

CHALMERS



Feedback control of pallet systems

*Master of Science Thesis in the Master Degree Programme
Systems, Control and Mechatronics*

Karl Johansson & Sebastian Rebolledo

Department of Signals & Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2014
Master's Thesis SSYX04:1

Abstract

There are several costs associated with automated material handling systems in today's industry. Some of these costs can be related to operational costs, factory space and production time. By introducing feedback control to a pallet system, sustainable aspects such as system life time, energy consumption, noise produced by handling equipment and running costs can all be improved. The strategy is to reduce chain velocity of conveyors, with the condition of maintaining a desired production rate. The goal was to implement and verify the described strategy in a physical system.

In the initial phase of developing a controller, testing and verification was done in a simulation environment. The approach was to, in a step by step manner, improve the controller by adding supporting methods until it showed a satisfying performance before implementation in a physical system.

The goal of this project was reached and it was confirmed that controlling the chain velocity of a conveyor while maintaining a desired production rate is highly possible.

The concept presented in this project is still far from being a complete product. Further development of this concept however, could result in a product where mentioned sustainable aspects are greatly improved.

Acknowledgements

We would like to extend a great deal of gratitude to the project supervisor, Maziar Mashaei, for many invaluable inputs and really productive discussions during this project. During the simulation part of the project, we received help and would also like to thank Tommy Fässberg and Erik Ramfelt. Regarding the physical implementation we owe a lot of gratitude to Fredrik Sannehed and Rolf Eriksson for giving us the possibility to implement our work in practice.

The Authors, Göteborg 29/04/14

Contents

1	Introduction	1
1.1	Background and problem description	1
1.2	X85 Pallet System	2
1.3	Control Strategy	5
1.4	Limitations of the work	7
2	Controller design based on simulation	8
2.1	Simulation model	8
2.2	Initialization	8
2.2.1	Tuning of the number of pallets	9
2.2.2	Cyclic loading section	10
2.3	PID	10
2.4	Velocity Averaging	12
2.5	CONWIP	13
2.6	Kalman	15
2.7	Variable setpoint method	17
2.8	Summary	20
3	Implementation	22
3.1	Modification of physical system	22
3.1.1	Connecting PC and PLC	22
3.1.2	PLC code	23
3.2	Controller performance	23
3.2.1	Tuning of DPID gains	23
3.2.2	Complete controller	24
3.2.3	CONWIP	25
3.2.4	Average velocity	26
3.2.5	Kalman Filter	28
3.3	Variable setpoint method	28
3.3.1	Variable setpoint without disturbance	28

3.3.2	Variable setpoint with disturbances	30
3.3.3	Online change of setpoint	32
3.4	Summary	33
4	Discussion	35
5	Conclusion	38
5.1	Future works	38
	Bibliography	40

1

Introduction

IN manufacturing plants, activities such as handling, buffering and locating procedures are very frequent. These procedures are performed by resources such as pallet and conveyor systems, handling robots and automated guided vehicles. All these kind of resources which receive commands from electronic devices can be grouped together and referred to as Automated Material Handling Systems(AMHSs).

In the design and control of material handling systems there are a lot of issues concerning operational and economical aspects. Operational cost (up to 50%), factory space(up to 55%) and production time(up to 87%), are all factors that can be associated to material handling systems according to [1, 2]. The above statements indicate that there are a lot of opportunities for the improvement of AMHSs. A smooth and gentle handling of material to enhance the system life time, decrease electrical energy consumption and noise produced by handling equipment, and reduce the running costs are important steps towards improving sustainable aspects of a manufacturing plant.

In this report we handle the mentioned sustainable objectives by reducing the chain velocity for a dynamic pallet system while maintaining a desired production rate.

1.1 Background and problem description

FlexLink® AB is a world leading provider of automated production flow solutions, [3]. FlexLink® designs and fabricates conveyor and pallet systems for various industry fields such as automotive, bearings, electronics, foods, tissue, and lab automation. This master project, which is a collaboration between FlexLink® and the Automation group at Chalmers university of technology, mainly focuses on pallet systems.

Pallet systems are automated material handling mechanisms used in modern production environments to transport, buffer, and locate a number of parts. To facilitate the handling and locating actions, the parts are loaded on standard carriers, pallets. A conveyor chain transports the pallet to an operating position close to the machine

conducting an operation. Then, the pallet is fixated by a locating module if the control system specifies an operation for the part on the pallet. After the operation is finished, the pallet is released to follow its route to the next operating point. At the last position in the handling path, the on board part is unloaded, and the empty pallet is directed to the loading section where a new part is loaded on the pallet. Accordingly, the pallets follow a closed loop path, transporting the parts.

The main issues when considering the control of a pallet system is to handle parts in specific positions at specific times. At these positions, the parts can be processed by machines, robots and workers. Finished products are obtained when all operation actions are completed. An improper control of a pallet system can lead to a significant decrease in the production rate (number of parts produced per hour). Hence, it is crucial to design a control strategy which guarantees a desired production rate, while minimizing the handling cost, energy consumption and handling noise. Most control strategies introduced in the literature and practice for pallet systems are based on off-line tuning of controllable parameters such as the number of pallets and chain velocity. However, there is a challenging problem that these strategies cannot properly cope with desired objectives in a dynamic production environment, considering various production scenarios, machine failures, and stochastic operation times.

This master project focuses on this problem and addresses the following objectives:

- A feedback controller for a pallet system should be designed to properly tune conveyor velocity online, regarding design objectives such as desired production(handling) rate, minimum energy consumption, and minimum part waiting times in the conveyor.
- The designed control should be robust under disturbances such as stochastic operation times.
- The control should be implemented in practice and its behavior should verified and validated for a real X85 pallet system.

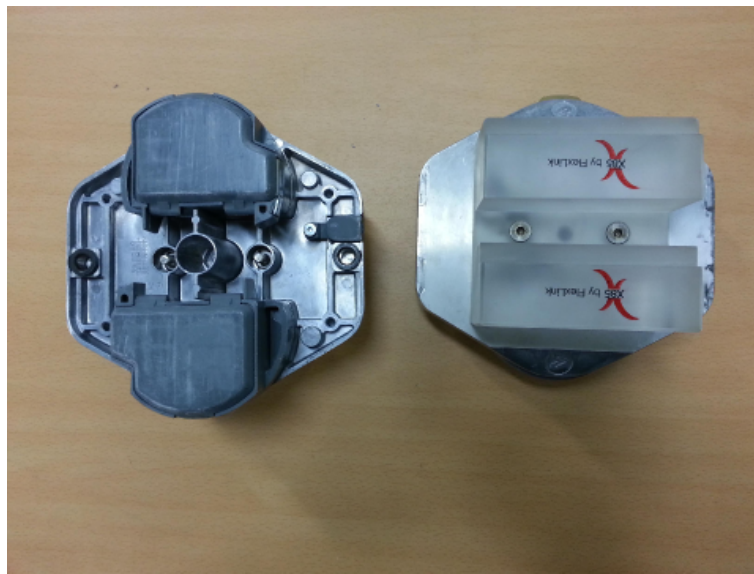
1.2 X85 Pallet System

The X85 pallet system has been developed and constructed by FlexLink® . It is built on configurable standard modules which contribute to the benefit of creating layouts for routing, buffering and positioning of pallets in a simple and fast manner, [4]. The modules of X85, which are used and mentioned throughout this report, are presented in Table 1.1.

The pallets used in X85 pallet system comes in four different sizes and with different properties. The type used in this project can be seen in Figure 1.1. The different properties of a pallet are dependent of the environment in which the pallet system should be used. Factors such as resistance to tough environmental conditions, low friction and high precision locating are all examples of properties that can be demanded for a pallet.

Table 1.1: X85 Pallet modules

Module	Function Description
Pallet	Transports parts through the system
Basic Locating	Locates a pallet for a process and block flow of other pallets
Stop	Stops a pallet for buffering or loading/unloading actions
Combiner	Merges and diverts pallets from two branch lines into one main line, and vice versa

**Figure 1.1:** Pallets used in this project

The plastic part seen on the right pallet in Figure 1.1 demonstrates the location where a product should be placed.

To be able to perform a machining operation on a part mounted on a pallet, the pallet needs to be disconnected from the conveyor and fixated. This operation is performed by the basic locating module shown in Figure 1.2.

Figure 1.2a shows an empty basic locating module, while Figure 1.2b shows the situation with a fixated pallet. The fixating action is performed by lifting the pallet towards a hold-down surface.

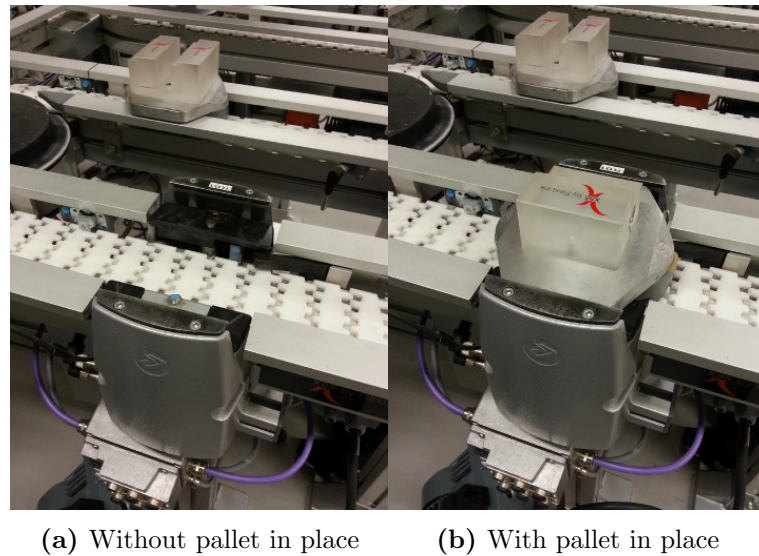


Figure 1.2: Basic locating module in system

To stop pallets at some selected positions along a conveyor line, pneumatic stops are used. Stopping actions can occur to avoid a collision of pallets on paths crossing each other or to buffer pallets for loading, unloading and locating activities. A stop is shown in Figure 1.3.

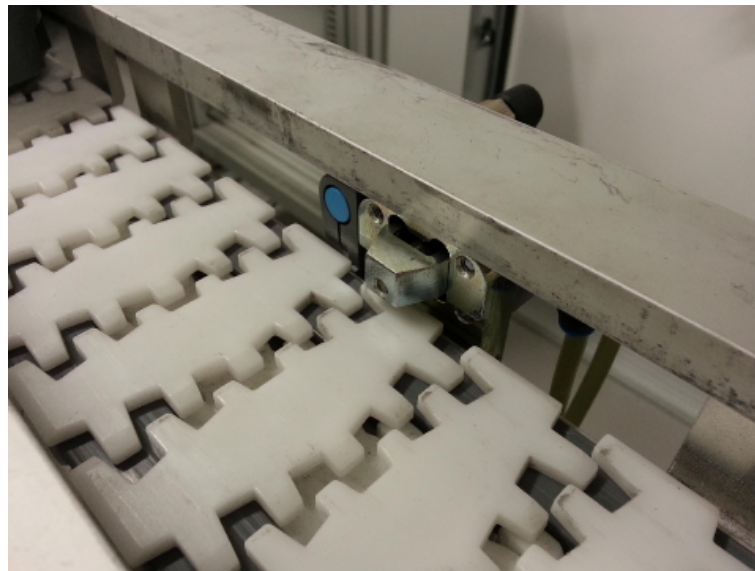


Figure 1.3: Pneumatic stop

To control the routing of the pallets throughout a X85 pallet system, a module called combiner is used. The module, which is shown in Figure 1.4, has an arm that can be

positioned in up to three directions, allowing pallets to follow different paths.

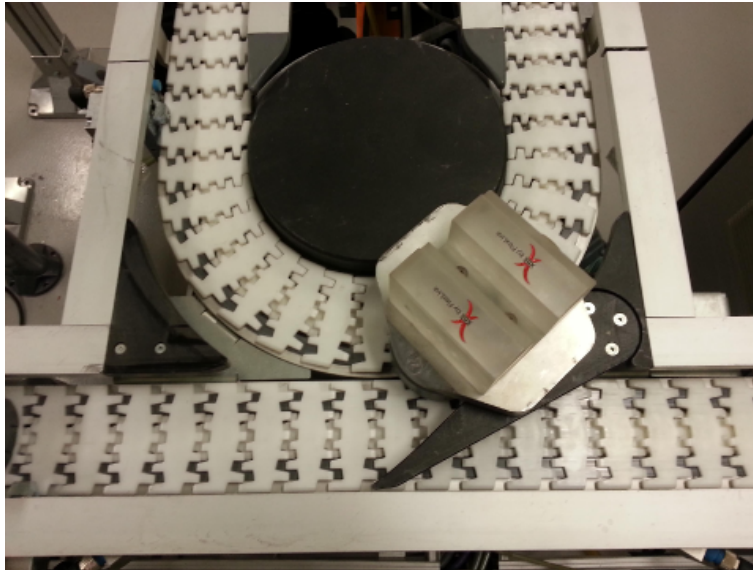


Figure 1.4: Combiner module

Pallets are not removed from a X85 pallet system but a carried part can be unloaded if all its related operations are finished. After unloading of a part, the empty pallet is loaded with a new part and start its journey through the system. This procedure conserves the number of pallets in the system. The described components have been assembled together with conveyor modules to form the complete rig shown in Figure 1.5.

Figure 1.5 shows the X85 pallet system together with the location of the components used. The largest loop in the system, Highway, is connected to the two smaller loops, Satellite 1 and 2, through the combiner modules.

Movement of pallets on the highway occurs clockwise while movement within the satellites occur counter clockwise. A chosen setup for this project uses a process time of 2 seconds for every basic locating module. Initially all pallets are released into the system from a release point located in the top right of Figure 1.5. When the system starts, pallets are released from this point until all pallets are circulating in the system.

1.3 Control Strategy

A block diagram illustrating the setup of the system with the controller and the pallet rig is shown in Figure 1.6. For each loop in the rig, there should be an associated controller to regulate the chain velocity of the loop. Hence, the controller unit in the figure is composed of three distributed regulators for Highway, Satellite 1, and Satellite 2. The control objective is to drive the system to a desired production rate, while minimizing the conveyor velocities as much as possible. To monitor the production rate in each loop, the duration between two consecutive pallet releasing actions in the related basic locating

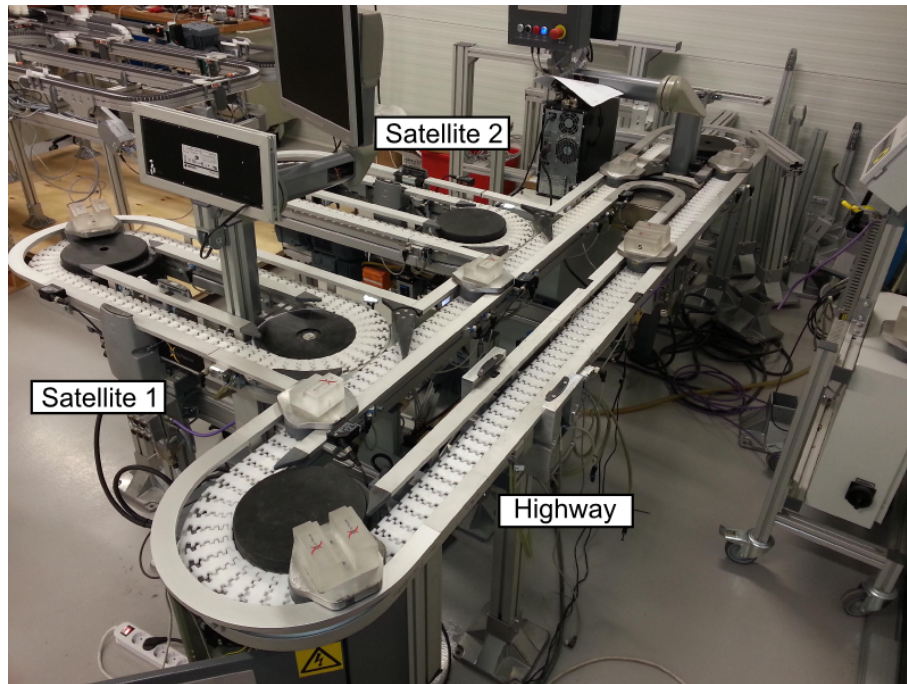


Figure 1.5: X85 pallet system

module(cycle time) is measured and considered as the output($y[k]$ for k :th cycle times). These measurements are then compared with a desired cycle time(setpoint $r[k]$) and the error($e[k]$) from their difference is fed to the controller to regulate the velocity($u[k]$).

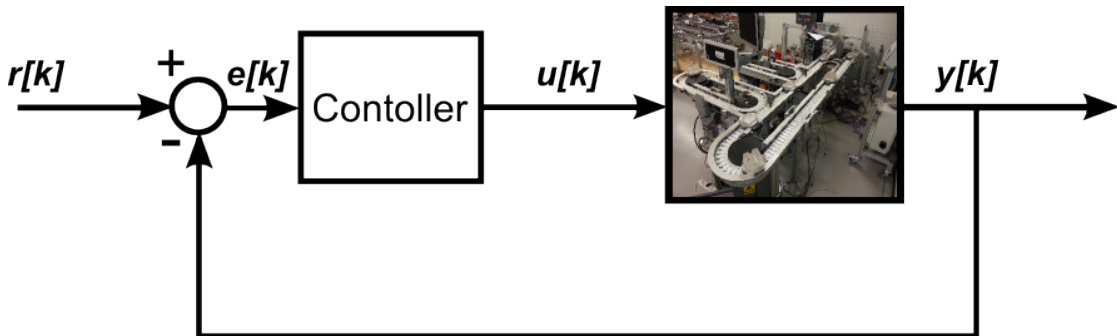


Figure 1.6: Block diagram of the system

In this project, the strategy was to develop the discussed control mechanism in a simulation environment and if the results were promising, the controller would be applied to the real system. The procedure for developing the controller in the simulation environment was considered to be step by step, improving the system performance. In each step after applying a control method, the results from the cycle time and velocities were analyzed to implement a new control method for the next step.

1.4 Limitations of the work

There exists a possibility to introduce components that can write and read tags on the pallets in order to direct different products to different routes through the system. The absence of these products on the provided rig however introduced the restriction for this project, to handle only one product type, following the same route.

2

Controller design based on simulation

IN this chapter, the controller is developed for a simulation model of the X85 pallet rig. The approach is to use three controllers, one for each loop in the system. The controllers are gradually improved using several methods which will be described in the following sections. Each method is described, and the related control block diagram is presented. Furthermore, the figures showing the profile of the cycle time and the chain velocity in time are depicted for each method.

2.1 Simulation model

Based on the described physical rig, a simulation model was created to be used for the development of the controller. The tool used for constructing the simulation model is *FlexLink Design Tool*(FLDT) which is an add-on to *3Dcreate* 2014 ([5]) containing conveyor modules and aids to easily assemble a conveyor model. The resulting model is shown in Figure 2.1.

The behavior and functionality of the models within FLDT were programmed with *Python*([6]) which is an object oriented programming language. This made manipulations and modifications of the model very flexible.

2.2 Initialization

Transition time happens from the moment a pallet system is started until it reaches to a steady-state mode when the system shows a stationary and cyclic behavior under constant profile chain velocities. In this duration which can be long, applying a dynamic speed control is not very advised because the system may show strong variations in velocity (hazard for chain drivers) as well as it may take a considerable amount of time

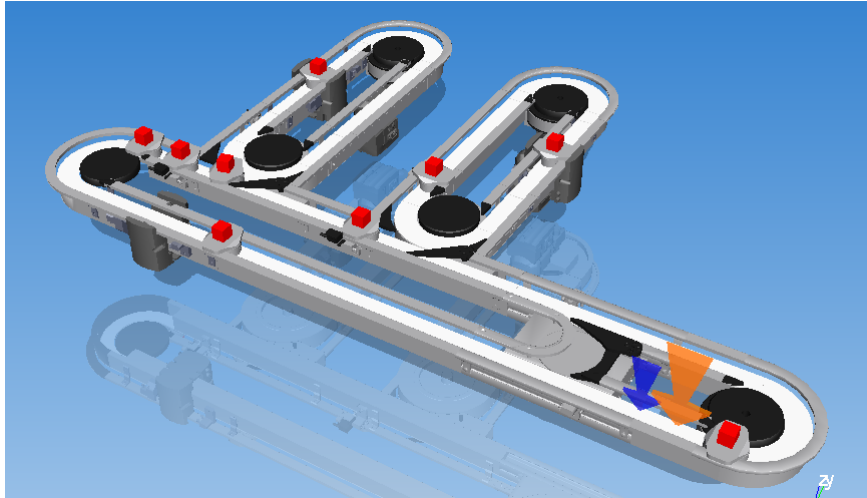


Figure 2.1: Model of the rig created in *FlexLink Design Tool*.

for the velocity to converge to a fixed value. Hence, it is important to initialize a pallet system properly to shorten the transition time. Proper tuning of the number of pallets and initiating cyclic behavior in the loading section are two approaches used in the simulation to find a reasonable choice of initial state for the system. These approaches are explained below.

2.2.1 Tuning of the number of pallets

An equation relating the number of pallets in the system, the cycle time and the time it takes for one pallet to go through the system was used([4]). The equation is as follows

$$C * N_p = T_p + T_t + W \quad (2.1)$$

Where,

- N_p is the number of pallets in the loop.
- C is the desired cycle time [s].
- T_p is the summarized time of all processes that the pallets pass through in the loop.
- T_t is the transportation time, calculated by L/V_c , where L is the length [m] of the loop and V_c is the speed [m/s].
- W is a waiting time, which in this case, is assumed to be zero.

There existed a constraint to only run with velocities between 5 and 35 m/min on the motors to avoid damaging the physical system. The procedure was thus to choose

a value between 5 and 35(14 for Highway and 12,5 for satellites) and calculate a fitting number of pallets needed in the system as follows

$$7 * N_p = 2 + \frac{6,1185}{14/60}$$

$$N_p \approx 5$$

Solving N_p and rounding upwards to the closest integer indicated that the highway needed five pallets. The corresponding calculations for the satellites resulted in two pallets for each satellite. The total number of pallets inserted in the system was thus nine. The chosen velocities for calculating the number of pallets could have been some other value within the constraint as well. The important thing was to calculate a fitting amount of pallets corresponding to the chosen initial velocities.

2.2.2 Cyclic loading section

The system was initialized with a distribution of pallets to force the system into a cyclic behavior faster. If all the pallets would have been released together in a queue, it would take some time for the system to reach a cyclic behavior. Hence a choice to release the pallets with an interval of six seconds was made. As for the chosen initial velocities, this choice ensured that the system was in the same state for every test case. The initial velocities and the corresponding number of pallets could equally have been chosen to some other values within the allowed range.

2.3 PID

A Proportional, Integral, and Differential (PID) controller is the most common form of feedback regulator which is widely used in industrial applications([7]). Due to the convenience of designing and implementing the PID regulator and its history of success in real applications, a PID controller has been selected as the key method for the regulation of the chain speed. The standard equation for a PID controller is as follows

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{d}{dt} e(t) \quad (2.2)$$

where $u(t)$ is the control variable (chain velocity) and K_p , K_i , and K_d are gains for proportional, integral, and differential parts acting on the error value, $e(t)$, (the difference between the desired and the obtained cycle time). These gains should be tuned properly to obtain a desired control response.

Since the output of the pallet system for each loop is discrete (controller is event oriented and gathers a new cycle time every time a new pallet is released from the basic locating module), a discrete PID which is Z-transform of (2.2) has been considered as the following.

$$U(z) = \left[K_p + \frac{K_i}{1 - z^{-1}} + K_d(1 - z^{-1}) \right] E(z) \quad (2.3)$$

rewriting (2.3) gives

$$U(z) = \left[\frac{(K_p + K_i + K_d) + (-K_p - 2K_d)z^{-1} + K_dz^{-2}}{1 - z^{-1}} \right] E(z) \quad (2.4)$$

From (2.4), the following entities can be defined

$$K_1 = K_p + K_i + K_d$$

$$K_2 = -K_p - 2K_d$$

$$K_3 = K_d$$

Then, (2.4) can be rearranged and transformed to a difference equation

$$u[k] = u[k - 1] + K_1e[k] + K_2e[k - 1] + K_3e[k - 2] \quad (2.5)$$

The setup of the system at this stage is illustrated in Figure 2.2.

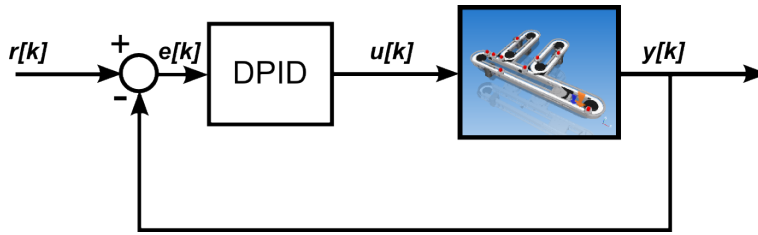


Figure 2.2: Block diagram illustrating the setup of the system

The input to the system, $r[k]$, is the desired cycle time chosen to be 7 seconds. The control signal $u[k]$ is the velocity sent to the motor, and the output $y[k]$ is the measured time between releasing two pallets from a basic locating module. The tuning of the DPID gain was done manually, and a wide range of values was investigated. The resulting behavior of the velocity for all these cases were quite similar. Figure 2.3 shows the chain velocity manipulated by the DPID controller and bounded between 5 m/min and 35 m/min.

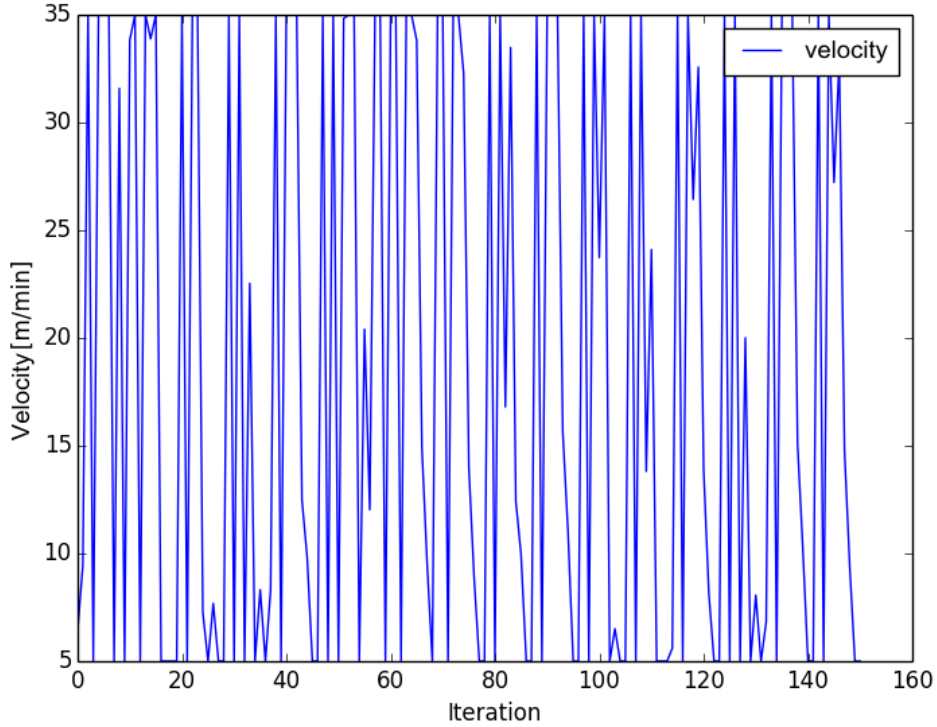


Figure 2.3: Velocity pattern controlled by DPID

The velocity oscillates between the upper and lower bounds as shown in the figure. This behavior was not desirable so further improvements had to be investigated.

2.4 Velocity Averaging

The control signal to the motor (denoted $u[k]$ in Figure 2.2) should be smoother in order to prevent damaging the motor and gain a satisfying behavior. To do this, the recursive function

$$V_k = \frac{V_{k-1} * (N - 1) + u_k}{N} \quad (2.6)$$

was used to calculate the average velocity of all obtained velocities until the current iteration k . Since the test performed in this project consisted of up to 150 released pallets, the method can be interpreted as using a window of the latest 150 values for calculating an average velocity (a modified version of this function have been used when "tuning" the length of the vector, to test other ranges than all previous values). The notations used in this equation are as follows:

- V_{k-1} is the average velocity of all calculated velocities until the iteration $k - 1$

- N is the number of iterations that have occurred.
- u_k is the velocity calculated by DPID for the current sample.

Figure 2.4 shows the modified block diagram including the averaging velocity method. The block marked with "AV" is where the method is applied and its output is fed to the plant.

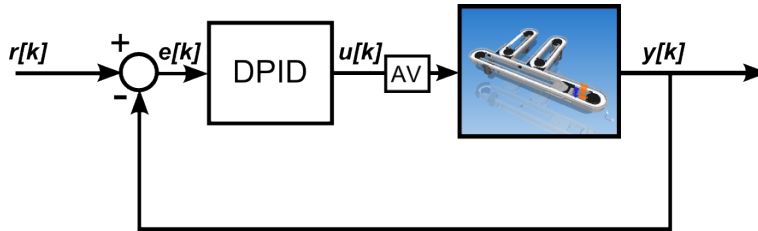


Figure 2.4: Block diagram including the velocity averaging method

The impact of AV method on the chain velocity is shown in Figure 2.5. Comparing to Figure 2.3, a significant improvement on reducing the variation of chain velocity is obtained. However, the velocity pattern is not yet smooth enough to be applied in practice for pallet systems. Hence, further improvement was needed to be able to reach a desired cycle time with a more stable behavior of the velocity.

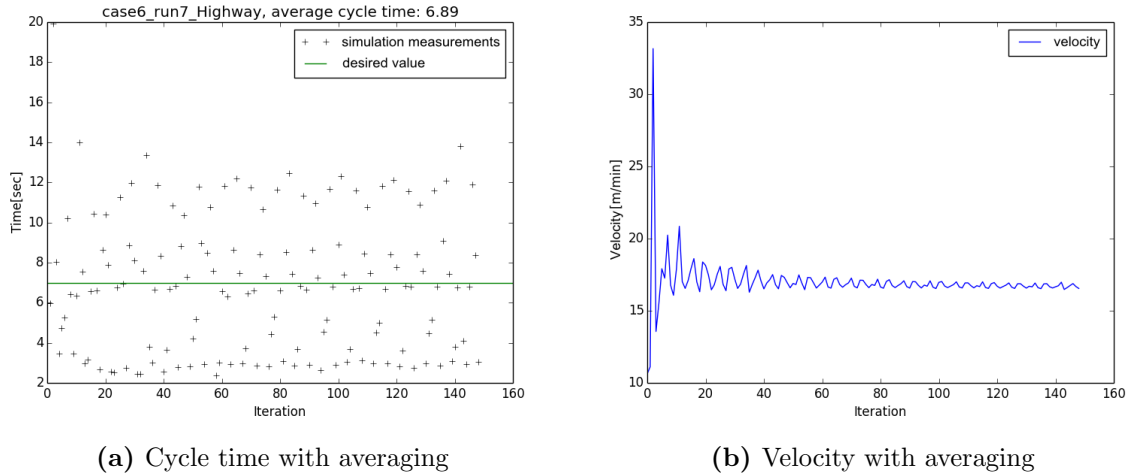


Figure 2.5: Velocity pattern manipulated by DPID controller with averaging method

2.5 CONWIP

CONstant Work-In-Process (*CONWIP*) control stands for an operational mechanism which constantly conserves the same number of parts in a production system. This

mechanism is a pull-based system which keeps a constant work in process by authorizing the entry of a new part to the system provided that another one exits the system ([8]).



Figure 2.6: CONWIP

The principles of this control are illustrated in Figure 2.6, where the nodes (q_0 - q_3) represent four work stations on a production line. The last station sends a signal requesting a new part for the line when a finished part leaves the line. The first station receives a new part as the response to the request without considering how many individual processes are involved in the line.

For the simulation, CONWIP control was individually applied for each loop. Technically, this was realized by forcing a certain number of pallets to always be in each satellite. The number of pallets has been calculated in Section 2.2. The highway should contain exactly five pallets and each satellite should contain two pallets. One pallet can leave a satellite if another pallet is available to enter the satellite. This mechanism contributes to a more evenly distribution of pallets in the system although it introduces some more queuing and waiting behavior for pallets.

The role of CONWIP control is depicted in the block diagram 2.7. In this block diagram, CONWIP mechanism is considered as a property of the plant rather than a part of the controller due to the fact that CONWIP has been directly implemented in the simulation environment *FlexLink Design Tool* (not in the external control python codes).

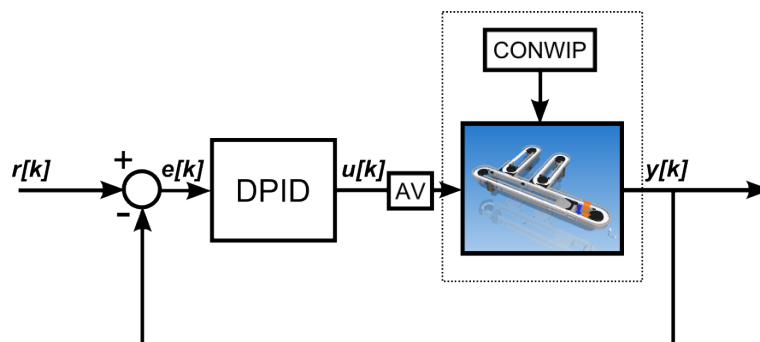


Figure 2.7: Block diagram illustrating CONWIP control applied to the system

The following figures indicate the cycle time and velocity measurements for Highway after applying CONWIP control.

Figure 2.8a shows that the cycle time measurements become more evenly distributed compared to Figure 2.5a in which the measurements are highly scattered. CONWIP

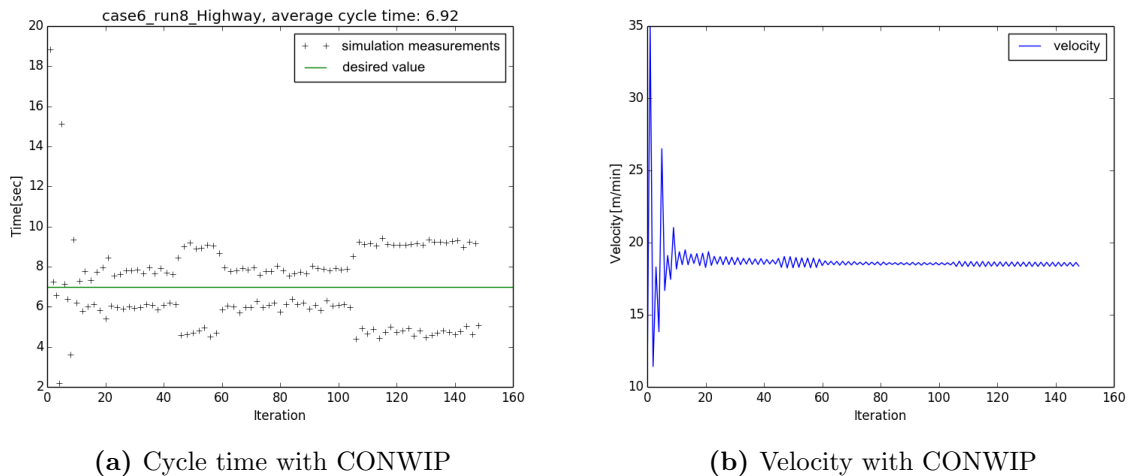


Figure 2.8: Plots with CONWIP implemented

helps the system and its output to be more stable and as a result the calculated average velocity that is shown in Figure 2.8b, to be less variant compared to Figure 2.5b. However, there are still strong fluctuations in the velocity in the early phase of running the system (at least until 20 iterations of releasing parts). Thus, the control strategy shown in Figure 2.7 should be more enhanced to be used for the physical system.

2.6 Kalman

The oscillations seen in figure 2.8b can be related to the distribution of measurements around the desired cycle time. A smoother behavior of the velocity might be achieved by using the average of the measurements as feedback to the controller. To estimate proper weights for this average, a Kalman filter was used. The Kalman filter is an efficient recursive method applying on the stream of measurements to optimally estimate the state(s) of a process even though the nature of the process is not precisely known([9]).

In each loop, pallets in the system arrives to the basic locating module with different interval times (cycle times). Spaces among various pallets are varied over time since a change in the velocity affects the distance between a moving pallet and one that is held at a locating module or in a queue. Assuming that the varying spaces have gaussian patterns, a Kalman filter was designed based on a simple periodic model of the system to estimate the average cycle time and feed it back to the DPID.

Before designing the filter, a simple and linear model of the pallet system was considered as follows

$$\begin{bmatrix} x_k^0 \\ x_k^1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1}^0 \\ x_{k-1}^1 \end{bmatrix} + w_k \quad (2.7)$$

$$z_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_k^0 \\ x_k^1 \end{bmatrix} + v_k \quad (2.8)$$

This model relates a pallet release time(from the basic locating module) at the iteration k , x_k^0 , based on the previous release time, x_{k-1}^0 , and cycle time x_{k-1}^1 under the process noise w_k . Furthermore, it assumes a cyclic behavior for the system ($x_k^1 = x_{k-1}^1$) and emphasizes on the fact that only release times are measured(z_k) directly and subjected to the measurement noise v_k . Some explanation can be made on the absence of the control signal in this model, which in a common Kalman filter should exist. As mentioned, the model assumes a stationary and cyclic behavior for the system and in that case, the control signal(chain velocity) remains constant.

Based on this model the prediction and update steps for the Kalman filter can be written as follows

$$\hat{x}_k^- = A\hat{x}_{k-1}^-; A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \hat{x}_k^- = \begin{bmatrix} x_k^0 \\ x_k^1 \end{bmatrix} \quad (2.9)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (2.10)$$

where P is the estimated error covariance and Q is the process noise covariance matrix. After the prediction step, the update step takes place as follows

$$K_k = \hat{P}_k^- H^T (H\hat{P}_k^- H^T + R)^{-1} \quad (2.11)$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (2.12)$$

$$P_k = (I - K_k H)P_k^- \quad (2.13)$$

where K_k is the Kalman gain at time step k , R the measurement noise covariance matrix and H is the matrix used in Equation 2.8. The estimate of the release time, \hat{x}_k^- , is subtracted from the real measurements, z_k , and weighted with the Kalman gain to produce the estimate \hat{x}_k . For further information about the Kalman filter the reader is referred to [9]. Figure 2.9 shows how the Kalman filter was implemented in the system. The signal $y[k]$ is fed back through the Kalman filter which estimates the average cycle time.

Even though the real measurements of the cycle time were noisy the Kalman filter provided a good estimate of the average cycle time(blue line) seen in Figure 2.10a(Highway), 2.11a(Satellite 1) and 2.12a(Satellite 2).

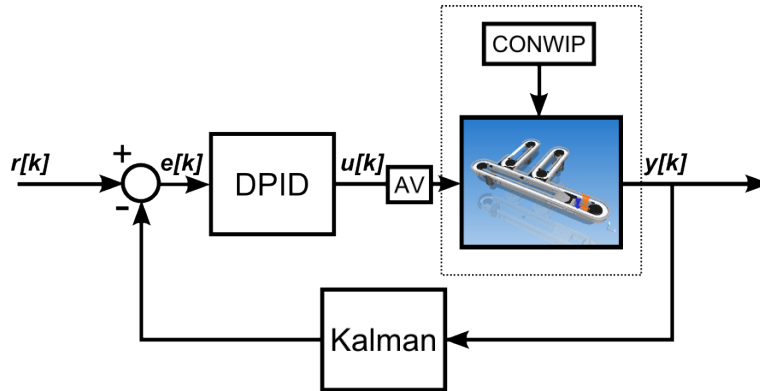
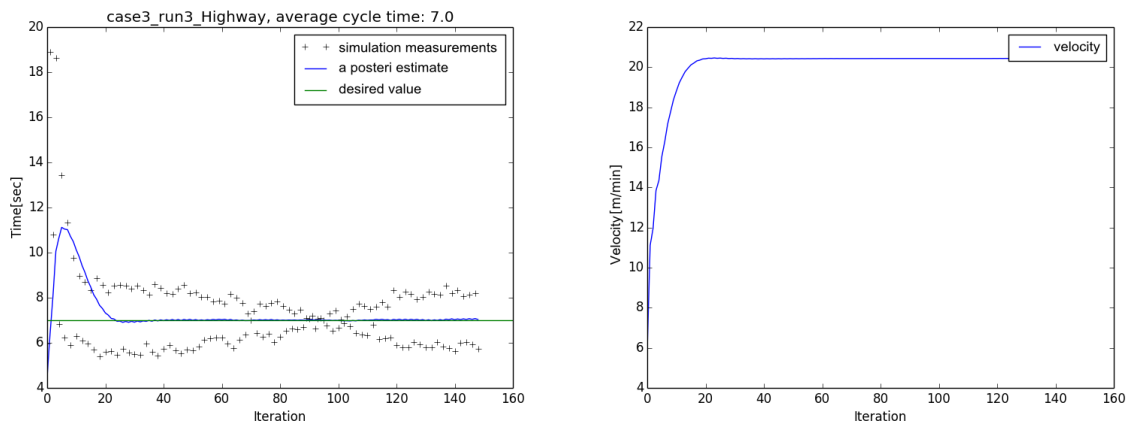


Figure 2.9: Block diagram describing the setup of the system with Kalman filter



(a) Cycle time using Kalman

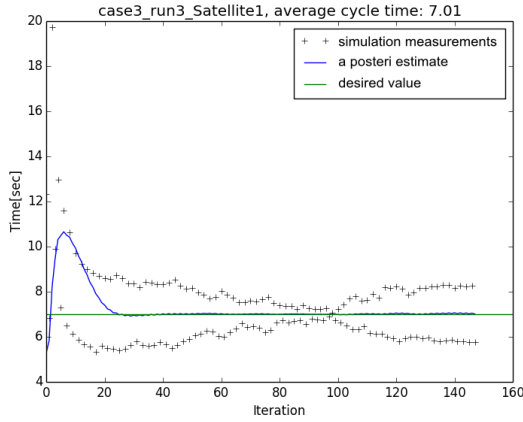
(b) Velocity plot using Kalman

Figure 2.10: Plots with Kalman filter implemented

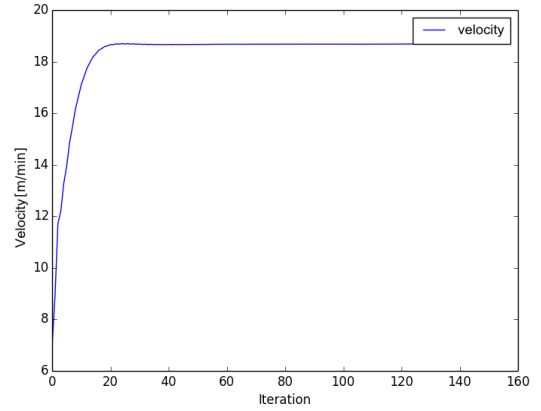
Using this technique resulted in a smoother velocity shown in Figure 2.10b, 2.11b and 2.12b. The fluctuations in the acceleration/deceleration had in this scenario been vastly decreased and the controller was considered stable enough to continue the testing on the physical rig.

2.7 Variable setpoint method

At *FlexLink*®, an algorithm has been developed to make the controller more sensitive and faster to respond to variations and uncertainties in the system. The idea of the variable setpoint method is to change the controller setpoint when the cycle time measurements are within a predetermined range from the desired one. This creates a bigger difference between the measurements and the setpoint, which in turn gives a larger control signal and a stronger reaction in the control signal. If the measurements are outside

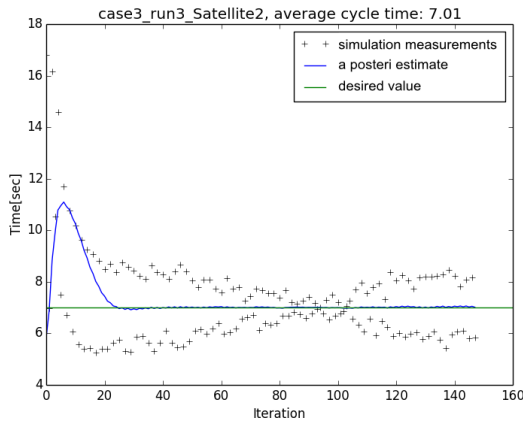


(a) Cycle time using Kalman, Satellite 1

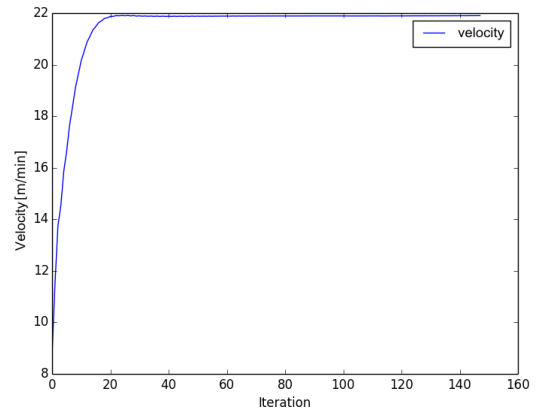


(b) Velocity plot using Kalman, Satellite 1

Figure 2.11: Plots of Satellite 1 with Kalman filter implemented



(a) Cycle time using Kalman, Satellite 2



(b) Velocity plot using Kalman, Satellite 2

Figure 2.12: Plots of Satellite 2 with Kalman filter implemented

the predetermined range the method resets the setpoint to its initial value. The variable setpoint method is presented in Algorithm 1

This method promotes less dependency for good tuning of DPID and better counteractions for disturbances, but at the same time a higher sensitivity is introduced in the controller. To show one of the benefits of the variable setpoint method some test cases were conducted with poor tuning of DPID gains. This was done to present the difference between the system behavior with and without the variable setpoint method implemented in the controller. The altered parameters for these test cases are seen in Table 2.1.

The first case was to run the system with the poorly tuned parameters and without

Algorithm 1 Variable setpoint method

```

Calculate error between current_setpoint and Kalman filter estimation
if |error| ≤ predetermined range then
    current_setpoint = error + predetermined_setpoint
else
    current_setpoint = predetermined_setpoint
end if
change controller setpoint to current_setpoint
    
```

Control Parameters		
K_p	K_d	K_i
1.2	1.5	2.82

Table 2.1: Parameters used in the DPID controller

the variable setpoint method.

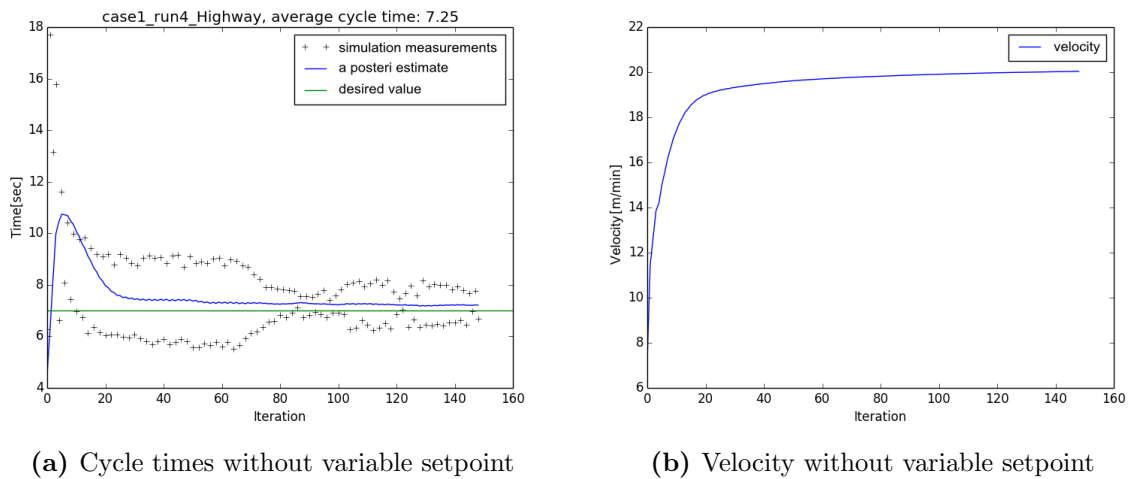


Figure 2.13: Plots with badly tuned controller without variable setpoint

Figure 2.13 shows the cycle time and velocity with a poorly tuned controller which result in an offset from the desired cycle time(7 seconds). Introducing the variable setpoint method with the bound value of 1 gives a better accuracy of the cycle time even though the DPID parameters are poorly tuned. The plots of the cycle time and velocity are shown in Figure 2.14.

Increasing the bound to three and five give the plots shown in Figure 2.15 and 2.16.

It can be seen that the larger the value of the bound is, the more sensitive the controller becomes, showing more oscillating patterns on both the average cycle time and the velocity.

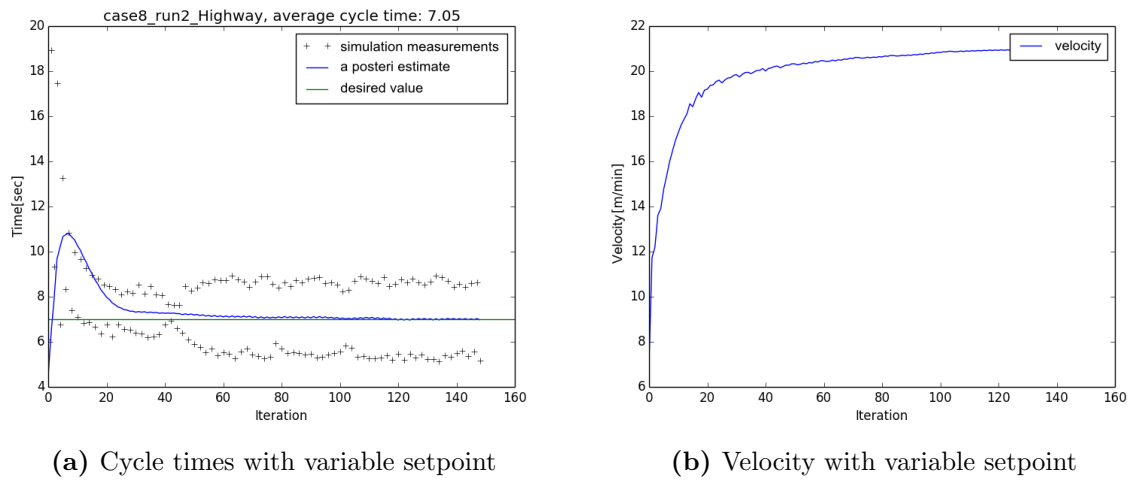


Figure 2.14: Plots with badly tuned controller using variable setpoint with bound of 1

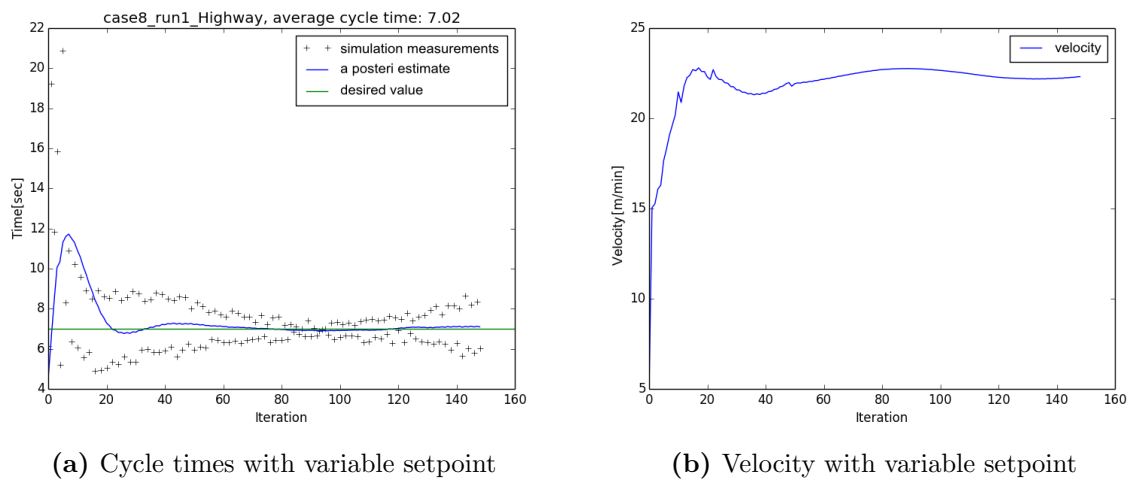
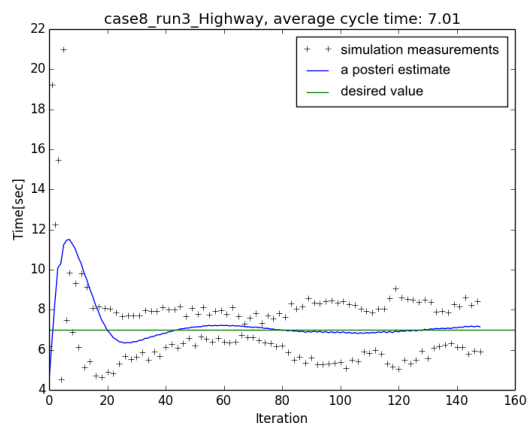


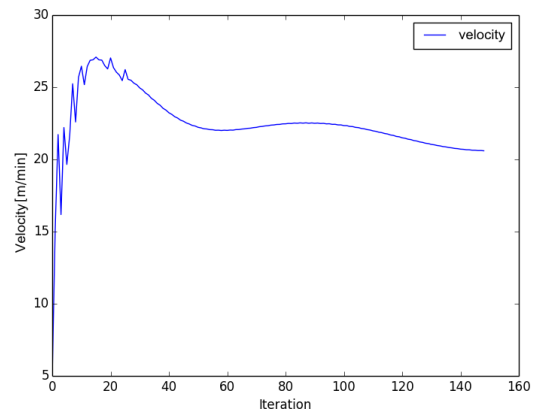
Figure 2.15: Plots with badly tuned controller using variable setpoint with bound of 3

2.8 Summary

The steps taken in the simulation environment have all been presented here. The initial part consists of the construction of the simulation model and initialization of the model to an initial state, followed by a description and results of several methods added to improve the controlling behavior. Including all the described methods (in this chapter) in the controller, the system behavior was considered to be stable enough to apply the controller to the physical system. The implementation on the physical system is presented in the next chapter.



(a) Cycle times with variable setpoint



(b) Velocity with variable setpoint

Figure 2.16: Plots with badly tuned controller using variable setpoint with bound of 5

3

Implementation

ONCE the controller had been developed in the simulation environment it was applied to the real physical system. A few changes, that were needed for the implementation in the physical rig, are described. Then, results from different test cases are presented. To avoid damaging the system, all methods mentioned in Chapter 2 were integrated in the controller. The necessity of each method was then verified by conducting a set of experiments where one method was excluded for each experiment.

3.1 Modification of physical system

To be able to reconstruct the behavior of the simulation model, on the physical rig, a number of factors had to be taken into consideration. The two major factors are described in this section.

3.1.1 Connecting PC and PLC

To be able to control the physical rig with the external controller on the PC, a connection between the PC and the PLC should be established. A common way to establish this connection is by using an OPC server/client setup. Due to the long trial period and experience at *FlexLink*® Matrikons OPC server([10]) was selected. The OPC server functionality was verified by reading and writing to chosen signals(such as velocities) used in the PLC.

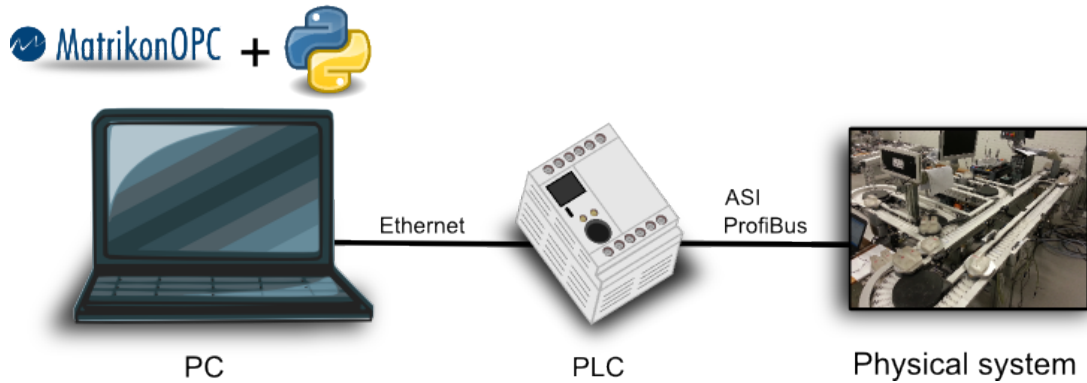


Figure 3.1: Schematic figure of the connection between PC and the physical system.

The controller was then connected directly to the OPC server, using the open source project OpenOPC for *Python*. This provides an easy way to create a communication link to an OPC server([11]) with *Python*. Figure 3.1 shows the connection between the physical system and the PC containing both the controller code and the OPC server.

3.1.2 PLC code

The behavior of the simulation model which is programmed using *Python* had to be copied to the physical system where the behavior is programmed using PLC code. The significant modifications that had to be implemented in the original code were the property of CONWIP and the ability to read cycle times.

3.2 Controller performance

In this section, the results from the experiments on the real rig are presented by figures showing both cycle times and velocities. Before these experiments were conducted, tuning of the DPID gains had to be done in order get a satisfying behavior.

3.2.1 Tuning of DPID gains

Similar to the tune procedure in the simulation, tuning for the real system was also done by trial and error.

Control Paramters		
K_p	K_d	K_i
1.2	3	5.82

Table 3.1: Parameters used in the DPID controller on the rig

Table 3.1 gives the values of the gains providing a desirable behavior for the system. From now on, these values are the ones used for test cases on the physical rig.

3.2.2 Complete controller

The following plots show the performance of the highway and satellites, using the controller implemented in the previous chapter.

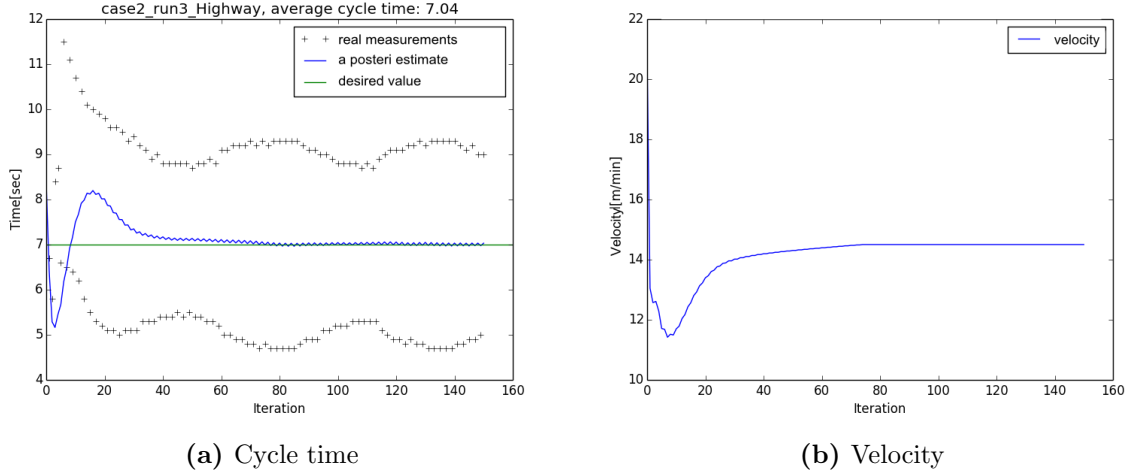


Figure 3.2: Plots showing the controllers performance on Highway

Figure 3.2 shows that the controller for Highway is able to steer the average cycle time to the desired value while maintaining a smooth behavior of the velocity.

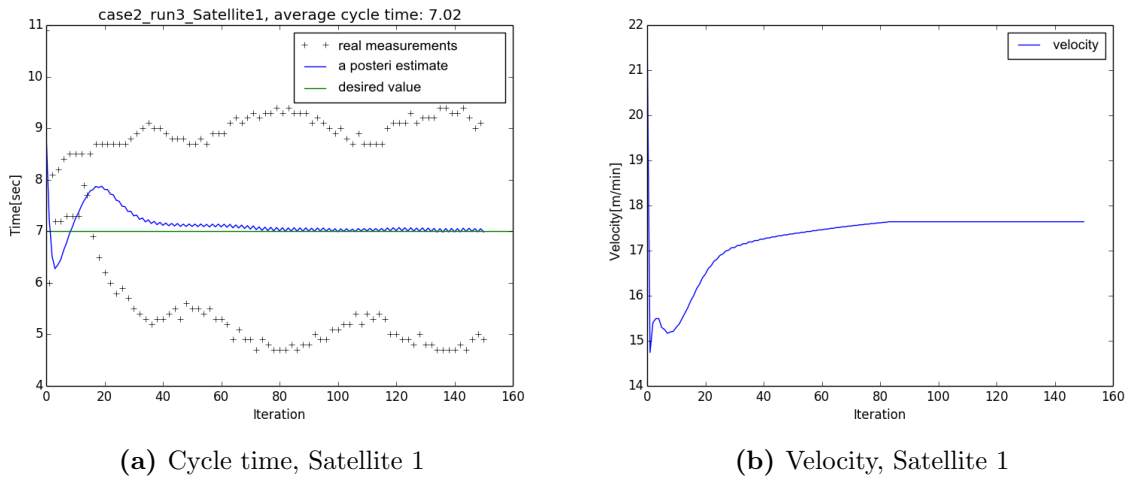


Figure 3.3: Plots showing the controllers performance in Satellite 1

The plots shown in Figure 3.3 and 3.4 present the results obtained from the Satellites 1 and 2. The behavior of the both Satellite 1 and 2 are very similar to that of Highway but with slightly higher velocities. For the following subsections the impact of each method(defined i Chapter 2) on the real controller performance is discussed. In these

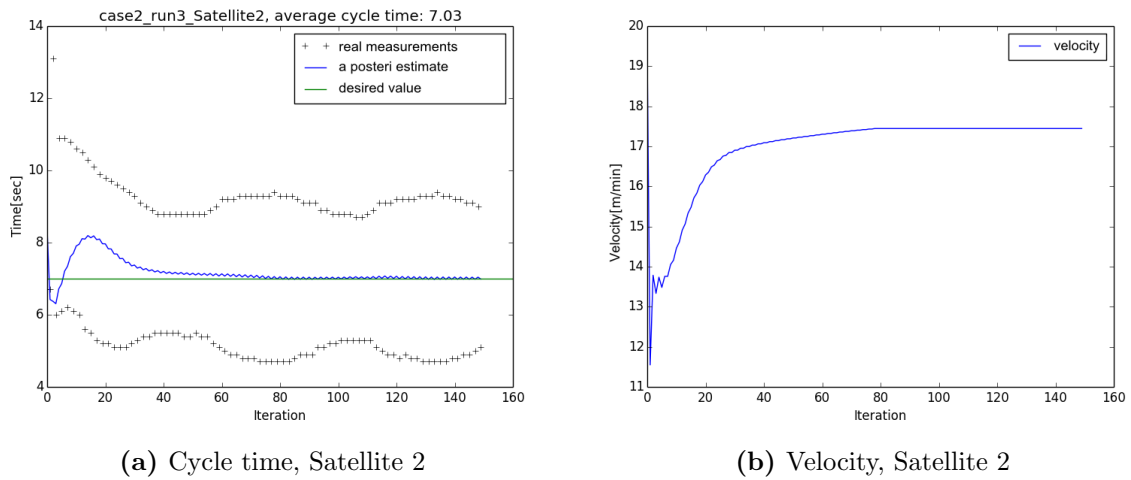


Figure 3.4: Plots showing the controllers performance in Satellite 2

discussions, due to the similarity of the controller performance for Highway and Satellite 1 and 2, only the resulting figures for Highway are demonstrated.

3.2.3 CONWIP

CONWIP was the first method which impact on the controller was investigated. As mentioned in Section 2.5, this method is mainly needed for the distribution management of the pallets. The proper distribution of pallets can effectively reduce the scattering of cycle times.

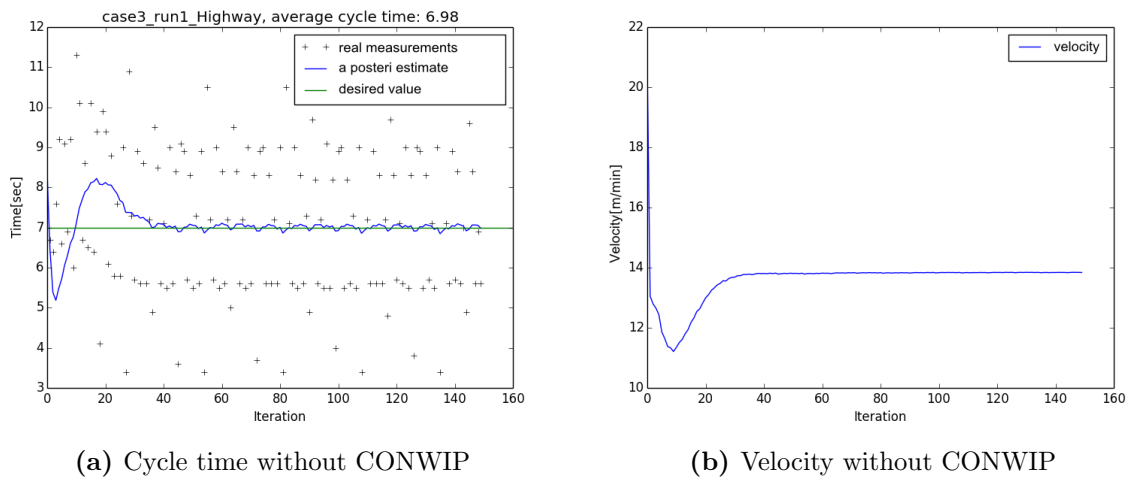


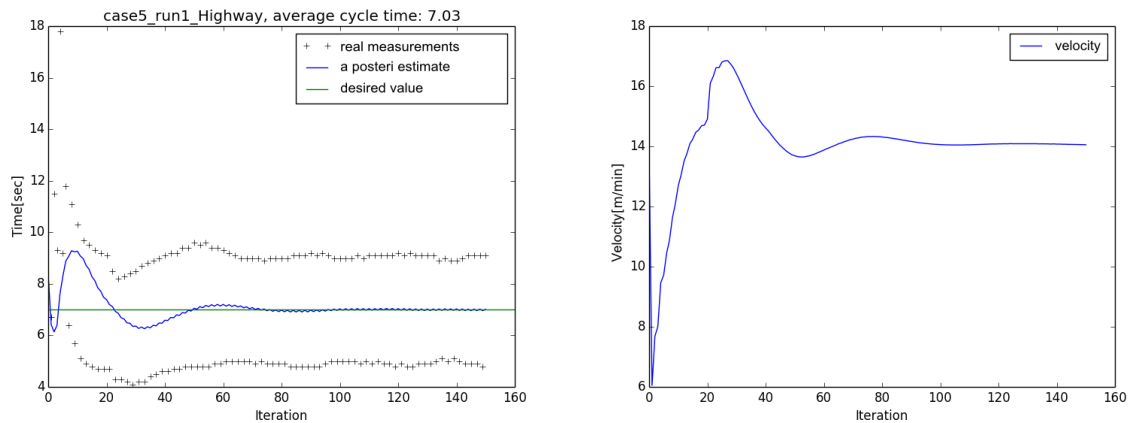
Figure 3.5: Plot with CONWIP disabled

Figure 3.5 illustrates the performance of the Highway without using CONWIP. The

cycle time measurements are very scattered around the setpoint of 7 seconds compared to 3.2a. The velocity still shows a smooth behavior with a value slightly lower than that shown in Figure 3.2b. The reason for the lower velocity is that CONWIP introduces waiting times in the system.

3.2.4 Average velocity

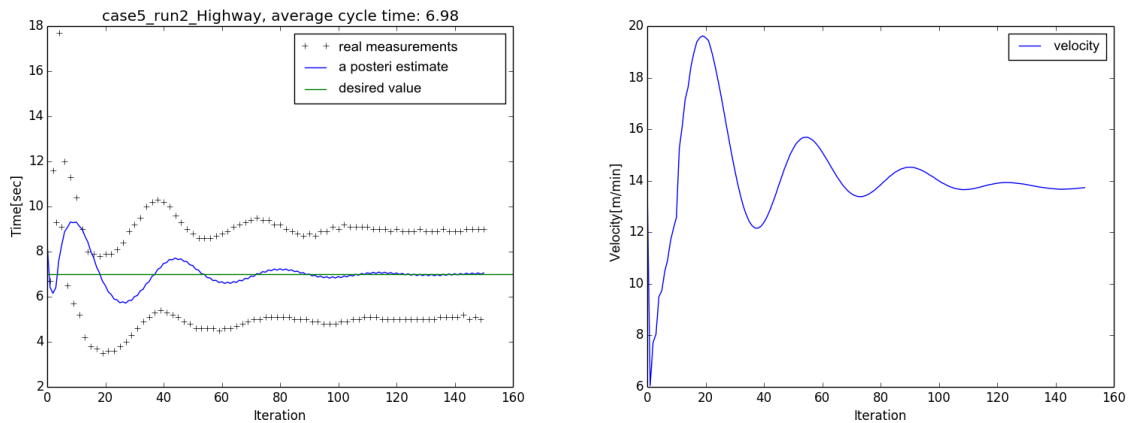
The implementation of averaging of velocity during the control design phase seemed crucial for the stability of the system. Due to safety reasons related to the physical system, the method was not completely removed while conducting test cases. The modifications made to prove the importance of this method was to test different ranges of previous values for calculating the average velocity.



(a) Cycle time with 20 values for average velocity (b) Velocity with 20 values for average velocity

Figure 3.6: Average velocity method using 20 values for Highway

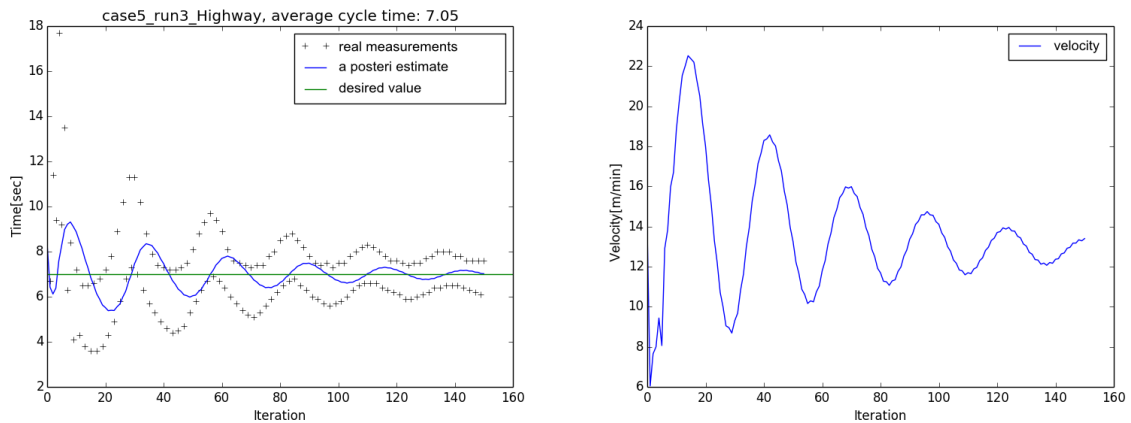
Figure 3.6 shows the performance of the system when the average velocity method only uses the last 20 values. The behavior of the velocity is more oscillating than the one presented in Section 3.2.2 for Highway.



(a) Cycle time with 10 values for average velocity (b) Velocity with 10 values for average velocity

Figure 3.7: Average velocity method using 10 values for Highway

The next range is the 10 latest values for the average calculation. The obtained results can be seen in Figure 3.7. Both the estimated cycle time and the velocity are more oscillating than those with a range using 20 values.



(a) Cycle time with 5 values for average velocity (b) Velocity with 5 values for average velocity

Figure 3.8: Velocity average method using 5 values

The last range is the 5 latest values. As shown in Figure 3.8, the results demonstrate highly oscillating behavior for both the cycle time and the velocity. At this point, it is possible to see a clear trend of what happens when the range of the average calculation is decreased. The lower the range, the more unstable the system becomes.

3.2.5 Kalman Filter

Implementation of the Kalman filter in Section 2.6 showed that the feedback of the average cycle time effectively stabilized the velocity. Removing the Kalman filter means that the real measurements are fed back to the controller instead of the estimated average cycle time. The result of this elimination is demonstrated in Figure 3.9b for the velocity.

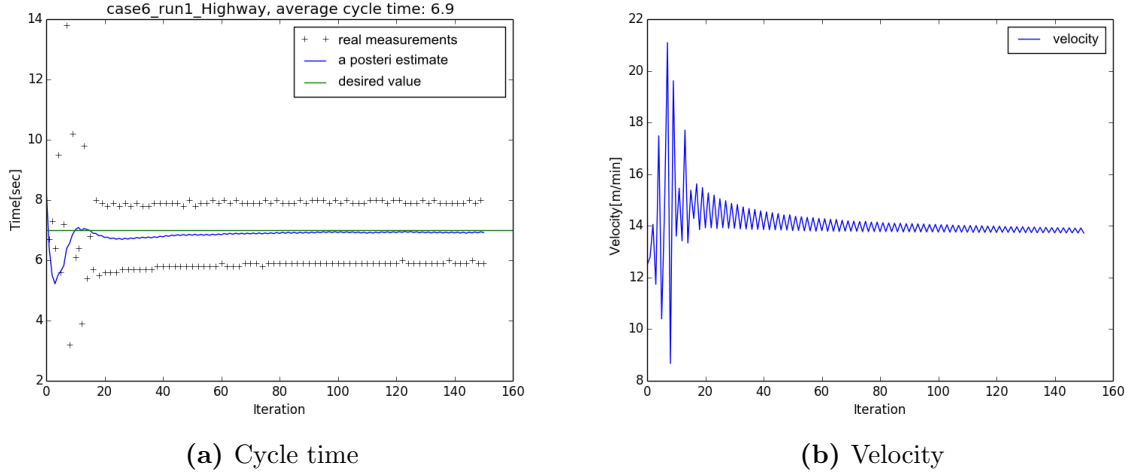


Figure 3.9: Characteristics of Highway when the Kalman filter does not give any feedback

The velocity averaging method makes the magnitudes of the oscillations shown in Figure 3.9b smaller as time passes but the behavior is not desirable. In Figure 3.9a there is a blue line showing the Kalman estimations for the cycle times. These estimations are not fed back to the system but are only plotted to show the estimated average of the measurements.

3.3 Variable setpoint method

In this section, results are presented using the variable setpoint method with and without added disturbance, to verify its functionality in the physical system. The ability of the system to handle online changing of the setpoint is also presented.

3.3.1 Variable setpoint without disturbance

Before exposing the system to disturbances, its control functionality with variable setpoint had to be investigated on the physical rig. The following plots shows the impact of using variable setpoint, with different bounds, without disturbances in the physical system.

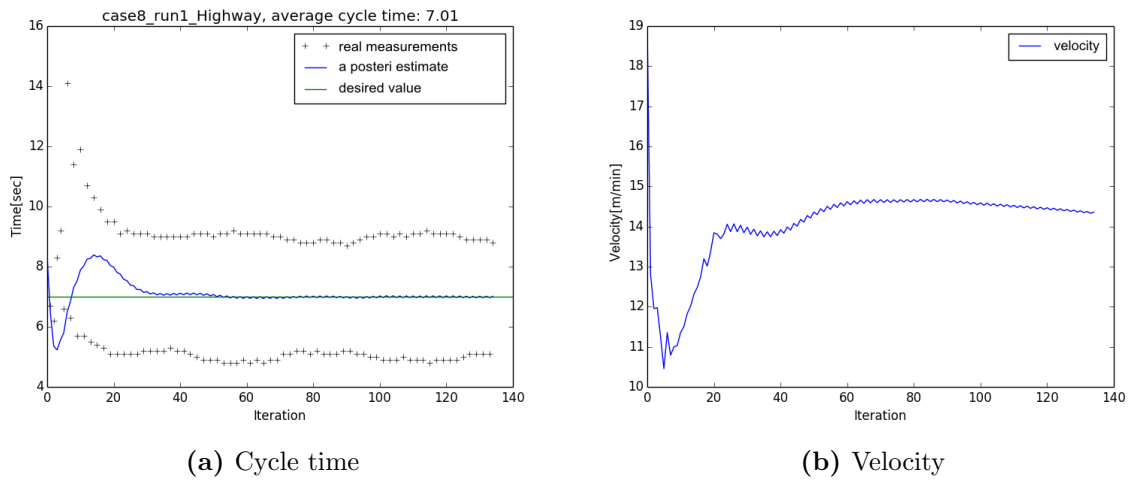


Figure 3.10: Highway performance with variable setpoint of bound 1

Figure 3.10 shows the system using a variable setpoint with a bound of 1 without disturbances added. The cycle time converge slightly faster than Figure 3.2a, but the related velocity shows a more sensitive behavior.

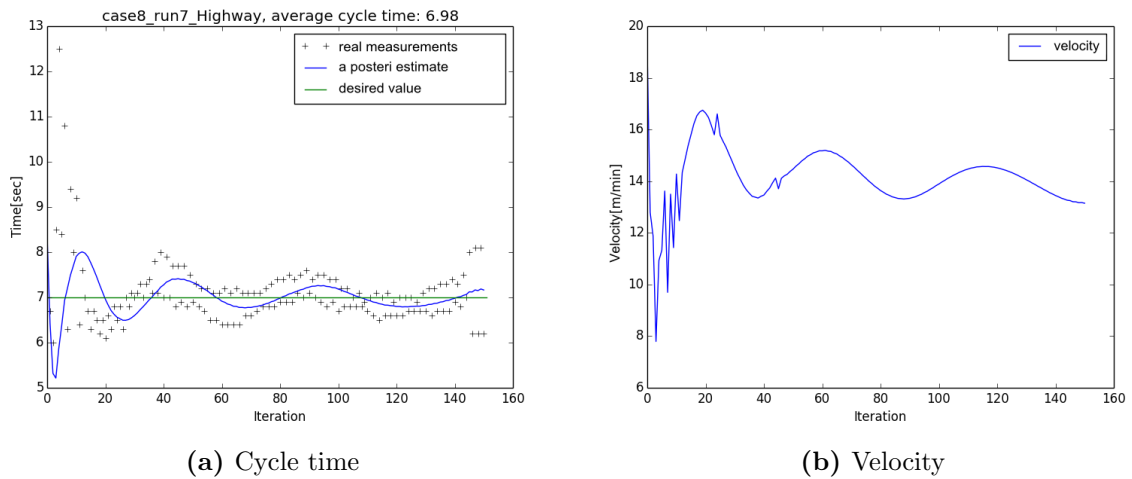


Figure 3.11: Highway performance with variable setpoint of bound 3

When the error bound of the variable setpoint method was increased both the cycle time and velocity, showed oscillations in their patterns, although the average cycle times were still in a desired range. These patterns are demonstrated in Figure 3.11 and 3.12 for the bound 3 and 5, respectively.

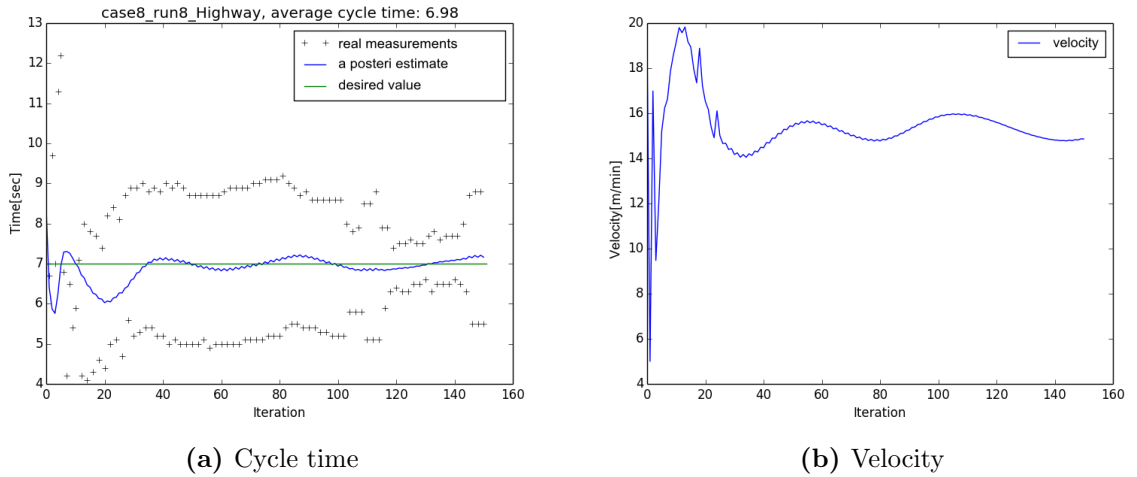


Figure 3.12: Highway performance with variable setpoint of bound 5

3.3.2 Variable setpoint with disturbances

By manually moving and holding pallets in the system, strong stochastic disturbances were introduced. These disturbances were active during 50 to 120 iterations of passing pallets through the basic locating modules. The first test case was conducted with zero error bound for the variable setpoint method followed by more test cases with the error bounds of 1, 3 and 5. The obtained result from these test cases are shown in Figures 3.13 to 3.16, respectively.

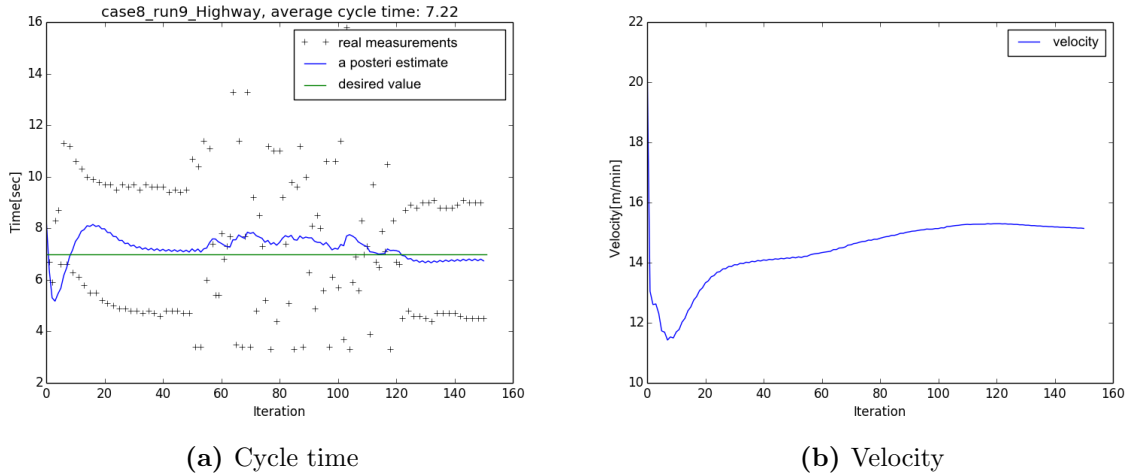


Figure 3.13: Highway performance under disturbances with variable setpoint of bound zero

In Figure 3.13a the average cycle time during the iterations exposed to the distur-

bances, is about 7.38. In Figure 3.13b, the velocity is smooth and does not react much to changes in the system.

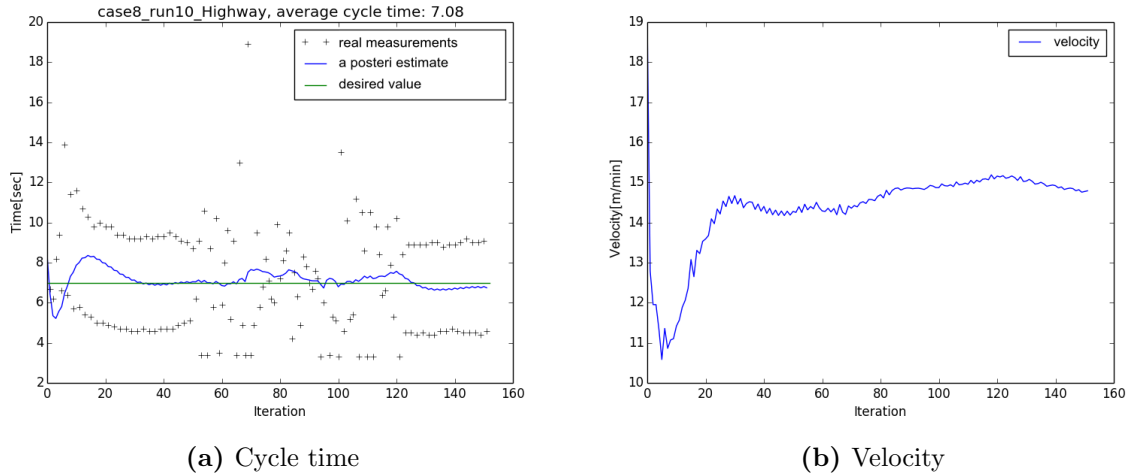


Figure 3.14: Highway performance under disturbances with variable setpoint of bound 1

Figure 3.14 shows that the method makes the system slightly more reactive which can be seen in Figure 3.14b, where the velocity rise time is faster than before. The average cycle time during 50 to 120 iterations is 7.22.

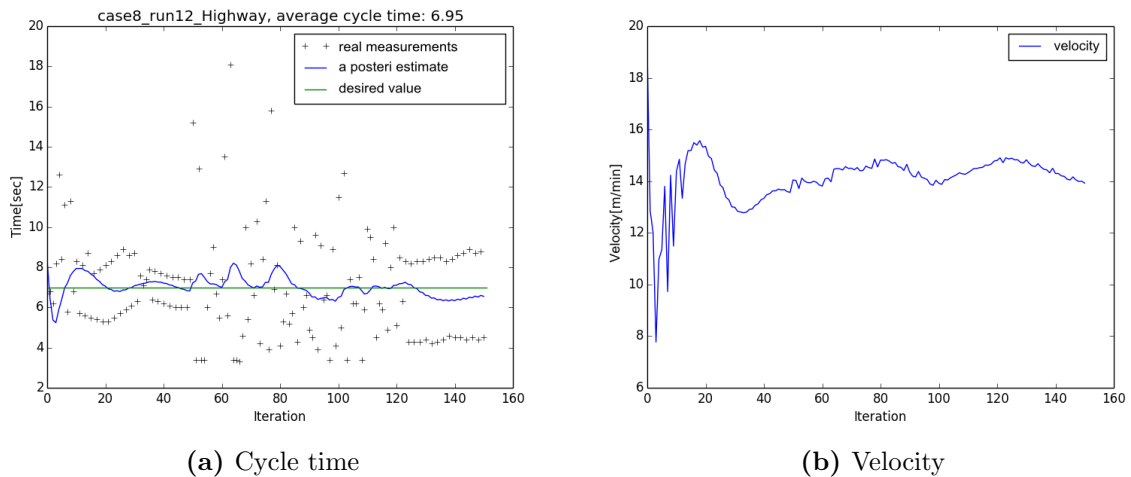


Figure 3.15: Highway performance under disturbances with variable setpoint of bound 3

The average cycle time for Figure 3.15 is 7.14 and the behavior is somehow more oscillating than the case using 1 as the error bound.

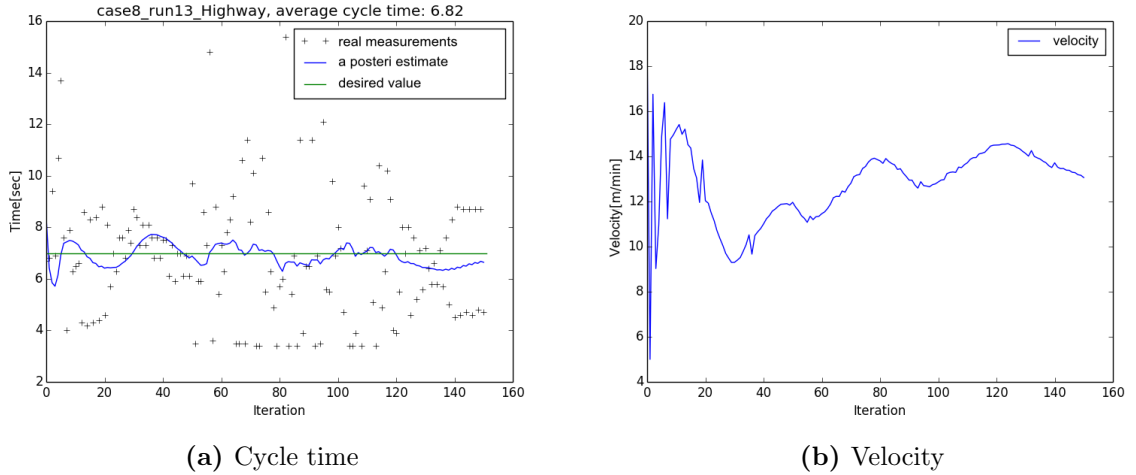


Figure 3.16: Highway performance under disturbances with variable setpoint of bound 5

Finally, the average cycle time for Figure 3.16 is 6.98. In Figure 3.16b, the velocity oscillates more than previous cases.

3.3.3 Online change of setpoint

These tests were performed to investigate the ability of the system to adapt when changing the desired cycle time online. The procedure was to let the system run with different cycle times for 75 pallets each. Initially the desired cycle time was seven seconds, then changed to 15 seconds, followed by 10 and then back to seven seconds. To give the controller the ability to react to changes of the desired cycle time, the earlier collected values in the average velocity method was reset with every new value of desired cycle time. As in the previous tests conducted when disturbances were introduced, these tests also consider the variable setpoint method with error bound of zero, 1, 3 and 5. The results can be seen in Figures 3.17 to 3.20.

Figure 3.17 shows how the system react and converge to changes in the desired cycle time. For every change of desired cycle time, Figure 3.17b shows large velocity changes. The speed of convergence varies for every change of desired cycle time.

Figure 3.18 shows the system performance when changing the desired cycle time with an error bound 1 for the variable setpoint method. It can be seen in Figure 3.18b that the peaks in the velocity changes are slightly higher than in Figure 3.17b.

Figure 3.19 shows that the rise time of the estimated cycle time are faster than the earlier cases but with a more oscillating behavior. It can be seen in Figure 3.19b that the peaks in the velocity change has a more oscillating behavior than both previous cases.

Figure 3.20 shows that the first change of desired cycle time gives a more oscillating behavior of the estimated cycle time while the other two changes have a smoother and better behavior. The velocity is slightly more oscillating.

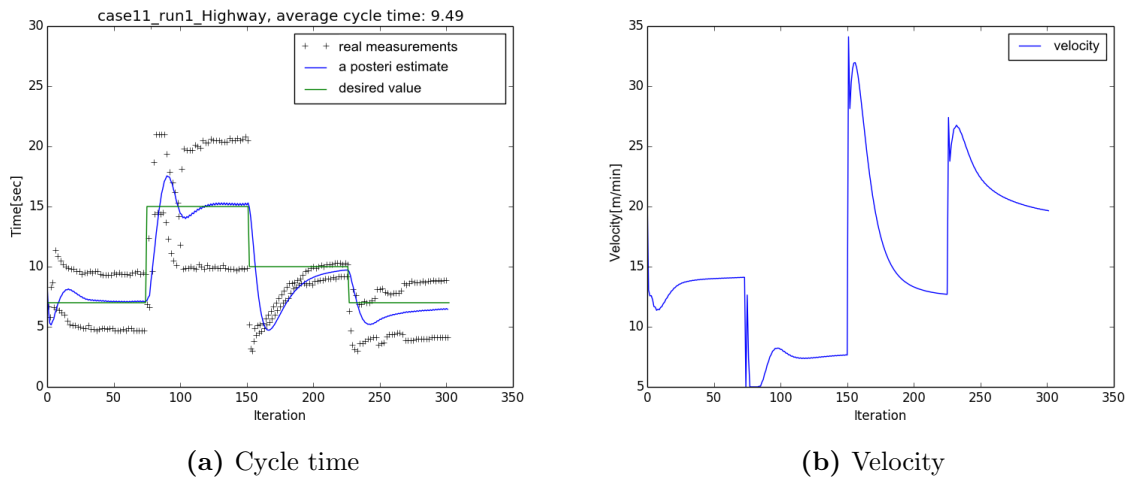


Figure 3.17: Highway performance when changing cycle time online, with variable setpoint of bound zero

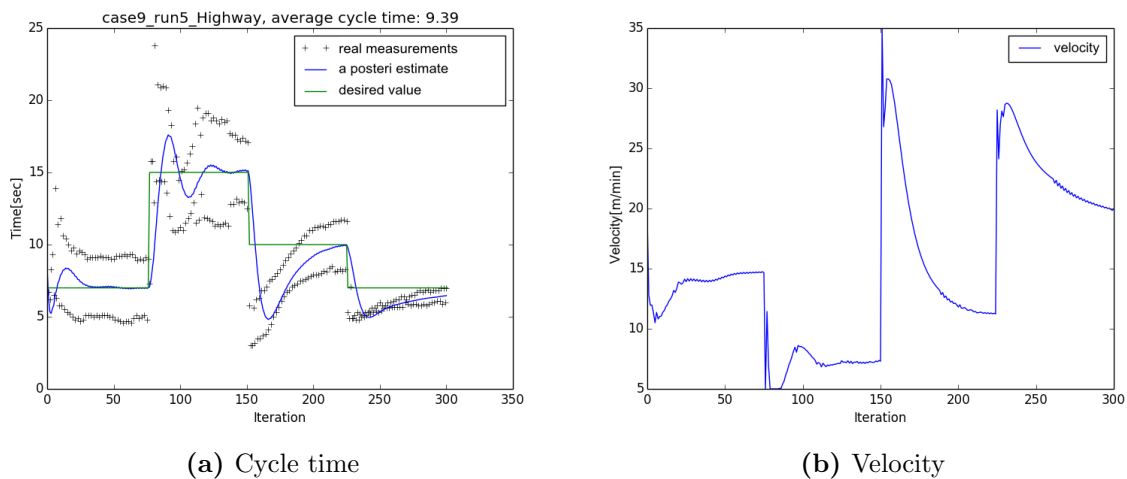


Figure 3.18: Highway performance when changing cycle time online, with variable setpoint of bound 1

3.4 Summary

This chapter handles the implementation of the controller in the provided X85 pallet system. Results from experiments in the physical system are presented, showing the necessity of each method described in Chapter 2. Also, performance when introducing disturbances and online changes of the desired cycle time are demonstrated. These results are discussed and concluded in the following chapters.

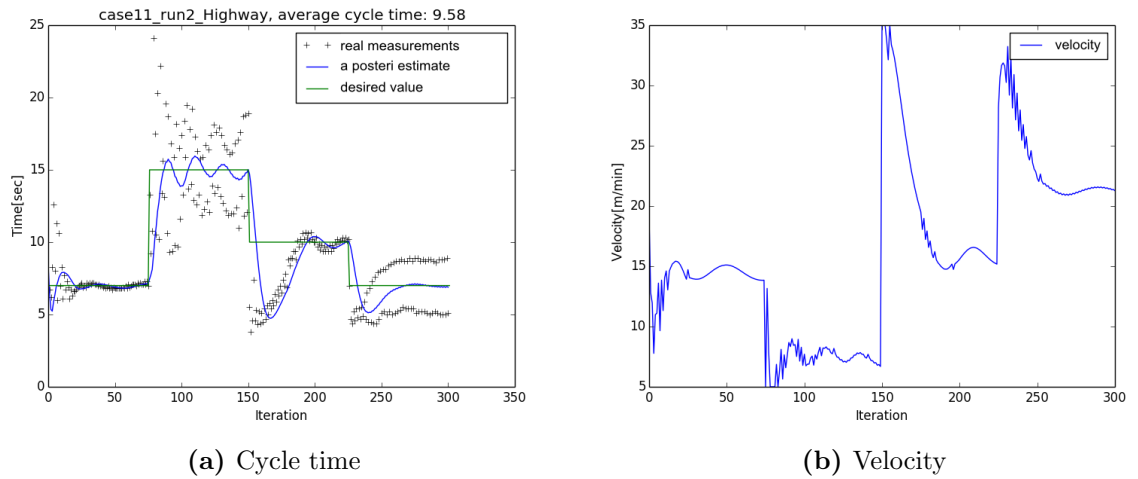


Figure 3.19: Highway performance when changing cycle time online, with variable setpoint of bound 3

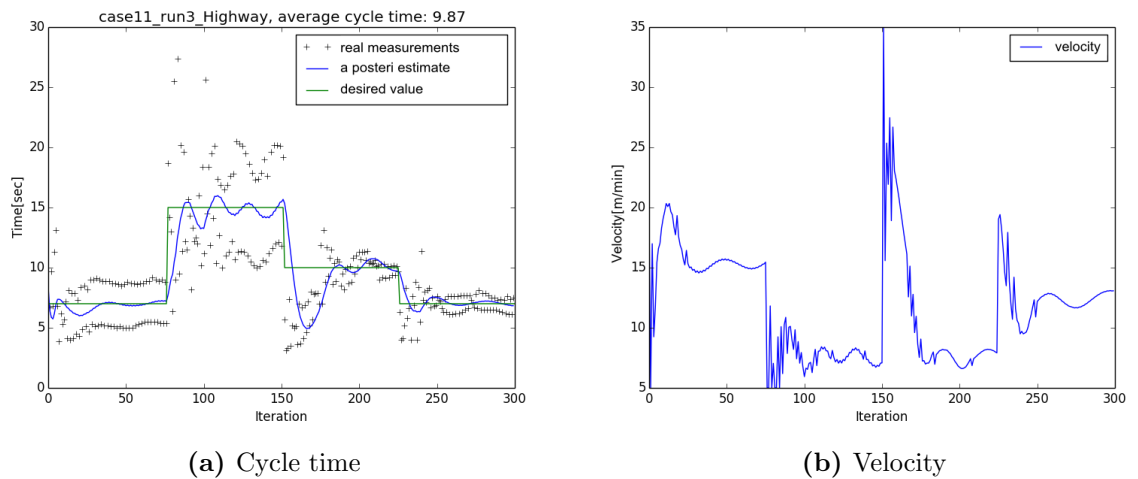


Figure 3.20: Highway performance when changing cycle time online, with variable setpoint of bound 5

4

Discussion

THE work presented in this report shows that the main goal, of achieving a smooth behavior of the velocity while maintaining a desired production rate, were fulfilled in both the simulation environment as well as in the physical system. This is shown in Figure 4.1(cycle times) and 4.2(velocities).

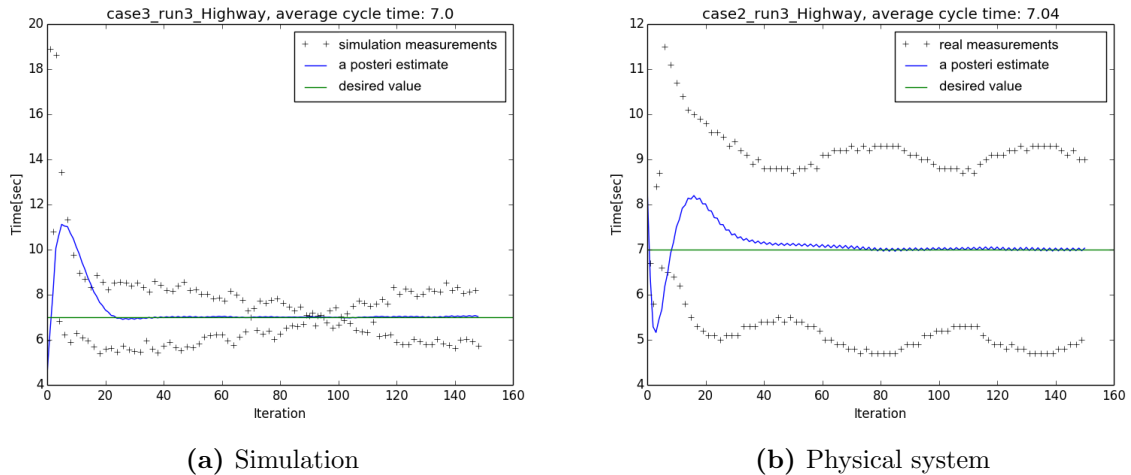


Figure 4.1: Highway cycle times for complete controller

In both cases the behaviors are satisfying and desirable, with the difference that the velocity is significantly higher in the simulation environment. This is very interesting regarding the fact that uncertainties in the physical system are much stronger and higher than those in the simulation environment. The reason for this performance improvement can be related to the waiting times that appeared in different locations along the conveyor lines for both systems. In the simulation environment, a queue of pallets was formed before Satellite 1, and in the physical rig a queue of pallets was formed between the

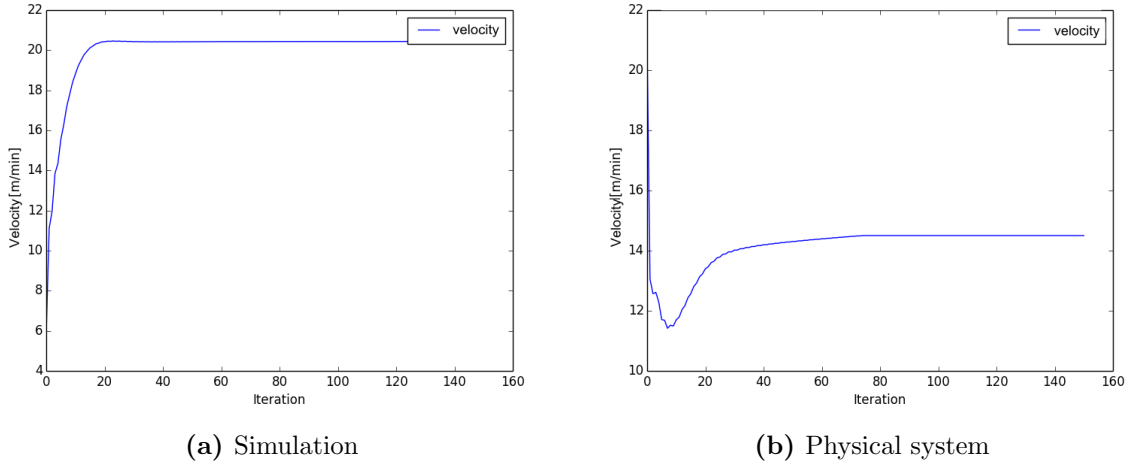


Figure 4.2: Highway velocity for complete controller

two satellites. A factor that can contribute to this difference is that changes in the simulation environment are instantaneous, while the physical systems have some inertia. This difference in the systems are very likely to affect the distribution of pallets and thus introduces different waiting times.

As mentioned earlier in the report, applying CONWIP control also introduces more waiting times in the system and the evaluation of the results rises the question of the necessity of CONWIP control. From the experiments, it was concluded that the implementation of CONWIP control resulted in an increased velocity but a more even distribution of pallets. This distribution is important for the robustness of the system, and the more complex the layout of a system becomes, the more important it is to maintain this even distribution. If CONWIP control is not maintained for such complex systems, disturbances may disrupt the cyclic behavior, resulting in a scattered flow of pallets and less accuracy in the average cycle time. CONWIP control might not be vital for all running modes of a pallet system. An option can be to implement CONWIP control for the startup of a system and then switch it off after the transition phase.

According to the results obtained from the simulation and rig for average velocity function and Kalman filter, these methods are crucial to maintain a smooth velocity. The averaging method has to be implemented in order to stabilize the behavior of the velocity. However, the domain of values used to calculate the average velocity can be varied. The narrower domain of values used, the more sensitive the system becomes. Thus, the domain of the averaging method should be determined according to the amount of disturbances that affect the system. For a system that is highly exposed to large disturbances, a larger domain of values should be used. The Kalman filter vastly decreases the oscillations in velocity by feeding back the estimated average cycle time. Another method that can be used instead of Kalman filter, is to feed back a calculated average of the real measurements. The benefits of using the Kalman filter instead of the calcu-

lated average is that the Kalman filter has the ability to filter noise/disturbances more effectively by considering a mathematical model for the cyclic behavior of the system.

Although, the variable setpoint method introduces larger oscillations in the velocity, the major strength of the method lies in its contribution to decrease a possible cycle time offset. Similar to CONWIP control, variable setpoint can be implemented for a certain running phase of the system. A possible option, can be to disable the method for the transition phase and apply it when the system shows a stationary cyclic behavior.

In online changes of the desired cycle time, large variations in the velocity appears. It is desirable to avoid these variations for the system safety issues. A possible solution can be to implement some more sophisticated ramping or averaging method for changing the setpoint.

5

Conclusion

THE main objective of this project has been successfully fulfilled. Automatic control of the chain velocity to achieve a smooth behavior while maintaining a desired production rate(cycle time) has been successfully implemented on a physical rig. For industrial pallet systems, this automated feedback control promises significant improvements of sustainable measures such as reduction of noise pollution, running cost and electrical energy consumption.

Furthermore, it has been observed and verified that *FlexLink Design Tool* provides an effective simulation environment for the control design. The similarity of the behavior between the simulation and the physical system, is very remarkable. This indicates that a large amount of time and effort can be saved by using FLDT as the environment for the control design procedure.

Even though, the concept presented in this thesis still demands more elaborations to realize an industrial control mechanism, the obtained results are very promising. Some recommendations for future works are presented below.

5.1 Future works

As mentioned in the discussion, the systems performance can benefit by applying CON-WIP control and the variable setpoint method over different running phases of a pallet system. These control modes should be further investigated to improve the controller design.

There is still a huge potential for the improvement of a pallet system behavior by applying more sophisticated control methods. In this project, the control parameter was the conveyor chain velocity. Other control parameters such as process times and queue sizes were not taken into consideration in the controller but recognized to be of great importance for the performance of the system. A more sophisticated control method such as the model predictive control(MPC) introduces the ability to add constraints to

the system which could prove beneficial.

There are cases that have not been taken into consideration during this project. A control parameter which has not been monitored is the total number of pallets in the system. For example, if one has the intention of increasing the desired cycle time to 60 seconds, there will be a need to remove some pallets in the current setup of the system. The opposite situation when trying to lower the cycle times, could require additional pallets added to the system. This control and possibility to vary of the total number of pallets is needed to increase the domain of possible cycle times. Finally the behavior of a controlled pallet system with different product types and routings has not been investigated. An enhancement of our control design framework will be achieved by implementing such general concepts.

Bibliography

- [1] J. Tompkins, J. White, Y. Bozer, J. Tanchoco, Facilities Planning, Third Edition, John Wiley & Sons, 2003.
- [2] E. Frazelle, Material handling: A thechnology for industrial competitiveness, Tech. rep., Georgia Institute of Technology (1986).
- [3] Flexlink (2013).
URL <http://www.flexlink.com/en/>
- [4] M. Mashaei, Sustainable Design and Control of Automated Material Handling Systems, Department of Signals and Systems, Chalmers University of Technology, 2012.
- [5] Visual Components (2014).
URL <http://www.visualcomponents.com/Products/3DCreate>
- [6] Python Software Foundation (2014).
URL <http://docs.python.org/2/tutorial/index.html#tutorial-index>
- [7] K. J. Åström, R. M. Murray, Feedback Systems, Princeton University Press, 2008.
- [8] P. Jonsson, S.-A. Mattsson, Manufacturing, Planning And Control, McGraw-Hill Education, 2009.
- [9] G. Welch, G. Bishop, An introduction to the kalman filter, Tech. rep., Department of Computer Science, University of North Carolina at Chapel Hill (2006).
- [10] Matrikon (2014).
URL www.matrikonopc.com
- [11] OpenOPC (2014).
URL <http://openopc.sourceforge.net/>