it considerably simplifies the VFE expression.

Subsection 4.5.1 establishes the control law by regarding a linear relationship between sensory input and action, see Eq. (4.44). Despite this inconsistency, since action may not change all measurement orders in the same manner, the simple inverse mapping from sensory input to action can be explained by biological evolution. For instance, the mapping from proprioception to action could be part of classical motor reflex arcs [Baltieri and C. Buckley 2019]. What is more, this mapping comprises an approximation in the same way as classical PID regulators provide inexact, but effective, control solutions.

6.1.2 Simulation Results

To test the correct implementation of the controller, simulation with fixed parameters and gains has been considered first. The attained results show that the control signal (i.e., the opening degree of the EEV) is able to drive the sensory input and the agent's states to the desired set-point, see Fig. 5.1. Furthermore, agent's states are able to follow the measurements through perception. Hence, a simple linear generative model can be enough to describe how observations are generated. Then, changes in measurement and process precisions are made to test the system's response. On the one hand, increasing measurement precisions imply more confidence in sensory input (i.e., less variance). As a result, PID gains rise (i.e., $\pi_{\tilde{z}}$). Enlarging temperature precision, π_z, entails a faster response, whereas an increment in higher orders of sensory input lead to more oscillations in agent's states. On the other hand, higher process precisions provoke agent's states to settle faster as shown in Fig. 5.3. In conclusion, larger precisions result in faster transitions as changes in the signal are less expected.

The second simulation part aims to infer the optimal PID gains. The hyperparameters are updated online once the agent's states are settled. That is, measurement precisions are changed every time hyperparameters are upgraded, thus modifying perception and action as depicted in Fig. 5.4. This updating scheme, proposed in [Baltieri and C. Buckley 2019] for the cruise control in a car, lead to instability in most of the simulations regarded in this thesis as subtle changes in the control law result in big changes in the outlet temperature. When an appropriate set of parameters is found, the system is able to react to changes in the set-point but cannot handle small disturbances, as illustrated in Fig. 5.6. Furthermore, it is shown that changing the updating scheme to open-loop allows inferring measurement noise properties, see Fig. 5.5. In other words, updating the hyperparameters without modifying the sensory precisions permit deducing the variance present in the sensory input. These hyperparameters, however, tend to large values and cannot be utilised for control as the response becomes unstable.

6.1.3 Tests Results

As mentioned in the beginning of this report, successful results in the physical setup could not be accomplished due to certain technical issues when carrying out the tests. Specifically, it was found out that in most cases the evaporator inlet temperature was higher than the outlet. When observing the compressor input and output no pressure rise was noticed. Moreover, in some other cases the solenoid valve[1] closed suddenly as the superheating temperature was below the 5 ºC threshold, thus impeding the flow of refrigerant through the evaporator.

Fig. 6.1 shows one of the performed tests, where a step in the opening degree of the EEV is applied ($0-100\%$). As can be seen, after applying the step, no changes in the temperature or in the pressure can be noticed. In addition, outlet evaporator temperature, $T_{e,o}$ is lower than the evaporating temperature, T_e. In the same manner, a pressure drop between the

[1] The solenoid valve is placed after the condenser outlet and before the EEV.

evaporating and the condensing pressure (i.e., P_e and P_c, respectively) indicates that the compressor may not be working properly, see Fig. 6.1d. Fig. 6.1b displays that refrigerant is flowing along the evaporator since the solenoid valve is completely open.

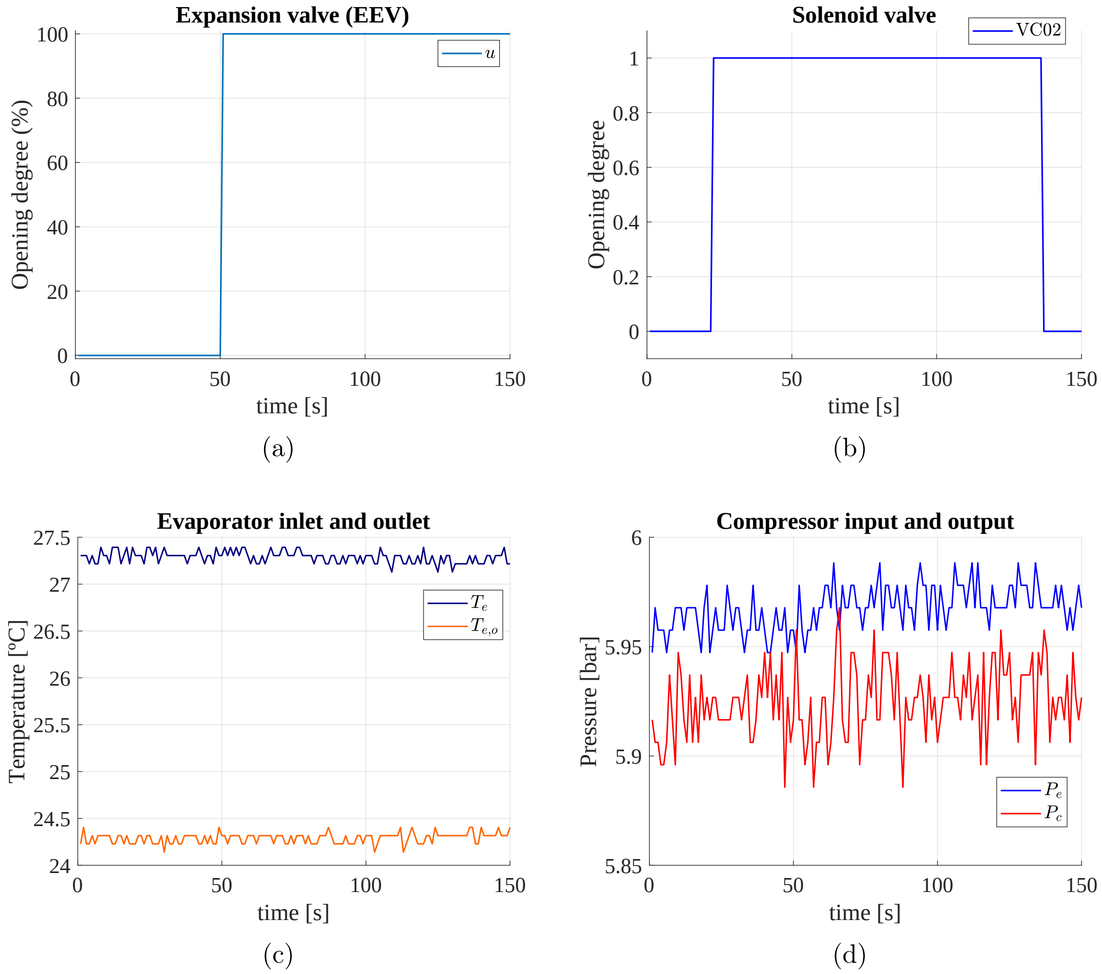


Figure 6.1: Test results after applying a step in the opening degree of the EEV at $t = 50$ s. (a) Opening degree of the EEV. (b) Solenoid valve. (c) Evaporating temperature (inlet), T_e and evaporator outlet temperature, $T_{e,o}$. (d) Evaporating pressure, P_e and condensing pressure, P_c.

In summary, the measured variables suggest that the compressor may not be working appropriately since not even simple tests could be carried out (e.g., OD-sweep to check the nonlinear relationship between the opening degree and the outlet temperature).

6.2 Conclusion

This section pretends to assess qualitatively the controller performance and robustness as well as extract several conclusions. For this purpose, recall the initial problem formulation:

Can PID control as a process of active inference be useful to regulate the evaporator outlet temperature in a refrigeration process?

To determine whether a controller behaves well, at least four requirements must be fulfilled [Baltieri and C. Buckley 2019]:

- Load disturbance response
- Measurement noise response
- Set-point response
- Robustness to model uncertainties

In the active inference framework, the first two items depend on the measurement precisions, while the other two rely on process precisions. This new PID approach has proven that the agent can drive the outlet temperature to its desired state in a noisy environment. Furthermore, the controller can respond to changes in the set-point. Despite the absence of process noise, the generative model takes into account uncertainties in the state's dynamics. These precisions are almost negligible which entails high process variance, see Table 5.1. Nevertheless, the system's response to load disturbances has not exhibited good behaviour.

Regarding the optimal tuning of the PID gains, the system's response was sensitive to subtle changes in the hyperparameters. As opposed to the cruise controller implemented in [Baltieri and C. Buckley 2019], here small changes in the control signal entail large variations in the outlet temperature which compromises the system's stability.

In conclusion, PID control as a process of active inference could be employed in a refrigeration system. However, further investigation is needed when inferring the hyperparameters so as to enhance the load disturbance response.

6.3 Future Work

This section includes some additional considerations that could help improving the attained simulation results as well as providing some ideas for further research.

Throughout this project, the control of the evaporator outlet temperature has been considered. Nonetheless, as mentioned in Section 3.1, controlling the superheating temperature is important since ensures the correct functioning of the compressor. Hence, the target would be to adapt the designed controller to regulate the superheating temperature instead.

Concerning simulation, the controller has been implemented in one of the simplest evaporator models: the Wiener-Hammerstein model. Despite obtaining some promising results, the inability to respond to load disturbances suggests that parameter tuning should be examined more thoroughly (i.e., learning rates). Moreover, to have a better perspective of the active inference framework, the regulator should be implemented in more complex models and compared with the classical PID controller.

Finally, more tests on the laboratory should be carried out to determine what caused the unsuccessful results. Once it is ensured that the refrigeration system works properly, the goal would be to prove whether the controller is suitable for this application.

Bibliography

Adkins, C. J. (1975). *Equilibrium thermodynamics*. English. 2d ed. McGraw-Hill London ; New York.

Åström, Karl Johan and Tore Hägglund (1995). *PID Controllers: Theory, Design, and Tuning*. English. ISA - The Instrumentation, Systems and Automation Society.

Baltieri, Manuel and Christopher Buckley (2019). "PID Control as a Process of Active Inference with Linear Generative Models". eng. In: *Entropy (Basel, Switzerland)* 21.3, pp. 257–.

Buckley, Christopher L. et al. (2017). "The free energy principle for action and perception: A mathematical review". In: *Journal of mathematical psychology* 81, pp. 55–79.

Dincer, Ibrahim (2017). *Refrigeration Systems and Applications*. eng. Chicester: John Wiley & Sons, Incorporated.

Friston, K.J (2008). "Variational filtering". eng. In: *NeuroImage (Orlando, Fla.)* 41.3, pp. 747–766.

Friston, Karl (2012). "A Free Energy Principle for Biological Systems". In: *Entropy (Basel, Switzerland)* 14.11, pp. 2100–2121.

Friston, Karl, Jérémie Mattout, et al. (2007). "Variational free energy and the Laplace approximation". eng. In: *NeuroImage (Orlando, Fla.)* 34.1, pp. 220–234.

Friston, Karl, Klaas Stephan, et al. (2010). "Generalised Filtering". eng. In: *Mathematical problems in engineering* 2010, pp. 1–34.

Iglesias, Pablo A. and Brian P. Ingalls (2009). *Control theory and systems biology*. eng. Cambridge, MA: MIT Press.

Larsen, Lars Finn Sloth (2006). "Model Based Control of Refrigeration Systems". English. PhD thesis.

Pio-Lopez, Léo et al. (2016). "Active inference and robot control: a case study". eng. In: *Journal of the Royal Society interface* 13.122.

Smith, Ryan, Karl Friston, and Christopher Whyte (Jan. 2021). "A Step-by-Step Tutorial on Active Inference and its Application to Empirical Data". In:

Sutton, Richard S. and Andrew G. Barto (2018). *Reinforcement learning : an introduction*. Second edition. Adaptive computation and machine learning. Cambridge, Massachusetts: The MIT Press.

Vinther, K et al. (2013). "Single temperature sensor superheat control using a novel maximum slope-seeking method". eng. In: *International journal of refrigeration* 36.3, pp. 1118–1129.

Vinther, Kasper (2014). "Data-Driven Control of Refrigeration System". English. PhD thesis.

Wiering, Marco and Martijn van Otterlo (2012). *Reinforcement Learning: State-of-the-Art*. 1. Aufl. Adaptation, learning, and optimization. Springer-Verlag.

A Appendix

A.1 Kullback-Leibler Divergence

This section proves that KL divergence is always greater or equal than zero. First, note that:

$$\ln a \leq a - 1, \quad \forall\, a > 0 \tag{A.1}$$

Then, consider:

$$-D_{\mathrm{KL}}\left(q(x) \mid\mid p(x)\right) \leq 0 \tag{A.2}$$

which also implies $D_{\mathrm{KL}}\left(q(x) \mid\mid p(x)\right) \geq 0$. Thus, applying the KL divergence definition:

$$-D_{\mathrm{KL}}\left(q(x) \mid\mid p(x)\right) = -\int q(x) \ln\left(\frac{q(x)}{p(x)}\right) dx = \int q(x) \ln\left(\frac{p(x)}{q(x)}\right) dx \tag{A.3}$$

Applying the inequality in Eq. (A.1), one gets:

$$-D_{\mathrm{KL}}\left(q(x) \mid\mid p(x)\right) \leq \int q(x) \left(\frac{p(x)}{q(x)} - 1\right) dx = \int p(x)\, dx - \int q(x)\, dx \tag{A.4}$$

It is assumed that densities are normalised, that is, they sum to one. Hence:

$$-D_{\mathrm{KL}}\left(q(x) \mid\mid p(x)\right) \leq 0 \tag{A.5}$$

A.2 Kalman Filter

In this section, the equations related to the Kalman filter (KF) algorithm are presented. The KF is a state estimator algorithm for observable linear systems that tries to guess a process state which is not directly measured, or which is measured with a lot of noise. As explained in Subsection 4.6.2, in order to construct the PID controller one needs the higher order derivatives of the variable to be controlled.

In the case under study, only the evaporator outlet temperature is accessible. Since this outcome is quantised and contains measurement noise, discrete differentiation will result in noisy estimates. Hence, the Kalman filter algorithm is employed instead.

A.2.1 Theory

The KF regarded in this report assumes a discrete time system model with additive white Gaussian noise, as shown in the following equation:

$$\begin{aligned} x_{k+1} &= \Phi x_k + \Gamma u_k + \omega_k, \qquad &\omega_k \sim \mathcal{N}(0, Q_k) \\ y_k &= H x_k + D u_k + z_k, \qquad &z_k \sim \mathcal{N}(0, R_k) \end{aligned} \tag{A.6}$$

where x_k, y_k and u_k correspond to the states, measured outputs and control signal at time step k, respectively. Process and measurement noise are denoted with ω and z. Both are assumed to be white Gaussian noise with covariance matrices Q_k and R_k, respectively. The KF consists of two steps, namely the measurement update and the time update, which are detailed next.

Measurement update

Measurement update after receiving y_k and u_k:

$$\begin{aligned} \hat{y}_{k|k-1} &= H\hat{x}_{k|k-1} + Du_k \\ \tilde{y}_{k|k-1} &= y_k - \hat{y}_{k|k-1} \\ K_k &\triangleq P_{k|k-1}H^\top \left(HP_{k|k-1}H^\top + R_k\right)^{-1} \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k\tilde{y}_{k|k-1} \\ P_{k|k} &= (I - K_kH)P_{k|k-1}(I - K_kH)^\top + K_kR_kK_k^\top \end{aligned} \tag{A.7}$$

where K_k is the Kalman gain. The state error covariance matrix is denoted with P. The subscript $k|k-1$ denotes the variable value at time step k given all samples until, and including, $k-1$.

Time update

Time update from k to $k+1$:

$$\begin{aligned} \hat{x}_{k+1|k} &= \Phi\hat{x}_{k|k} + \Gamma u_k \\ P_{k+1|k} &= \Phi P_{k|k}\Phi^\top + Q_k \end{aligned} \tag{A.8}$$

A.2.2 Implementation

As stated previously, the measured variable corresponds to the evaporator outlet temperature, $T_{e,o}$. To estimate up to second order derivatives, the chosen states are:

$$x = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^\top = \begin{bmatrix} T_{e,o} & \dot{T}_{e,o} & \ddot{T}_{e,o} & \dddot{T}_{e,o} \end{bmatrix}^\top \tag{A.9}$$

Furthermore, $T_{e,o}$ derivatives higher than second order are assumed to be white Gaussian noise. Thus, the system model can be expressed as:

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} &= \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{A} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \omega \\ y &= \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}}_{H} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \end{aligned} \tag{A.10}$$

However, the system dynamics in Eq. (A.10) are represented in continuous time. To implement the discrete KF, one needs to discretise the model. In this thesis, the forward Euler method is regarded. Hence, the state dynamics are as follows:

$$x_{k+1} = \underbrace{(I + T_s A)}_{\Phi} x_k \tag{A.11}$$

where I is the identity matrix and T_s the sampling time. This leads to the discrete state matrix, Φ, presented in Eq. (A.6). Considering a sampling time of 1 second, the state matrix can be determined:

$$\Phi = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.12}$$

Due to the difficulty of obtaining a model of the noise involved in the process, Q_k and R_k are seen as tuning parameters. After several trials, the chosen values are:

$$R_k = 10 \qquad\qquad Q_k = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 10^{-4} & 0 & 0 \\ 0 & 0 & 10^{-4} & 0 \\ 0 & 0 & 0 & 10^{-4} \end{bmatrix} \tag{A.13}$$

Fig. A.1 depicts the attained results for the estimated evaporator outlet temperature and its two first time derivatives.

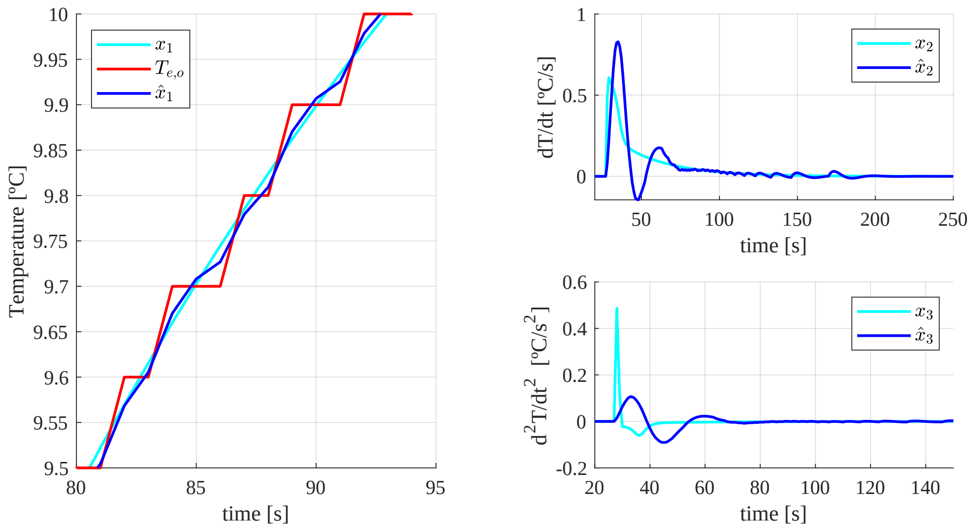


Figure A.1: Kalman filter estimates (blue) given the evaporator outlet temperature (red) from the evaporator model simulation .

The states plotted in cyan (i.e., x_1, x_2 and x_3) represent the temperature and its continuous-time derivatives without input and output quantisations so as to compare with the KF estimates. The evaporator outlet temperature, $T_{e,o}$, corresponds to the measured variable. As it can be seen, the temperature estimate, $\hat{x}_1$, follows appropriately the simulated true state, x_1. If one decreases the covariance measurement value, R_k, the KF filter would trust more the measurements and, thus, the state estimate would follow the measured temperature, $T_{e,o}$, leading to a possible overfitting. The second estimate, $\hat{x}_2$, is able to follow the true value with certain confidence. However, the second time derivative estimate, $\hat{x}_3$, does not follow reliably x_3 during the transient phase. This could be solved by increasing the diagonal values in Q_k, which would imply trusting less the model (i.e., increase of process' variances). Nevertheless, this is at the expense of obtaining noisy estimates (see $\hat{x}_2$ between $t = 100$ and $t = 200$ seconds in Fig. A.1).

A.3 Active Inference Implementation

This section provides some specific information regarding the implementation of the PID controller as a process of active inference.

A.3.1 Action and Perception Discretisation

The perception and action updates shown in Eq. (4.37) and Eq. (4.41) represent the rate of change of μ and u, respectively. This is equivalent to taking the temporal derivative of the mentioned variables. Therefore, to attain the appropriate expressions of μ and u one needs to integrate. The numerical integration employed in this thesis is the forward Euler method. Thus, the gradient-descent scheme for perception and action can be written as:

$$\tilde{\mu}^{\tau+1} = \tilde{\mu}^{\tau} + T_s \left(\dot{\tilde{\mu}}^{\tau} - \eta \nabla_{\tilde{\mu}} F \right) \tag{A.14}$$

$$u^{\tau+1} = u^{\tau} + T_s \left(-\kappa \sum \frac{\partial \tilde{y}}{\partial u^{\tau}} \nabla_{\tilde{y}} F \right) \tag{A.15}$$

As explained in [K.J Friston 2008], it is worth noting the difference between the temporal derivatives and the higher orders of motion in the generalised coordinates (i.e., $\tilde{\mu} = \{\mu, \dot{\mu}, \ddot{\mu}, ...\}$). The former can be seen as computing the motion of a point (i.e., the rate of change), whereas a point in generalised coordinates can be regarded as encoding the instantaneous trajectory of a variable. However, the rate of change $\mu^{\tau+1}$, for instance, is not necessarily the motion encoded by $\dot{\mu}$ (although it will be under Hamilton's principle of stationary action).

Another important aspect lies in the relationship between observations and action. Eq. (4.44) assumes a positive linear interaction (i.e., $\partial \tilde{y} / \partial u = 1$). Nevertheless, in the evaporator case it is seen that the more the EEV opens the lower the temperature is. Thereby, this linear relationship needs to be expressed in the following form:

$$\frac{\partial y}{\partial u} = \frac{\partial \dot{y}}{\partial u} = \frac{\partial \ddot{y}}{\partial u} = -1 \tag{A.16}$$

A.3.2 Implementation in Simulink

To implement the Kalman filter and the PID controller in the Simulink environment, the user-defined MATLAB Function block is utilised. Simulations are run with Euler solver and a fixed-step of size 1 second. Since previous sample values are required to apply the forward Euler integration method (e.g., action, perception or Kalman estimates), Unit Delay blocks are employed. Fig. A.2 depicts the block diagram in Simulink of the evaporator model and the controller. The written code in the MATLAB Function can be found in Listing A.1.

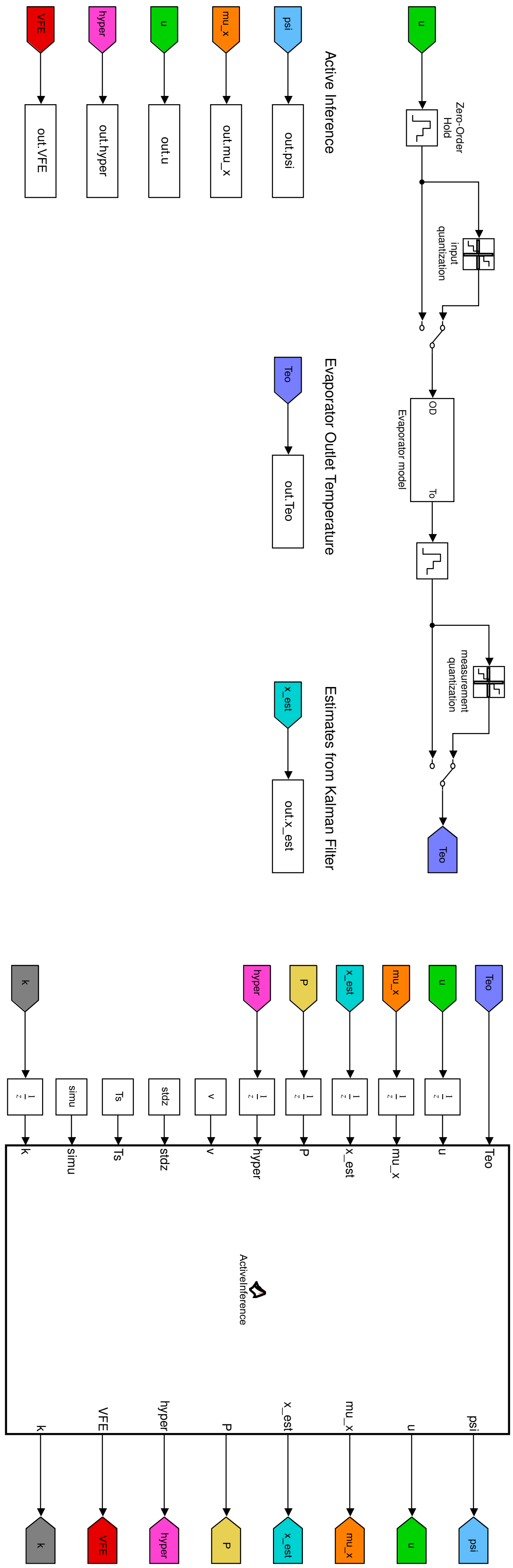


Figure A.2: Block diagram of the evaporator model and the active inference framework.

```matlab
function [ psi, u, mu_x, x_est, P, hyper, VFE, k ] =
    ActiveInference(Teo, u, mu_x, x_est, P, hyper, v, stdz, Ts, simu
    , k)

%% Kalman Filter
% Parameters
A = [ 0 1 0 0 ;       % State matrix
      0 0 1 0 ;
      0 0 0 1 ;
      0 0 0 0 ];
H = [ 1 0 0 0 ];      % Output matrix
I = eye(4);

R = 10;               % Measurement covariance
Q = [ 0.1 0 0 0  ;    % Process covariance
      0 1e-4 0 0 ;
      0 0 1e-4 0 ;
      0 0 0 1e-4 ];

% Measurement update
y_est = H*x_est;
y_err = Teo - y_est;
K = P*H'/(H*P*H' + R);
x_est = x_est + K*y_err;
P = (I - K*H)*P*(I - K*H)' + K*R*K';

% Time update
x_est = (I + Ts*A)*x_est;
P = (I + Ts*A)*P*(I + Ts*A)' + Q;

%% Active Inference
% Parameters
eta = 0.1;                  % Perception learning rate
kappa = 0.01;               % Action learning rate
rho = 0.005;                % Hyperparameter learning rate
lambda = 1.5;               % Hyperparameter damping term

alpha = 10^5;               % Linear generative model parameter

piw = 1e-9*[ 0.05 1 1 ];    % Process precisions

% Measurement precisions update
piz = exp([ hyper(1, 1) hyper(2, 1) hyper(3, 1) ]);

% Measurements
psi = x_est(1:3) + stdz.*randn(3, 1);

% Perception
mu_x(1:3) = mu_x(1:3) + Ts*(mu_x(2:4) - eta*(-piz'.*( psi - mu_x
    (1:3)) + alpha*piw'.*(mu_x(2:4) + alpha*(mu_x(1:3) - v)) + [ 0 ;
     piw(1:2)'.*(mu_x(2:3) + alpha*(mu_x(1:2) - v(1:2))) ]));
```

```
% Action
u = u + Ts*kappa*(piz*(psi - mu_x(1:3)));

% Action saturation (0-100%)
if u > 100
    u = 100;
else
    if u < 0
        u = 0;
    end
end

% Tuning hyperparameters
k = k + 1;
if k > 4e3/Ts && k < 15e3/Ts && simu
    hyper(:, 1) = hyper(:, 1) + Ts*hyper(:, 2);
    hyper(:, 2) = hyper(:, 2) + Ts*(-1/2*rho*(exp(hyper(:, 1)).*(
        psi - mu_x(1:3)).^2 - 1) - lambda*hyper(:, 2));
end

% Variational Free Energy
VFE = 1/2*(piz*(psi - mu_x(1:3)).^2 + piw*(mu_x(2:end) + alpha*(
    mu_x(1:3) - v)).^2 - log(piz(1)*piz(2)*piz(3)*piw(1)*piw(2)*piw
    (3)));
```

Listing A.1: MATLAB Function code.