The impact of event-triggered control on the energy consumption of a legged robot

Bastiaan Oosterhuis

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

The impact of event-triggered control on the energy consumption of a legged robot

Master of Science Thesis

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

Bastiaan Oosterhuis

August 25, 2016

Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS) · Delft University of Technology

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

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS) for acceptance a thesis entitled

The impact of event-triggered control on the energy consumption of a legged robot

by

Bastiaan Oosterhuis

in partial fulfillment of the requirements for the degree of

Master of Science Embedded Systems

Dated: August 25, 2016

Supervisor(s):

dr.ir. Manuel Mazo Jr.

Reader(s):

dr.ir. Tamás Keviczky

dr.ir. Przemysław Pawełczak

Abstract

In event-triggered control (ETC) the control task is only executed when needed to ensure performance or stability. A well known property of ETC is that it can reduce the number of control task executions while retaining similar performance. A large amount of research is focused on how ETC can be used to achieve energy savings in wireless networked control systems. However, the effect of ETC on the actuation energy consumption and whether it could be used as a strategy to achieve actuation energy savings has been unexplored.

This thesis presents a study of the impact of ETC on the energy consumption of a legged robot. The focus is on the control task of balancing the robot on top of its legs. To perform this study, a system allowing real-time control and energy monitoring of the robot is developed. This system allowed an experimental comparison between a standard periodic time-triggered controller and a periodic event-triggered controller.

Three experiments were performed that enabled time-triggered control and ETC to be compared during transients. The results show that ETC is capable of affecting the energy consumption while reducing the number of control updates. During the first two experiments the system was brought into a transient state by starting the system from nonzero initial conditions. During the third experiment a disturbance was applied on the control input of one leg pair. In the first experiment the energy consumption during ETC increased with 3.0% while in the second and third experiment the energy consumption decreased with 3.2% and 7.8% respectively. These differences occurred while reducing the number of control updates by at least 35.9%. Analysis of the results suggest that the differences in energy consumption are due to the different control input during ETC combined with the effect of unmodeled dynamics.

Table of Contents

List of Figures

List of Tables

Preface

This thesis presents the work that I performed at the Delft Center for Systems and Control (DCSC) towards obtaining my master's degree in Embedded Systems. My choice to specialize in embedded control systems naturally led me towards performing my thesis project at DCSC.

After getting in touch with dr.ir. Manuel Mazo and dr. Gabriel Lopes I got introduced to the idea of controlling legged robots in an event-triggered way. The question whether or not event-triggered control (ETC) can be used to achieve actuation energy savings for a legged robot, or for other systems in general, is unexplored. The knowledge gained along the way and perhaps a positive answer to this question can greatly promote the use of ETC outside of its classical application of networked control systems. This thesis project explored the effects of ETC on the energy consumption of a legged robot and resulted in insights that could be used in further research on using ETC to achieve energy savings.

I would like to sincerely thank my supervisor Manuel Mazo for his support and guidance. His clever suggestions, constructive criticism and positive attitude are very much appreciated. I would also like to thank Gabriel Lopes for introducing me together with Manuel into the world of legged robots and event-triggered control.

Furthermore I would like to thank both dr.ir. Tamás Keviczky and dr.ir. Przemysław Pawełczak for taking their time to evaluate my thesis and participate as members of my thesis committee.

Finally, I would like to thank my family and girlfriend. Their encouragement and support is greatly appreciated and helped me stay motivated throughout the years.

Bastiaan Oosterhuis
Delft, The Netherlands
August 17, 2016

Chapter 1

Introduction

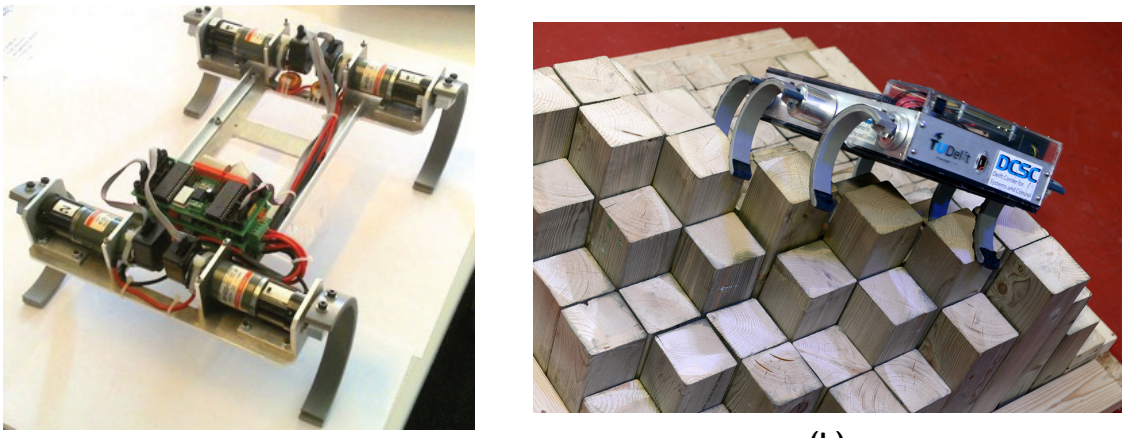
1-1 Legged robots

In urban search and rescue (USAR) situations rescuers often deal with dangerous environments such as collapsed buildings, trapped victims and rough terrains. To increase the odds of survival it is essential that trapped victims are localized and saved as fast as possible. Unfortunately, participating in rescue missions can be of great risk for rescue workers. Employing USAR robots instead of humans can help in saving lives while limiting the danger of rescue operations. These robots are designed to perform a variety of tasks such as: crawling through or removing debris, reconnaissance and mapping, operating in hazmat conditions and supplying or aiding victims.

In 1998 the RHex project started which focused on building a robotic platform that is capable of traversing rugged terrains. The RHex robots consist of six legs and are designed to be mechanically simple which promotes reliability and robustness [1]. These types of robots have one rotary actuator for each leg and are capable of performing locomotion. The use of semicircular legs instead of wheels allows them to traverse different types of terrain and to perform a wide range of tasks such as: running [1], flipping [2], climbing stairs [3] and leaping [4]. The versatile nature of these type of legged robots make them suitable for use in exploration or search and rescue missions.

To be able to assist in operations it is important that search and rescue robots are power and computationally autonomous. Due to this (often) battery powered nature, the energy consumption is an important factor. A lower energy consumption of the robot can increase mission time and make the robot more successful during operations.

Inspired by the RHex, two power and computational autonomous legged robot platforms were developed at the TU Delft. The RQuad, shown in Fig. 1-1a, is a four legged robot and the Zebro, shown in Fig. 1-1b, is a six legged robot. The robots have been designed for educational and research purposes and are used for studying and developing novel control methodologies.



(a) (b)

Figure 1-1: Power and computational autonomous legged robots developed at the TU Delft. (a) The four legged RQuad robot. (b) The six legged Zebro robot.

1-2 Event-triggered control

In most control applications the control task consists of continuously or periodically calculating and updating the control input. However, execution of the control task when the system is operating as desired is a waste of resources. The main idea behind event-triggered control (ETC) is to execute the control task only when a certain event has occurred. These events are generated by an event-triggering mechanism which ensures that the control task is only executed when needed to ensure stability or performance.

The publication of [5] and [6] showed that ETC can decrease the processor utilization while maintaining similar or even better performance. Reducing the processor utilization in resource constrained embedded control systems allows other tasks to finish quicker, and could help in reducing costs by allowing cheaper hardware. Despite the potential benefits, most control systems are implemented in a periodic time-triggered fashion.

Over the last few years the interest in ETC has grown. This can be explained by the increase of networked or distributed control systems. In these systems, reducing the number of control task executions reduces the number of transmissions. The reduction in transmissions lead to lower bandwidth utilization and lower energy consumption in wireless systems. This increasing interest in ETC caused a rapid development of theory that systematically analyses event-triggered control, e.g. [7, 8, 9, 10, 11, 12, 13, 14, 15].

Although still limited, the number of applications and experimental results of ETC are growing as well. Examples of applications are: attitude stabilization of a quadrotor [16] and controlling a double tank setup [17]. The results are clear: a significant reduction in control computations resulting in less communication in distributed or networked systems.

ETC of legged robots can offer several benefits. A decrease in processor utilization in absence of events means that more CPU time will be available for other tasks. If the other tasks can finish quicker the total mission time could be reduced. Besides potentially saving energy due to a shorter mission time, ETC can have a direct impact on the actuation energy consumption due to differences in the control input.

1-3 Problem Statement

A lot of research on ETC is focused on how energy can be saved by reducing the number of transmissions in wireless networked control systems (e.g. [18, 17, 19]). Whether or not ETC can affect the actuation energy consumption and if it could be used to achieve energy savings in the future is yet unexplored.

Due to their often battery powered nature, legged robots make for an interesting subject to study what effect ETC has on their energy consumption. Performing this research can aid in deciding when to employ event-triggered control outside of its traditional application of networked or distributed control systems.

This leads to the following research question of this thesis:

What is the impact of event-triggered control on the actuation energy consumption of a legged robot?

Since ETC of legged robots is unexplored the focus will be on a basic but fundamental control task: Balancing the robot on top of its legs. The RQuad is the legged robot on which this thesis is focused and a schematic depiction of the control task viewed from the side of the robot is shown in Fig. 1-2. The focus will be on gaining insight in how current ETC techniques affect the energy consumption rather than necessarily designing an energy saving event-triggered controller.

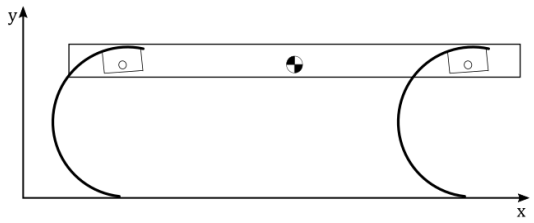


Figure 1-2: Schematic depiction of the control task viewed from the side of the robot.

In order to answer the research question the following objectives are defined:

- Develop a system that allows real-time control and energy monitoring of the RQuad
- Design and implement a time-triggered controller for the RQuad
- Design and implement an event-triggered controller for the RQuad

A large part of the thesis work will focus on developing a system that allows real-time control and monitoring of the RQuad. This system should allow easy implementation of control algorithms and provide means for energy monitoring. To determine the effect of ETC on the energy consumption a standard time-triggered controller will be designed and implemented first. This time-triggered controller will serve as a base case for the comparison with event-triggered control. Next, an event-triggered controller will be designed using existing techniques from the literature and will be compared with the time-triggered controller.

1-4 Thesis outline

This thesis is structured as follows: The required preliminary knowledge of ETC will be presented in Chapter 2. Chapter 3 will discuss the modeling of the RQuad. Next, Chapter 4 will discuss control design. After control design the design of the system will be addressed in Chapter 5. Chapter 6 will discuss the experiments and their results. Finally, conclusions and suggestions for future work will be given in Chapter 7.

Chapter 2

Preliminaries

This chapter will discuss the preliminaries of event-triggered control. First, an overview of the different aspects and prior work in the field of event-triggered control will be given. This will be followed by a more technical discussion regarding the design of event-triggered controllers.

2-1 Event-triggered control

2-1-1 Overview

An event-triggered control scheme generally consists of two elements: a feedback controller and an event-triggering mechanism [20]. The feedback controller computes the control input and the event-triggering mechanism (ETM) decides when the controller has to recompute and update the input, i.e. when an event should be triggered. Fig. 2-1 shows a basic ETC scheme where the ETM is responsible for closing the control loop by generating events. Event-triggered controllers are mostly designed using the emulation approach. In this approach the feedback controller is designed first without taking event-triggered control into account after which the event-triggering mechanism is designed [10] .

The ETM is responsible for monitoring the system state or output and should generate an event when the state or output deviates too much from a certain threshold. The term continuous event-triggered control (CETC) is used when the triggering condition is checked continuously and the term periodic event-triggered control (PETC) when the triggering condition is checked periodically [11]. In PETC the ETM checks at every sampling instance if an event should be triggered. PETC can be used to mimick CETC by implementing it on a digital platform with a sufficiently high sampling frequency. A benefit of PETC over CETC is that there is a guaranteed minimum inter-event time, which is at least the sampling period [11]. This is not always guaranteed for CETC, where for some triggering mechanisms Zeno behaviour might occur [10, 15]. Since the event-triggered controller will be implemented on a digital platform, the focus in this thesis will be on implementing PETC.

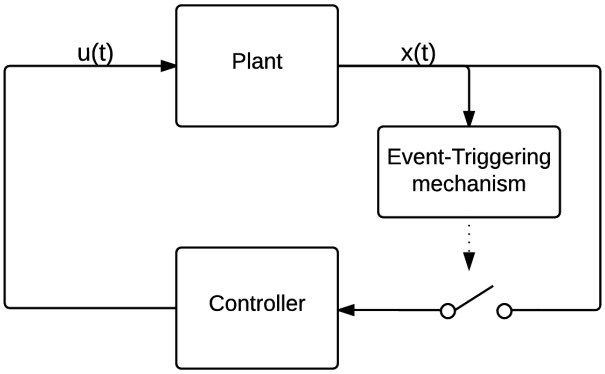


Figure 2-1: Event-triggered control scheme.

The most common approach in ETC is to keep the control input constant between two events (zero-order hold). Besides zero-order hold other approaches exist where a control input generator is used. The control input generator uses model-based predictions to generate the control input. The ETM which has knowledge of the state also runs the model-based prediction and triggers an event when the prediction deviates too much from the real state. Different model based approaches are discussed in e.g. [7, 12, 14, 21]. These approaches are less common and will not be used in this thesis since they mimic regular control and therefore a smaller difference in the control or actuation signal is expected than for zero-order hold.

An ETC scheme can be further classified as centralized or decentralized. In centralized event-triggered control, a central node is used which has all the state or output information needed to determine if an event should be triggered. In decentralized event-triggered control, sensors and actuators can be physically distributed and not all sensor nodes have access to the full state information. Decentralized systems have led to the study of decentralized triggering mechanisms where a node can trigger an event based on local information [7, 10, 22, 23]. Due to the centralized nature of the test platform, which is the RQuad, the focus of this thesis will be on centralized ETC.

Since the system state is not always available for feedback, output-based event-triggered control can be used. The output-based event-triggering mechanisms can be classified into direct output based and observer based (dynamic) [20]. The observer can be implemented at the event-triggering mechanism and run at the same speed as the triggering mechanism, or it could be implemented at the controller where only the measurements transmitted during events can be used to reconstruct the state.

2-1-2 Periodic event-triggered control

This section will discuss the design and analysis techniques for periodic event-triggered control following the approach in [11].

The following linear time-invariant plant is considered

$$\dot{x} = A_p x + B_p \hat{u} + B_w w \tag{2-1}$$

where $x \in \mathbb{R}^{n_x}$ is the system state, $\hat{u} \in \mathbb{R}^{n_u}$ is the input to the plant, and $w \in \mathbb{R}^{n_w}$ is the disturbance.

In the case of normal periodic control the following state-feedback controller is generally used

$$\hat{u}(t) = Kx(t_k), \qquad \text{for} \quad t \in (t_k, t_{k+1}] \tag{2-2}$$

where t_k represents k-th sampling instant and the sampling period is given by $h = t_{k+1} - t_k$.

Instead of updating the control input given by (2-2) every sampling instance, PETC can be employed to update the control input only when an event has occurred. The controller (2-2) can then be changed into

$$\hat{u}(t) = K\hat{x}(t), \qquad \text{for} \quad t \in \mathbb{R}_+ \tag{2-3}$$

where $\hat{x}(t)$ will remain constant in absence of events and is given for $t \in (t_k, t_{k+1}]$ by

$$\hat{x}(t) = \begin{cases} x(t_k), & \text{when} \quad \mathcal{C}(x(t_k), \hat{x}(t_k)) > 0 \\ \hat{x}(t_k), & \text{when} \quad \mathcal{C}(x(t_k), \hat{x}(t_k)) \leq 0 \end{cases} \tag{2-4}$$

Basically, $\hat{x}(t)$ is the value of the state for which the latest event has occurred, i.e. the most recent state known to the controller at time t. In (2-4), $\mathcal{C}$ represents the triggering condition that determines whether or not the state and therefore the control input should be updated.

In [11] a general class of quadratic triggering conditions is introduced of the following form

$$\mathcal{C}(\xi(t_k)) = \xi^T(t_k) Q \xi(t_k) > 0 \tag{2-5}$$

where $\xi = \begin{bmatrix} x^T & \hat{x}^T \end{bmatrix}^T \in \mathbb{R}^{n_\xi}$ with $n_\xi = 2n_x$ and where $Q \in \mathbb{R}^{n_\xi \times n_\xi}$ is a symmetric matrix. This framework can be used to express different kinds of event-triggering conditions by choosing a suitable matrix for Q.

Furthermore, the behavior of a PETC system can be captured by modeling it as an impulsive system [10, 11]. In [11] the system is transformed into an impulsive system by defining the following matrices

$$\bar{A} = \begin{bmatrix} A_p & B_p K \\ 0 & 0 \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} B_w \\ 0 \end{bmatrix}, \quad J_1 = \begin{bmatrix} I & 0 \\ I & 0 \end{bmatrix}, \quad J_2 = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \tag{2-6}$$

then the complete PETC system can be formulated as an impulsive system as follows

$$\begin{aligned} \begin{bmatrix} \dot{\xi} \\ \dot{\tau} \end{bmatrix} &= \begin{bmatrix} \bar{A}\xi + \bar{B}w \\ 1 \end{bmatrix}, \qquad \text{when} \quad \tau \in [0, h] \\ \begin{bmatrix} \xi^+ \\ \tau^+ \end{bmatrix} &= \begin{cases} \begin{bmatrix} J_1 \xi \\ 0 \end{bmatrix}, & \text{when} \quad \xi^T Q \xi > 0, \tau = h \\ \begin{bmatrix} J_2 \xi \\ 0 \end{bmatrix}, & \text{when} \quad \xi^T Q \xi \leq 0, \tau = h \end{cases} \end{aligned} \tag{2-7}$$

This formulation of a PETC system represents that at each sampling instance $\tau = h$ it is checked whether or not an event should be triggered. When at a sampling instance $\tau = h$ an event occurs, i.e. when $\xi^T Q \xi > 0$, the state $\xi = \begin{bmatrix} x^T & \hat{x}^T \end{bmatrix}^T$ is updated as follows

$$\xi^+ = \begin{bmatrix} x^+ \\ \hat{x}^+ \end{bmatrix} = J_1 \xi = \begin{bmatrix} x \\ x \end{bmatrix} \tag{2-8}$$

and when no event occurs $\xi^T Q \xi \leq 0$ the state is kept constant as follows

$$\xi^+ = \begin{bmatrix} x^+ \\ \hat{x}^+ \end{bmatrix} = J_2 \xi = \begin{bmatrix} x \\ \hat{x} \end{bmatrix}. \tag{2-9}$$

At each sampling instance the state τ is set back to 0 to be able to keep track of the time between samples.

The goal is to design an event-triggering condition of the form given by (2-5) that renders the closed loop system of the combination of (2-1), (2-3), (2-4) and (2-5) stable while minimizing the number of controller updates required. Where the following definition of global exponential stability is used.

Definition 1. *"The PETC system (2-7) is said to be globally exponentially stable (GES), if there exist $c > 0$ and $\rho > 0$ such that for any initial condition $\xi(0) = \xi_0 \in \mathbb{R}^{n_\xi}$ all corresponding solutions to (2-7) with $\tau(0) \in [0, h]$ and $w = 0$ satisfy $||\xi(t)|| \leq ce^{-\rho t}||\xi_0||$ for all $t \in \mathbb{R}_+$. In this case we call ρ a (lower bound on the) decay rate."* [11]

To accomplish this goal three different analysis approaches are described in [11] to design and analyze the PETC systems, namely the impulsive system approach, piecewise linear system approach and the perturbed linear (PL) system approach. Of the three approaches the piecewise linear system approach is computationally the most complex but yields the least conservative results, followed by the impulsive system approach and finally the perturbed linear approach.

In this section the simplest but most conservative of the three approaches will be discussed, the perturbed linear systems approach. This approach is valid for so called relative triggering mechanisms, for example

$$||\hat{x}(t_k) - x(t_k)|| > \sigma ||x(t_k)|| \tag{2-10}$$

where $||x||$ represents the 2-norm and $\sigma > 0$ is a constant. This triggering mechanism can be cast into the framework of (2-5) by defining

$$Q = \begin{bmatrix} (1-\sigma^2)I & -I \\ -I & I \end{bmatrix}. \tag{2-11}$$

The stability analysis for the perturbed linear approach is based on the following discretized version of the plant

$$x_{k+1} = (A + BK)x_k + BKe_k \tag{2-12}$$

where $e_k = \hat{x}_k - x_k$ represents the measurement error.

The following result is obtained for the system given by (2-12):

Theorem 1. *"Suppose the PL system (2-12) admits a storage function $\bar{V}(x) = x^T \bar{P} x$ with $\bar{P}$ a symmetric positive definite matrix for supply rate $-\theta^2||x||^2 + ||e_k||^2$ with $\theta > 0$, i.e. the dissipation inequality*

$$\bar{V}(x_{k+1}) - \bar{V}(x_k) \leq -\theta^2||x_k||^2 + ||e_k||^2, \quad k \in \mathbb{N} \tag{2-13}$$

is satisfied for any disturbance sequence $\{e_k\}_{k\in\mathbb{N}}$ and all corresponding solutions $\{x_k\}_{k\in\mathbb{N}}$. Then the PETC system (2-7) with Q as in (2-11) is GES for any $0 < \sigma < \theta$ " [11]

In [11] the existance of a storage function used to prove GES of the PETC system has been recast in checking the feasibility of the following set of LMIs:

$$\begin{bmatrix} \bar{P} - (A+BK)^T \bar{P}(A+BK) - \theta^2 I & -(A+BK)^T \bar{P}(BK) \\ -(BK)^T \bar{P}(A+BK) & I - (BK)^T \bar{P} BK \end{bmatrix} > 0$$
$$\bar{P} > 0 \tag{2-14}$$

By maximizing the value of θ^2 with respect to the LMIs of Eq. (2-14) one can find the largest σ that yields the largest time between events while keeping the system stable. This corresponds to a convex optimization problem.

2-1-3 Event-triggering mechanisms

Designing an event-triggered controller following the emulation approach mostly consists of designing an appropriate event-triggering mechanism. For CETC it is important to have a guaranteed positive nonzero time between events to ensure that the ETC scheme actually yields benefits. For PETC there is already a guaranteed inter-event time which is equal to the sampling period. However, if events are generated at the same rate at which the event-triggering mechanism is evaluated, the same behavior as in time-triggered control is obtained. In this situation ETC is not yielding any benefits and will only complicate matters.

The most popular triggering mechanisms can be captured using the following general representation:

$$t_{i+1} = \inf\{t > t_i \mid \rho(||e(t)|| > \sigma||x(t)|| + \beta\} \tag{2-15}$$

where t_{i+1} is the next event time and t_i is the time of the latest event. This representation could be used to capture both continuous or periodic triggering mechanisms. The most commonly used triggering mechanisms are:

- Relative event triggering: $0 < \sigma < 1$ and $\beta = 0$. See e.g. [8, 23, 14].
- Absolute event triggering: $\sigma = 0$ and $\beta > 0$. See e.g. [21]
- Mixed event triggering: $0 < \sigma < 1$ and $\beta > 0$. See e.g. [10]

In general, a relative triggering mechanism is robust when the system state is far from the origin. However, when the system is operating close to the origin many events could be generated in the presence of arbitrarily small external disturbances or measurement noise [15].

When using absolute triggering, the absolute term β prevents in case of small disturbances or noise that many events will be triggered when the state is close to the origin. However, when the system is far from the origin many events can be generated [15].

The advantages of relative and absolute triggering are combined in a mixed triggering mechanism. The absolute term prevents many events from being triggered when close to the origin and the relative term prevents many events from being triggered when far from the origin [15].

Chapter 3

Modeling the RQuad

In this chapter a model for the RQuad will be derived. This model will be used during control design and allows application of the model-based PETC design technique discussed in Section 2-1-2.

3-1 Modeling assumptions

During the RHex project different models of varying complexity have been derived using the Newton-Euler approach [1, 2, 4, 24]. Due to the similar nature of the RHex and RQuad, the assumptions used to model the RHex provide a good starting point to derive a model for the control task under study using the Euler-Lagrange method.

The following assumptions used to model RHex type robots will be used to model the RQuad:

1. The mass of a leg is negligible relative to the mass of the body [1, 2, 4, 24]
2. The legs are rigid bodies [2, 25]
3. The legs do not slip [1, 4, 24, 25]

The first assumption is straightforward since the weight of a leg is small compared to the weight of the body (<2%). For most RHex type robots the legs are designed to be slightly compliant. The compliant legs allow the storage and release of energy during locomotion [26]. Since no locomotion will be performed the compliancy of the legs will be ignored. The third assumption is used because the RQuad has rubber coating on the legs which increases their grip.

Furthermore, due to the configuration of the motors where their rotation axes are parallel and aligned with the body of the robot it is expected that the motion will be mostly in the fore/aft direction of the robot. This will lead to an additional assumption used to model the robot:

4. The balancing motion is primarily planar

By using this assumption a planar 2 leg model can be derived instead of a 4 leg model. In such a model the legs will not be controlled individually but in pairs in similar fashion as in [2, 4, 25].

The numbering for the leg/motor pairs that will be used throughout this thesis is shown in Fig. 3-1. Leg 1 represents the front left leg of the robot, leg 2 the front right leg, leg 3 the back left leg and leg 4 the back right leg. Legs 1 and 2 will be called to front leg pair and legs 3 and 4 the back leg pair.

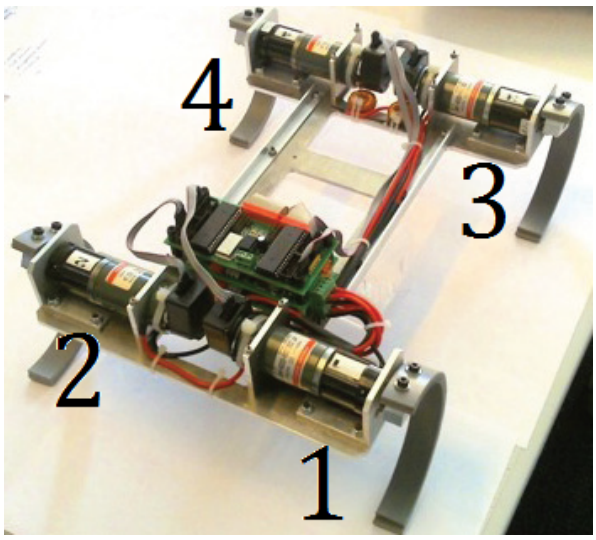


Figure 3-1: Leg numbering for the RQuad

3-2 Four-bar linkage

A four-bar linkage as shown in Fig. 3-2 is a commonly used mechanical building block consisting of four links connected in a loop by joints. One of the links is fixed and three of the links can move. In [27] and [28] a four-bar linkage mechanism is analyzed and the dynamic equation describing the motion of the link is presented. Fig. 3-2 shows the four-bar linkage discussed in [27]. For each link ($i = 1, 2, 3$) l_i represents the length of the link, m_i represents the mass of a link and l_{ci} represents the distance from the joint of a link to its centre of mass. Furthermore, the absolute angles of links 1, 2 and 3 with respect to the horizontal are denoted by θ, α and ϕ respectively.

Because of the assumption that the legs do not slip, the distance between the legs will remain constant and the robot has similarities to a four-bar linkage. Fig. 3-3 shows how the robot can be modeled as a four-bar linkage by making the following observations:

- The back leg pair (legs 3 and 4) represent link 1
- The body of the robot represents link 2
- The front leg pair (legs 1 and 2) represent link 3
- The distance between the leg pairs on the ground represents link 0

The joint connecting links 2 and 3 will be called joint 1 and the joint connecting links 1 and 2 will be called joint 2.

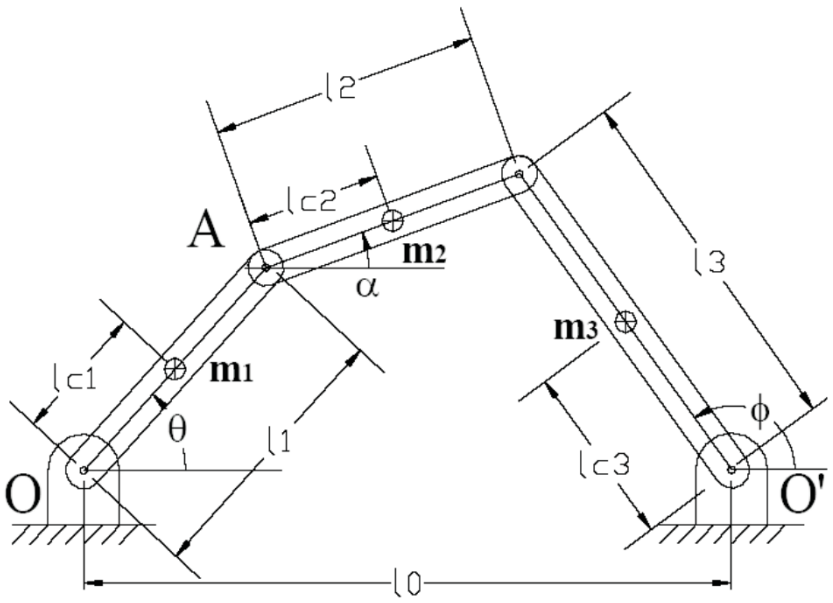


Figure 3-2: Four-bar linkage. Reproduced with permission from [27]

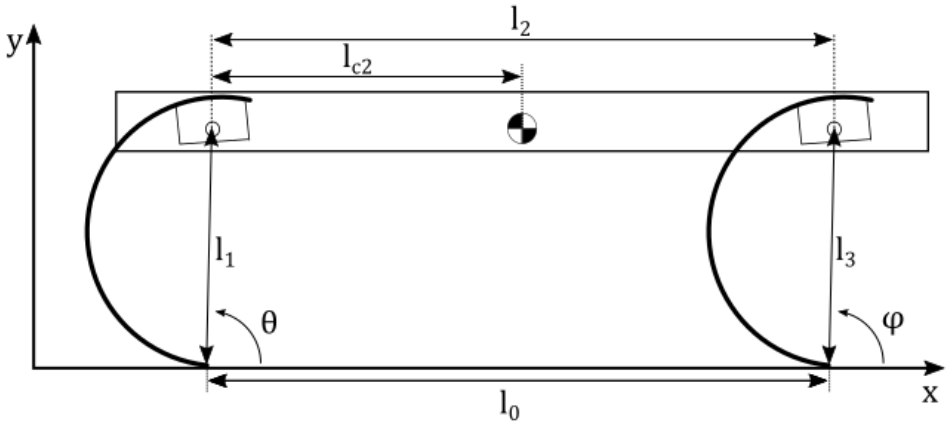


Figure 3-3: RQuad modeled as a four-bar linkage.

3-3 Kinematic analysis

In [27] the four-bar linkage is described using the coordinates θ, α and ϕ. These angles are not independent and are related through the following constraint equations:

$$-l_1 \cos\theta - l_2 \cos\alpha + l_0 + l_3 \cos\phi = 0 \tag{3-1}$$

$$-l_1 \sin\theta - l_2 \sin\alpha + l_3 \sin\phi = 0 \tag{3-2}$$

Following [27] the constraint equations can be used to express α and ϕ in terms of θ in order to describe the system with just one coordinate. This yields the following expressions for α and ϕ of which the derivation can be found in appendix A-1

$$\phi(\theta) = 2 \cdot \arctan\left(\frac{-k_1 \pm \sqrt{k_1^2 + k_2^2 - k_3^2}}{k_3 - k_2}\right) \tag{3-3}$$

$$\alpha(\theta, \phi) = \arctan\left(\frac{-l_1 \sin\theta + l_3 \sin\phi}{l_0 - l_1 \cos\theta + l_3 \cos\phi}\right). \tag{3-4}$$

where

$$k_1 = -2 l_1 l_3 \sin\theta \tag{3-5}$$

$$k_2 = 2 l_3 (l_0 - l_1 \cos\theta) \tag{3-6}$$

$$k_3 = l_0^2 + l_1^2 - l_2^2 + l_3^2 - 2 l_0 l_1 \cos\theta. \tag{3-7}$$

By differentiating the constraint equations given by (3-1) and (3-2) with respect to time the following expressions for the velocities are obtained [27, 28]

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} S_1(\theta, \alpha, \phi) \\ S_2(\theta, \alpha, \phi) \end{bmatrix} \dot{\theta} \tag{3-8}$$

where

$$S_1(\theta, \alpha, \phi) = \frac{l_1 \sin(\phi - \theta)}{l_2 \sin(\alpha - \phi)} \tag{3-9}$$

$$S_2(\theta, \alpha, \phi) = \frac{l_1 \sin(\alpha - \theta)}{l_3 \sin(\alpha - \phi)}. \tag{3-10}$$

Note that even though α and ϕ are present in (3-9) and (3-10) they can be expressed purely in terms of θ by applying (3-3) and (3-4). Furthermore the derivation details of (3-8) can be found in Appendix A-1.

3-4 Equation of motion

Next, by following [27] and [28] the Lagrangian is computed after which the the Euler-Lagrange equation is applied. This leads to the following equation of motion

$$M(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) = Q_\theta \tag{3-11}$$

where Q_θ represents the generalized forces, $M(\theta)$ the inertia terms and $V(\theta, \dot{\theta})$ represents the Coriolis and gravity terms. The full expressions of $M(\theta)$ and $V(\theta, \dot{\theta})$ along with their derivations details can be found in Appendix A-2.

The generalized forces Q_θ are given by

$$Q_\theta = \sum_{i=1}^{2} \mathbf{f}_i \frac{\partial r_i}{\partial \theta} \tag{3-12}$$

where r_i represent the location of the points on which the forces act and $\mathbf{f}_i$ represent the forces acting on the system. The position where the motor and friction torque is acting for joint 1 is given by

$$r_1 = (2\pi - \phi + \alpha)\hat{k}. \tag{3-13}$$

And the position where the torques are acting on joint 2 is given by

$$r_2 = ((\pi - \theta) + \alpha)\hat{k}. \tag{3-14}$$

This yields the following expression for the generalized forces

$$\begin{aligned} Q_\theta &= (\tau_1 - \tau_{j1})(-\frac{\partial \phi}{\partial \theta} + \frac{\partial \alpha}{\partial \theta}) + (\tau_2 - \tau_{j2})(-1 + \frac{\partial \alpha}{\partial \theta}) \\ &= (\tau_1 - \tau_{j1})(S_1 - S_2) + (\tau_2 - \tau_{j2})(S_1 - 1) \end{aligned} \tag{3-15}$$

where τ_1 represents the torque exerted by the motors of leg pair 1 (motors 1 and 2) and τ_2 represents the torque exerted by the motors of leg pair 2 (motors 3 and 4). Furthermore, the joint friction torques are given by

$$\tau_{j1} = 2b_j(\dot{\alpha} - \dot{\phi}) = 2b_j(S_1 - S_2)\dot{\theta} \tag{3-16}$$

$$\tau_{j2} = 2b_j(\dot{\alpha} - \dot{\theta}) = 2b_j(S_1 - 1)\dot{\theta} \tag{3-17}$$

where b_j represents the viscous joint friction coefficient of one motor.

If the system state is defined as $x = \begin{bmatrix} \theta & \dot{\theta} \end{bmatrix}^T$ the dynamics of Eq. (3-11) can be represented in the following general form

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = f(x) = \begin{bmatrix} x_2 \\ \frac{1}{M(x_1)}(Q_\theta - V(x_1, x_2)) \end{bmatrix} \tag{3-18}$$

3-5 Motor model

To incorporate the motor dynamics the following model is used [25]

$$\begin{aligned} \tau &= N_g \tau_m \\ \tau_m &= \frac{K_t}{R}(V - K_e \dot{\theta}_m) \\ \dot{\theta}_m &= N_g \dot{\gamma} \end{aligned} \tag{3-19}$$

where τ is the torque applied to a joint, τ_m is the torque delivered by the motor, V is the input voltage, N_g is the gear ratio, K_t is the torque constant, K_e is the back EMF constant and $\dot{\gamma}$ is the angular velocity of a joint. Since there is a coil in series with the motor, $R = R_m + R_c$, which is the resistance of the rotor plus the resistance of the coil. The coil in series with the motor will be discussed in Section 5-3.

The torque delivered by the motors at the joints is then given by

$$\begin{aligned} \tau_1 &= N_g \frac{K_t}{R}(u_1 - K_e N_g(\dot{\alpha} - \dot{\phi})) \\ &= N_g \frac{K_t}{R}(u_1 - K_e N_g(S_1 - S_2)\dot{\theta}) \end{aligned} \tag{3-20}$$

$$\begin{aligned} \tau_2 &= N_g \frac{K_t}{R}(u_2 - K_e N_g(\dot{\alpha} - \dot{\theta})) \\ &= N_g \frac{K_t}{R}(u_2 - K_e N_g(S_1 - 1)\dot{\theta}) \end{aligned} \tag{3-21}$$

since the torque at each joint is generated by two motors $\frac{u_1}{2}$ represents the input voltage for each motor on joint 1 and $\frac{u_2}{2}$ is the input voltage for each motor on joint 2.

3-6 Linearized model

To apply the PETC design techniques presented in [11] the model given by (3-18) will be linearized. The system will be linearized for $l_0 = l_2$, i.e. the legs are as wide apart as the body link. This will result in $\theta = \phi$ and $\alpha = 0$. The relationship given by (3-8) will become

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \dot{\theta}. \tag{3-22}$$

Next the equation of motion (3-11) can be simplified to

$$2J_1 \ddot{\theta} - \frac{\partial G}{\partial \theta} = -\tau_1 - \tau_2 + \tau_{j1} + \tau_{j2} \tag{3-23}$$

where

$$J_1 = \frac{1}{2} m_2 l_1^2 \tag{3-24}$$

$$\frac{\partial G}{\partial \theta} = -m_2 l_1 g \cos\theta. \tag{3-25}$$

By simplifying (3-23) and applying (3-20), (3-21), (3-16) and (3-17) the following equation is obtained:

$$\ddot{\theta} + (2N_g^2 \frac{K_t K_e}{R m_2 l_1^2} + \frac{4b_j}{m_2 l_1^2})\dot{\theta} + \frac{g}{l_1}\cos\theta = -N_g \frac{K_t}{R m_2 l_1^2}(u_1 + u_2) \tag{3-26}$$

which can be represented in the form $\dot{x} = f(x)$ as follows

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{g}{l_1}\cos x_1 - (2N_g^2 \frac{K_t K_e}{R m_2 l_1^2} + \frac{4b_j}{m_2 l_1^2})x_2 - N_g \frac{K_t}{R m_2 l_1^2}(u_1 + u_2) \end{bmatrix} \tag{3-27}$$

The system given by (3-27) can be linearized to obtain a linear system of the following form

$$\dot{x}_\Delta = A x_\Delta + B u_\Delta \tag{3-28}$$

for a certain equilibrium state and equilibrium input given by x_{eq} and u_{eq} respectively. Furthermore, $x_\Delta = x - x_{eq}$, $u_\Delta = u - u_{eq}$ and the system matrices A and B are defined as follows

$$A = \partial_x f(x_e, u_e) \tag{3-29}$$

$$B = \partial_u f(x_e, u_e). \tag{3-30}$$

This leads to the following system matrices

$$A = \begin{bmatrix} 0 & 1 \\ \frac{g}{l_1}\sin x_{eq1} & -(2\frac{N_g^2 K_t K_e}{R m_2 l_1^2} + \frac{4b_j}{m_2 l_1^2}) \end{bmatrix} \tag{3-31}$$

$$B = \begin{bmatrix} 0 & 0 \\ -N_g \frac{K_t}{R m_2 l_1^2} & -N_g \frac{K_t}{R m_2 l_1^2} \end{bmatrix}. \tag{3-32}$$

For a specific equilibrium state x_{eq} the equilibrium input can be computed as follows

$$(u_1 + u_2) = \frac{\frac{g}{l_1}\cos x_2}{\frac{N_g K_t}{R m_2 l_1^2}}. \tag{3-33}$$

The system parameters are shown in Table 3-1. The motor parameters are taken from the manufacturers datasheet [29], b_j is taken from [25] which is obtained using nonlinear system identification of the RQuad and R_c is measured using the four wire Kelvin measurement technique.

Table 3-1: Parameters of the RQuad.

Parameter	Value	Unit
l_1	$5.25 \cdot 10^{-2}$	m
g	9.81	m/s^2
m_2	2.517	kg
N_g	35	-
K_t	$3.15 \cdot 10^{-2}$	Nm/A
K_e	$3.15 \cdot 10^{-2}$	Nm/A
R_m	5.21	Ω
R_c	1.02	Ω
b_j	$1 \cdot 10^{-4}$	Nms

Chapter 4

Control design

This chapter discusses the design of a periodic and event-triggered controller based on the model derived in Chapter 3. The periodic controller will serve as the base case for the comparison with event-triggered control. Furthermore, the periodic controller will be used to design the event-triggering mechanism of the event-triggered controller using the emulation approach.

4-1 Linear quadratic regulator

The design of the controller starts with the following linear model

$$\begin{aligned} \dot{x}_\Delta &= Ax_\Delta + Bu_\Delta \\ y &= Cx_\Delta. \end{aligned} \tag{4-1}$$

where

$$A = \begin{bmatrix} 0 & 1.0000 \\ 93.0891 & -14.0761 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ -6.3772 & -6.3772 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}. \tag{4-2}$$

These A and B matrices are obtained by using (3-31) and (3-32) for the following equilibrium state $x_{eq} = \begin{bmatrix} 1.4855 & 0 \end{bmatrix}^T$ and $u_{eq} = \begin{bmatrix} -0.6239 & -0.6239 \end{bmatrix}^T$ and the parameters listed in Table 3-1. The matrix C represents that only the first state is directly measurable. The equilibrium angle $x_{eq1} = \theta_{eq}$ is chosen to be approximately 5° smaller than a straight angle. For this angle the robot is standing on the tip of its semicircular legs which makes its resemblance to the four-bar linkage closer and will cause a higher equilibrium input which might make differences between ETC and periodic control more apparent in terms of energy consumption.

To design a control law the linear quadratic regulator (LQR) method will be used. A feedback law of the form

$$u_\Delta = -Kx_\Delta \tag{4-3}$$

will be designed that renders the closed loop

$$\dot{x}_\Delta = (A - BK)x_\Delta \tag{4-4}$$

asymptotically stable while minimizing the following cost function

$$J = \int_0^\infty x(t)^T Q x(t) + u(t)^T R u(t) dt. \tag{4-5}$$

The weighting matrices Q and R have been chosen to reflect that fast convergence of the robot angle is required while allowing higher angular velocities and control input. The weighting matrices are shown below

$$Q = \begin{bmatrix} 1000 & 0 \\ 0 & 5 \end{bmatrix} \tag{4-6}$$

$$R = \begin{bmatrix} 0.02 & 0 \\ 0 & 0.02 \end{bmatrix}. \tag{4-7}$$

By using the MATLAB function "lqr" the following feedback gain is obtained

$$K = \begin{bmatrix} -165.5809 & -11.2326 \\ -165.5809 & -11.2326 \end{bmatrix}. \tag{4-8}$$

For reference tracking the feedforward gain G should be calculated to satisfy the following equation

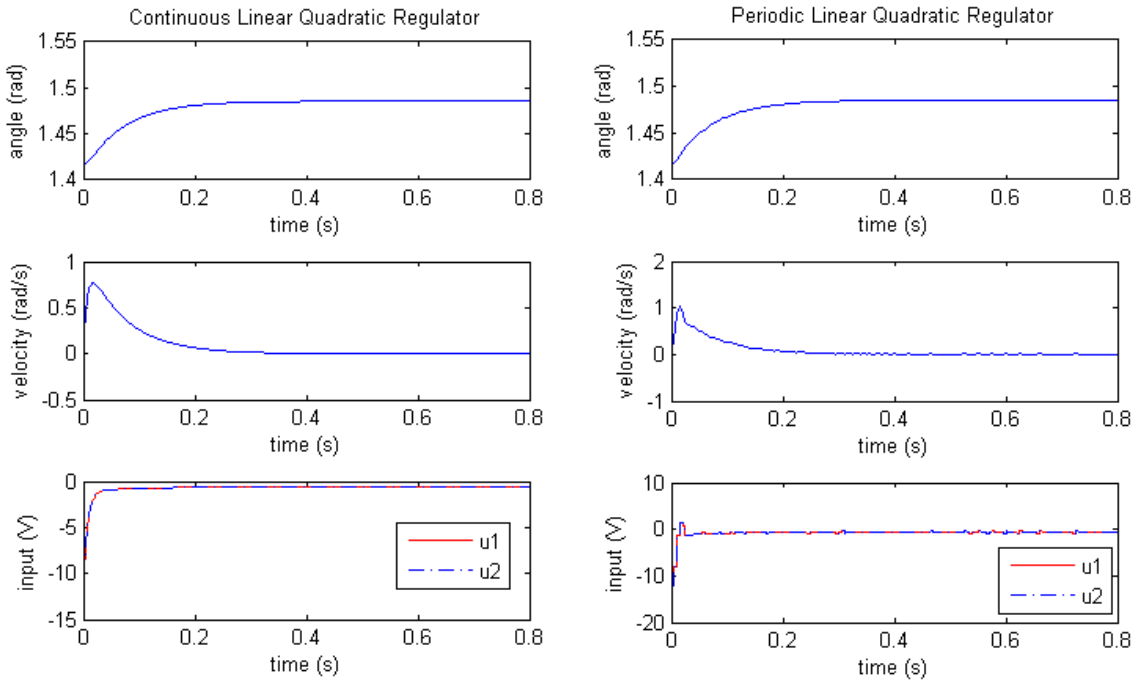
$$[C(A - BK)^{-1}B]G = 1 \tag{4-9}$$

which is satisfied by using the following feedforward gain

$$G = \begin{bmatrix} -158.2822 \\ -158.2822 \end{bmatrix}. \tag{4-10}$$

Fig. 4-1a shows the simulation results for the LQR controller with full state feedback. The first graph shows $x_1 = \theta$, the second graph shows $x_2 = \dot{\theta}$ and the third graph shows the control inputs. The simulation is performed in Simulink and the controller is applied to the nonlinear RQuad model given by (3-18). From the perturbed initial conditions the system reaches its equilibrium state within 0.5 s.

Fig. 4-1b shows the simulation results for the periodic controller. The angular velocity is obtained by backward differentiation of the angle and the limited resolution of the encoders (70000 counts per rotation) is included in the simulation as well. The sampling frequency is chosen to be 200 Hz which corresponds to 31 samples during the rise time of the closed loop system. Additionally, it has been verified that the eigenvalues of the discrete time closed loop system $(A_d - B_d K)$ are within the unit circle. It can be seen that the performance of the periodic controller is similar to the continuous one although the limited resolution of the measured angle causes some noise on the angular velocity.



(a) Continuous lqr controller

(b) Periodic lqr controller

Figure 4-1: Linear quadratic regulator simulation results

4-2 Event-triggering mechanism

To design the event-triggered controller the PETC approach discussed in [11] will be used. Since the design approach is emulation based we start out with the controller designed in the previous section given by (4-3). This controller will then be turned into an event-triggered controller by designing an event-triggering mechanism.

The event-triggering mechanism will be of the relative kind and is repeated below for convenience. If it turns out that many events will be generated when the system is operating close to the origin, the triggering mechanism will be appended with an absolute term as suggested by the results presented in [15].

$$||\hat{x}(t_k) - x(t_k)|| > \sigma||x(t_k)|| \tag{4-11}$$

To determine σ the perturbed linear systems approach is used. By solving the following optimization problem σ can be determined by picking $0 < \sigma < \theta$

$$\max \theta^2 \qquad \text{subject to}$$
$$\begin{bmatrix} \bar{P} - (A+BK)^T\bar{P}(A+BK) - \theta^2 I & -(A+BK)^T\bar{P}(BK) \\ -(BK)^T\bar{P}(A+BK) & I - (BK)^T\bar{P}BK \end{bmatrix} > 0 \tag{4-12}$$
$$\bar{P} > 0.$$

This optimization problem has been solved in MATLAB using the LMI solver function "mincx". This resulted in $\theta = 0.071$. Since the results of the perturbed linear system approach have shown to be conservative in [11] the maximum value for σ is chosen: $\sigma = 0.071$. Since the event-triggering mechanism is designed using the assumption of full state feedback, the differentiation of the angle is performed at the same rate as the event-triggering mechanism which mimicks availability of the full state.

Fig. 4-2 shows the simulation results for the event-triggered controller for $\sigma = 0.071$. The angular velocity is obtained by backward differentiation and quantization effects are included. As can be seen in the fourth graph the event-triggering mechanism generates 88 events which are 45% less events than the periodic controller. Even though less control updates are occurring the controller still stabilizes the system.

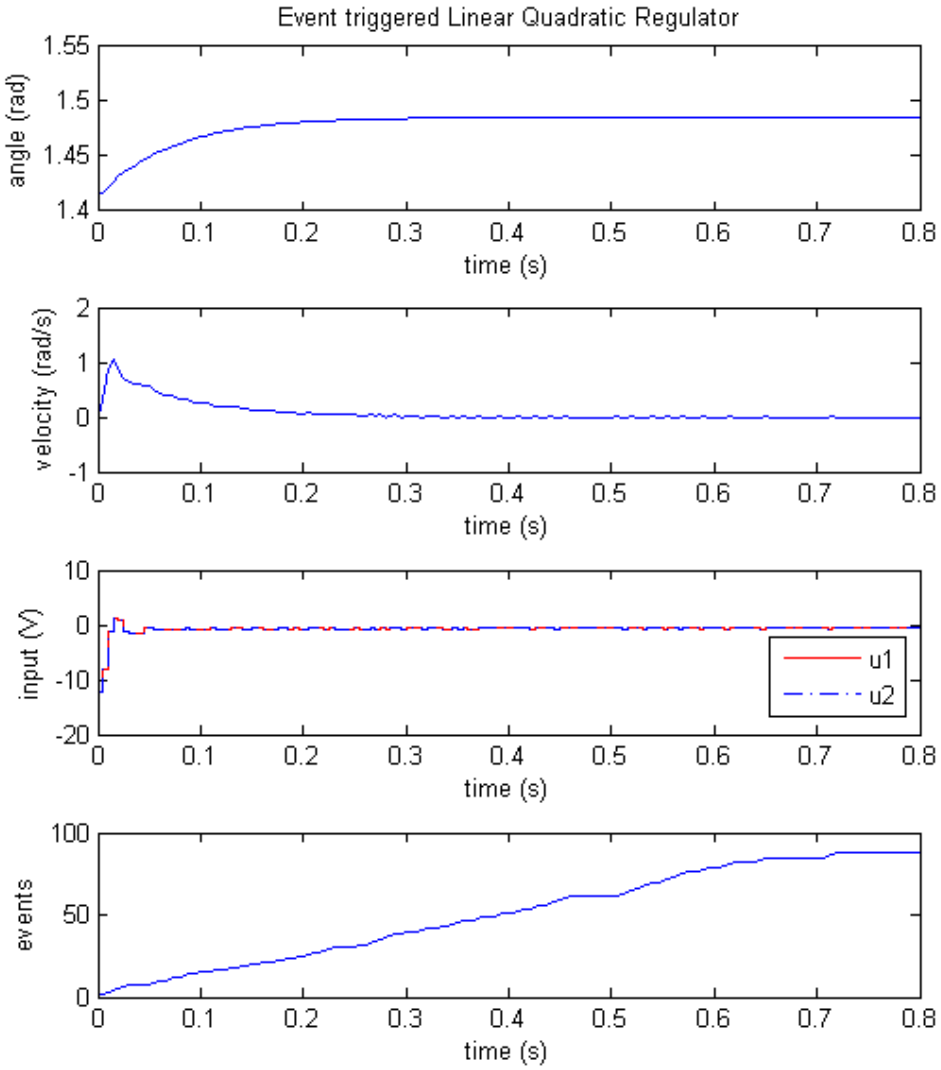


Figure 4-2: Simulation results for the event-triggered linear quadratic regulator.

Chapter 5

System Design

This chapter will discuss the design of the system that allows control and energy monitoring of the RQuad. This system implements the controllers designed in Chapter 4 and enables them to be compared in order to determine the effect of ETC on the energy consumption.

5-1 System overview

Fig. 5-1 gives an overview of the main hardware components of the system.

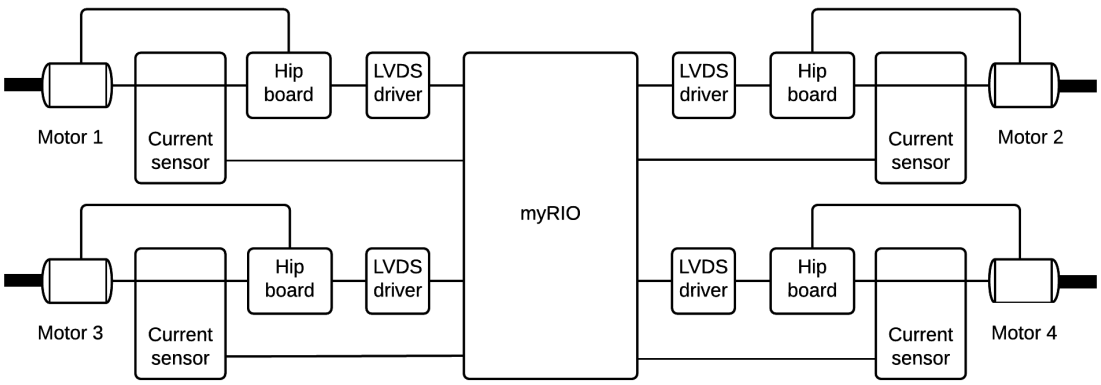


Figure 5-1: Schematic overview of the system

The embedded system that will be used to control the robot is the National Instruments myRIO-1900. The myRIO is a hardware/software platform designed to introduce students with the LabVIEW reconfigurable I/O (RIO) architecture [30]. The LabVIEW RIO Architecture consists of a processor, FPGA, inputs, outputs and design software that allows programming of the processor and FPGA graphically. The RIO architecture is schematically

depicted in Fig. 5-2 and allows the processor to interface with analog and digital in- or outputs by using the FPGA. Besides interfacing with I/O the FPGA can be used to offload the processor or to perform tasks which require precise timing. The myRIO contains an Xilinx Zynq-7010 all programmable system on a chip. This chip includes a dual-core ARM Cortex-A9 processor and an Artix-7 FPGA. The processor works at a clock rate of 667 MHz and runs NI Linux Real-Time OS.

The myRIO will be used to control the robot because the combination of a real-time OS and FPGA allows for real-time processing and control. Another important reason to use the myRIO is that the option to program it graphically using LabVIEW allows students with different levels of programming experience to easily make use of the robot in the future. This allows continuation of the research and further development of the system. A high level description of the software used for controlling and monitoring the RQuad will be given in Section 5-4

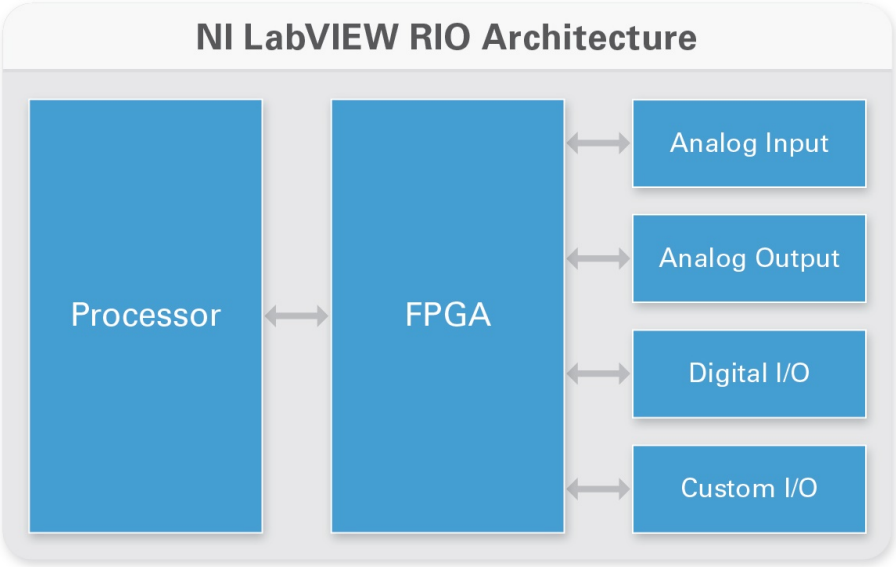


Figure 5-2: Schematical depiction of the LabVIEW RIO Architecture. Graphic provided by National Instruments (ni.com/myrio)

To control the motors of the RQuad, four motor controllers called Hip boards will be used. The Hip boards are developed at DCSC and are used for controlling the motors of the Zebro. The most important functions of the Hip boards are control of a motor using Pulse Width Modulation (PWM) and counting encoder pulses. Besides being used in the Zebro the Hip boards have been used in various other projects as well. Implementing drivers for the Hip boards in LabVIEW can be valuable for future projects where real-time control is required, or to potentially control the Zebro using LabVIEW in the future. In order to communicate with the Hip boards using the myRIO FPGA low voltage differential signalling drivers are needed (LVDS). The implementation of the Hip boards will be discussed in Section 5-2.

The final components in Fig. 5-1 are the current sensors. The current sensors are used for measuring the currents through the motors which is used to monitor the energy consumption of the robot. The design of the current sensors will be discussed in Section 5-3.

5-2 Motor controllers

5-2-1 Hip board

Fig. 5-3 shows the Hip board which is developed at DCSC and is capable of driving a motor with a maximum current of 30 A. Besides PWM control of a motor the hip board contains a temperature sensor, current sensor and 2 encoder pulse counters.

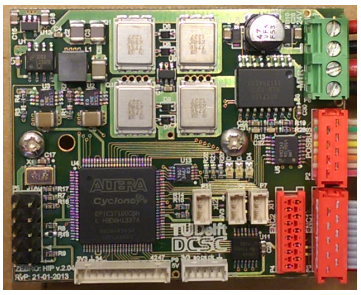


Figure 5-3: Picture of the Hip motorcontroller board developed at DCSC

In order to communicate with the Hip board a serial communication channel called Posbus is used. Posbus, developed at DCSC, allows communication between a master (myRIO) and a slave (Hip board). Three signals are used to communicate over the Posbus: SerialClock, SerialOut and SerialIn. SerialOut is used to transmit data from the master to the slave, SerialIn is used to transfer data from the slave to the master. The SerialClock signal is used for synchronization during transmission but is also used as the clock signal that drives the FPGA on the Hip board. All communication over the Posbus is initiated by the master and data is transmitted on the falling edge and captured on the rising edge of the SerialClock.

The master starts a transmission by making the SerialOut signal high for one period. After this start bit, all the bytes of the packet are transmitted least significant bit (LSB) first. After the complete packet has been transmitted, the data is processed on the Hip board once the SerialOut signal is made low by the master. When the master expects a reply, the slave will transmit its data after the master has finished transmitting its package and has made the SerialOut signal low.

The Hip board also contains an internal 48 MHz clock which is used to check whether or not there is still communication with the master. If the SerialClock signal has not changed for 50 clock ticks of the internal 48 MHz clock, the Hip board will reset itself.

Appendix C gives a description of the structure of the messages that are used in this thesis project to communicate with the Hip board. The most important data sent from the master to slave are: the PWM duty cycle, motor direction and a reset signal for the encoder pulse counters. The most important data sent from the slave to the master are the Hip board status and the encoder counts.

5-2-2 Implementation

The Hip board is designed to operate at a clock frequency of 48MHz and in order to communicate at this speed the FPGA of the myRIO is used. For programming on the FPGA, LabVIEW has developed a special structure called a single-cycle timed loop (SCTL). The SCTL has the property that all functionality placed inside the loop is guaranteed to execute within one clock period if the compilation succeeds. The SCTL is basically a while loop that executes every clock period.

A disadvantage of the SCTL is that it is executed once per period and no distinction is made between the rising and falling edge of the clock. Since data is transmitted on the falling edge and captured on the rising edge this poses an issue. To solve this problem a clock multiplier is used which can multiply the FPGA clock signal to create a clock signal of 80 MHz.

Fig. 5-4 shows how the 80 MHz clock is used to generate the 40 MHz clock signal. Within one 40 MHz clock period the code in the 80MHz SCTL executes twice. One period can be used as the high part of the 40 MHz clock (A) and the next period can be used as the low part of the 40 MHz clock (B). The start of each period of the 80 MHz clock is used as either a falling or rising edge of the 40 MHz clock.

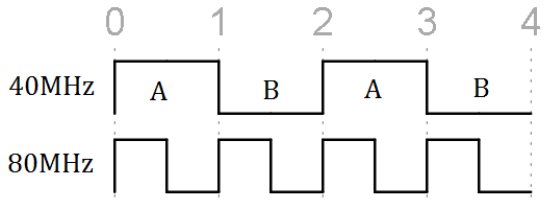


Figure 5-4: Clock signals that drive the communication with the hip boards.

The tasks performed in period A corresponding to the high part of 40MHz clock are:

- Make SerialClock high
- Read and buffer the input (SerialIn)
- Determine the next bit to transmit

The next bit to transfer is computed in clock cycle A. This allows it to be transmitted immediately when clock cycle B starts which corresponds with the falling edge of the 40MHz clock.

The tasks performed in period B corresponding to the low part of the 40MHz clock are:

- Make SerialClock low
- Output the data (SerialOut)
- Process the received SerialIn

In this period the bit that is calculated in period A is transmitted and the input received in period A is processed.

Figure 5-5 shows the block diagram describing the implementation of the drivers for the Hip boards on the FPGA of the myRIO. Each block represents a SCTL and the execution rate is indicated in the schematic. Furthermore, all blocks in the schematic execute in parallel. Since there are four Hip boards, one for each motor, four Posbus blocks are present on the FPGA. The rate at which the Posbus blocks communicate with the Hip boards is controlled by the Posbus control block that can trigger a transmission. This means that each time the Posbus Control block triggers a transmission the Posbus blocks send a message to, and receive a message from the Hip Boards. Currently the Posbus control block triggers transmissions at a rate of 25 kHz but a different rate can be selected by the processor. Every time a transmission is triggered the latest value written to a Posbus block is sent to the Hip board.

The Posbus blocks are implemented using LabVIEW functions inside the SCTL combined with combinational VHDL code. The VHDL code has been adapted from the Hip board source code and is used to serialize and deserialize the data.

The processor can write data to each Posbus block which is transmitted to the Hip board when the Posbus control block triggers a transmission. All messages sent to the Hip boards have a bit set that indicates to the Hip board that the master expects a reply. This means that in normal operation each transmission is followed by the reception of a message. The raw data received by the Posbus blocks is decoded in the Posbus control block and can be read out by the processor.

The Hip boards were originally designed to run at a clock speed of 48 MHz. However, communication at 48MHz was not reliable, therefore 40MHz communication is used. This is most likely caused by the current way of wiring where relatively long wires are used to allow for flexibility during the experimental implementation phase. The lower clock frequency lowers the PWM frequency from 22 kHz to 18.33 kHz.

Furthermore, the Hip boards are designed to transmit and receive low-voltage differential signals (LVDS). The circuitry used to convert the digital signals of the myRIO into LVDS signals is similar as the circuitry used on the Hip board. A schematic of this circuit is given in appendix C.

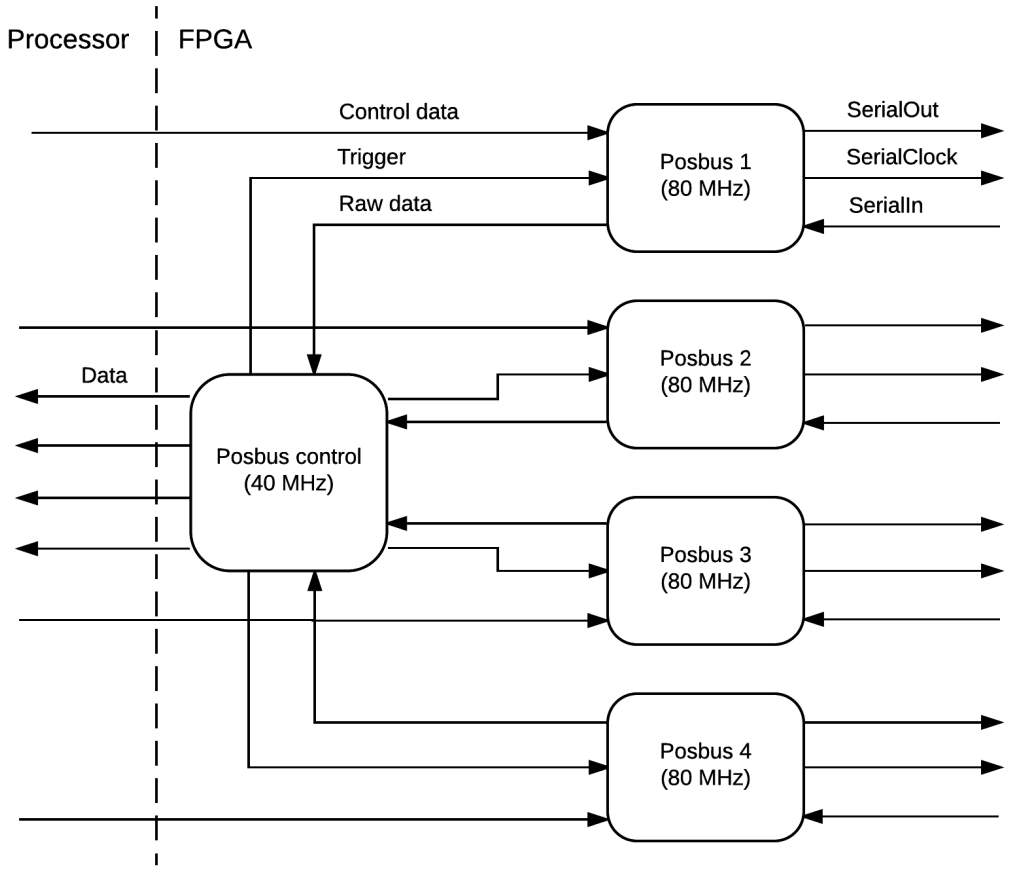


Figure 5-5: Block diagram of the implementation of the drivers for the Hip board motor controllers on the FPGA of the myRIO.

5-3 Power measurements

5-3-1 Current ripple

The DC motors of the RQuad are controlled using unimodal PWM by the Hip boards. To reduce the current ripple caused by the PWM scheme an extra coil is present in series with each DC motor. This extra inductance adds to the inductance of the windings of the motor and acts as a filter.

The total inductance of the combination of the coil and motor is given by

$$L = L_m + L_s \tag{5-1}$$

where the motor inductance $L_m = 0.985$ mH was taken from the datasheet [29] and the inductance of the extra coil, $L_s = 3.9$ mH, was measured with an LCR meter at a frequency of 10 kHz. This results in a total inductance of $L = 4.885$ mH. Furthermore, as already mentioned in Section 3-5 the resistance of the motor and coil combination equals $R = 6.23\ \Omega$.

This combination will act as a low pass filter with its cutoff frequency given by

$$f_c = \frac{R}{2\pi L} = 202.3\ Hz \tag{5-2}$$

With a rolloff of 20 dB/decade this will result in an attenuation of approximately 40 dB of the fundamental pwm frequency of 18.33 kHz.

By making use of the relation

$$V = L\frac{di}{dt} \tag{5-3}$$

the current ripple during steady state, when disregarding the resistance of the motor and coil is equal to

$$I_\Delta = \frac{V_{DC} - V}{L} D T_s = \frac{V}{L}(1 - D)T_s \tag{5-4}$$

where V_{DC} represents the PWM voltage (18 V), V represents the back EMF voltage, D is the PWM duty cycle and T_s is the PWM period. The resistance of the motor and coil is ignored since the PWM period is smaller than the electrical time constant ($T_s << \tau = \frac{L}{R}$) [31]. By then solving for V the following expression is obtained for the current ripple

$$I_\Delta = \frac{V_{dc}}{L} D(1 - D)T_s. \tag{5-5}$$

By dividing the ripple current by the average current (V_{avg}/R) the relative ripple error can be calculated and is shown in Fig. 5-6. The average PWM voltage V_{avg} is given by

$$V_{avg} = DV_{DC} \tag{5-6}$$

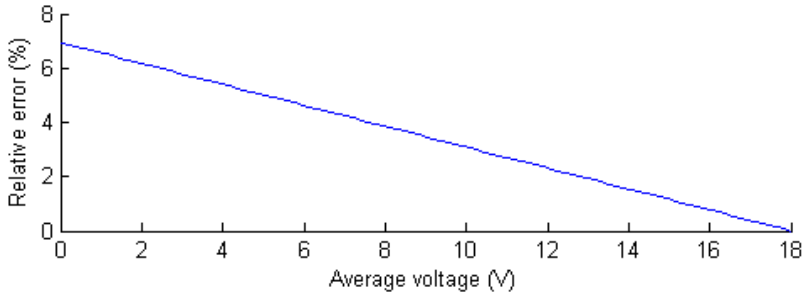


Figure 5-6: Relative current ripple error for the average PWM voltage

Even though the maximum current ripple occurs at 50% duty cycle, the maximum relative ripple error occurs at low duty cycles and is approximately 7%. Due to the filtered nature of the current caused by the inductance of the motor and the additional coil, the following equation will be used to calculate the power

$$P = V_{avg} I. \tag{5-7}$$

In (5-7) I is the measured current and V_{avg} is the average PWM voltage according to (5-6) which can be calculated from the duty cycle. The energy consumption can then by obtained by numerically integrating the power.

5-3-2 Current sensor

Each Hip board contains an Allegro ACS710 hall-effect based current sensor. The sensing circuitry on the Hip boards has been designed for measuring currents up to 30 A. The sensitivity of the current sensor is 56 mV/A and its output is connected to a 12 bit 3.3 V ADC. For the initial conditions of the simulations in Chapter 4 the maximum current will be lower than 1 A because the maximum voltage applied to each motor is lower than 6 V. For this maximum current the sensor on the Hip board will output just 56 mV. The resolution of approximately 14.4 mA will be too low for small control inputs, therefore new sensors will be designed that are more suitable for the RQuad.

Each Hip board uses an H-bridge to control the motors. A schematic of this H-bridge is shown in Fig. 5-7. Three common approaches for sensing the current through the motors are low side, high side and directly on the motor. For high side current sensing the current is measured that flows from the supply V_{DC} to the H-bridge. For low side current sensing the current flowing from the H-bridge to the ground is measured. Finally, with direct sensing on the motor, the current flowing through the motor or between terminals A and B of Fig. 5-7 is sensed. Since the H-bridge is part of the PCB, the only option is to perform direct sensing on the motor and measure the current between terminals A and B.

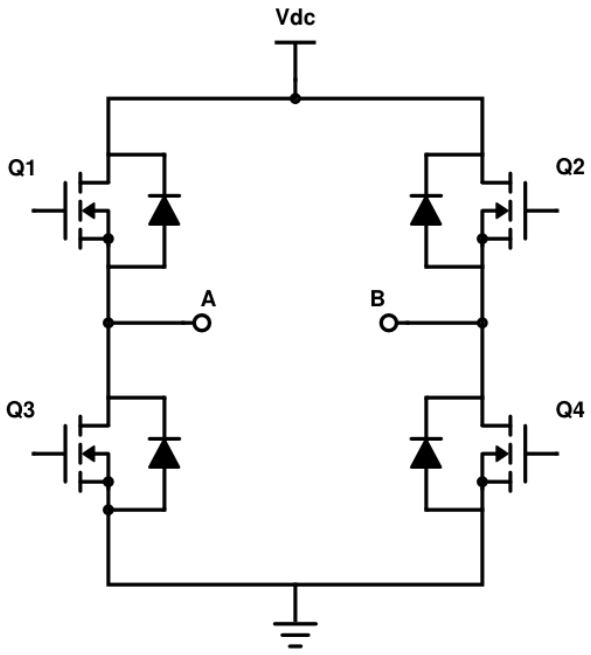


Figure 5-7: Schematic of the H-bridge used on the Hip board to control the motors

To measure the current a shunt resistor will be used. This method is based on Ohm's law and is used for measuring low currents where indirect techniques (e.g. Hall effect sensors) are commonly used for higher currents ($> 5\ A$). By placing a shunt resistor in series with the load a voltage drop over the resistor will be present that is directly proportional to the load current. This voltage can then be amplified using a difference amplifier. A disadvantage of this method is that the shunt resistor dissipates power and influences the amount of current flowing through the circuit. Therefore, a small shunt resistor compared to the load resistance should be selected.

Fig. 5-8 shows how a current sense resistor can be placed in series with a motor in order to measure the motor current. It is important to note that in the motor symbol represents the DC motor and the additional coil in series.

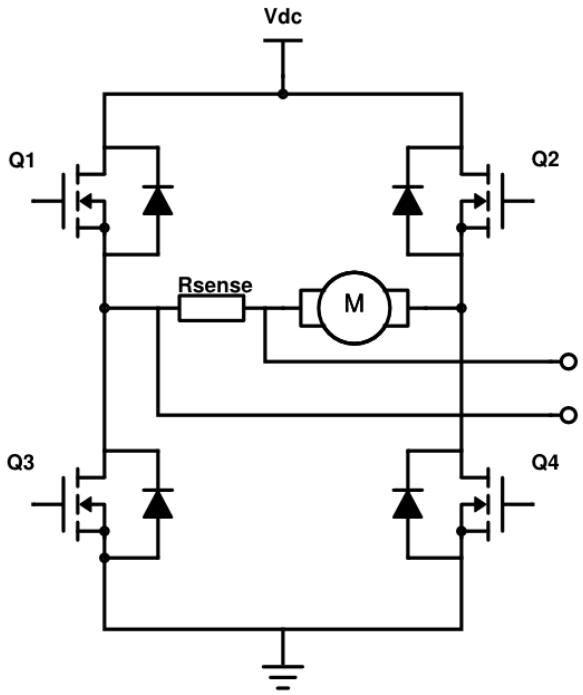


Figure 5-8: Schematic of the placement of the current sense resistor in series with the DC motor.

A current shunt monitor is used to amplify the voltage over the current sense resistor. Due to the switching nature of the output signal the amplifier must be able to withstand common-mode voltages from the ground up to the supply voltage (V_{DC}). To amplify the differential signal the INA282 current shunt monitor has been chosen [32]. This sensor is selected since it has, to the authors best knowledge, the widest common mode range while having the lowest input offset error of max $70\mu V$. The gain of the current shunt monitor is 50. The most significant parameters of the current shunt monitor can be found in Appendix B.

The most important factors in deciding what resistor value to use are the input offset error and the maximum current to be measured. A higher resistance allows lower currents to be measured more accurately while a lower resistance allows larger currents to be measured. Based on the equilibrium input (u_{eq}) and the maximum control input for the simulations in Chapter 4, the range for which the current sensors will be designed is chosen to be 50-1000 mA. If for other applications higher currents need to be measured, the current sensors embedded on the Hip boards could be used.

The outputs of the current sensors will be connected to the 12-bit ADCs of the myRIO which accept input voltages between 0-5 V. Since the INA282 will be used for bidirectional current sensing, the sensor will output 2.5 V when no current is being measured. For a maximum current of 1 A the maximum allowable resistance value can be calculated as follows:

$$R_{max} = \frac{2.5}{GI_{max}} \tag{5-8}$$

where G is the gain of the current shunt monitor and I_{max} is the maximum current. This results in a maximal allowable resistor value of 50 $m\Omega$. The most dominant error term when

measuring low currents is the input offset voltage. For a maximum input offset of 70 μV the error for a current of 50 mA and a sense resistor of 50 $m\Omega$ is given by

$$e_{V_{os}} = \frac{V_{os}}{I_{load} R_{sense}} = 2.8\% \tag{5-9}$$

Fig. 5-9 shows the total relative error in the current measurement. The operating conditions along with details of the error calculation can be found in appendix B. It can be seen that the typical error is lower that 5% from approximately 9.2 mA upwards and the maximum error value is lower than 5% from 32.5mA and upwards.

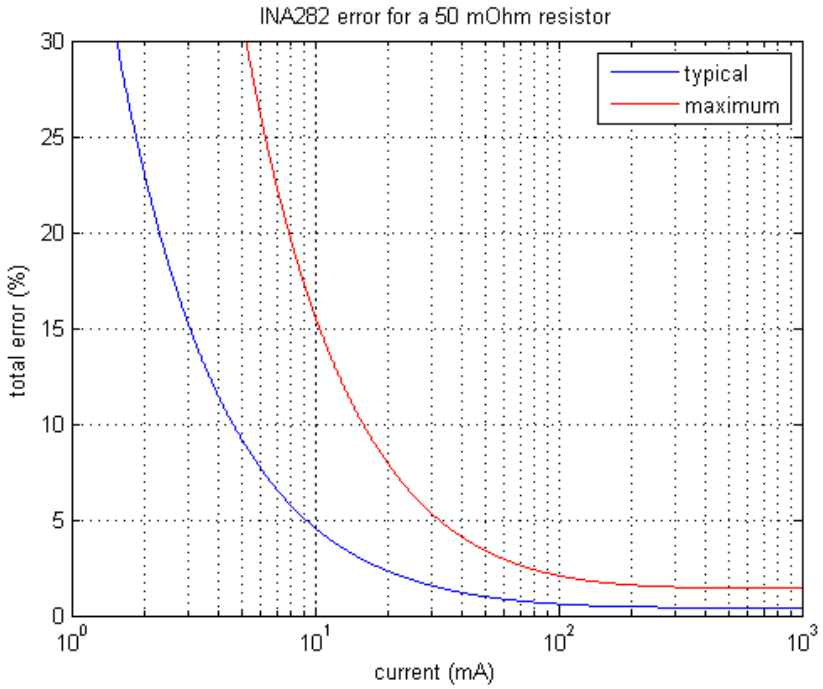


Figure 5-9: Total relative error in the current measurement for the INA282 current shunt monitor

Fig. 5-10 shows the circuit diagram illustrating how the INA282 is used for bidirectional current sensing. The terminals of the shunt resistor will be connected to pins +IN and -IN. The supply voltage is 5 V and is supplied by the myRIO. By connecting one of the two reference pins to V+ and the other to ground the output is set at half of the supply voltage.

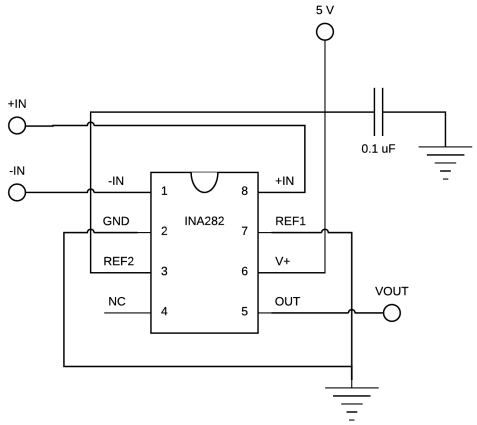


Figure 5-10: INA282 current shunt monitor with the power supply split over the reference pins

The outputs of the current shunt monitors are sampled and passed through a 5 point median filter at a rate of 90 kHz. The median filter is used to prevent samples being obtained at switching instants of the H-bridge to prevent short switching transients to influence the measurements. The median filtered values are then filtered at a rate of 18 KHz by a 4th order butterworth low-pass filter with a cutoff frequency of 200 Hz. All filtering is performed by the FPGA.

5-4 Software Architecture

5-4-1 State machine

The main behavior of the system can be described by the state diagram shown in Fig. 5-11.

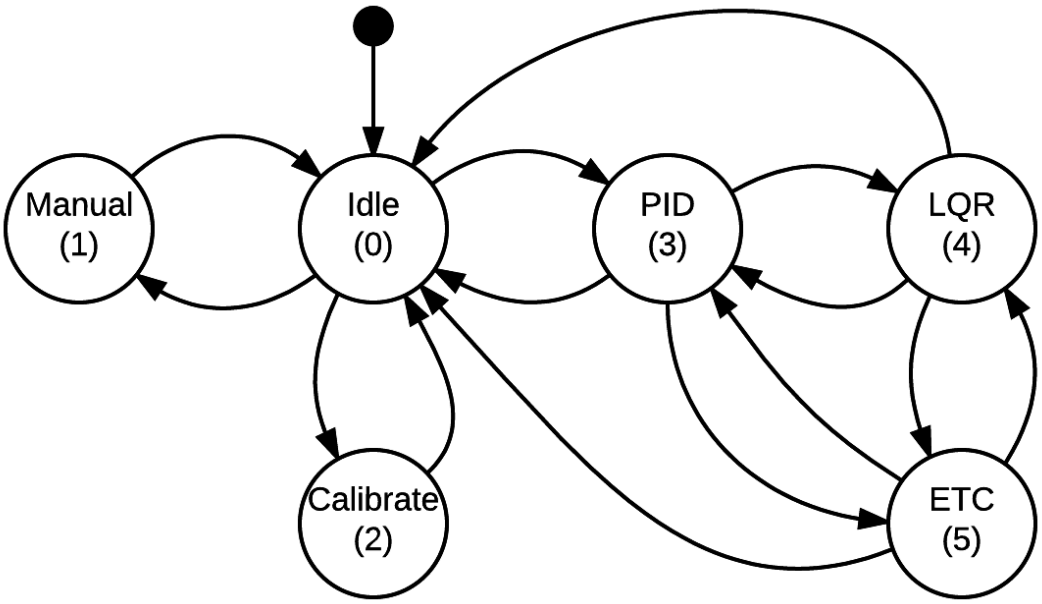


Figure 5-11: State diagram for the RQuad

The current states are mainly implemented for use in this thesis project. However the functionality can be easily extendend or adapted by adding or removing states.

Idle

The system always starts in the *Idle* state. In this state all the motors are turned off and the system is safe. Besides being the starting state of the robot it is also the desired end state. From *Idle* it is possible to perform a switch to the *Manual*, *Calibrate* or the *PID* state. Although not indicated in the diagram, the switch to *PID* is only possible when the system is calibrated. From any state a switch to the safe *Idle* state can be made. Furthermore, when an error is received from the Hip boards or when the connection seems lost, a switch to *Idle* is always performed to ensure safety.

Manual

When the system is in the *Manual* state all the Hip boards and thus the motors can be controlled individually by the user. From *Manual* it is only possible to transition to the safe

Idle state.

Calibrate

In the *Calibrate* state the controller applies a low voltage to the motors for six seconds after which a command is send to the Hip board to reset the encoder pulse counters. This low voltage allows all four of the legs to rotate until they hit a surface. Within six seconds all legs should be able to rotate and hit the surface. After this process the encoders are calibrated. A depiction of the calibration process is given in Fig. 5-12. After the system is calibrated a switch is automatically performed back to the *Idle* state.

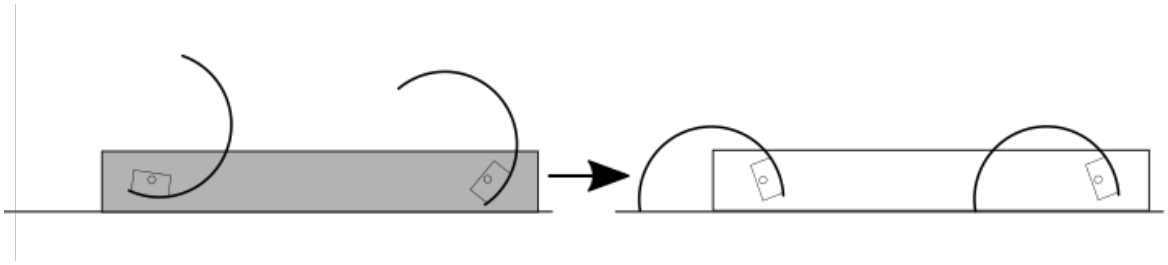


Figure 5-12: Side view of the calibration process. The grey image indicates the uncalibrated robot and the white image the calibration point where all legs touch the surface.

PID

In the *PID* state all four legs are individually controlled using PD controllers. This state is used to have the robot stand up in a position close to the point for which the system model derived in Chapter 3 is linearized. From *PID* a switch can be made back to *Idle* or a switch can be made to the *LQR* or *ETC* state.

LQR

In the *LQR* state the robot is controlled using the LQR controller designed in Section 4-1. From *LQR* it is possible to switch to *Idle*, *ETC* or *PID*.

ETC

In the ETC state the robot is controlled using the event-triggered controller as designed in Section 4-2. This state is similar to *LQR* except an event-triggering mechanism is used to determine when a control update should take place. From the *ETC* state a switch can be made to *Idle*, *LQR* or *PID*.

5-4-2 Tasks

Each task that is running on the myRIO is implemented using a LabVIEW structure called a timed loop. The functions inside a timed loop execute periodically at a specified period. Fig. 5-13 gives an overview of all the tasks, their execution rate and their relations with other tasks. The relative priorities of each task is shown in Fig. 5-14. The control task runs at the highest priority and the display task runs at the lowest priority. To transfer data between tasks in a nonblocking deterministic way, LabVIEW single element real-time FIFOs are used. Multiple readers of this FIFO receive the same value and the FIFO returns the same value to readers until a new value is written to the FIFO.

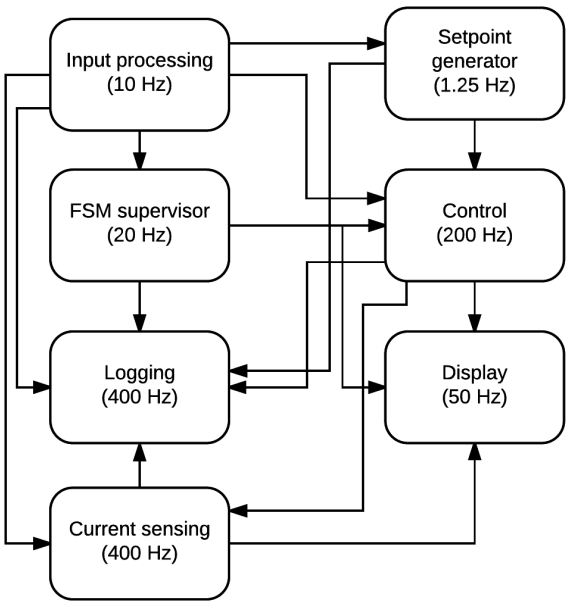


Figure 5-13: Tasks running on the processor of the myRIO.

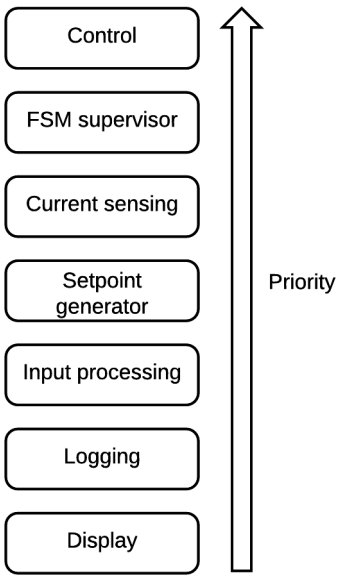


Figure 5-14: Relative priorities of the tasks running on the processor of the myRIO

FSM supervisor

The FSM supervisor is responsible for keeping track of the system state and performing state switches. This task enforces the state diagram as discussed in Section 5-4-1. It checks whether the state switch selected by the user is valid and performs the switch if allowed. Furthermore, it checks whether or not errors are received from the Hip board to transition to the *Idle* state if needed.

Control task

The control task is responsible for control of the RQuads motors by writing data to the FPGA which will send the commands to the Hip boards. Depending on the system state different

control functionality is implemented. In the states *PID*, *LQR* and *ETC*, feedback control is implemented and the control loop takes care of reading and processing the motor encoder inputs, calculating the control input and passing them to the FPGA which sends a control message to the motors. The control tasks uses the setpoint provided by the setpoint generator and different control parameters such as PID gains can be changed through the user input task. The control task will be further discussed in Section 5-4-3.

Current sensing

The current sensing tasks takes care of measuring the current. It uses the filtered ADC measurements of the FPGA to calculate the current. To calculate the power the control input for each motor is multiplied with the measured current. This power is then integrated using a LabVIEW point by point integrator block, which uses Simpson's method, in order to monitor the energy. Furthermore, through the input processing task the user can calibrate the current sensors. Since the FPGA low-pass filters the current measurements with a cutoff frequency of 200 Hz the current sensing task runs at 400 Hz to prevent aliasing.

Setpoint generator

The setpoint generator task takes care of generating the setpoints for the controller. The user can select different setpoints through the input processing task or can start different setpoints sequences. These setpoint sequences are used for the experiments which will be discussed in Section 6-1.

Logging

The logging task is implemented in order to log relevant experiment data to a usb stick plugged into the myRIO. This data can be used to visualize and analyze the results after the experiments have been performed. The data is logged as a comma separated text file which can be easily analyzed using Matlab. The user can start logging manually or the setpoint generator can start the logging during experiments. The logger mainly logs data from the control task and the current sensing task.

Input processing

The input processing task is used to read and process user input. The most important user inputs are the desired new FSM state, controller parameters and commands to start or stop the logging of data. These commands are given through the host PC which is connected to the myRIO over USB.

Display

The display task is used to display basic relevant information to the user. The most important data that is being visualized are the FSM state, system state (x), control input, ETC events, measured currents and Hip board data. This data is displayed on the host PC which is connected to the myRIO over USB.

5-4-3 Controller implementation

Fig. 5-15 shows a block diagram of the controller when the system is in the LQR state. This control loop is implemented by the control task running on the processor of the myRIO and is executed at a rate of 200 Hz.

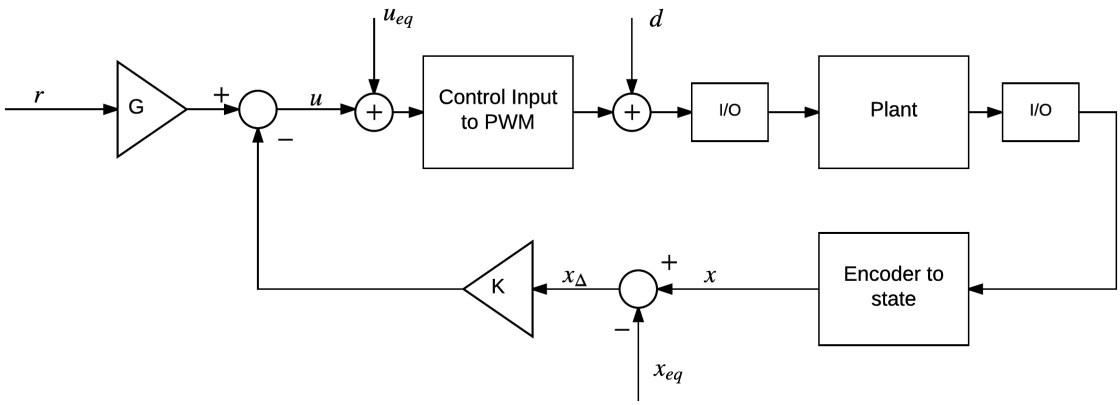


Figure 5-15: Block diagram illustrating the periodic control loop.

Each execution of the control tasks starts with reading out the most recent encoder data from the FPGA. This encoder data is then used to calculate the system state as follows

$$x_1(k) = \pi - \frac{(enc_2 + enc_{\frac{\pi}{2}} - enc_{up}) + (-enc_1 + enc_{\frac{\pi}{2}} - enc_{up})}{2} \frac{2\pi}{N} \tag{5-10}$$

where enc_1 and enc_2 are the encoder pulse counts for motor 1 and motor 2 respectively. The number of counts per rotation is $N = 70000$, the encoder counts corresponding with $\frac{\pi}{2}$ is given by $enc_{\frac{\pi}{2}} = 17500$. The encoder value corresponding with the legs being in a right angle with respect to the robots body, after the calibration procedure, is given by $enc_{up} = 56300$.

The second state is computed as follows

$$x_2(k) = \frac{x_1(k) - x_1(k-1)}{h}. \tag{5-11}$$

After computing the state $x(k)$, the state for the linear system, $x_\Delta(k)$, is obtained by subtracting the equilibrium state. After the control inputs are computed they are converted into PWM duty cycles. Since there are two control inputs, one for each leg pair, the control input is split in half and the PWM duty cycle is computed for each motor. Furthermore, the sign of the control input u is used to calculate the appropriate motor direction. Optionally, a disturbance on the control input can be added which can be used for experiments. After converting the control input the values are written to the FPGA which sends them to the Hip boards.

When the system is in the ETC state the control loop shown in Fig. 5-16 is used. There are two main differences between the control loop for the ETC state and the LQR state. The first difference is that the event-triggering mechanism is evaluated after the state is computed.

The event-triggering mechanism checks whether a control update is needed by evaluating (4-11). When an event occurs the control input is calculated, converted to PWM and written to the FPGA. When no event occurs all these steps are omitted and the FPGA retains its old control inputs. The event-triggering mechanism is currently implemented on the processor, however it would be better to implement it on the FPGA which will help to truly offload the processor and save more computation time. An implementation on the processor allows easier adaptation and testing of the triggering-mechanism. This is the reason why it is currently not implemented on the FPGA, since every change of the code requires a compilation. The second difference between the control loop of the LQR and ETC state is that no reference signal can be used since the design of the event-triggering mechanism following [11] is applicable for when the state is converging to the origin.

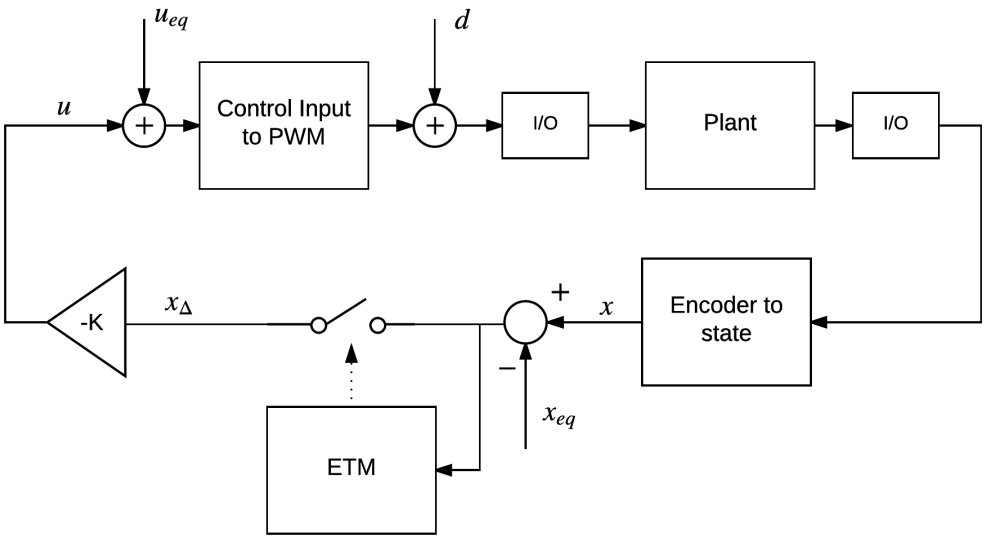


Figure 5-16: Block diagram illustrating the event-triggered control loop.

5-4-4 Profiling results

Table 5-1 shows the profiling results for the tasks discussed in Section 5-4-2. The maximum time for each task was obtained by measuring the maximum execution time that occurred within 60 seconds. Furthermore, the mean execution times were calculated by the display task by averaging the execution times of 3000 executions. For tasks running at a rate lower than the display task, the mean execution times were calculated by averaging the times of the executions that occur within 60 seconds. During profiling of a task it was temporarily assigned the highest priority. Since file I/O is non-deterministic only the average time of the logging task is reported.

The execution time for the control task has been measured for different states of the system. The part of the control task that is spent on reading the encoders, calculating the control input and writing the control input to the FPGA is also reported for the control task during the LQR and ETC states. These times are indicated in Table 5-1 as S2A (sensors to actuators). The execution times presented for the event-triggered controller represent the worst case where both the event-triggering mechanism is invoked and a control update is triggered every time.

Table 5-1: Profiling results of the tasks running on the processor of the myRIO.

Task	mean time (μs)	Max time (μs)	Frequency (Hz)	Worst case utilization (%)
Controller:				
Idle	341.27	468	200	9.36
PID	378.25	550	200	11.00
LQR	501.47	773	200	15.46
-S2A	*315.9*	*552*	-	-
ETC	557.84	776	200	**15.52**
-S2A	*368.75*	*602*	-	-
FSM super.	181.44	233	20	**0.47**
Current sens.	266.26	381	400	**15.24**
Setpoint gen.	186.45	275	1.25	**0.034**
Input proc.	365.29	482	10	**0.48**
Logging	655.95	-	400	**26.24**
Display	440.18	686	50	**3.43**
Worst case:				**61.41**

Chapter 6

Experiments

In this chapter the experiments and their results will be discussed in order to determine how event-triggered control impacts the energy consumption. First, the experiments used to compare periodic and event-triggered control will be discussed followed by a presentation and analysis of the obtained results. Finally, this chapter finishes with a discussion of the results and their implications.

6-1 Experiment design

When no disturbances are present and when the system is in steady state both the periodic and event-triggered controller will apply the same control input. Except the difference in the number of events, there will be no difference in terms of actuation energy consumption. To compare ETC and periodic control in terms of energy consumption the measurements should be performed during transients.

Three experiments will be conducted that allow periodic and event-triggered control to be compared during transients. Each experiment will be performed for periodic and ETC where the periodic results will serve as a base case for the comparison. Besides using the event-triggering mechanism designed in Section 4-2 with $\sigma = 0.071$, the experiments will also be performed for $\sigma = 0.100$ to see the effects of a less conservative triggering mechanism. However, the main analysis will focus on the formally designed event-triggering mechanism with $\sigma = 0.071$.

Experiment 1 and 2

In experiment 1 and 2 the system will be put in a transient state by starting the experiment from nonzero initial conditions. From these initial conditions the controllers will steer the system towards the equilibrium state. During this resulting transient the energy consumption will be measured. Both experiments are schematically depicted in Fig. 6-1. In experiment 1,

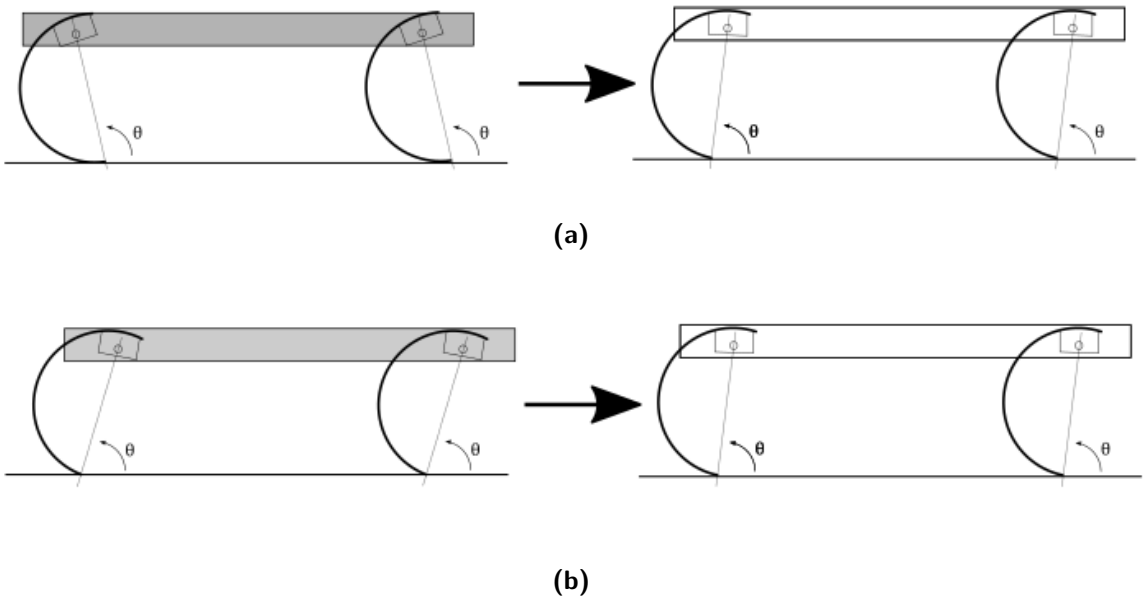


Figure 6-1: Schematical depiction of experiment 1 and 2 where the experiment starts from nonzero initial conditions (grey image) and the controller has to steer the system towards the equilibrium state. (a) Experiment 1: the robot starts out from $\theta = x_{eq1} + 0.07$ rad. (b) Experiment 2: the robot starts out from $\theta = x_{eq1} - 0.07$ rad.

depicted in Fig. 6-1a, the system will start from an initial condition 0.07 rad (4°) away from its equilibrium angle. In experiment 2, depicted in Fig. 6-1b the system will start -0.07 rad from its equilibrium angle.

Fig. 6-2 shows the flow chart illustrating how a sequence of consecutive runs for experiment 1 and 2 are performed. After starting the experiment the robot is brought into the required starting position by setting the appropriate reference signal. When the robot is in the required starting position the reference signal is set to zero and energy measurements will be performed during the resulting transient. After the system is again in equilibrium the energy measurements are stopped and the experiment will be repeated until the required number of experiments are performed. The event-triggering mechanism is not designed for reference tracking, therefore during ETC the system will be brought to the initial starting state by forcing events which will result in the same behavior as the periodic controller.

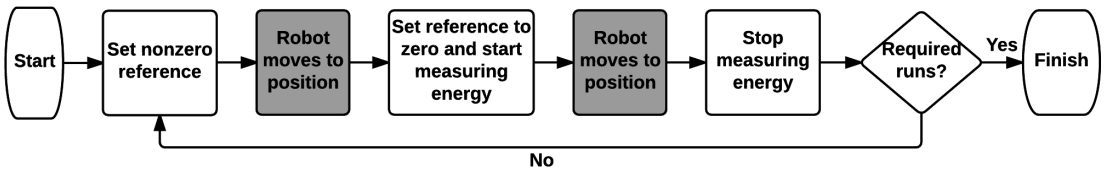


Figure 6-2: Flow chart illustrating how a sequence of experimental results is obtained for experiment 1 and 2.

Experiment 3

In experiment 3 the robot will be brought into a transient state by applying a disturbance pulse on the control input of its back leg pair. This experiment is interesting since it will give insight in how the controllers responds to a disturbance when the state is close to the equilibrium. The system starts out in its equilibrium state and the energy will be measured from when the disturbance is applied until the system is again in steady state. Since the disturbance input affects the energy consumption of the motors of the back leg pair, only the energy consumption of the motors of the front leg pair will be used for the comparisons.

For all three experiments the robot will be brought to the equilibrium state using the PID controller after which a switch to either the LQR or ETC state is made. Furthermore, in order to compare the performance of the controllers during experiments the Root Mean Square Error (RMSE) will be used given by (6-1).

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(x_{eq1} - x_1)^2} \tag{6-1}$$

6-2 Results experiment 1

For each controller seven sequences of 20 runs of experiment 1 have been performed, resulting in a total 140 runs per controller. Fig. 6-3 shows the evolution of the state, the control input and the energy consumption for a sequence of experiments obtained for the periodic controller. The figure is shown to illustrate what a sequence of runs of an experiment looks like. First, the system is brought into the starting position after which the reference signal is set to zero. The energy measurements are performed only during the transient when the controller is steering the system from the nonzero initial conditions towards the equilibrium state.

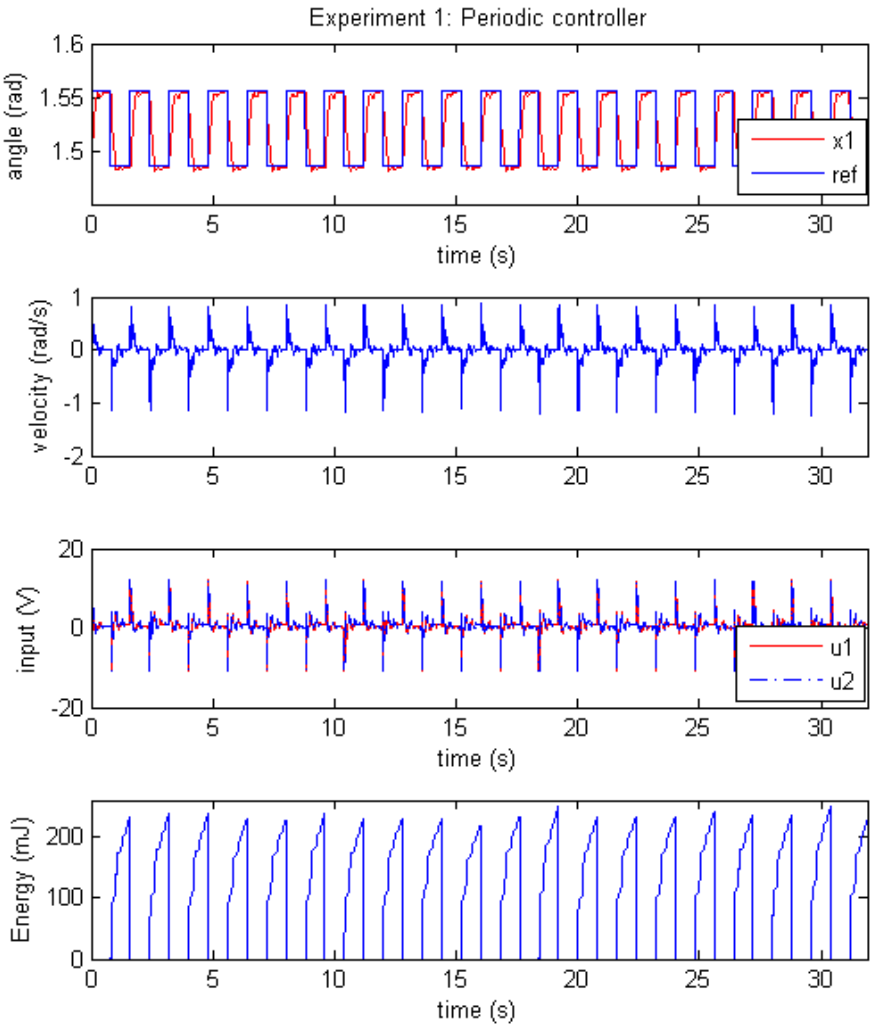


Figure 6-3: Results for a sequence of 20 consecutive executions of experiment 1 for the periodic controller.

Fig 6-4 shows the number of events and the total energy consumption for each run of experiment 1. The periodic controller has 160 events during each run which is equal to the number of periods that occur during the experiment time. The number of events generated by the event-triggered controllers during the experiments are both clustered around approximately 100. The energy consumed during periodic control is spread around the average of 234.79 mJ and the energy consumed during event-triggered control is spread around 241.85 mJ and 248.19 mJ for $\sigma = 0.071$ and $\sigma = 0.100$ respectively. Histograms showing the distribution of the energy usage during experiment 1 can be found in Appendix D-1.

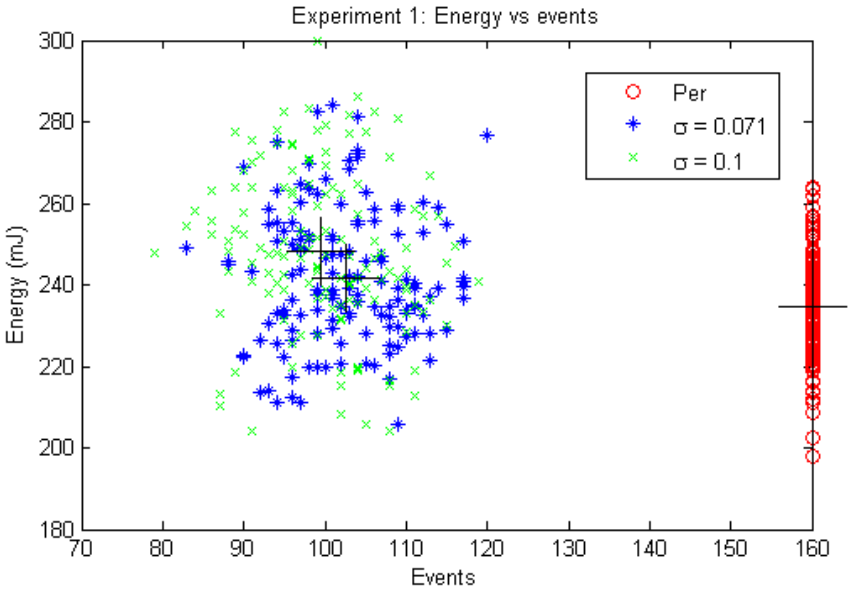


Figure 6-4: Total energy consumption and number of events for each of the 140 runs of experiment 1.

The results of experiment 1 are summarized in Table 6-1. Both event-triggered controllers reduce the number of control updates while having an increased energy consumption compared to the periodic controller.

Table 6-1: Average results of 140 runs of experiment 1. The bold entries correspond with the results obtained during periodic control which is used as a base case for the comparison with event-triggered control. In the last column SS represents the steady state error.

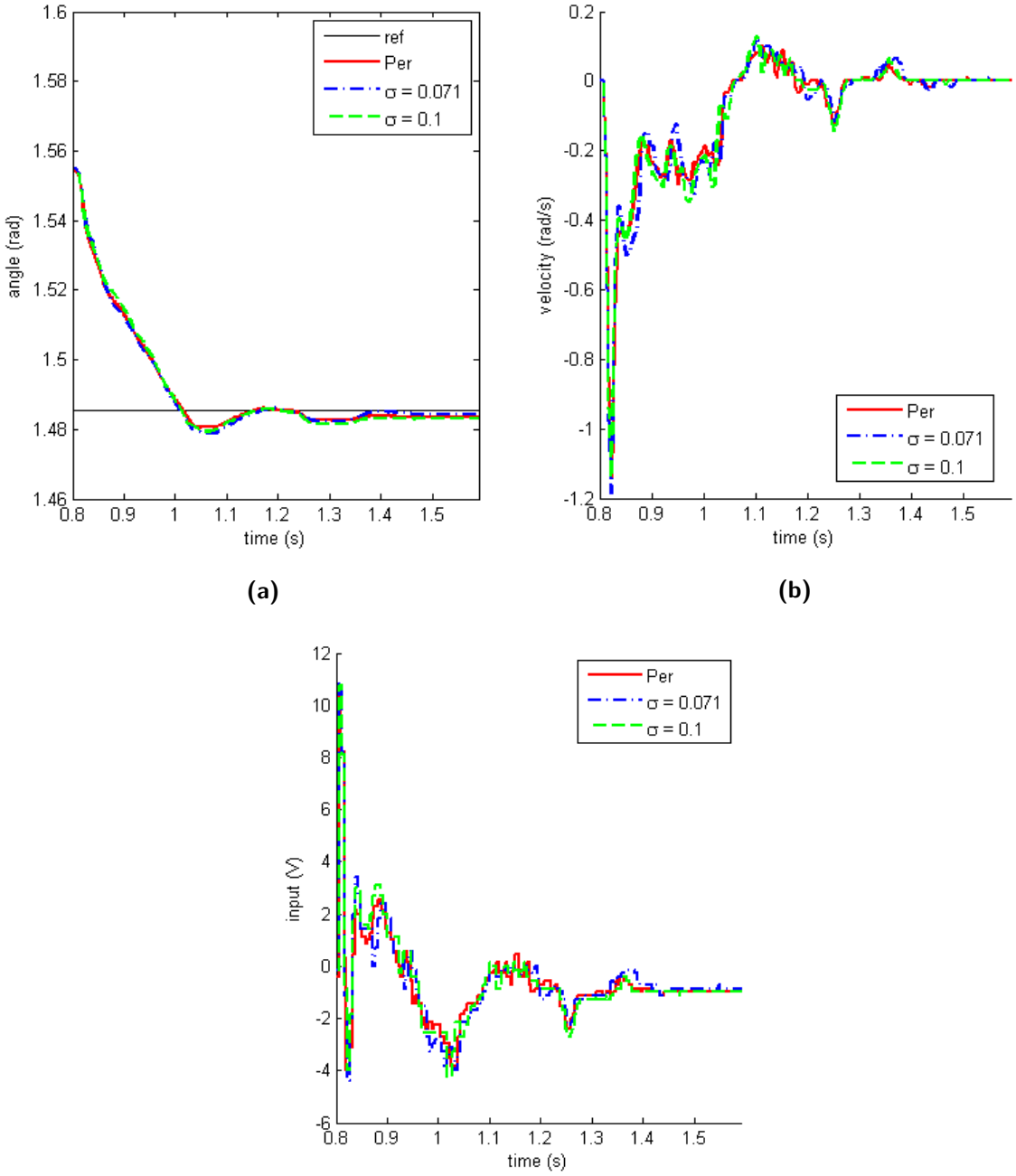
Contr.	σ	Events	Energy (mJ)		std. dev. (mJ)	RMSE ($\cdot 10^{-3}$ rad)	max abs. overshoot ($\cdot 10^{-3}$ rad)	abs. SS err. ($\cdot 10^{-3}$ rad)
Per	-	**160**	**234.79**		**12.25**	**18.1**	**4.8**	**0.89**
ETC	0.071	102.5	241.85	+3.0%	16.61	18.1	5.5	0.65
ETC	0.100	99.5	248.19	+5.7%	19.51	18.2	5.9	0.51

The amount of information that can be extracted from plots showing full experiment sequences such as Fig. 6-3 is limited. In order to analyze why differences in the average results occur, plots showing a single run of the experiment for each controller will be used.

The evolution of the states and the control input for one run of the experiment is shown in Fig. 6-5. The evolution of the first state shows that overshoot occurs when the controller steers the system towards the equilibrium state. This overshoot is then followed by a decaying oscillation. When the system is approaching the equilibrium state the absolute velocity starts to decrease between 0.85-0.9 s. When the velocity decreases, the backlash in the motor/gearbox combination causes the robot to shoot through. The combination of the backlash and the compliancy of the legs seem to cause this oscillation. The control input shown in Fig. 6-5c shows that the controller responds to the overshoot and steers the system back towards the equilibrium state.

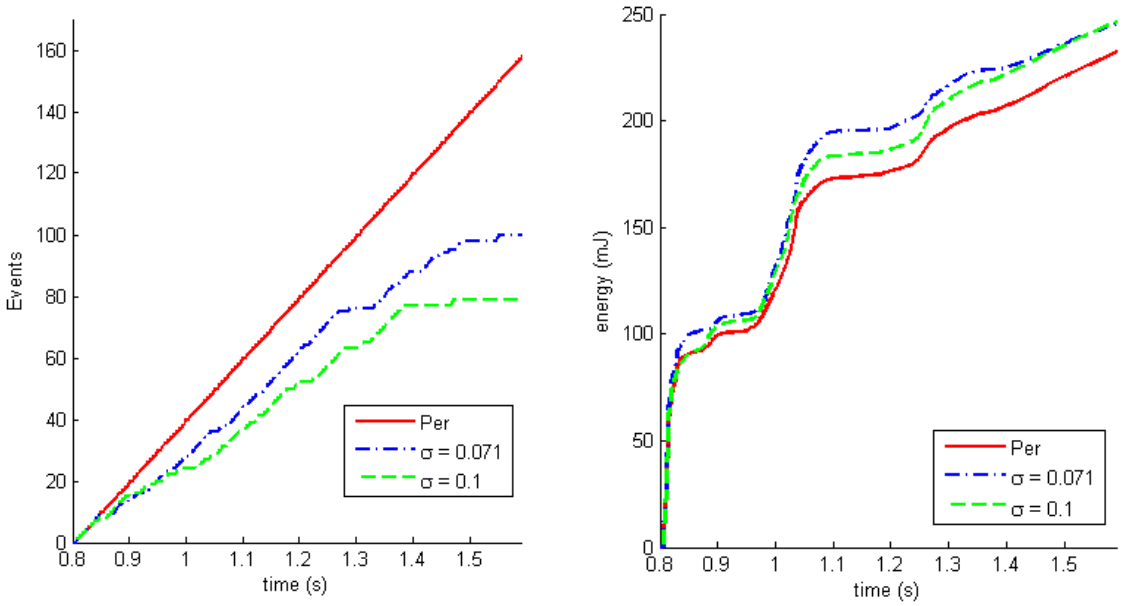
Fig. 6-5 shows the number of events and the energy consumption during the same experiment run as the results shown in Fig. 6-6. During the first 50 ms of the experiment similar behavior occurs for all three controllers since the number of generated events roughly equals the number of periods. After the first part of the experiment both event-triggered controllers start generating events at a lower rate than the rate at which periods occur. The energy consumption for the three controllers, shown in Fig. 6-6b, has slight differences. The first difference occurs between 0.85-0.9 s and seems to correspond with the differences in the initial conditions of the experiment, where the experiment for the controller with $\sigma = 0.071$ starts slightly further from the equilibrium angle, followed by the controller with $\sigma = 0.100$ and finally the periodic controller starts closest to the equilibrium angle.

The biggest difference in energy consumption occurs during the overshoot between 1-1.1 s shown in Fig. 6-5a. The overshoot for both event-triggered controllers is higher which results in higher absolute control inputs and more control effort required to steer the system back towards the equilibrium state. It seems that when the system is moving towards the equilibrium state, a reduction in the number of control updates causes higher absolute control inputs to be applied for longer periods of time. This results in a higher angular velocity which helps explain why higher overshoot occurs during ETC. Table 6-3 shows that on average the overshoot during of experiment 1 is indeed higher during ETC and seems to be the main reason for the difference in the average energy consumption.



(a) **(b)** **(c)**

Figure 6-5: Results of one run of experiment 1 for each of the three controllers. (a) Evolution of the first state $x_1 = \theta$. (b) Evolution of the second state $x_2 = \dot{\theta}$. (c) Control input where only one of the two control inputs (u_1) for each controller is shown since they are equal ($u_1 = u_2$)



(a) **(b)**

Figure 6-6: Events and energy for one run of experiment 1. (a) Evolution of the number of events. (b) Evolution of the energy.

6-3 Results experiment 2

The number of events and total energy consumption during each run of experiment 2 is shown in Fig. 6-7. Just as in experiment 1 the results are obtained by performing seven sequences of 20 runs. The number of events for the event-triggered controllers with $\sigma = 0.071$ and $\sigma = 0.100$ are clustered around approximately 56 and 52 respectively. The energy consumed during periodic control is spread around 502.90 mJ and the energy consumed during event-triggered control is spread around an average of 486.71 mJ and 477.15 mJ for $\sigma = 0.071$ and $\sigma = 0.100$ respectively. Histograms showing the distribution of the energy usage during experiment 2 can be found in Appendix D-2.

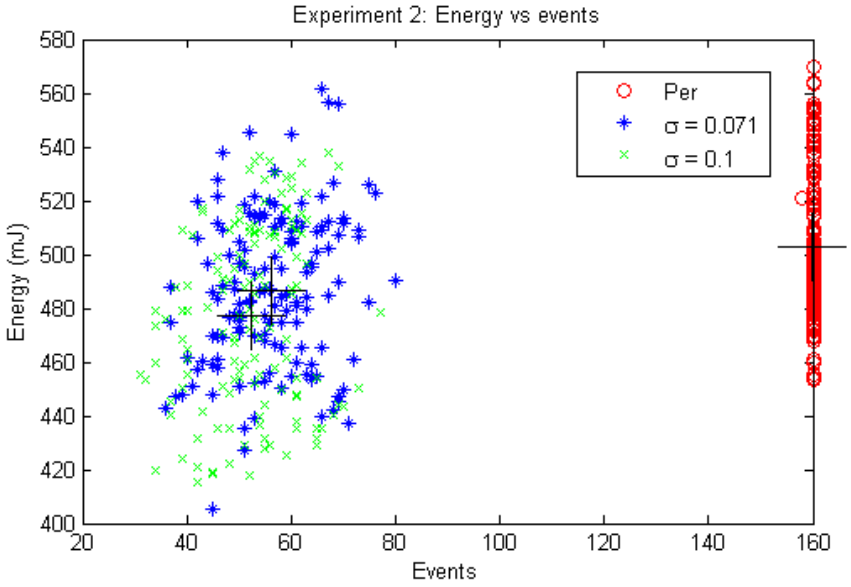


Figure 6-7: Energy consumption and number of events for each of the 140 runs of experiment 2.

The results of experiment 2 are summarized in Table 6-2. On average both event-triggered controllers reduce the number of control updates while obtaining a decrease in energy consumption compared to the periodic controller. Furthermore, the average RMSE during ETC is lower than during periodic control.

Table 6-2: Average results of 140 runs of experiment 2. The bold entries correspond with the results obtained during periodic control which is used as a base case for the comparison with event-triggered control. In the last column SS represents the steady state error.

Contr.	σ	Events	Energy (mJ)		std. dev. (mJ)	RMSE ($\cdot 10^{-3}$ rad)	abs. SS err. ($\cdot 10^{-3}$ rad)
Per	-	**160**	**502.90**		**26.45**	**20.9**	**6.91**
ETC	0.071	56.3	486.71	-3.2%	28.73	20.6	6.32
ETC	0.100	52.3	477.15	-5.1%	34.12	20.5	6.06

Fig. 6-8 shows the evolution of the state and the control input while Fig. 6-9 shows the corresponding evolution of the number of events and energy during one run of experiment 2. There is a clear steady state error for all three controllers as shown in Fig. 6-5a. The magnitude of the steady state error during experiment 2 is higher than during experiment 1. Possible effects that might cause this steady error are the weight of the legs and static or dynamic friction components of the motor/gearbox combinations, such as Coulomb friction. These effects seem to be of less influence during experiment 1 where the robot is moving downwards and the weight of the legs, which is not modelled, might help reduce the effect of friction.

Although smaller than in experiment 1, there is still a small oscillation in the angle before the system reaches its steady state. The influence of backlash is smaller during experiment 2 since the robot is moving upwards towards the equilibrium angle. Therefore, the oscillation seems to be caused mainly by the compliancy of the legs.

Besides a difference in the steady state error, Fig. 6-8 shows that there is a clear difference in the evolution of the states. Fig. 6-9a shows that approximately 0.1 s after starting the experiment, the event-triggered controllers start generating less events than the periodic controller. This results in a higher and more constant velocity between 0.9-0.96 s, shown in Fig. 6-8b. This effect causes the angle to approach the equilibrium angle from 0.9 s and onwards in a more linear fashion until events start occurring and the controller takes action to slow down the robot which is shown in Fig. 6-8c after 0.95 s.

Due to the generation of lesser events, higher absolute control inputs are applied for a longer period of time compared to the periodic controller which results in a higher angular velocity. This higher velocity seems to allow the event-triggered controllers to reach the equilibrium state more closely which results in a lower steady state error and therefore a lower absolute input voltage being applied to the system. This lower input voltage in steady state results in lower energy consumption for the event-triggered controllers as shown in Fig. 6-9b. Table 6-2 shows that the steady state error during ETC is on average indeed lower which seems to be the main reason for the energy savings obtained during ETC.

Fig. 6-10 shows a plot of the RMSE and the total energy consumption for each run of experiment 2. Clearly a higher RMSE results in a higher energy consumption. This confirms that, on average, the better performance achieved during ETC, results in a lower energy consumption during this experiment.

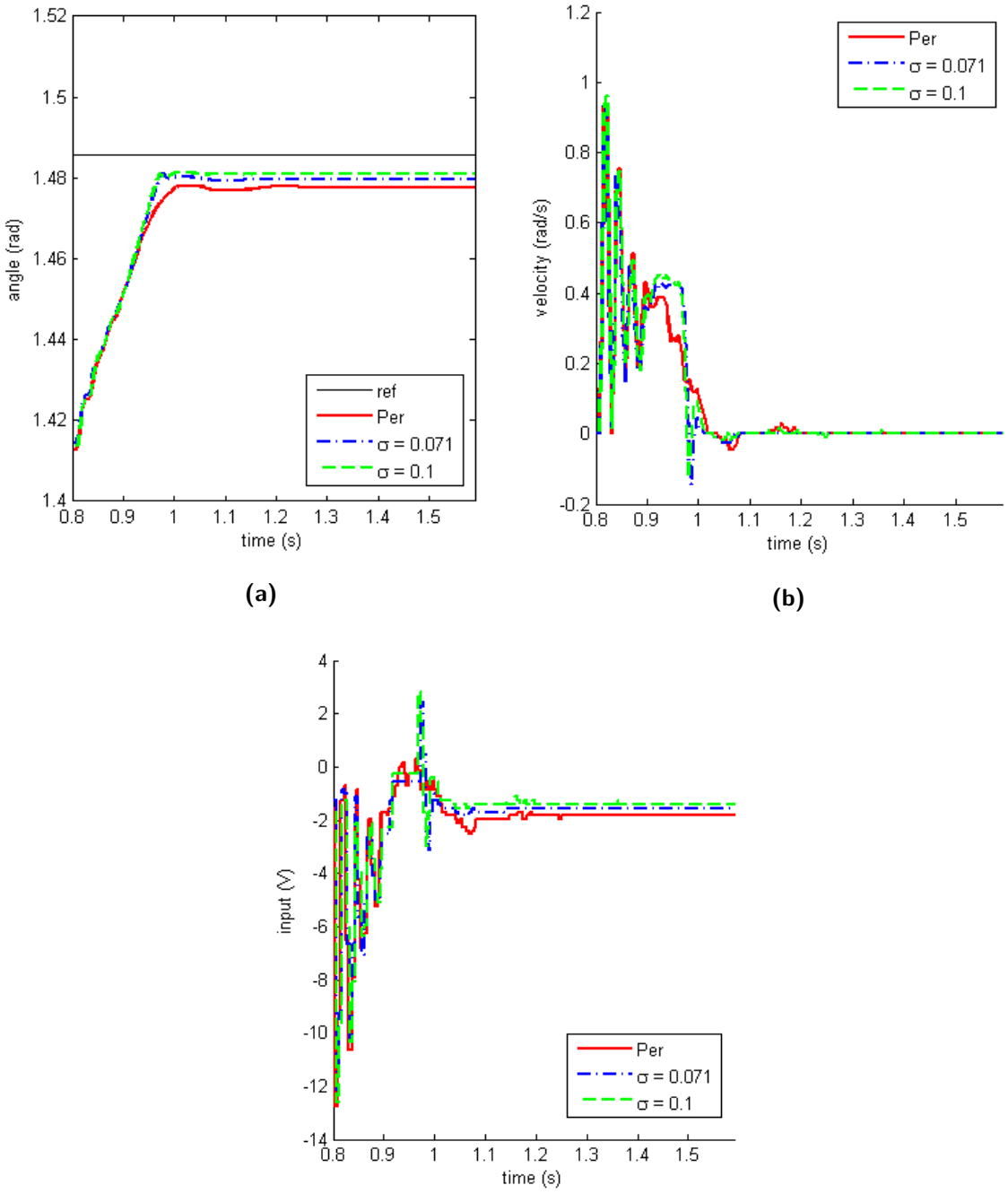
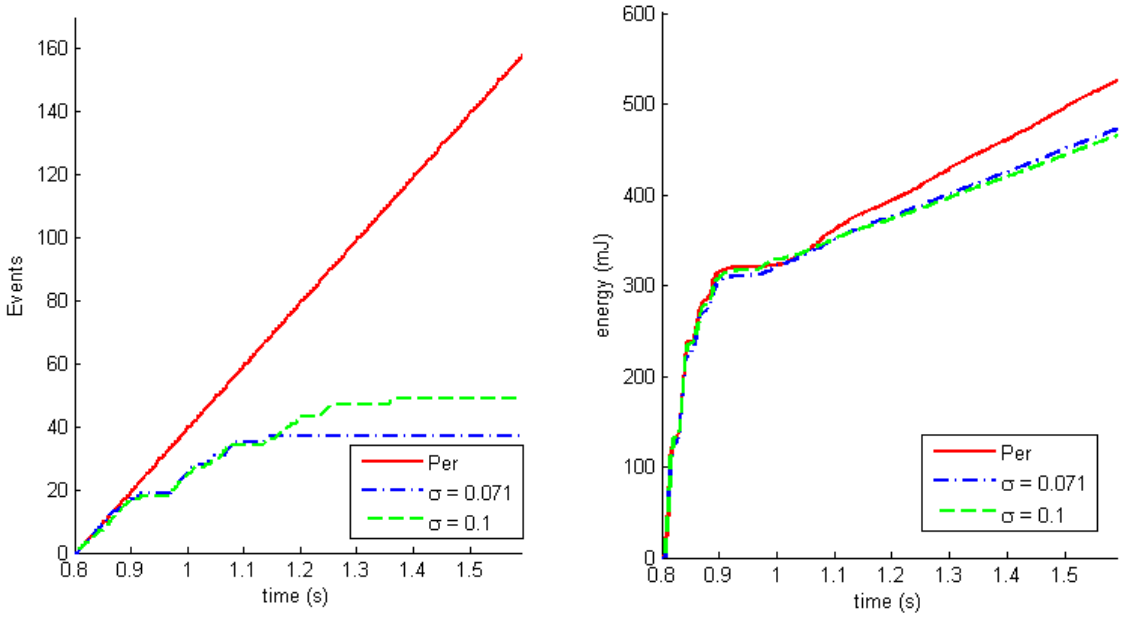


Figure 6-8: Results of one run of experiment 2 for each of the three controllers. (a) Evolution of the first state $x_1 = \theta$. (b) Evolution of the second state $x_2 = \dot{\theta}$. (c) Control input where only one of the two control inputs (u_1) for each controller is shown since they are equal ($u_1 = u_2$)



(a) **(b)**

Figure 6-9: Events and energy for one run of experiment 2. (a) Evolution of the number of events. (b) Evolution of the energy.

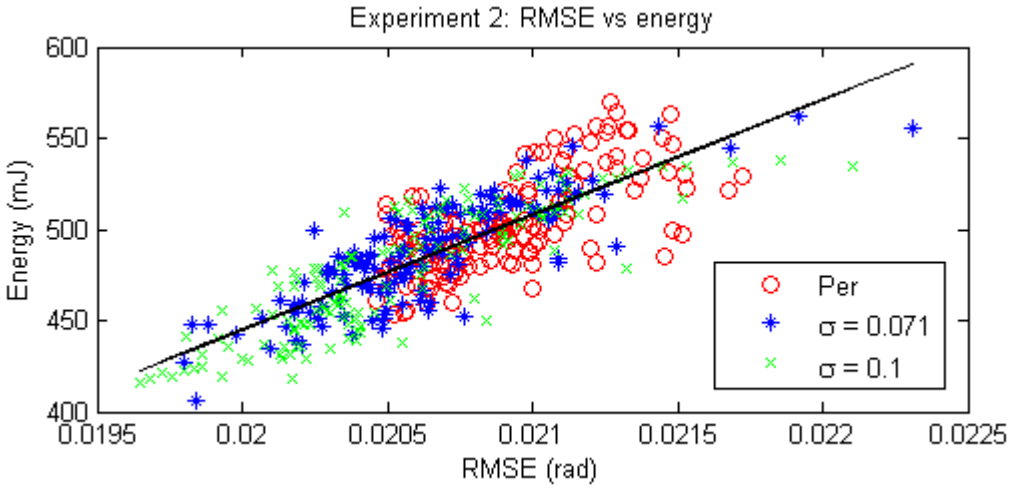


Figure 6-10: RMSE and the total energy consumption for each of the 140 runs of experiment 2.

6-4 Results experiment 3

Similarly as for the previous experiments the results are obtained by performing seven sequences of 20 runs of experiment 3. Fig. 6-11 shows that for both event-triggered controllers the number of generated events are clustered around approximately 96. The energy consumed during periodic control is spread around 47.87 mJ and the energy consumed during event-triggered control is spread around an average of 44.16 mJ and 42.26 mJ respectively. In this section only the energy consumption of the motors of the front legs is presented since the disturbance is applied on the control input of the back legs. Histograms showing the distribution of the energy usage during experiment 3 can be found in Appendix D-3.

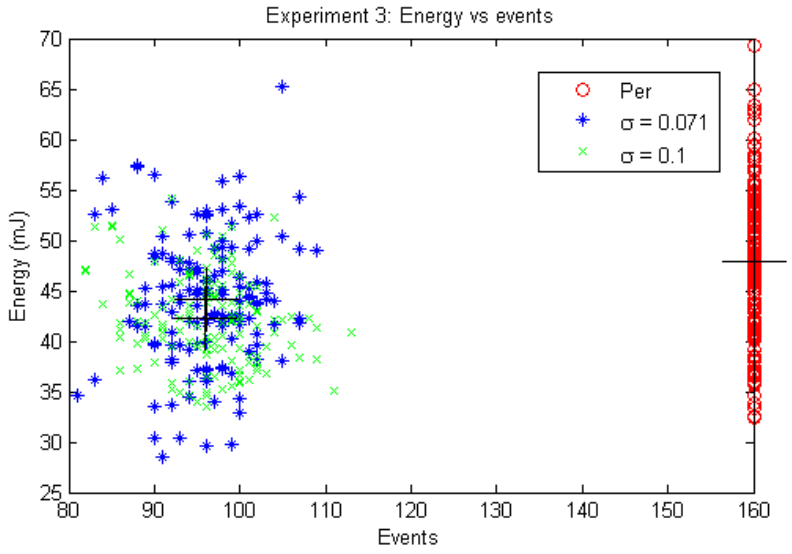


Figure 6-11: Energy consumption and number of events for each of the 140 runs of experiment 3.

The results of experiment 3 are summarized in Table 6-3. On average, both event-triggered controllers reduce the number of control updates while reducing the energy consumption compared to periodic control. Furthermore, both controllers obtained a lower RMSE than the periodic controller.

Table 6-3: Average results of 140 runs of experiment 3. The bold entries correspond with the results obtained during periodic control which is used as a base case for the comparison with event-triggered control. In the last column SS represents the steady state error

Contr.	σ	Events	Energy (mJ)		std. dev. (mJ)	RMSE ($\cdot 10^{-3}$ rad)	max abs. overshoot ($\cdot 10^{-3}$ rad)	abs. SS err. ($\cdot 10^{-3}$ rad)
Per	-	**160**	**47.87**		**7.84**	**2.45**	**6.1**	**1.62**
ETC	0.071	96.03	44.16	-7.8%	6.48	2.42	5.8	1.79
ETC	0.100	95.97	42.26	-11.7%	4.64	2.24	5.6	1.35

The evolution of the state, control input and the disturbance pulse that is added on the control input of the back leg pair is shown in Fig. 6-12. The corresponding evolution of the events and energy is shown in Fig. 6-13.

After the disturbance is applied, the system deviates form its starting position and enters a transient state. The controllers react to this sudden disturbance and try to steer the system back towards the equilibrium state. First, the system deviates towards an angle higher than the equilibrium angle. The controller tries to counteract this movement and stops this motion. It seems that the combination of the control action, the compliancy of the legs and possibly the effect of backlash cause the system to overshoot its equilibrium angle at around 1 s.

During the first 0.2 s the energy consumption for each of the three controllers is similar as shown in Fig. 6-13b. The most obvious difference in energy consumption shown in Fig. 6-13b occurs from 1.2 s and onwards and corresponds with a higher steady state error for the periodic controller. Similarly as in experiment 2 it seems that a higher absolute velocity is achieved during ETC which allows the system to get closer to the equilibrium state. Even though the results presented Fig. 6-13 show that the steady state error for periodic control is larger than for ETC, this is not the case on average as shown by Table 6-3. Furthermore, the absolute average steady state error in experiment 3 is smaller than during experiment 2 so the slight differences between the errors in Table 6-3 will have lesser effect. It is nevertheless possible that outliers with above average steady state error, such as shown in Fig. 6-12a, do have an effect on the average energy consumption.

The second difference in energy consumption shown in Fig. 6-13b occurs at around 1.1 s. This difference is harder to explain however it seems that during ETC a higher absolute velocity is obtained during the overshoot that occurs after 1 s. This higher velocity can be caused by the small reduction in control updates since when the system is moving away from the equilibrium angle a lower absolute control input is applied by the controller for a longer period during ETC than during periodic control. A speculative explanation as to why this could lead to a reduction in energy consumption is that when this downward movement away from the equilibrium is stopped the compliant legs store some energy of this higher velocity which helps propel the robot more easily towards the equilibrium state.

Fig. 6-14 shows a plot of the RMSE and the total energy consumption for each run of experiment 3. The results show that a higher RMSE results in a higher energy consumption. A possible explanation for the average higher RMSE of the periodic controller is that more overshoot occurs on average which is shown in Table 6-3.

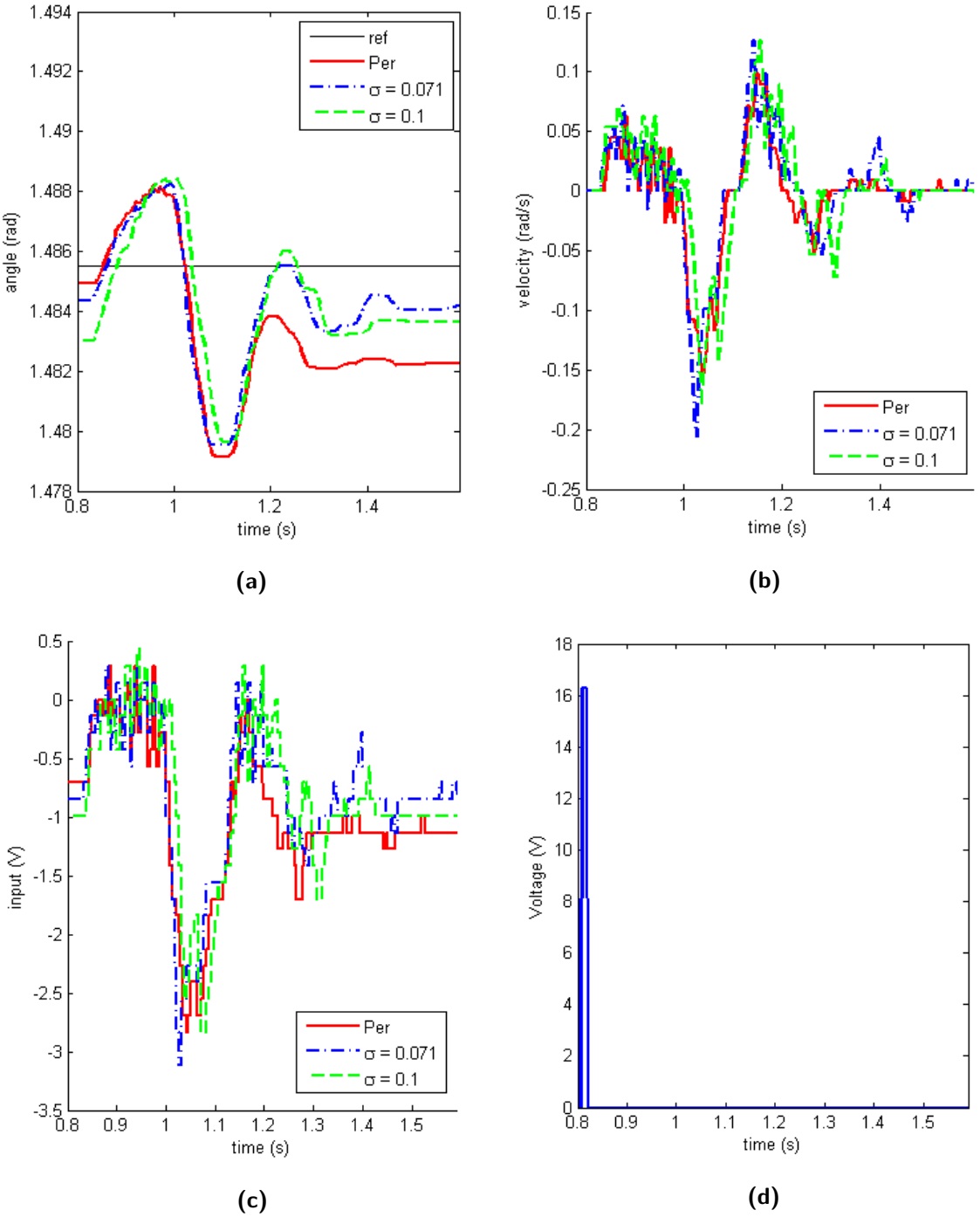
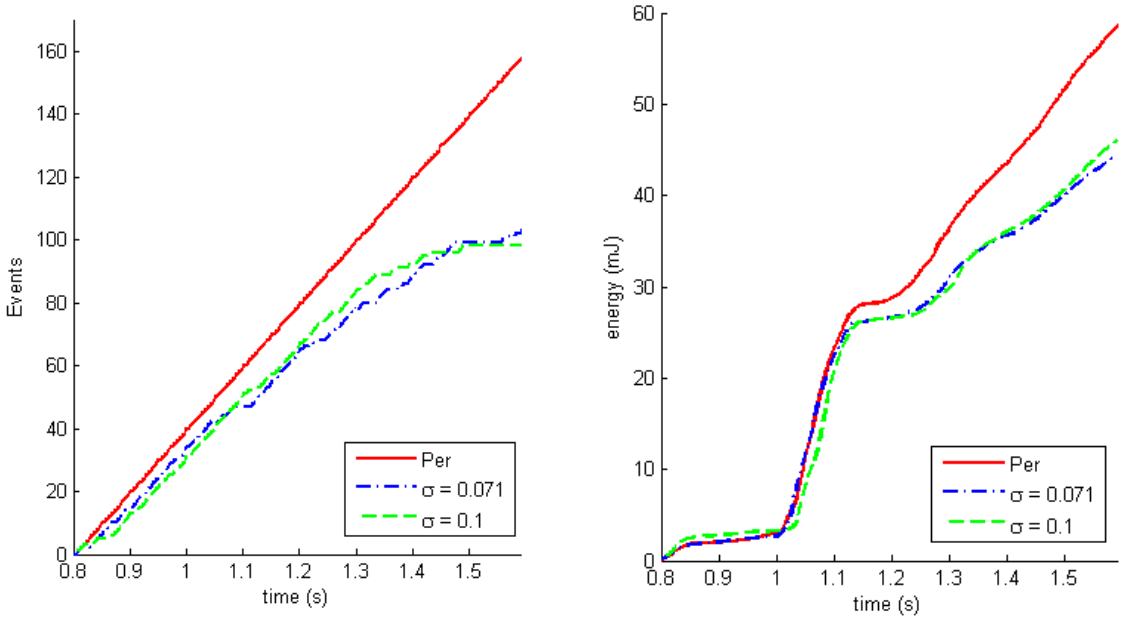


Figure 6-12: Results of one execution of experiment 3 for each of the three controllers. (a) Evolution of the first state $x_1 = \theta$. (b) Evolution of the second state $x_2 = \dot{\theta}$. (c) Control input where only one of the two control inputs (u_1) for each controller is shown since they are equal $(u_1 = u_2)$ (d) Disturbance added to the control input of the back leg pairs.



(a) **(b)**

Figure 6-13: Events and energy for one execution of experiment 3. (a) Evolution of the number of events. (b) Evolution of the energy.

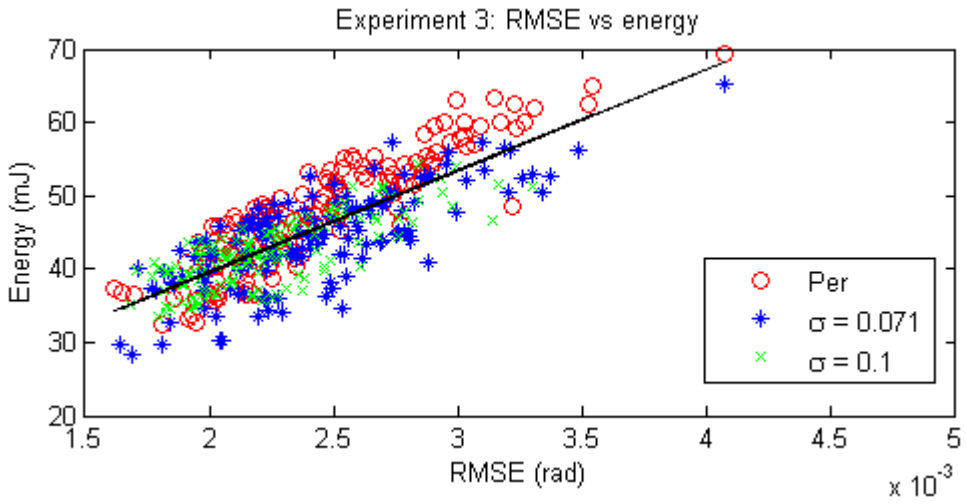


Figure 6-14: RMSE and the energy consumption for each of the 140 runs of experiment 3

6-5 Discussion

After analyzing the results it seems that the higher angular velocity obtained during ETC is the main effect causing differences in the energy consumption. A reduction in the number of control updates during transients results in a different control input sequence. When the system is moving towards the equilibrium state, this reduction causes higher absolute control inputs to be applied for longer periods of time. Similarly, when the system is moving away from the equilibrium state, this reduction causes lower absolute inputs to be applied for longer amounts of time. In both cases this results in a higher angular velocity.

In the first experiment higher overshoot occurred during ETC which increased the energy consumption by 3.0%. In the second experiment the equilibrium state was reached more closely during ETC which reduced the energy consumption by 3.2%. Finally, in the third experiment the energy consumption during ETC decreased with 7.8%. A speculative explanation for why differences in the energy consumption occur during the third experiment is that the compliancy of the legs, together with the higher absolute velocity obtained during ETC, helped propel the system towards the equilibrium state with less control effort. The above mentioned results are for the most conservative event-triggered controller designed in Chapter 4 with $\sigma = 0.0710$. Overall, it seems that differences in energy consumption are caused by the higher velocity obtained during ETC combined with the effects of unmodeled dynamics such as friction (static and dynamic), backlash and the compliancy of the legs. In absence of these effects it is likely that the energy consumption of ETC and periodic control would be more similar.

During all three experiments ETC reduced the number of control updates on average with at least 35.9% . These results were expected and confirm the results of this property presented in the literature (e.g. [6, 8, 11]). The significance of this reduction becomes more clear when realizing that these reductions were achieved during transients. When combined with the savings that would occur in steady state, the overall processor utilization will be reduced even more. With respect to legged robots this reduction can have several benefits. It allows cheaper hardware to be used or it allows extra processing time for other tasks which could result in indirect energy savings as well. These savings can be achieved if faster completion of other tasks can reduce the overall time that the robot has to be balanced or controlled. To fully exploit the potential reduction in processor utilization, it would be more effective to implement the event-triggering mechanism on the FPGA in order to truly offload the processor.

A reduction in the total energy consumption and a reduction in the number of control updates, as observed in the second and third experiment, is the best case scenario and justifies the use of ETC. A slight increase in energy consumption of 3.0% but a reduction in the number of control updates of 35.9 %, as observed in the first experiment for $\sigma = 0.071$, can still justify the use of ETC. This mainly depends on whether or not the extra processing time that is being gained can be used effectively and outweighs the increase in energy consumption.

When comparing the two event-triggered controllers, one with $\sigma = 0.071$ and one with $\sigma = 0.100$, the maximum difference in the average number of generated events during the experiments was 4. The expectation would be that the less conservative triggering mechanism would yield a higher reduction in terms of events. Using a higher sampling frequency would possibly increase the differences between the two event-triggered controllers in terms

of the number of events. Furthermore, the results indicate that the relative triggering mechanisms did not cause events being generated at a high rate when the system was close to the equilibrium. This justifies the use of a relative triggering mechanism.

Despite the experiments being repeated a large number of times, analyzing the individual experiment runs is difficult due to the distribution of the results. Several factors contributed to this distribution of the results. The backlash, which is slightly different for each motor/gearbox combination, combined with variations in the shapes, compliancy and mounting of each leg adds a random factor to the experiments. Furthermore, these factors make it difficult to ensure the same initial conditions for each experiment run. These variations can cause differences in the energy consumption during an experiment and can result in different event-triggering behavior during ETC. By running each experiment a large number of times it is expected that the effect of these variations are partially averaged out. The insights gained by analyzing the average results, combined with the results of individual experiment runs seem to justify the observed differences in the average results.

Chapter 7

Conclusions and future work

The goal of this thesis was to study the effect of event-triggered control on the energy consumption of a legged robot. The study was performed using the RQuad and was focused on the control task of balancing the robot on top of its four legs.

To achieve this goal a standard time-triggered controller was designed which would serve as a base case for the experimental comparison with event-triggered control. After deriving and linearizing a planar two leg model for the robot, a standard periodic controller was designed using the LQR method. This periodic controller was then used to design a periodic event-triggered controller with a relative event-triggering mechanism.

To perform the experimental comparison, a system was developed that allowed real-time control and energy monitoring of the RQuad. The embedded system used to control the robot is a National Instruments myRIO and is programmed using LabVIEW graphical design software. The FPGA of the myRIO is used to interface with the hardware of the robot while the processor runs the control tasks. In order to monitor the energy consumption the motor currents needed to be measured. This was achieved using shunt resistors in series with each DC motor of the RQuad.

Three experiments were performed that allowed the energy consumption of the controllers to be compared during transients. In the first two experiments the robot started from nonzero initial conditions and the energy measurements were performed during the resulting transients. In the third experiment a disturbance was added to the control input of the back leg pair after which the energy consumption of the front leg pair was measured. All the experiments were repeated 140 times for periodic and event-triggered control.

The impact of ETC on the energy consumption of the robot was found to be different for each experiment. In the first experiment ETC increased the average energy consumption with 3.0% while during the second and third experiment it decreased the average energy consumption with 3.2% and 7.8% respectively. During all three experiments ETC was capable of reducing the number of control updates significantly with at least 35.9% while retaining similar performance in terms of RMSE. The main effect observed was that a higher absolute angular velocity was achieved due to the reduction in control updates during ETC. This effect

combined with unmodeled dynamics such as friction, backlash and the compliancy of the legs seem to contribute to the differences in the energy consumption.

A slight increase in energy consumption while reducing the number of control updates significantly, as observed in the first experiment, can make ETC appropriate depending on whether or not the reduced processor utilization can be used effectively for other purposes. A decrease in energy consumption while reducing the number of control updates, as was observed during the second and third experiment, justifies the use of ETC for legged robots.

In this thesis the effect of event-triggered control on the energy consumption of a legged robot has been explored. The work in this thesis and the developed system can be used to continue research of event-triggered control of legged robots. Some suggestions for future work:

- To get a more complete overview of the effects of ETC it would be interesting to study the impact of different, perhaps mechanically applied, external disturbances. This can sharpen the understanding of the general effects that occur during ETC which could help working towards a strategy for exploiting the positive effects.
- Currently, events are triggered by the periodic control task that runs on the processor. To fully exploit the benefits of ETC it would be interesting to work towards a truly event-triggered architecture where for example the FPGA can interrupt the processor to have it perform the control task in a more event-triggered way.
- Study how the positive effects of ETC and the differences observed between the two event-triggered controllers can be exploited in order to work towards an overall energy saving ETC approach. Perhaps some sort of mechanism that uses different triggering rules in different parts of the state space or a mechanism that triggers events in an adaptive way.
- The study in this thesis was performed for a single control task. Studying different control tasks and implementing them in an event-triggered way can lead to new insights in ETC. Furthermore, working towards a fully event-triggered robot capable of performing a variety of tasks in an event-triggered way allows a study of whether or not secondary energy savings can actually be achieved.
- A more general direction would be to look into how ETC responds to different types of unmodelled dynamics or how it could be used as a way to deal with them.

Appendix A

Modelling

A-1 Angle derivation

To express the four-bar linkage angles ϕ and α in terms of θ the following approach is taken in [27]. By summing the squares of the constraint equations given by (3-1) and (3-2) the following equation is obtained

$$k_1 \sin\phi + k_2 \cos\phi + k_3 = 0 \tag{A-1}$$

where k_1, k_2 and k_3 are given by (3-5), (3-6), (3-7) respectively.

Following [27] (A-1) is solved by defining

$$t = \tan\frac{\phi}{2} \tag{A-2}$$

$$\sin\phi = \frac{2t}{1+t^2} \tag{A-3}$$

$$\cos\phi = \frac{1-t^2}{1+t^2} \tag{A-4}$$

If the above equations are substituted in (A-1) the following quadratic equation is obtained

$$(k_3 - k_2)t^2 + (2k_1)t + (k_3 + k_2) = 0 \tag{A-5}$$

of which the solution is given by

$$t = \frac{-k1 \pm \sqrt{k_1^2 + k_2^2 - k_3^2}}{k_3 - k_2}. \tag{A-6}$$

By then substituting (A-6) in (A-2) equation (3-3) can be derived. And finally (3-4) can be derived by dividing (3-2) by (3-1).

By differentiating the constraint equations given by (3-1) and (3-2) the following expressions for the velocities are obtained

$$\underbrace{\begin{bmatrix} l_2 \sin\alpha & -l_3 \sin\phi \\ -l_2 \cos\alpha & l_3 \cos\phi \end{bmatrix}}_{\mathrm{M}} \begin{bmatrix} \dot{\alpha} \\ \dot{\phi} \end{bmatrix} = \underbrace{\begin{bmatrix} -l_1 \sin\theta \\ l_1 \cos\theta \end{bmatrix}}_{\mathrm{K}} \dot{\theta}. \tag{A-7}$$

From (A-7) the velocities $\dot{\alpha}$ and $\dot{\phi}$ can be expressed in terms of the velocity $\dot{\theta}$

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\phi} \end{bmatrix} = M^{-1} K \dot{\theta} = \begin{bmatrix} S_1(\theta, \alpha, \phi) \\ S_2(\theta, \alpha, \phi) \end{bmatrix} \dot{\theta} \tag{A-8}$$

where

$$S_1(\theta, \alpha, \phi) = \frac{l_1 \sin(\phi - \theta)}{l_2 \sin(\alpha - \phi)} \tag{A-9}$$

$$S_2(\theta, \alpha, \phi) = \frac{l_1 \sin(\alpha - \theta)}{l_3 \sin(\alpha - \phi)}. \tag{A-10}$$

A-2 Equation of motion

By following [27] and [28] the Lagrangian will be computed after which the equation of motion for the RQuad is derived by applying the Euler-Lagrange equation.

The Lagrangian of the system is given by (A-11) where T is the total kinetic energy V is the total potential energy.

$$\mathcal{L} = T - V \tag{A-11}$$

The total kinetic energy of the system is given by

$$T = \frac{1}{2}(m_2 ||v_{c2}||^2 + I_2 \dot{\alpha}^2) \tag{A-12}$$

where I_2 is the moment of inertia of link 2 and where

$$||v_{c2}||^2 = l_1^2 \dot{\theta}^2 + l_{c2}^2 \dot{\alpha}^2 + 2 l_1 l_{c2} \cos(\theta - \alpha) \dot{\theta} \dot{\alpha}. \tag{A-13}$$

The total potential energy of the system is given by

$$V = m_2 g y_{c2} \tag{A-14}$$

where g is the gravitational acceleration and

$$y_{c2} = l_1 \sin\theta + l_{c2} \sin\alpha. \tag{A-15}$$

By then subtracting the total potential energy from the total kinetic energy the following expression for the Lagrangian is obtained:

$$\mathcal{L} = J_1 \dot{\theta}^2 + J_2 \dot{\alpha}^2 + P_1 C_1 \dot{\theta} \dot{\alpha} + G \tag{A-16}$$

where

$$J_1 = \frac{1}{2} m_2 l_1^2 \tag{A-17}$$

$$J_2 = \frac{1}{2}(m_2 l_{c2}^2 + I_2) \tag{A-18}$$

$$P_1 = m_2 l_1 l_{c2} \tag{A-19}$$

$$C_1 = \cos(\theta - \alpha) \tag{A-20}$$

$$G = -m_2 g l_1 \sin\theta - m_2 g l_{c2} \sin\alpha. \tag{A-21}$$

Next the velocities $\dot{\alpha}$ and $\dot{\phi}$ can be eliminated using (A-8). This results in the following expression for the Langragian

$$\mathcal{L} = (J_1 + J_2 S_1^2 + P_1 C_1 S_1)\dot{\theta}^2 + G. \tag{A-22}$$

The equation of motion can then be derived by applying the Euler-Langrage equation which is given by

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}}\right) - \frac{\partial \mathcal{L}}{\partial \theta} = Q_\theta \tag{A-23}$$

where Q_θ represents the generalized forces and the partial derivatives, derived following [27], are given by:

$$\frac{\partial \mathcal{L}}{\partial \dot{\theta}} = 2(J_1 + J_2 S_1^2 + P_1 C_1 S_1)\dot{\theta} \tag{A-24}$$

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}}\right) = {} & 2\left(J_1 + J_2 S_1^2 + P_1 C_1 S_1\right)\ddot{\theta} + \\ & 2\left(2J_2 S_1\left(\frac{\partial S_1}{\partial \theta} + \frac{\partial S_1}{\partial \alpha}\frac{\partial \alpha}{\partial \theta} + \frac{\partial S_1}{\partial \phi}\frac{\partial \phi}{\partial \theta}\right) + P_1\left(C_1\left(\frac{\partial S_1}{\partial \theta} + \frac{\partial S_1}{\partial \alpha}\frac{\partial \alpha}{\partial \theta} + \frac{\partial S_1}{\partial \phi}\frac{\partial \phi}{\partial \theta}\right)\right.\right. \\ & \left.\left. + S_1\left(\frac{\partial C_1}{\partial \theta} + \frac{\partial C_1}{\partial \alpha}\frac{\partial \alpha}{\partial \theta}\right)\right)\right)\dot{\theta}^2\end{aligned} \tag{A-25}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \theta} = {} & \left(2J_2 S_1\left(\frac{\partial S_1}{\partial \theta} + \frac{\partial S_1}{\partial \alpha}\frac{\partial \alpha}{\partial \theta} + \frac{\partial S_1}{\partial \phi}\frac{\partial \phi}{\partial \theta}\right) + P_1\left(C_1\left(\frac{\partial S_1}{\partial \theta} + \frac{\partial S_1}{\partial \alpha}\frac{\partial \alpha}{\partial \theta} + \frac{\partial S_1}{\partial \phi}\frac{\partial \phi}{\partial \theta}\right)\right.\right. \\ & \left.\left. + S_1\left(\frac{\partial C_1}{\partial \theta} + \frac{\partial C_1}{\partial \alpha}\frac{\partial \alpha}{\partial \theta}\right)\right)\right)\dot{\theta}^2 + \frac{\partial G}{\partial \theta} + \frac{\partial G}{\partial \alpha}\frac{\partial \alpha}{\partial \theta}.\end{aligned} \tag{A-26}$$

This leads to the following expression for the equation of motion

$$\begin{aligned}& M(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) = \\ & 2\left[J_1 + J_2 S_1^2 + P_1 C_1 S_1\right]\ddot{\theta} + \\ & \left[2J_2 S_1\left(\frac{\partial S_1}{\partial \theta} + S_1\frac{\partial S_1}{\partial \alpha} + S_2\frac{\partial S_1}{\partial \phi}\right) + P_1\left(C_1\left(\frac{\partial S_1}{\partial \theta} + S_1\frac{\partial S_1}{\partial \alpha} + S_2\frac{\partial S_1}{\partial \phi}\right) + S_1\left(\frac{\partial C_1}{\partial \theta} + S_1\frac{\partial C_1}{\partial \alpha}\right)\right)\right]\dot{\theta}^2 \\ & - \frac{\partial G}{\partial \theta} - S_1\frac{\partial G}{\partial \alpha} = Q_\theta\end{aligned} \tag{A-27}$$

where

$$\frac{\partial S_1}{\partial \theta} = -\frac{l_1 \cos(\phi - \theta)}{l_2 \sin(\alpha - \phi)} \tag{A-28}$$

$$\frac{\partial S_1}{\partial \alpha} = -\frac{l_1 \sin(\phi - \theta)\cos(\alpha - \phi)}{l_2 \sin^2(\alpha - \phi)} \tag{A-29}$$

$$\frac{\partial S_1}{\partial \phi} = -2\frac{l_1 \sin(\alpha - \theta)}{-l_2 + l_2 \cos(2\alpha - 2\phi)} \tag{A-30}$$

$$\frac{\partial S_2}{\partial \theta} = -\frac{l_1 \cos(\alpha - \theta)}{l_3 \sin(\alpha - \phi)} \tag{A-31}$$

$$\frac{\partial S_2}{\partial \alpha} = 2\frac{l_1 \sin(\phi - \theta)}{-l_3 + l_3 \cos(2\alpha - 2\phi)} \tag{A-32}$$

$$\frac{\partial S_2}{\partial \phi} = \frac{l_1 \sin(\alpha - \theta)\cos(\alpha - \phi)}{l_3 \sin^2(\alpha - \phi)} \tag{A-33}$$

$$\frac{\partial C_1}{\partial \theta} = \sin(\alpha - \theta) \tag{A-34}$$

$$\frac{\partial C_1}{\partial \alpha} = -\sin(\alpha - \theta) \tag{A-35}$$

$$\frac{\partial G}{\partial \theta} = -m_2 l_1 g \cos\theta \tag{A-36}$$

$$\frac{\partial G}{\partial \alpha} = -m_2 g l_{c2} \cos\alpha. \tag{A-37}$$

Appendix B

Current sensor error calculation

In [32] it is specified how the total error can be calculated for the INA282 current sense monitor. Several error terms depending on the operating conditions affect the total measurement error.

Additional input offset voltage will be present due to the common mode voltage. The input offset voltage due to the common mode voltage can be calculated as follows

$$V_{OS\_CM} = \frac{1}{10^{\frac{CMRR}{20}}}(V_{CM} - 12V) \tag{B-1}$$

where $CMRR$ is the common mode rejection ratio in dB and V_{CM} is the common mode voltage.

The reference voltage is used as the voltage corresponding to zero current. The additional offset voltage caused by the reference voltage can be calculated using the following equation

$$V_{OS\_REF} = RVRR(2.048V - V_{REF}) \tag{B-2}$$

where $RVRR$ is the reference voltage rejection ratio in $\mu V/V$.

The total input offset voltage is given by the following equation

$$V_{OS\_Total} = \sqrt{(V_{OS})^2 + (V_{OS\_CM})^2 + (V_{OS\_REF})^2}. \tag{B-3}$$

The error due to the input offset voltage is given by

$$e_{V_{OS}} = \frac{V_{OS\_Total}}{V_{SENSE}} \times 100 \tag{B-4}$$

where V_{SENSE} is the voltage over the resistor that is used to measure the current.

The total error is then given by

$$e_{Total} = \sqrt{(e_{V_{OS}})^2 + (G_{\text{err}})^2 + (\text{Lin}_{\text{err}})^2} \tag{B-5}$$

Where G_{err} is the gain error and Lin_{err} is the nonlinearity error

The specifications of the INA282 of which some are used to calculate the error are shown in Table B-1 and are taken from the manufacturers datasheet [32]. The operating conditions used to calculate the error in the current measurements are $V_{REF} = 2.5$ V, $V_{CM} = 18$ V and V_{sense} is varied in order to calculate the error for a range of current.

Table B-1: Specifications of the INA282 current shunt monitor

parameter	Symbol	min	typ	max	unit
Supply voltage	$V+$	2.7	-	18	V
Bandwidth	BW	-	10	-	KHz
Common mode voltage	V_{CM}	-14	-	+80	V
Input offset voltage	V_{OS}	-	$\pm$20	$\pm$70	μV
Gain	G	-	50	-	-
Gain error	G_e	-	$\pm$0.4%	$\pm$1.4%	-
Common mode rejection ratio	$CMRR$	120	140	-	dB
Reference voltage rejection ratio	$RVRR$	-	$\pm$25	$\pm$75	$\mu V/V$
Nonlinearity error	L_e	-	$\pm$0.01%	-	-

Appendix C

Posbus

All data is transmitted over the Posbus in packets containing a LCS (length/command/status) byte, zero or more data bytes and a CRC byte. Fig. C-1 shows a schematical depiction of a Posbus package. Each byte is serialized LSB first.

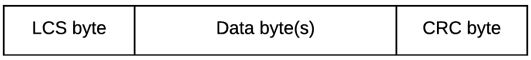


Figure C-1: Packet structure for packets transmistted over the Posbus

The first six bits of the LCS byte, depicted in Fig. C-2 are used to encode the size of the packet. The master uses bit 7 to indicate whether or not it expects a reply from the slave, bit 8 is used for sending and receiving universal commands. Bit 7 is used by the slave to indicate whether a receive error has occurred.

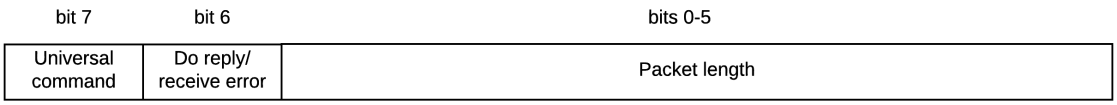


Figure C-2: Structure of the LCS byte

Fig. C-3 shows the packet that is used to control the Hip board. The structure and data that is transmitted in messages to and from the Hip board as presented in this text have been obtained by analyzing the VHDL source code of the Hip boards.

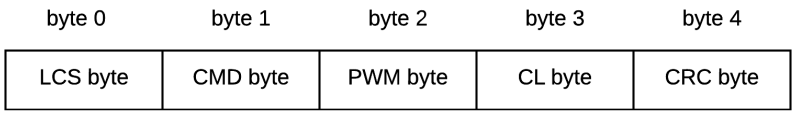


Figure C-3: Control message that can be used to control the Hip board

The structure of the CMD byte is shown in Fig. C-4

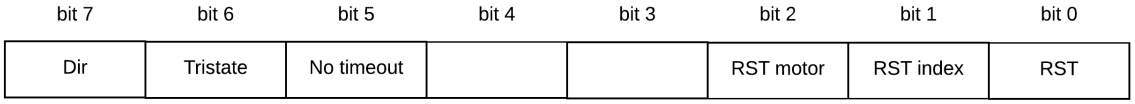


Figure C-4: Structure of the CMD byte

In the CMD byte the "Dir" bit sets the motor direction, "Tristate" can be used to turn all the FETs in the H-bridge off, "No timeout" sets whether the Hip board will stop generating PWM if no valid command is received every 120 ms (at 40MHz SerialClock), "RST motor" is used to allow an external signal to reset the encoder pulse counter, "RST index" resets the pulse counter on an index pulse of the encoder and "RST" directly resets the pulse counters.

The message sent from the Hip board to the master is shown in Fig. C-5

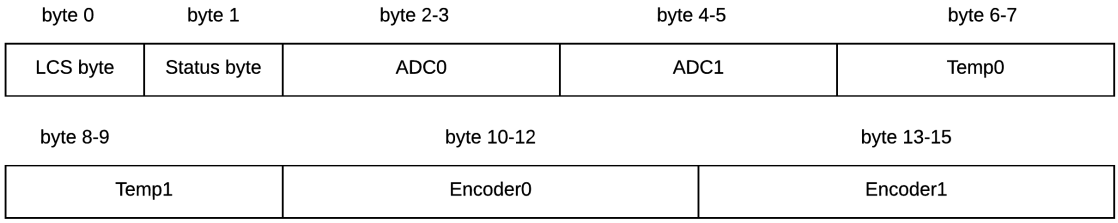


Figure C-5: Data message sent from the Hip board to the master

The structure of the status byte is shown in Fig C-6

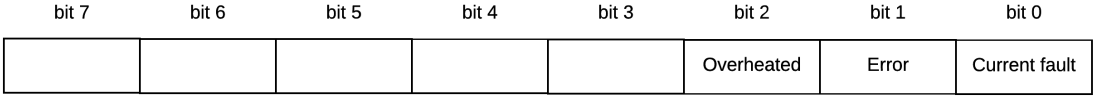


Figure C-6: Structure of the status byte

Furthermore Fig. C-5 shows that there are two ADC channels of which only ADC1 is used to measure the current. There are also two temperature channels of which only temp0 is

used. From the encoder channels only Encoder0 is used. The unused channels can be used to extend the Hip boards functionality.

Fig. C-7 shows the ciruit that is used to communicate with the Hip boards over the Posbus using the DS90LV049 LVDS driver from Texas Instruments [33]. Since there are four Hip boards to communicate with, four of these circuits are used. The LVDS drivers are powered by the 3.3 V output of the myRIO. The connections with the myRIO are as follows: SerialClock is connected to Din1, SerialOut is connected to Din2 and SerialIn is connected to Rout2. Rout1 is unconnected. Both receiving channels are terminated with a 100Ω resistor. C1 and C2 are bypass capacitors of respectively $0.001\mu F$ and $0.1\mu F$.

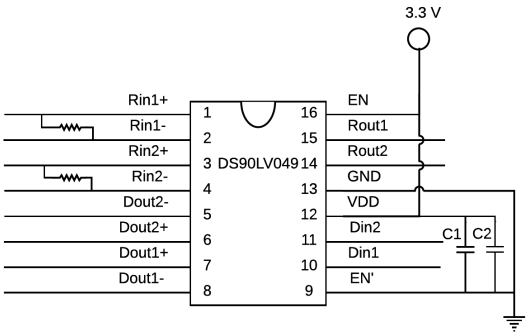


Figure C-7: LVDS driver circuit used for communicating over the Posbus

Appendix D

Experimental results

D-1 Experiment 1

Fig. D-1 shows for each controller the distribution of the total energy consumption for each run of experiment 1.

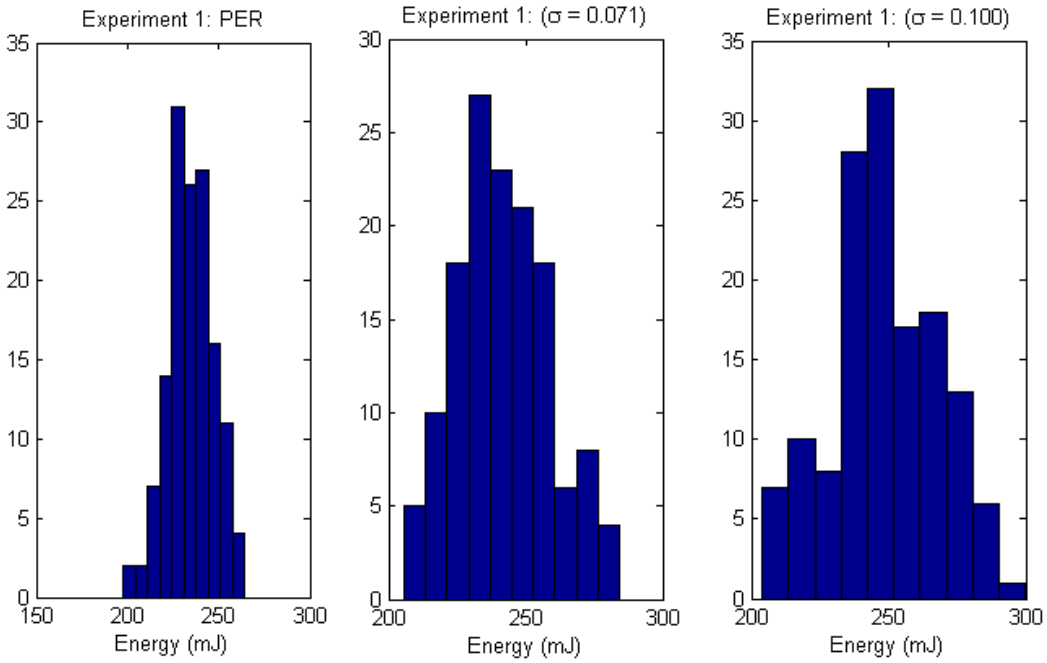


Figure D-1: Histograms showing the distribution of the energy consumption for 140 runs of experiment 1. (a) Periodic controller. (b) Event-triggered controller with $\sigma = 0.0710$. (c) Event-triggered controller with $\sigma = 0.100$.

D-2 Experiment 2

Fig. D-2 shows for each controller the distribution of the total energy consumption for each run of experiment 2.

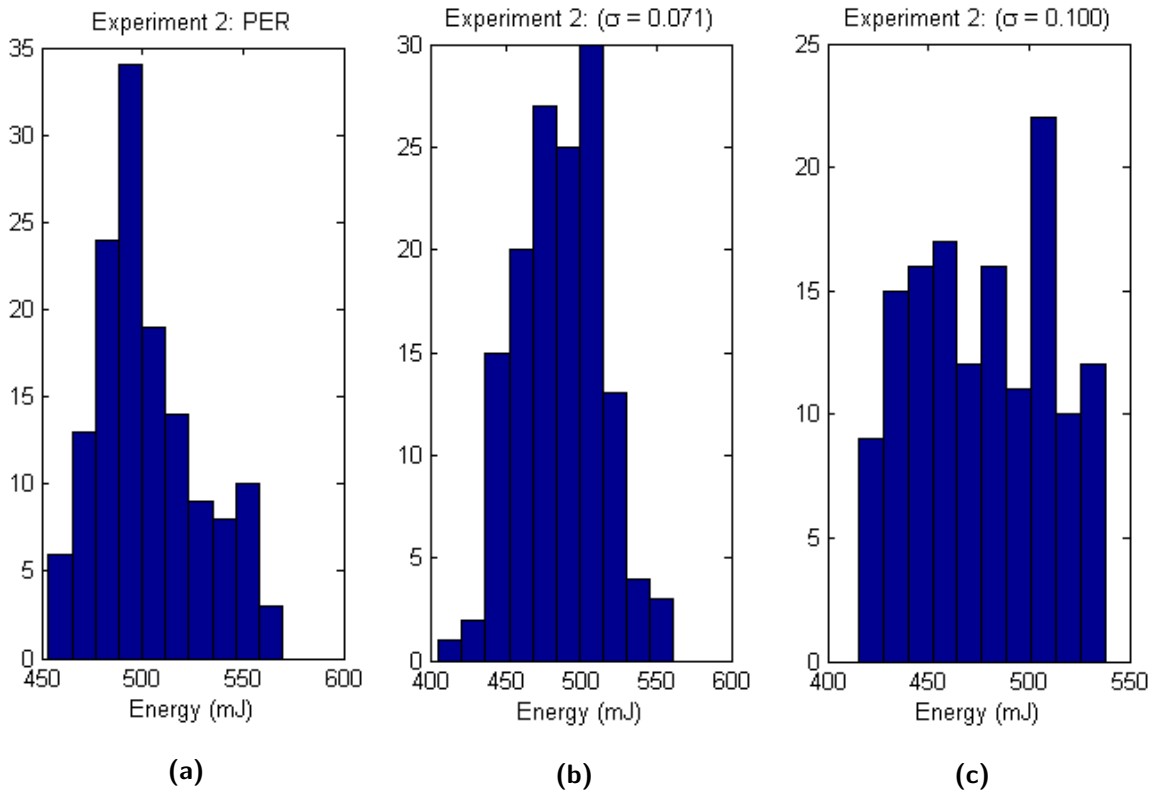


Figure D-2: Histograms showing the distribution of the energy consumption for 140 runs of experiment 2. (a) Periodic controller. (b) Event-triggered controller with $\sigma = 0.0710$. (c) Event-triggered controller with $\sigma = 0.100$.

D-3 Experiment 3

Fig. D-3 shows for each controller the distribution of the total energy consumption for each run of experiment 3.

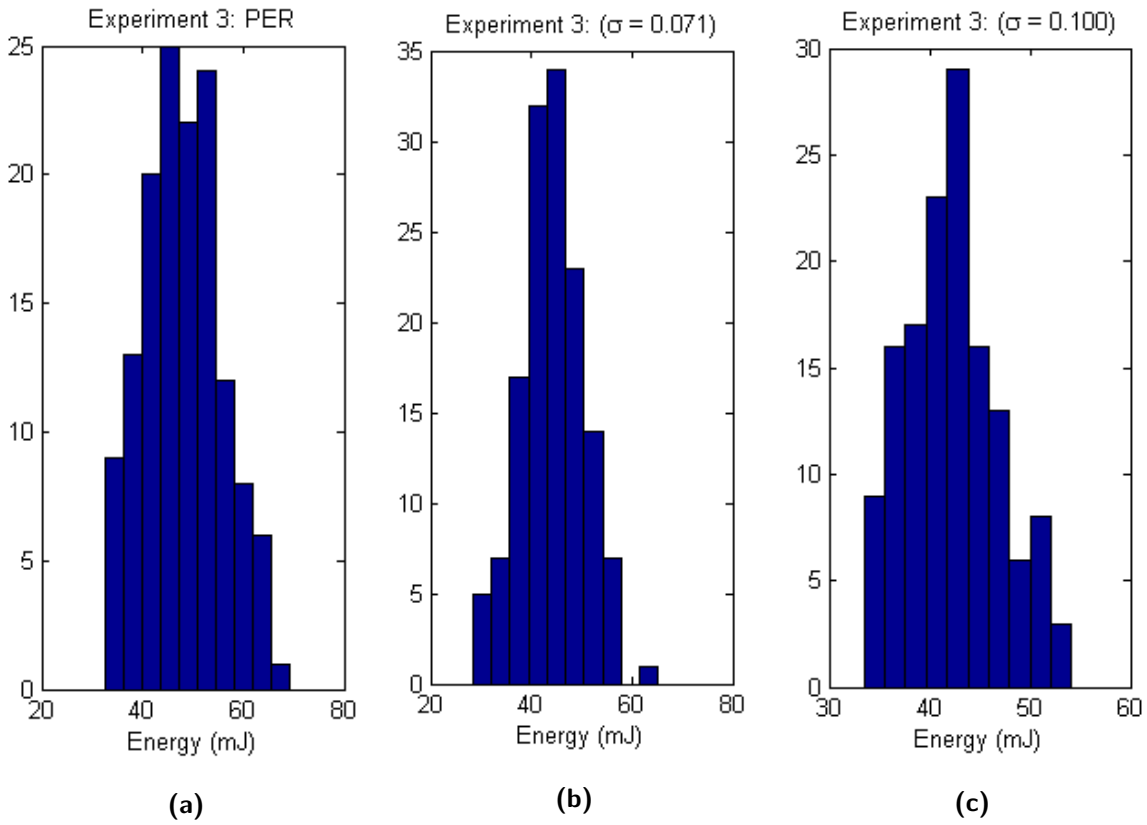


Figure D-3: Histograms showing the distribution of the energy consumption for 140 runs of experiment 3. (a) Periodic controller. (b) Event-triggered controller with $\sigma = 0.0710$. (c) Event-triggered controller with $\sigma = 0.100$.

Bibliography

[1] U. Saranli, "RHex: A Simple and Highly Mobile Hexapod Robot," *Int. J. Robotics Research*, vol. 20, pp. 616–631, July 2001.

[2] U. Saranli, A. Rizzi, and D. E. Koditschek, "Model-Based Dynamic Self-Righting Maneuvers for a Hexapedal Robot," *Int. J. of Robotics Research*, vol. 23, pp. 903–918, Sept. 2004.

[3] E. Z. Moore, D. Campbell, F. Grimminger, and M. Buehler, "Reliable stair climbing in the simple hexapod 'RHex'," in *Proc. Int. Conf. Robotics and Automation (ICRA 2002)*, vol. 3, (Wash. DC), pp. 2222 – 2227, 2002.

[4] Y. C. Chou, K. J. Huang, W. S. Yu, and P. C. Lin, "Model-based development of leaping in a hexapod robot," *IEEE Trans. Robot.*, vol. 31, pp. 40–54, Feb. 2015.

[5] K. J. Åström and B. Bernhardsson, "Comparison of periodic and event based sampling for first-order stochastic systems," in *Proc. IFAC World congr.*, pp. 301–306, 1999.

[6] K. E. Årzén, "A simple event-based PID controller," in *Preprints IFAC World Congr.*, vol. 18, pp. 423–428, 1999.

[7] J. K. Yook, D. M. Tilbury, and N. R. Soparkar, "Trading computation for bandwidth: Reducing communication in distributed control systems using state estimators," *IEEE Trans. Control Syst. Technol.*, vol. 10, pp. 503–518, July 2002.

[8] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Trans. on Autom. Control*, vol. 52, pp. 1680–1685, Sept. 2007.

[9] T. Henningsson, E. Johannesson, and A. Cervin, "Sporadic event-based control of first-order linear stochastic systems," *Automatica*, vol. 44, pp. 2890–2895, Nov. 2008.

[10] M. C. F. Donkers and W. P. M. H. Heemels, "Output-based event-triggered control with guaranteed $\mathcal{L}_\infty$-gain and improved and decentralized event-triggering," *IEEE Trans. Autom. Control*, vol. 57, pp. 1362–1376, June 2012.

[11] W. P. M. H. Heemels, M. C. F. Donkers, and A. R. Teel, "Periodic event-triggered control for linear systems," *IEEE Trans. Autom. Control*, vol. 58, pp. 847–861, Apr. 2013.

[12] W. P. M. H. Heemels and M. C. F. Donkers, "Model-based periodic event-triggered control for linear systems," *Automatica*, vol. 49, pp. 698–711, Mar. 2013.

[13] R. Postoyan, P. Tabuada, D. Nesic, and A. Anta, "A framework for the event-triggered stabilization of nonlinear systems," *IEEE Trans. Automat. Control*, vol. 60, pp. 982–996, Apr. 2015.

[14] E. Garcia and P. J. Antsaklis, "Model-based event-triggered control for systems with quantization and time-varying network delays," *IEEE Trans. Autom. Control*, vol. 58, pp. 422–434, Feb. 2013.

[15] D. P. Borgers and W. P. M. H. Heemels, "Event-Separation Properties of Event-Triggered Control Systems," *IEEE Trans. Autom. Control*, vol. 59, pp. 2644–2656, Oct. 2014.

[16] J. F. Guerrero-Castellanos, J. Téllez-Guzmán, S. Durand, *et al.*, "Attitude stabilization of a quadrotor by means of event-triggered nonlinear control," *J. Intell. and Robotic Syst.*, vol. 73, pp. 123–135, 2014.

[17] J. Araujo, M. Mazo, A. Anta, P. Tabuada, and K. H. Johansson, "System architectures, protocols and algorithms for aperiodic wireless control systems," *IEEE Trans. Ind. Informat.*, vol. 10, pp. 175–184, Feb. 2014.

[18] M. Mazo and P. Tabuada, "On event-triggered and self-triggered control over sensor/actuator networks," in *Proc. IEEE conf. Decision and Control*, pp. 435–440, 2008.

[19] F. Altaf, J. Araújo, A. Hernandez, H. Sandberg, and K. H. Johansson, "Wireless event-triggered controller for a 3d tower crane lab process," in *IEEE Mediterranean Conf. Control & Automation*, pp. 994–1001, 2011.

[20] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *IEEE 51st Ann. Conf. Decision and Control (CDC)*, (Maui, Hawaii), pp. 3270–3285, 2012.

[21] J. Lunze and D. Lehmann, "A state-feedback approach to event-based control," *Automatica*, vol. 46, pp. 211–215, Jan. 2010.

[22] M. Mazo and P. Tabuada, "Decentralized event-triggered control over wireless sensor/actuator networks," *IEEE Trans. Autom. Control*, vol. 56, pp. 2456–2461, Oct. 2011.

[23] X. Wang and M. D. Lemmon, "Event-triggering in distributed networked control systems," *IEEE Trans. Autom. Control*, vol. 56, pp. 586–601, Mar. 2011.

[24] D. Panagou and H. Tanner, "Modeling of a Hexapod Robot ; Kinematic Equivalence to a Unicycle," Tech. Rep. UDME TR-2009-0001, Dept. Mech. Eng., Univ. of Delaware, Newark, DE, Apr. 2009.

[25] E. Najafi, R. Babuška, and G. A. D. Lopes, "Balancing a Legged Robot Using State-Dependent Riccati Equation Control," in *Proc. 19th IFAC World Congr.*, vol. 19, pp. 2177–2182, 2014.

[26] M. Buehler, U. Saranli, D. Papadopoulos, and D. E. Koditschek, "Dynamic locomotion with four and six legged robots," in *Proc. Int. Symp. Adaptive Motion of Animals and Machines*, (Montreal, Canada), August 2000.

[27] C. Tang, "Lagrangian dynamic formulation of a four-bar mechanism with minimal coordinates." unpublished, Mar. 2006.

[28] W. J. Zhang, W. Li, and L. S. Guo, "Integrated design of mechanical structure and control algorithm for a programmable four-bar linkage," *IEEE/ASME Trans. Mechatron.*, vol. 4, pp. 354 – 362, Dec. 1999.

[29] maxon motor, *S 2332 ⌀32 mm, grafietborstels, 15 W*, May 2013.

[30] National Instruments, "From student to engineer: Preparing future innovators with the ni labview rio architecture," Apr. 2014. [Online]. Available: http://www.ni.com/white-paper/52093/en/.

[31] maxon motor, "Pwm-scheme and current ripple of switching power amplifiers," Aug. 2000. Company white paper.

[32] Texas Instruments, *INA28x High-Accuracy, Wide Common-Mode Range, Bidirectional Current Shunt Monitors, Zero-Drift Series*. Dallas, Tex., May 2015.

[33] Texas Instruments, *DS90LV049 3V LVDS Dual Line Driver with Dual Line Receiver*. Dallas, Tex., Apr. 2013.