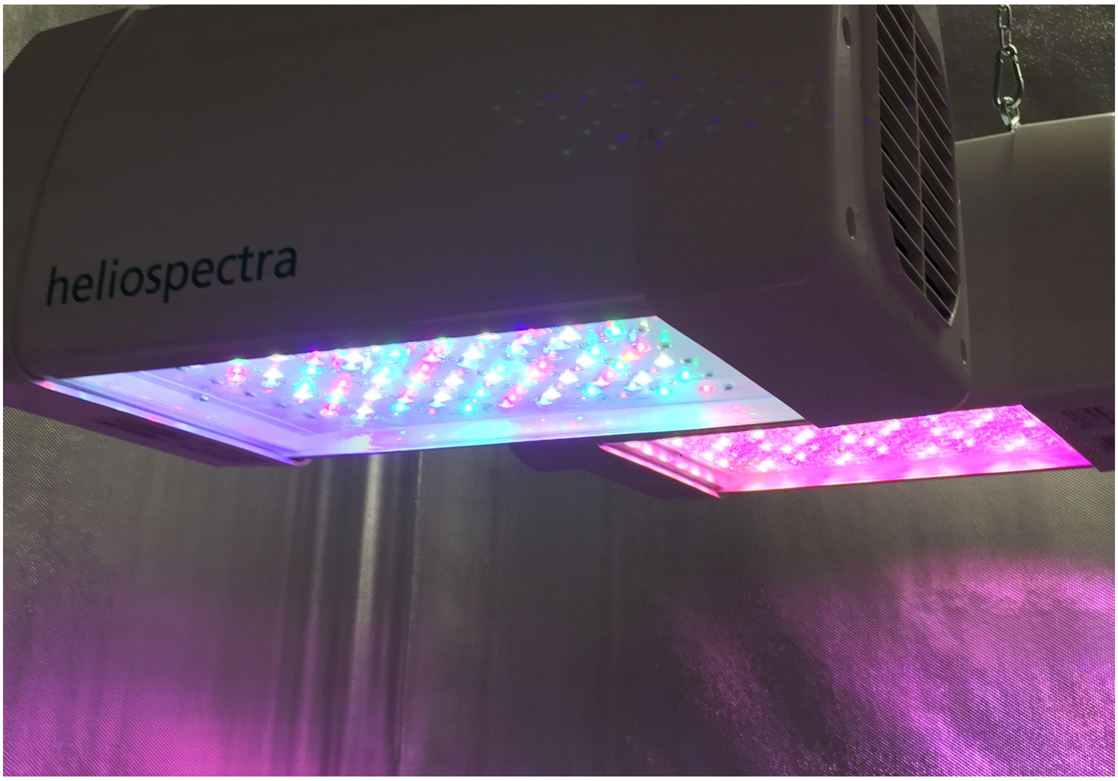
CHALMERS



Feedback control of LED-lamps for simulating varying natural sunlight

Master's Thesis in Systems, Control and Mechatronics

JOHAN FRÖBERG
ANTON LINDAHL

Department of Signals & Systems
Division of Automatic Control, Automation and Mechatronics
Chalmers University of Technology
Gothenburg, Sweden 2015
Master's Thesis EX046/2015

MASTER'S THESIS IN SYSTEMS, CONTROL AND MECHATRONICS

Feedback control of LED-lamps for simulating varying natural sunlight

JOHAN FRÖBERG & ANTON LINDAHL

Department of Signals and Systems
Division of Automatic Control, Mechatronics and Automation
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2015

Feedback control of LED-lamps for simulating varying natural sunlight
JOHAN FRÖBERG & ANTON LINDAHL

Master's thesis EX046/2015
Department of Signals and Systems
Division of Automatic Control, Mechatronics and Automation
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone: +46 (0)31-772 1000

Cover:
Heliospectra lamps RX30 and LX60XC during a simulation.

Gothenburg, Sweden 2015

Abstract

LED-based greenhouse lighting enables direct control of the light spectrum received by plants. An effect of this is that a lot of research is being conducted on the effects of different light conditions on plant growth. Experiments in such research requires a high accuracy and repeatability which is difficult to achieve with open loop control due to external disturbances and variations in experiment setups. The wide spectrum produced by such lamps also gives the possibility to mimic the light spectrum produced by the sun.

In this master thesis, a feedback control system is designed and implemented using a spectrometer placed amongst the plants for measuring incoming light. The system is designed for four different types of lamps that are equipped with 3, 4, 7 and 9 different LED colors. The lamps are modelled as MIMO systems due to the fact that the different LEDs produce light spectrums that overlap each other. Models are acquired using system identification. Two different control systems are implemented and tested, a control scheme using multiple integral controllers with a decoupling network and an LQ controller with integral action. The complete system is designed and implemented using Matlab, but also implemented on a Raspberry Pi in order to achieve a stand alone product.

Tests determined that the version using decoupling and multiple integral controllers worked best. The result is a robust control system that is able to counteract sizable disturbances in order to create the light spectrum desired by the user. The system is also able to simulate varying natural sunlight by using references created either by measurements of real sunlight or calculated with external programs. However, due to hardware limitations, multiple lamps must be used to recreate the full intensity of sunlight. It is also possible to use the system in a greenhouse to compensate lack of natural sunlight in bad weather, or to enhance the intensity of certain colors of light to stimulate plant growth.

Keywords: LED lamps, controller, plant growth, spectrometer, MIMO, decoupling network

Preface

The following thesis work has been conducted at Heliospectra AB in collaboration with Chalmers University of Technology under supervision of Torsten Wik during the period January 2015 - June 2015.

Acknowledgements

We would like to thank Daniel Bånkestad at Heliospectra and Torsten Wik at Chalmers for guiding us throughout our work. We would also like to thank the employees at Heliospectra for welcoming and supporting us during our time at the company.

Johan Fröberg and Anton Lindahl, Gothenburg

Contents

Nomenclature

DC Direct current

IT Integration time

I/O Input/output

LED Light-emitting diode

LQI Linear quadratic integral

MIMO Multiple Input Multiple Output

PAR Photosynthetic active radiation

PID Proportional integral derivative

PR Pixel resolution

PWM Pulse-width modulation

RPI Raspberry Pi, a single-board computer

SNR Signal-to-noise ratio

SR Spectral range (of grating)

UDP User datagram protocol

1

Introduction

The increase in world population places large demands on food production. A large part of the food production is the growing of crops, fruit and vegetables. There is also a need for alternatives to non-renewable energy sources, an area where algae as biofuel has become increasingly popular. In order to meet these demands the efficiency of the growth itself must be increased. There is currently a lot of research being done on the photosynthesis of plants and the effects of different parameters such as light conditions, temperature, pH levels and so on. In order to conduct such research it is of great importance to be able to perform experiments with a high level of control. It is also of interest to study the effects of dynamic lighting such as circadian cycles in a closed environment. Lucker et al. [1] have set up such a changing environment. With varying light conditions, temperature, pH levels and so on to mimic a natural environment. However the setup only uses a single white LED which means that only the light intensity is mimicked and not the spectrum.

Heliospectra is a company based in Gothenburg that produces LED-based greenhouse lamps. The lamps are equipped with roughly 240 LEDs of several different colors. This gives a very detailed control of the light spectrum and enables tailor made lighting conditions specific for certain types of plants. The lamps are controlled by a computer and it is possible to set each individual color to a specific intensity as well as to run a schedule simulating for example a daylight cycle. The current control is purely open loop which means that the growth light will be susceptible to disturbances.

Experiments are mostly conducted in growth chambers with reflective walls that distort the light, but there are also external sources of disturbance such as natural sunlight from windows and light leakage from adjacent growth chambers. These factors affect the precision and repeatability of experiments but can be compensated for using a feedback control system. In addition differences in experiment setup will change the intensity the plants receive, which is also something the feedback control system should compensate for.

The main purpose of this master thesis project is to design and implement such a control system. Using a spectrometer placed among the plants which measures the wavelengths and intensity of incoming light, a feedback loop can be created. The different LED colors in the lamps produce light spectrums that overlap each other, which means that the lamps must be modelled as multiple input multiple output (MIMO) systems. The wide spectrum of light produced by the lamps gives the opportunity to use reference spectrums mimicking that of the sun. This enables simulation of daylight cycles which is an interesting aspect in plant research. Otherwise the references used by the control system are specified by biologists, for example in terms of photosynthetic active radiation (PAR), which is the wavelength interval 400-700 nm of light that has the most effect on plants photosynthesis. A feedback control system such as this is also useful in a greenhouse environment since it is able to compensate the natural sunlight on a cloudy day. It can also be used to adjust the spectrum of light received by plants to optimize plant growth parameters influenced by spectral quality.

The spectrometer used in the feedback system is connected to a computer via USB. The length of the USB cable restricts the placement of the spectrometer since it is not possible to place a computer near the growth chamber due to irrigation and humidity. The spectrometer itself is also sensitive to such conditions. This will be solved by implementing the control system on a Raspberry Pi which is encapsulated together with the spectrometer in a waterproof box for protection. Information can then be retrieved via a Wifi connection to the Raspberry Pi.

There has been some work done on similar systems. For example, in Schuurmans and Pashley [2] a feedback loop is implemented to combine red, blue and green LEDs to create a white light. Here the application is to compensate for variability in the LEDs but aspects of this work can be used as an inspiration for this project. Previous master thesis students at Heliospectra have also attempted a simple PID controller for the lamps (Guerrerro, Soukhakian [3]). No consideration was then taken to the spectral overlap which made the controller difficult to stabilize. Employees at Heliospectra also do research on the effects of different light conditions on plants in order to determine what a suitable light spectrum is. For example in Bånkestad [4], plant fluorescence and reflectance are studied as a mean to determine plant responses given different light conditions.

2

System overview

This chapter describes the lamp-spectrometer system in full at a high level of abstraction. It includes all subsystems as well as a short description of a Raspberry Pi (RPI) implementation of the system. Note that this chapter only aims to give an overview of the system, details regarding each part of the system are presented later on.

2.1 Lamp-spectrometer feedback control system

Four different types of lamps are used that differ in number of colored LEDs. The lamp types have 3, 4, 7 and 9 different LED colors. However, because of the similarity of the lamp types only two of them are presented, the one with 4 LED colors and the one with 9 LED colors. Two different variants of the same spectrometer are used, which differ in the size of slit where light enters the measurement tool. A PC is used to control the lamps via Wi-fi connection to a router which has an ethernet connection to the lamp. The PC is also used to read measurements from the spectrometer via USB as well as to run the control system in Matlab. The functionality of the system is presented as a flow chart in Figure 2.1.

The system starts out with the selection of a lamp type, i.e the number of LED colors. This is followed by a search for an available lamp at a predefined IP adress. If a lamp is available a connection is established. When a connection exists the spectrometer is initialized. The initialization includes loading a calibration file, measuring bias, calculating an initial integration time (the collection time for the spectrometer), and creating a wrapper object for handling spectrometer functions. After this a controller initialization routine is executed, which initializes all parameters needed for the control system. An input/output (I/O) gain matrix system model is also created at this time. Finally, the references to be used in the system are created and the control system is started.

The controller loop starts out by reading the spectrum produced by the lamp and external environment. The spectrum is converted to a usable unit, split up and integrated in color partitions determined by the lamp type. The integration time is updated in order to ensure a reasonable signal-to-noise ratio (SNR) after which the control algorithm is run. The controller calculates the needed increment to reach the desired spectrum and adds this to the current setting of the lamp. Depending on the calculated control signal, parts of the I/O gain matrix is changed due to the fact that the lamps use pulse-width modulation (PWM) or direct current (DC) at different input settings with a major change in the gain matrix as a consequence. The control loop contains an inner loop that is used to control the total photosynthetic active radiation (PAR) irradiance in case of dominant external lighting conditions.

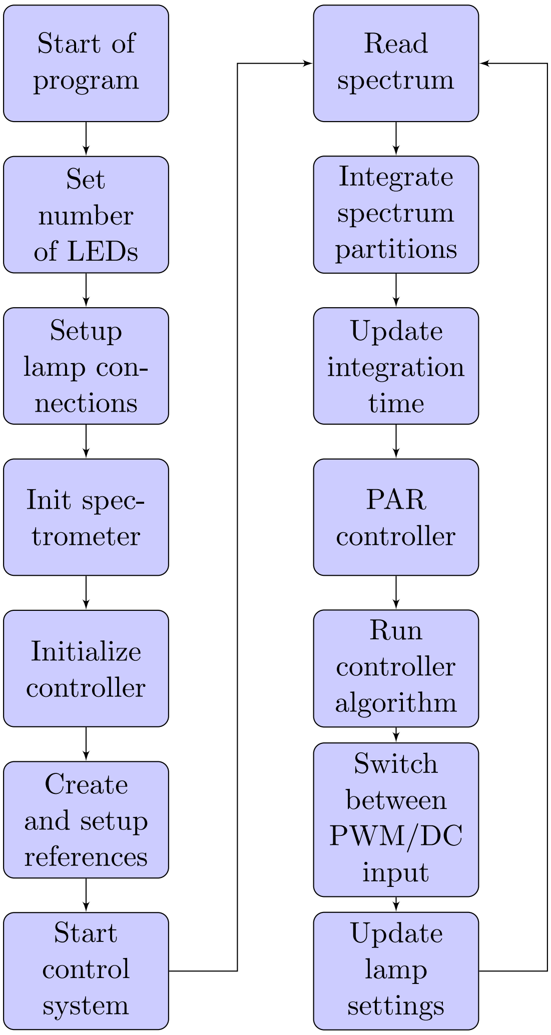


Figure 2.1: System overview.

2.2 Raspberry Pi version

The RPI version of the system does not differ much from the one in Matlab. The main difference is that the control system is disconnected from a PC and thus does not use the Wi-fi connection to a router. Instead the RPI controls the lamp directly via an ethernet cable. The spectrometer is still connected via USB but to the RPI instead of a PC. Since the RPI is to be placed in a lab environment, preferably in close proximity to the plants, it is difficult to observe the system on screen since it will require a long cable. However, continuous transmission of measurements are streamed from the RPI over Wi-fi and can easily be picked up by a PC for observation in an external program.

3

Light spectrum measurement

In order to develop a feedback control system for the lamps the output from the lamps needs to be measured. This is done using a spectrometer placed under the lamp. The spectrometer needs to be configured with respect to distance between lamp and spectrometer as well as for the intensity the lamp runs on. This chapter describes the hardware of the spectrometer as well as configurations, unit conversions and SNR.

3.1 The STS-VIS spectrometer

The measurements are made with an STS-VIS spectrometer from the company Ocean Optics [5] (see Figure 3.1). This particular spectrometer has a wavelength range of 350-800 nm which covers the range of interest in this project.

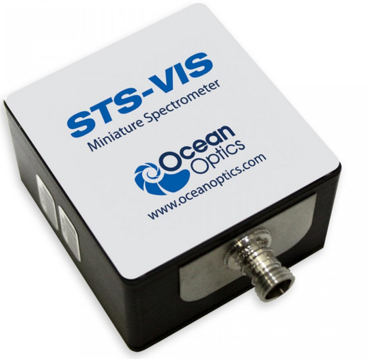


Figure 3.1: The STS-VIS spectrometer, image courtesy of Ocean Optics [5].

To be able to collect as much light as possible a cosine corrector (optical diffuser) is attached to the spectrometer (Figure 3.2). This increases the field of view for the spectrometer to 180°, making it able to collect all the light directed at the plants.

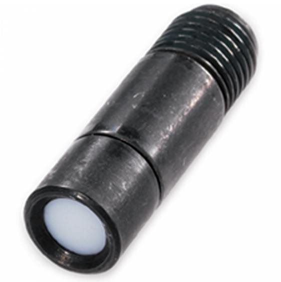


Figure 3.2: Attachable cosine corrector, image courtesy of Ocean Optics [6].

The spectrometer measures incoming light by collecting photons, sorting them with respect to wavelength and converting them to an electric charge which is read as a digital intensity number, also referred to as count. The counts are specific for a small wavelength interval, called a pixel. These intervals have varying wavelength widths (around 0.45-0.48 nm) which needs to be accounted for in the unit conversion. According to Ocean Optics the spectrometer should have its digital intensity numbers in the region of 80-90 % of the maximum count in order to achieve a reasonable SNR [7]. For the STS-VIS the maximum count is 16383 (14 bits). The number of counts depend on the light intensity, the duration the spectrometer collects data and the temperature of the detector board of the spectrometer. The time the spectrometer collects data is called integration time and ranges from 10 μs to 10 s. For a low light intensity it is necessary to measure longer in order for the spectrometer to collect enough photons. For higher light intensities the spectrometer is "filled up" faster and the integration time should thus be set shorter. If the integration time is set too high with respect to the light intensity the measurements will be saturated at the maximum digital intensity number making them incorrect and more or less unusable.

There are ways of increasing the counts read by the spectrometer other than changing the integration time. Depending on the application, the spectrometer can be built with different slit sizes. The size of the slit determines the flow of light entering the spectrometer. The slit can have a width that varies between 10 μm to 200 μm. A larger slit width will allow more light to enter which will increase the counts. This will however also lower the optical resolution of the measurement. A better optical resolution means that the wavelength of a photon can be determined with higher precision.

The optical resolution of a spectrometer is determined by three factors:

1. Spectral range of grating (SR) [nm], which is 338-824 nm for the specific spectrometer used (100 μm slit).
2. Number of pixels, 1024.

3. Pixel resolution for spectrometer and slit (PR, in pixels).

The optical resolution is measured at full width at half maximum (FWHM) in *nm*. The pixel resolution is not available from the manufacturer which makes this calculation unavailable. However statistical calculations of several spectrometers of the same type states that the optical resolution is around 6 nm for an STS-VIS with 100 μm slit and 1.5 nm for an STS-VIS with 25 μm slit according to the data sheet [8].

Since the measurement is integrated over each color (Section 4.2) the optical resolution will only affect the endpoints of the integral intervals. The optical resolution is therefore not that important and a large slit size can be used. Two different STS-VIS spectrometers have been used throughout the project, one with a slit size of 25 μm and one with 100 μm. For the smaller slit size, measurements are slow at low light and the integration time needs to be over 10 seconds to have a good maximum peak. However, using the spectrometer with 100 μm slit reduces the integration time. The increase from 1.5 nm to 6 nm in optical resolution affects the system so little that it is negligible.

3.2 Unit conversion

Light used for growing plants is mostly measured in photon irradiance or photon flux density, in the unit $mole\ photons/m^2s$. To achieve the conversion from the digital intensity numbers to irradiance, the speed of light, Planck's constant and Avogadro's constant are needed. With Planck's constant the energy, in *joules*, for a single photon (i) is obtained through the equation

$$E_i = \frac{h \cdot c}{\lambda}, \tag{3.1}$$

where c is the speed of light, λ is the wavelength for the particular photon and h is Planck's constant. The wavelength for each pixel is given by the spectrometers wavelength calibration. The total energy in each pixel can be calculated by first converting the counts to energy (*joules*). This is done using an instrument-specific calibration file for absolute irradiance, which was constructed by measuring a calibration lamp of known intensity (HL3p, Ocean Optics [9]). The number of photons is then obtained with the equation

$$photons = \frac{E_i}{joules}. \tag{3.2}$$

With the use of Avogadro's constant the number of photons can be transformed into *mole photons* with the equation

$$mole\ photons = \frac{photons}{N_A}, \tag{3.3}$$

where N_A is Avogadro's constant. To get the irradiance unit the collection area in m^2 is needed. The area is calculated as

$$A = \pi(\frac{3900 \cdot 10^{-6}}{2})^2, \tag{3.4}$$

where $3900 \cdot 10^{-6}$ is the diameter in m for the used cosine corrector. The final equation is then

$$Irradiance = \frac{mole\ photons}{A \cdot IT}, \tag{3.5}$$

where IT is the integration time in seconds. With these calculations the photon irradiance in $mole\ photons/m^2s$ is obtained.

3.3 Calibration and tuning

In order to successfully use the measurement data, the spectrometer needs to be tuned. The raw measurement has a bias over the whole spectrum which must be removed. This is known as dark spectrum correction. The bias changes with the integration time as well as with temperature. By measuring in complete darkness this bias can be obtained for the different integration times that are used. A cap is put on the spectrometer to minimize the amount of light that the spectrometer receives. The spectrometer is also placed in a dark room. By measuring the bias for different integration times a look-up table is created. For noise characterisation purposes the bias is measured 10 times at some integration times and a mean value is calculated at each one to evaluate the effect of the dark spectrum. In Figure 3.3 the mean of a typical dark spectrum is shown.

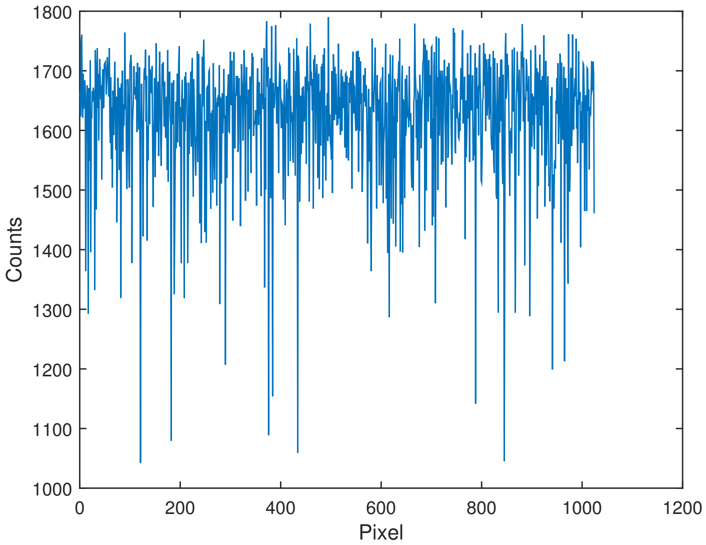


Figure 3.3: Bias measured in complete darkness.

Looking at the plot it can be seen that some pixels vary significantly from the rest of the spectrum. However, this phenomenon is not random. By looking at the variance it is determined that the variance of these pixels over 10 samples are only around 15 counts which means that this bias can easily be removed. Since the dark spectrum has such low variance, it is only measured once at each integration time used in the final system implementation. A lookup table is created and used to subtract the bias at runtime. This makes it possible to quickly compensate the dark spectrum for every new setup.

In order to get a good measurement at runtime the integration time needs to be dynamic. The integration time is therefore changed depending on the operating point of the light intensity. According to the data sheet the number of counts is a linear function of the integration time with an error of ±1 % [8]. This is verified by measuring the maximum count at different integration times. Given this information, an algorithm to find a suitable integration time is designed that uses the linear relation between number of counts and integration time.

The pixel with the highest count is always used as a basis for the calculations. If the highest number is below 80-90 % of the maximum, the integration time is increased. The increase in integration time is then calculated with the proportional gain that is needed to get the count inside the 80-90 % region. The aim is then to get 85 % of the maximum counts with the use of the algorithm. The used algorithm is

$$IT_{New} = IT_{Current} * (0.85 * Counts_{Max}/Count_{HighestMeasurement}). \tag{3.6}$$

Since the linearity relation only has an error of about ±1%, a good integration time should be reached after only one calculation. However with this error as well as the dark spectrum bias and other noise sources, large changes often requires a couple of calculations. If the highest count is above 90 % but not over the maximum counts an integration time can be found directly using Equation (3.6). If it is higher than what is possible the spectrometer will return the highest value. Then there is no way of knowing how large the maximum count actually is. Because of this the integration time is decreased a whole second to minimize the time it takes to find a suitable integration time. If the integration time then overshoots and becomes too small, Equation (3.6) is used, and a suitable integration time is found at the next iteration instead.

Another problem is that the noise increases with the integration time. Ocean Optics defines the maximum integration time for the STS-VIS to be 10 seconds [8]. This is because with this integration time the noise comprises 50 % of the total signal (conditioned that the signal is 85 % of the maximum). However this was found to be incorrect, the actual noise is in fact just 25 % of the total signal at said conditions. In this application situations arise where 10 seconds is not enough to reach the desired 80-90 % region. However, because of the high amount of noise at ITs over 10 s this is set as a limit in the final system implementation since it still gives a decent measurement.

3.4 Noise characteristics

The system is affected by noise coming from various sources, such as electrical noise from the A/D converter and stray light that has been scattered into the wrong parts of the detector. This section is about the characterization of the noise present in the system. The signal to noise ratio (SNR) is dependent on temperature, integration time and amount of light that is obtained in a measurement. An increase in temperature will make the spectrometer increase its count because of an increase in thermal energy. These counts will increase proportionally to the integration time. For example, at 35 °C the counts increase with about 50 counts each second, and at 60 °C the counts increase with about 150 each second (according to the data sheet [8]). Tests have been made to see how much this affects the readings.

Figure 3.4 shows the behavior of the STS-VIS when applying a change in temperature and integration time. Temperatures used are around the average operating temperature of the spectrometers detector board. Measurements are done in complete darkness.

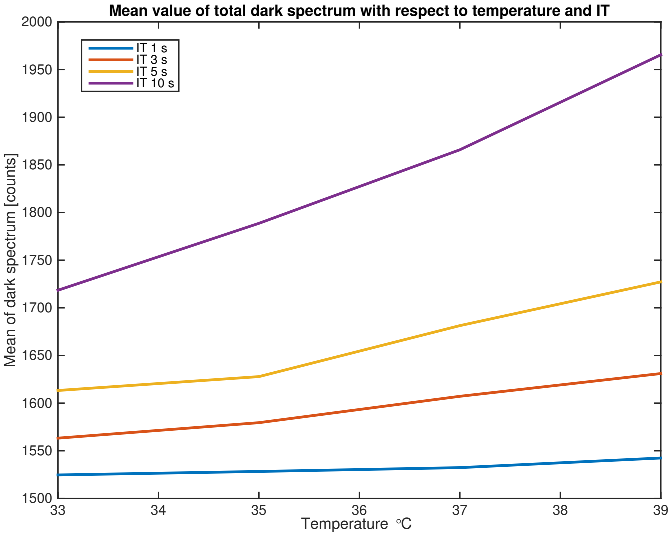


Figure 3.4: STS-VIS characteristics.

As seen in the figure the STS-VIS is slightly dependant on the detector board temperature. However the maximum change in counts is roughly 230 counts (at IT 10 s). Running the system with a continuous calculation of IT with respect to the counts, and assuming a steady count of 80-90 % (ca 13000-14500 counts), the effect of a temperature change of 6 °C is less than 2 %. Based on this knowledge it is reasonable to only measure the dark spectrum at one single temperature, and thereby neglect the influence of temperature on dark current. Additionally, in most applications related to plant research the operating temperature will be fairly stable.

In order to quickly create the lookup table for the dark spectrum, and since the difference in effect of the integration time is small, there is no need for a high resolution of dark spectra. Therefore, the dark spectrum is only sampled at a resolution of 0.5 seconds. Taking more measurements would make the readings more accurate, but the small gain in accuracy will however not notably affect the system performance.

Looking at the SNR for a measurement with fixed intensity on the white LEDs and varying integration time Figure 3.5 can be obtained. These measurements are made with the dark current removed and the temperature held constant. It can be seen that the difference in percent of maximum counts does not change the SNR for the signal for most wavelengths. However it is seen that when going far above the recommended integration time of 10 seconds the SNR is decreased. The reason for this is presumed to be a change in the noise characteristics due to unknown factors. Note that the white light does not affect all wavelengths which is why the SNR is very low at the edge wavelengths.

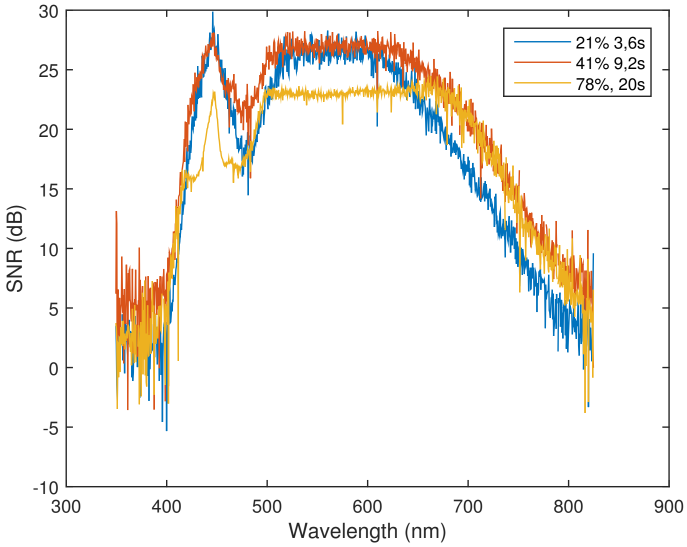


Figure 3.5: Measured SNR using a fixed setting on the white LED group at different integration times (seconds). The percentage is of the maximum count and the time stands for the integration time used.

Even though the change in SNR is small for different count levels, the recommendation from Ocean Optics to stay inside the 80-90 % interval of the maximum count is followed.

4

Modelling

This chapter describes the lamps that are to be controlled, how they work with input signals and how each LED group in the lamp affects the spectrum. Models of the lamps are described, determined by system identification and evaluated for use in the control system.

4.1 The lamps

Heliospectra produces a number of different lamps with a varying number of colored LED groups. This section gives a brief introduction to different lamp types, note that there exist additional types that are not used and therefore not mentioned. The lamps all have a power supply socket and an ethernet socket for control via a web page or an external program. The lamps are controlled by setting a desired intensity which essentially is an arbitrary number ranging from 0 to 1000 for each group of colored LEDs. Most of the lamps also have two types of input, pulse width modulation (PWM) and direct current (DC) voltage. The lamps are set with PWM at low intensities because a low DC input may cause a complete shutdown of some LEDs. The intensity threshold where the lamps switch from PWM to DC is different for each LED color and lamp type but is always in the range of 80-150 intensity. A typical lamp produced by Heliospectra can be seen in Figure 4.1.

Figure 4.1: A Heliospectra lamp.

Heliospectras lamp type LX60XC has four different groups of LEDs, namely blue, white, red and far-red.

Figure 4.2 shows the relative maximum spectrum of this lamp, the dashed line depicting the total spectrum that is measured when all the LEDs of the lamp are simultaneously set to maximum intensity. The other lines depict measurements taken when the groups of LED colors are individually set to maximum intensity.

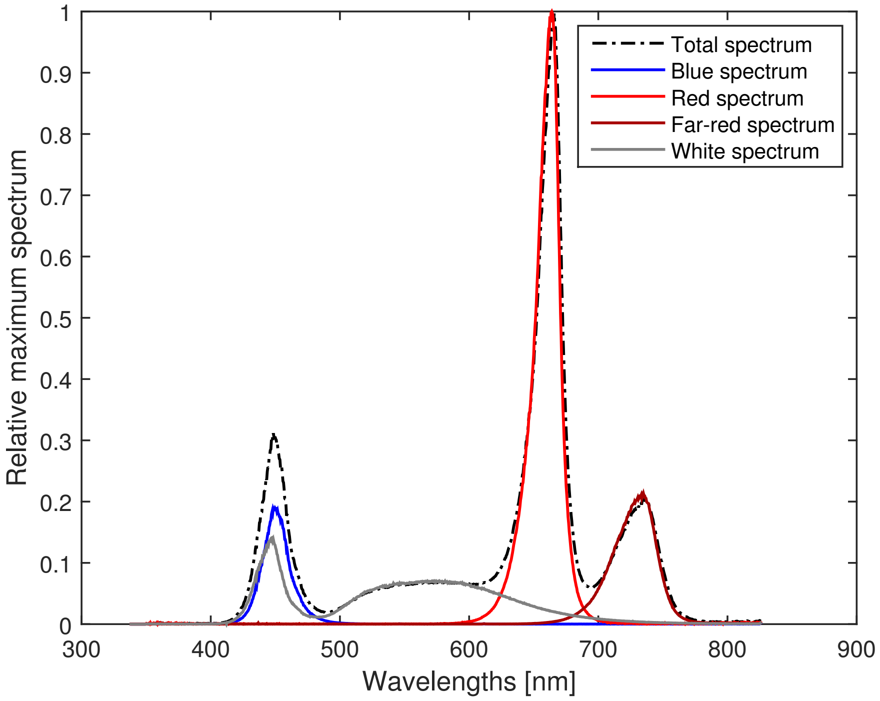


Figure 4.2: Relative maximum spectrum of the LX60XC.

As seen in the figure the spectrum produced by the white LEDs overlap the spectrum of all the other LEDs, there is also some overlap between the two different types of red. It can also be seen that the white LED produces a large peak in the blue spectrum. This is because the white LED is actually a blue LED coated with phosphor that scatters the wavelengths of the light in order to create white light. However a large amount of blue light still remains. It can also be seen that the red LED has a much larger output span than the other colors.

The most advanced lamp that is used has nine different groups of LEDs (type RX30). The colors are UV, violet, indigo, blue, green, orange, red, far-red and white. The spectrum of this lamp can be seen in Figure 4.3.

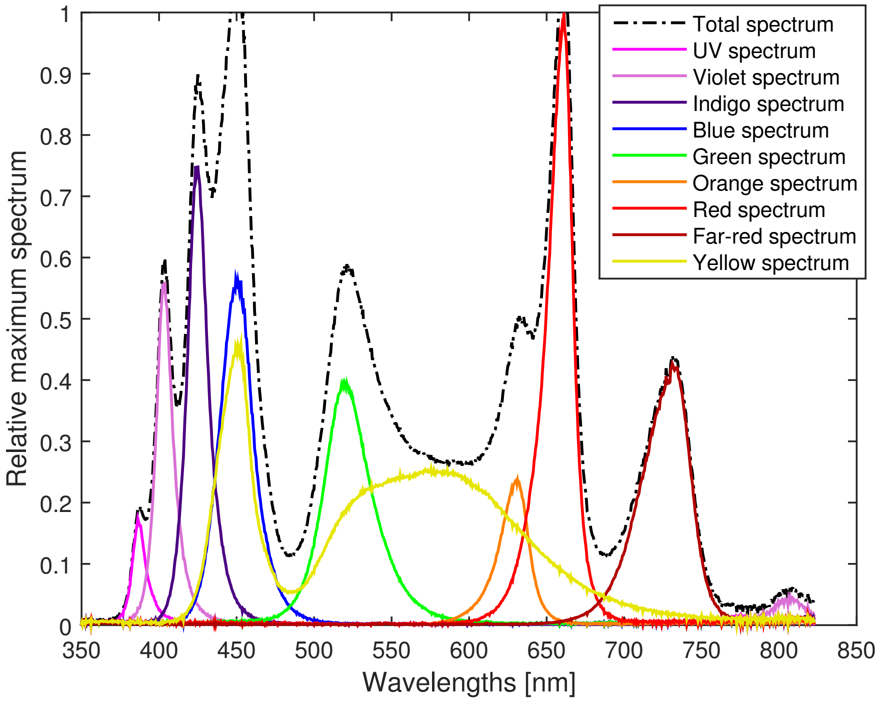


Figure 4.3: Relative maximum spectrum of the RX30.

As expected from looking at the spectrum this lamp produces it is more complicated to control since there are a lot more overlaps of the different spectra to consider.

4.2 Static gain model

The lamps are modelled with the same number of inputs as output. The inputs are the intensity settings of each LED group and the outputs are the integrated irradiance of all wavelength partitions affected by corresponding input. The reason for using the integration of a partition rather than for example the peaks of each LED group as the output, is that the integration is certain to include all the light present in the used wavelengths. Using integrated spectra as outputs also reduces the effects of noise. When selecting these partitions it is important to cover as much of each LED groups spectrum as possible in order to get a fair estimation of each LED groups capacity. A faulty selection of the spectrum partitions will also result in a non-linearity in the model due to the shape of the spectra of the LED groups.

This means, for example, that four different integrated irradiance partitions must be selected for the 4 LED group lamp. Looking at Figure 4.2 it can be seen that the lamp completely covers the wavelengths from 400 to 800 nm and the four different LED colors roughly covers an interval of 100 nm each with the exception of the white light. However, the white light is the only one that is capable of controlling the partition 500-600 nm. The selected spectrum partitions are:

- Blue: 400-500 nm
- Green: 500-600 nm
- Red: 600-700 nm
- Far-red: 700-800 nm

Each LED group controls its own color and the white LEDs control the green waveband. The lamp model is then simply the ratio between integrated spectrum of a wavelength partition in irradiance (output) and intensity setting (input). This ratio is measured on the physical system to achieve a model of the system.

The intensity for each LED group is incremented from setting 10 to setting 1000 using a step size of 10 and the integrated irradiance is measured at each step. The integration time of the spectrometer is evaluated at each incrementation and recalculated to ensure an acceptable SNR. Note that the wavelength range of the LED groups is not exclusive to its respective waveband, for example the red LED group leaks into the green waveband and the white LED group is spread across the entire spectrum. Because of this behavior the effects of the intensity setting of one LED group must be measured for all the output partitions.

The result of this is a matrix K_λ, estimated as the mean value of the measured slopes. This represents the output/input relation, i.e *integrated spectrum/intensity*. The K_λ-matrix is then used as the model describing the relation between input intensity u and output integrated spectrum y for all color combinations according to

$$y = K_\lambda \cdot u. \tag{4.1}$$

Specifically, the model for the lamp with 4 LED groups is

$$\begin{bmatrix} y_b \\ y_r \\ y_f \\ y_g \end{bmatrix} = \begin{bmatrix} K_{bb} & K_{br} & K_{bf} & K_{bw} \\ K_{rb} & K_{rr} & K_{rf} & K_{rw} \\ K_{fb} & K_{fr} & K_{ff} & K_{fw} \\ K_{gb} & K_{gr} & K_{gf} & K_{gw} \end{bmatrix} \begin{bmatrix} u_b \\ u_r \\ u_f \\ u_w \end{bmatrix}, \tag{4.2}$$

where indices b, r, f, w and g correspond to the colors blue, red, far-red, white and green respectively. The first index states the output waveband corresponding to the element and the second index states the input LED group. The model will change depending on a number of factors in the setup, such as placement of the spectrometer in relation to the lamp and reflections caused by the surroundings. Because of this a new model needs to be built the first time a new setup is used. Note also that there are scenarios where the elements of K_λ change during an experiment. One such scenario is for example when seeds bloom into green plants, the K_λ elements corresponding to the green outputs will then increase because of plant reflections and the model will no longer hold true. Depending on the effect such a change in the model has on the system it might be necessary to remodel the system, however the control system is able to handle large model errors. Clearly the K_λ-model can easily be extended to handle additional LED groups.

For use in an LQ control system (presented in Section 5.3) the system is also modelled as a discrete state space model as:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \tag{4.3}$$

The state vector x for the system is the integrated spectrum for different colors and the input vector u is the intensity for the different lamps. For the simplest lamp with the blue, red, far-red and white colored LEDs the complete state space model can be described as:

$$\begin{aligned} \begin{bmatrix} x_b(k+1) \\ x_r(k+1) \\ x_f(k+1) \\ x_g(k+1) \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_b(k) \\ x_r(k) \\ x_f(k) \\ x_g(k) \end{bmatrix} + \begin{bmatrix} K_{bb} & K_{br} & K_{bf} & K_{bw} \\ K_{rb} & K_{rr} & K_{rf} & K_{rw} \\ K_{fb} & K_{fr} & K_{ff} & K_{fw} \\ K_{gb} & K_{gr} & K_{gf} & K_{gw} \end{bmatrix} \begin{bmatrix} u_b \\ u_r \\ u_f \\ u_w \end{bmatrix} \\ \begin{bmatrix} y_b(k) \\ y_r(k) \\ y_f(k) \\ y_g(k) \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_b(k) \\ x_r(k) \\ x_f(k) \\ x_g(k) \end{bmatrix}. \end{aligned} \tag{4.4}$$

The 9 LED groups lamp is modelled in the same way. However the white LEDs are now directly linked to a partition of yellow light at wavelengths 535-620 nm since a green LED group is now available. The complete list of wavelength partitions is:

- UV: 350-390 nm
- Violet: 390-413 nm
- Indigo: 413-435 nm
- Blue: 435-485 nm
- Green: 485-535 nm

- Yellow: 535-620 nm
- Orange: 620-637 nm
- Red: 637-685 nm
- Far-red: 685-800 nm

The I/O relation for the 9 LED groups lamp is then

$$\begin{bmatrix} y_u \\ y_v \\ y_i \\ y_b \\ y_g \\ y_o \\ y_r \\ y_f \\ y_y \end{bmatrix} = \begin{bmatrix} K_{uu} & K_{uv} & K_{ui} & K_{ub} & K_{ug} & K_{uo} & K_{ur} & K_{uf} & K_{uw} \\ K_{vu} & K_{vv} & K_{vi} & K_{vb} & K_{vg} & K_{vo} & K_{vr} & K_{vf} & K_{vw} \\ K_{iu} & K_{iv} & K_{ii} & K_{ib} & K_{ig} & K_{io} & K_{ir} & K_{if} & K_{iw} \\ K_{bu} & K_{bv} & K_{bi} & K_{bb} & K_{bg} & K_{bo} & K_{br} & K_{bf} & K_{bw} \\ K_{gu} & K_{gv} & K_{gi} & K_{gb} & K_{gg} & K_{go} & K_{gr} & K_{gf} & K_{gw} \\ K_{ou} & K_{ov} & K_{oi} & K_{ob} & K_{og} & K_{oo} & K_{or} & K_{of} & K_{ow} \\ K_{ru} & K_{rv} & K_{ri} & K_{rb} & K_{rg} & K_{ro} & K_{rr} & K_{rf} & K_{rw} \\ K_{fu} & K_{fv} & K_{fi} & K_{fb} & K_{fg} & K_{fo} & K_{fr} & K_{ff} & K_{fw} \\ K_{yu} & K_{yv} & K_{yi} & K_{yb} & K_{yg} & K_{yo} & K_{yr} & K_{yf} & K_{yw} \end{bmatrix} \begin{bmatrix} u_u \\ u_v \\ u_i \\ u_b \\ u_g \\ u_o \\ u_r \\ u_f \\ u_w \end{bmatrix}, \tag{4.5}$$

where indices u, v, i, b, g, o, r, f, w and y correspond to the colors UV, violet, indigo, blue, green, orange, red, far-red, white and yellow respectively.

4.3 System identification

Measurements of the input/output relation according to the method presented in Section 4.2 are done for each lamp type. The results are presented below as grid plots representing the real measured I/O relation as well as the estimated relation as a linear equation $y = K_{\lambda}x$. The grid plots are built the same way as the K_{λ}-matrix, i.e each row corresponds to the integrated spectrum partition as a function of the different input intensities for each LED group.

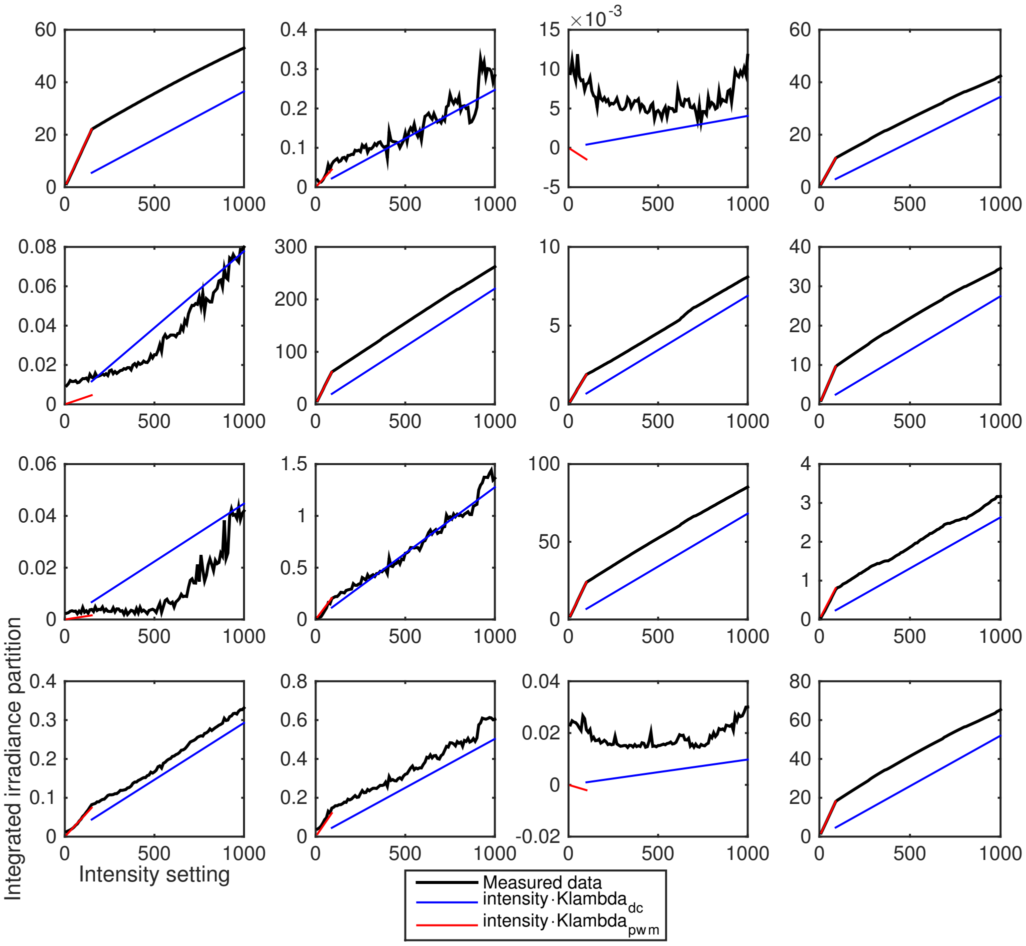


Figure 4.4: The input output relations of a lamp with 4 LEDs: blue, red, far-red and white. Note the different scales of the y-axes.

The grid plot in Figure 4.4 depicts the I/O relations for a lamp model with 4 LED colors and it can be seen in the figure that the relation is built up by two nearly linear lines. The lower segment of the line corresponds to intensity settings at which the lamp uses PWM. As stated before this setting ranges from 0 to around 80-150 intensity depending on the LED group. Above this threshold the intensity is set using DC voltage. Because of this behavior two separate K_λ matrices are introduced for the two intensity ranges.

Since there is a high grade of linearity it is of little effect to use this relation as a nonlinear function in practice, therefore a piece-wise linear estimation is used in the system. The linear estimations are based on the form $y = kx + m$ without the constant term m, which gives the linear estimation for DC voltage an offset. This will however only affect the system at the specific input where the switch between PWM and DC is since the controller uses increments. Looking closely at Figure 4.4 it can be seen that several of the plots show close to zero values. By cross-correlating this information to the spectrum in Figure 4.2 it can be determined that this is purely noise and the K_λ-matrix is therefore corrected according to:

$$\begin{bmatrix} K_{bb} & 0 & 0 & K_{bw} \\ 0 & K_{rr} & K_{rf} & K_{rw} \\ 0 & K_{fr} & K_{ff} & K_{fw} \\ 0 & 0 & 0 & K_{gw} \end{bmatrix}. \tag{4.6}$$

The grid plot in Figure A.1 (in the appendix) depicts the I/O relations for a lamp model with 9 LED colors. As seen in the Figure this lamp has a saturation at each LED group at intensity setting 800. This is taken care of by only using the unsaturated measurements when modelling the K_λ-matrix. The control signals for this lamp are also saturated at 800 intensity in the control system. Cross-correlation of the spectrum plot and the I/O relation plot is used to correct the K_λ-matrix according to:

$$\begin{bmatrix} K_{uu} & K_{uv} & K_{ui} & K_{ub} & 0 & 0 & 0 & 0 & 0 \\ K_{vu} & K_{vv} & K_{vi} & K_{vb} & 0 & 0 & 0 & 0 & K_{vw} \\ K_{iu} & K_{iv} & K_{ii} & K_{ib} & K_{ig} & 0 & 0 & 0 & K_{iw} \\ K_{bu} & K_{bv} & K_{bi} & K_{bb} & K_{bg} & 0 & 0 & 0 & K_{bw} \\ 0 & 0 & K_{gi} & K_{gb} & K_{gg} & K_{go} & 0 & 0 & K_{gw} \\ 0 & 0 & 0 & 0 & K_{og} & K_{oo} & K_{or} & K_{of} & K_{ow} \\ 0 & 0 & 0 & 0 & 0 & K_{ro} & K_{rr} & K_{rf} & K_{rw} \\ 0 & 0 & 0 & 0 & 0 & K_{fo} & K_{fr} & K_{ff} & K_{fw} \\ 0 & 0 & 0 & 0 & K_{yg} & K_{yo} & K_{yr} & 0 & K_{yw} \end{bmatrix}. \tag{4.7}$$

Because of the high grade of linearity very few measurements are needed to model the system. A new modeling script is therefore used, which only uses three different intensities for each of the piece-wise linear sections. This modelling script only takes a few minutes to run and will still be a good estimation of the system. This is also advantageous when setting up the system to run in a new environment, since a new tailor made model will quickly be created for that exact setup.

4.4 Model errors

As mentioned before, in order to speed up the process of determining the model of the system at startup, only a few measurements are taken for each LED color. Since the

K_λ-matrix is calculated as the mean value of several measurement ratios using more samples will result in a more accurate model. However, taking a lot of measurements takes a long time (especially for the 9 LED color lamp) and the acceptable error margins for the plant models are rather high, which makes it possible to use only a few samples for this calculation. To investigate how large the errors due to sparse sampling are, a comparison was made for the 4 LED groups lamp. The "fast" modelling uses only 3 samples at each input setting (PWM and DC), namely 50, 60, 70 for PWM and 300, 500 and 600 for DC. The "exact" modelling uses every 10th intensity setting from 10 all the way to 1000. The following values are obtained for DC:

$$K_{\lambda DC} = \begin{bmatrix} 0.0404 & -0.0001 & 0.0001 & 0.0392 \\ 0.0001 & 0.2485 & 0.0080 & 0.0307 \\ 0.0001 & 0.0006 & 0.0738 & 0.0029 \\ 0.0002 & 0.0003 & 0.0000 & 0.0571 \end{bmatrix} \tag{4.8}$$

and

$$FastK_{\lambda DC} = \begin{bmatrix} 0.0410 & 0.0002 & -0.0000 & 0.0394 \\ -0.0001 & 0.2496 & 0.0077 & 0.0308 \\ -0.0002 & 0.0010 & 0.0744 & 0.0023 \\ 0.0000 & 0.0005 & -0.0001 & 0.0580 \end{bmatrix}, \tag{4.9}$$

which has the percentage difference of

$$Difference_{\%} = \begin{bmatrix} 1.5522 & 0.1134 & 0.1609 & 0.7335 \\ 0.5757 & 0.4300 & 0.4945 & 0.4602 \\ 0.6273 & 0.1625 & 0.7944 & 0.7780 \\ 0.3811 & 0.0587 & 0.1691 & 1.8006 \end{bmatrix}. \tag{4.10}$$

As can be seen the difference is very small with a maximum of 1.8 % and a mean of 0.58 %. This has very little effect on the system and using the "fast" modelling makes it possible to model a new setup very fast.

4.5 Inverse square law

In general, light behaves according to the inverse-square law when it comes to distance,

$$Irradiance \propto \frac{1}{distance^2} \tag{4.11}$$

which states that the irradiance of the light is inversely proportional to the square of the distance to the light source. Because of this behavior it is theoretically possible to factor out a distance dependant proportionality constant γ from the K_λ-matrices. This constant could then be calculated for a new setup where the distance between lamp and spectrometer is simply measured and plugged into the model. This will then circumvent the need to identify a model by measurements for a new setup. The γ-constant is factored out simply by dividing the entire K_λ-matrix by its largest value.

The γ-constant for the system is modelled by making the K_λ measurements for different distances between spectrometer and lamp. Measurements are taken at the distances 10, 30, 60 and 90 cm in a reflective chamber.

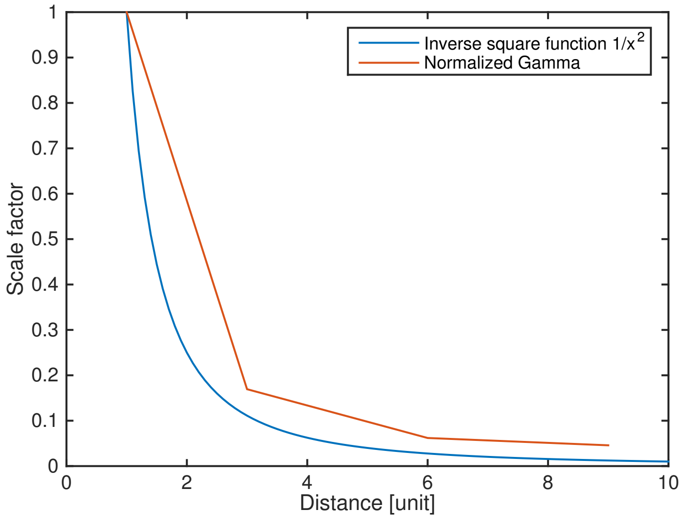


Figure 4.5: Normalized γ-constant factored out from K_λ at 10, 30, 60 and 90 cm compared to the inverse square function.

Figure 4.5 shows the behavior of the γ-constant which roughly follows the behavior of the inverse square law. Factoring out the γ-constant from each of the K_λ-matrices for the different distances should in theory yield exactly the same matrix. However due to external factors such as reflections and light leakage there is an error which had a maximum of 15 %.

4.6 Results of modelling

As seen in Section 4.4 the difference between a rigorous system modelling and a fast one is not large. Because of this and uncertainties in the γ-constant (Section 4.5) it is decided that the final control system will always measure and create a new model for every new setup or experiment, using only a few measurements.

5

Control

This chapter presents studies of the control of overlapping spectrum partitions as a MIMO system. An LQ controller with integral action (LQI) is presented, as well as a basic integral action controller using a decoupling network. Both of these controllers handle windup and saturate the control signal at the maximum lamp setting of 1000 or 800 depending on the lamp type. Since the lamp has very rapid dynamics the integration in both controllers is based only on samples, no consideration is taken to the true sample time. This is only possible because of the rapid dynamics. The continuous change in integration time would otherwise require the controllers to be tuned for the longest integration time. This would however result in a very slow controller for shorter integration times.

The effects of the different input settings PWM and DC are also presented in this chapter, as well as an extension of the control system that aims to control the total PAR irradiance.

5.1 Decoupled system

Interactions between the different LEDs on the same output might make a stable control difficult to achieve. However multi-variable control theory have been handled before. Boksenbom et al. (1949) [10] proposed the idea of decoupling control. The aim of the decoupling control is to make a multi-variable system non-interacting between individual inputs and outputs. This is done by using pre and post compensation such that the transfer function matrix becomes diagonal. This will make the relation between input and output one-to-one. Rearranging the system in this way makes it possible to use multiple SISO controllers to control a MIMO system.

A multivariable system can, according to Mesarovic, M.D. (As stated in the book by Liu, [11]), be described as P-canonical and V-canonical plants. These two forms are just different ways of formulating the system and both can be used for every system. The P-canonical form describes the system as each output being affected by all the inputs, while the V-canonical form describes the system as each output being affected by not only its input but also by all other outputs. The model described in Section 4.2 is naturally described in the P-canonical form which is therefore used.

When decoupling a system a decoupling network is created. There are different ways of calculating the elements in the decoupling network. In Liu's book [11] different ways of making the decoupling elements for both P-canonical and V-canonical systems are described. However the examples use specific systems and argue the different benefits of some decoupling systems. For the LED lamp system the transfer matrix is simple, a static gain, which makes simple solutions possible. This system uses the inverse of the K_λ-matrix to decouple the system. Figure 5.1 shows the system with the controller and the added decoupling network.

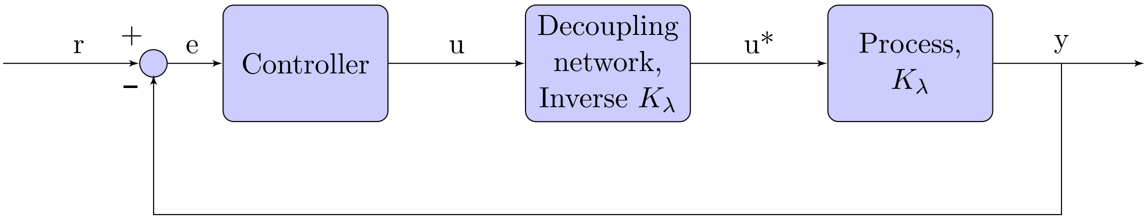


Figure 5.1: Decoupling system

5.2 Integral controller

This controller uses parallel integral controllers that each handle one partition of the spectrum in combination with the decoupling network described in Section 5.1. The reason for only using an integral controller is because the speed of the controller is less important than stability and ease of tuning. Imagine a scenario where plants are grown using the light control system, the system would then be set to run for several days or weeks. In that case, designing a controller that reaches a setpoint in the shortest time possible is simply not meaningful. With the use of only integral action the controller will still reach its setpoints in a few seconds but without risking unnecessary increases in sensitivity that comes with introducing proportional and derivative action.

The equation for the process, as stated in Section 4.2, is

$$y = K_\lambda u, \tag{5.1}$$

where y is the output vector, K_λ is the gain matrix and u is the input vector. To achieve a system where there is no coupling the gain matrix needs to be diagonal. This is done by adding the decoupling network. The control signal u from the controller will then be used as the input to the decoupling network and its output u^* will be the input to the plant defined as

$$u^* = K_\lambda^{-1} u. \tag{5.2}$$

If this output is used in the system a one-to-one relationship is obtained for the system since

$$\begin{aligned} y &= K_\lambda u^* \\ &= K_\lambda K_\lambda^{-1} u \\ &= Iu, \end{aligned} \tag{5.3}$$

where I is an identity matrix. The process is then clearly as simple as possible and discrete integral controllers can be added for each input/output pair. Using the time shift operator q, the plant and integral controller can be described as

$$\begin{aligned} y &= Iq^{-1}u \\ (1 - q^{-1})u &= K_i(r - y), \end{aligned} \tag{5.4}$$

where Ki is the integral action gain and r is the reference vector. Combining these two equations the system is described as

$$y = K_i I \frac{q^{-1}}{1 - q^{-1}}(r - y). \tag{5.5}$$

Chosing K_i to be scalar, the same controller is used for all input-output relations, i.e.

$$\frac{y}{r} = \frac{K_i}{(q - 1) + K_i}. \tag{5.6}$$

The closed loop system gives information about the behavior of the controller. As can be seen the system has poles in $q = 1 - K_i$. The system is stable iff the poles are within the unit circle for a discrete system. This makes the requirement for stability

$$K_i < 2 \tag{5.7}$$

for all states. The fastest controller (deadbeat) has all its poles in zero and should theoretically reach its setpoints in one sample [12]. This is achieved by setting

$$K_i = 1 \tag{5.8}$$

for all SISO controllers. However since there is some noise in the system a lower K_i could be beneficial to achieve a better robustness.

Figure 5.2 shows the behavior of the decoupled integral control scheme for the 4 LED lamp and K_i set to 1. As can be seen the references are almost reached in one sample.

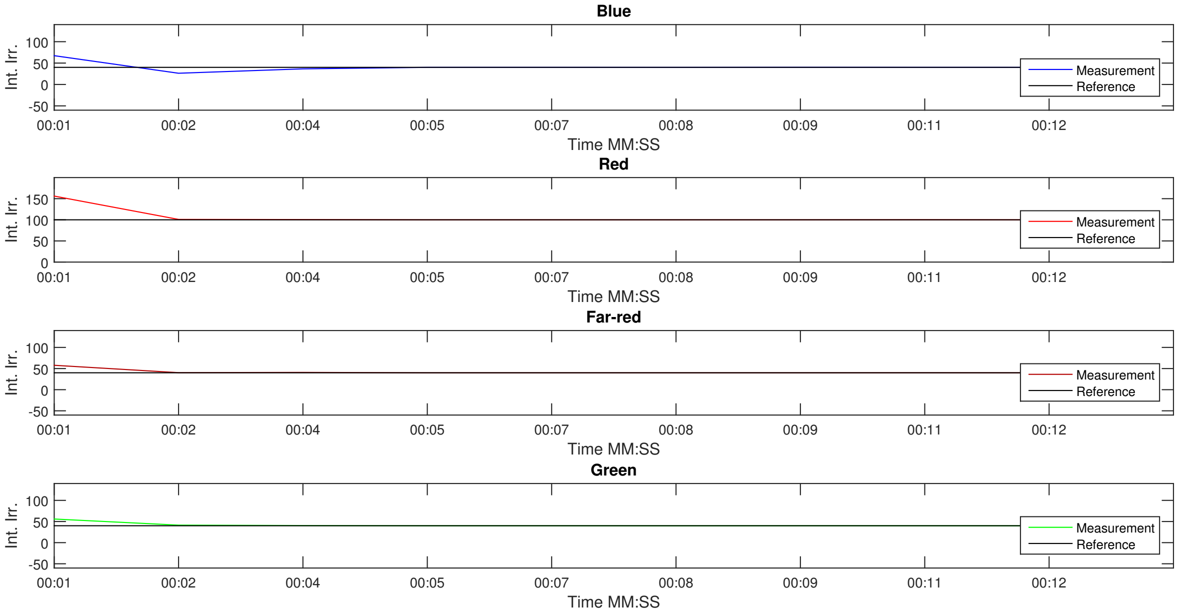


Figure 5.2: Integral controller test.

5.3 LQI control

For a case with overlapping light spectra a LQI controller is implemented acting on the state space model describing the overlapping behavior. The LQI controller is defined as

$$u = -Lx. \tag{5.9}$$

In order to achieve the desired integral action the state space model (for the 4 LED lamp) is augmented with integral states according to:

$$\begin{bmatrix} x_b(k+1) \\ x_r(k+1) \\ x_f(k+1) \\ x_g(k+1) \\ x_{bi}(k+1) \\ x_{ri}(k+1) \\ x_{fi}(k+1) \\ x_{gi}(k+1) \end{bmatrix} = \begin{bmatrix} A & 0_{m\times 4} \\ -C & I_{m\times 4} \end{bmatrix} \begin{bmatrix} x_b(k) \\ x_r(k) \\ x_f(k) \\ x_g(k) \\ x_{bi}(k) \\ x_{ri}(k) \\ x_{fi}(k) \\ x_{gi}(k) \end{bmatrix} + \begin{bmatrix} B \\ 0_{4\times n} \end{bmatrix} \begin{bmatrix} u_b(k) \\ u_r(k) \\ u_f(k) \\ u_w(k) \end{bmatrix} + \begin{bmatrix} 0_{n\times m} \\ I_{n\times m} \end{bmatrix} \begin{bmatrix} r_b(k) \\ r_r(k) \\ r_f(k) \\ r_g(k) \end{bmatrix} \tag{5.10}$$

where m is the number of rows in the A-matrix and n is the number of columns in the B-matrix. x_{bi}, x_{ri}, x_{fi} and x_{gi} are added for handling integration of the error. This particular LQI control scheme uses integral action as a means for tracking the reference. By inspection of the augmented state-space model it can be seen that

$$x_i(k+1) = -Cx(k) + x_i(k) + r \tag{5.11}$$

which means that

$$x_i(k+1) = r - y + x_i(k). \tag{5.12}$$

Since the LQ controller aims at $x_i = 0$ the the controller will track the reference.

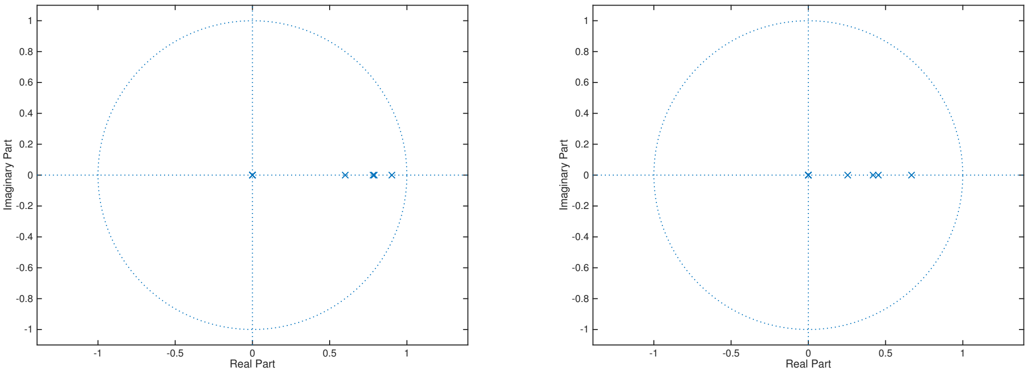
The Matlab command 'dlqr' is used to calculate a gain matrix L for the control system using the weight matrices Q and R in Equations (5.13) and (5.14), respectively.

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 50 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 20 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 50 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 50 \end{bmatrix} \tag{5.13}$$

$$R = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix} \tag{5.14}$$

The Q-matrix contains the weights of the different states in the system. The top four rows correspond to weights on the original states and the bottom four correspond to the introduced integral states. The integral states are weighted higher than the original states resulting in a slower but more stable system than a vice versa weighting. The R-matrix holds the weights of the control signal.

Applying these weights and calculating the closed loop system results in the pole placement seen in Figure 5.3a for the DC K_λ-matrix.



(a) Using K_λ-matrix for DC input.

(b) Using K_λ-matrix for PWM input.

Figure 5.3: Pole placements of the LQI controller for the 4 LED groups lamp.

More specifically the poles are placed at:

1. 0.6009
2. 0.9026
3. 0.7810
4. 0.7886
5. -0.0000
6. 0.0000
7. 0.0000
8. 0.0000

The same weights are used for the PWM part of the K_λ-matrix, which places the poles according to Figure 5.3b, or more specifically:

1. 0.6681
2. 0.2549
3. 0.4195
4. 0.4532
5. -0.0000
6. 0.0000
7. 0.0000
8. 0.0000

As seen in the figure all poles are within the unit circle which means that the system is stable. Figure 5.4 shows the behavior of the LQI control system.

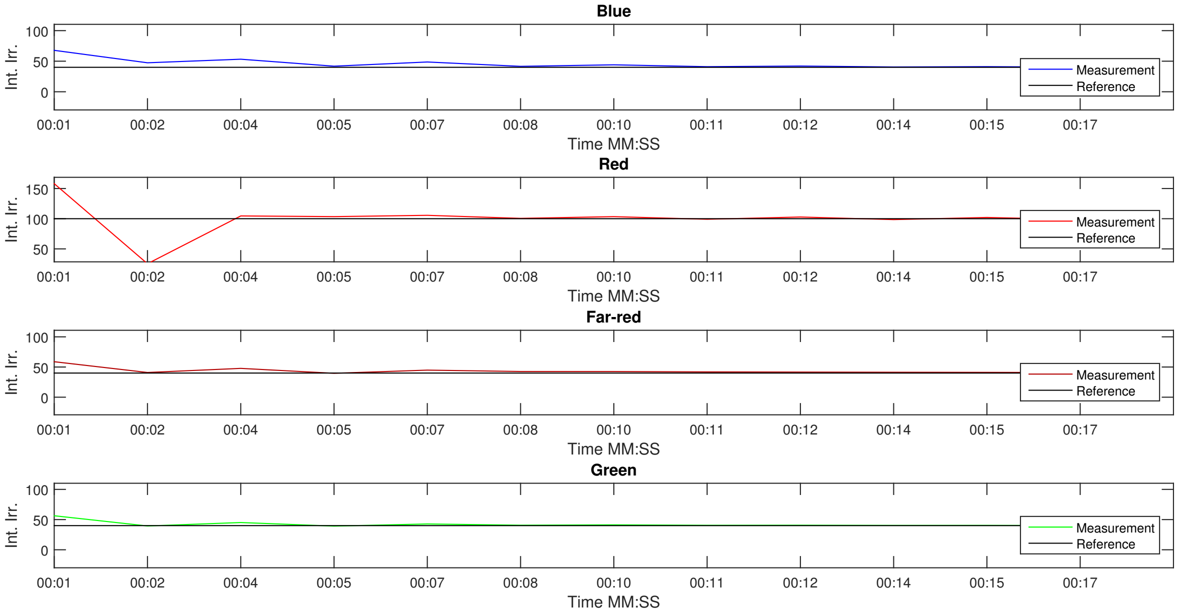


Figure 5.4: A test run of the LQI control system.

As seen in the figure, reaching the references takes a bit longer for the LQI controller than for the integral action decoupling controller (Figure 5.2). However, the LQI controller might not be optimally tuned.

5.4 Correcting for PWM/DC

Since the lamps use both PWM and DC for input, the controller needs to adapt accordingly. Whether PWM or DC is used is determined by what intensity is set to the LED group. The threshold intensity limit is slightly different for all the lamps as can be seen in Figures 4.4 and A.1. The input for each LED group is determined after the controller update. The change from PWM/DC to the other is therefore delayed one sample.

A change from PWM/DC to the other will make the system overshoot. This can make the system oscillate if the reference is close to or on the point where the switch occurs. Each switch will cause an overshoot which in turn will cause a new switch. This phenomenon is seen in Figure 5.5 for the integral controller.

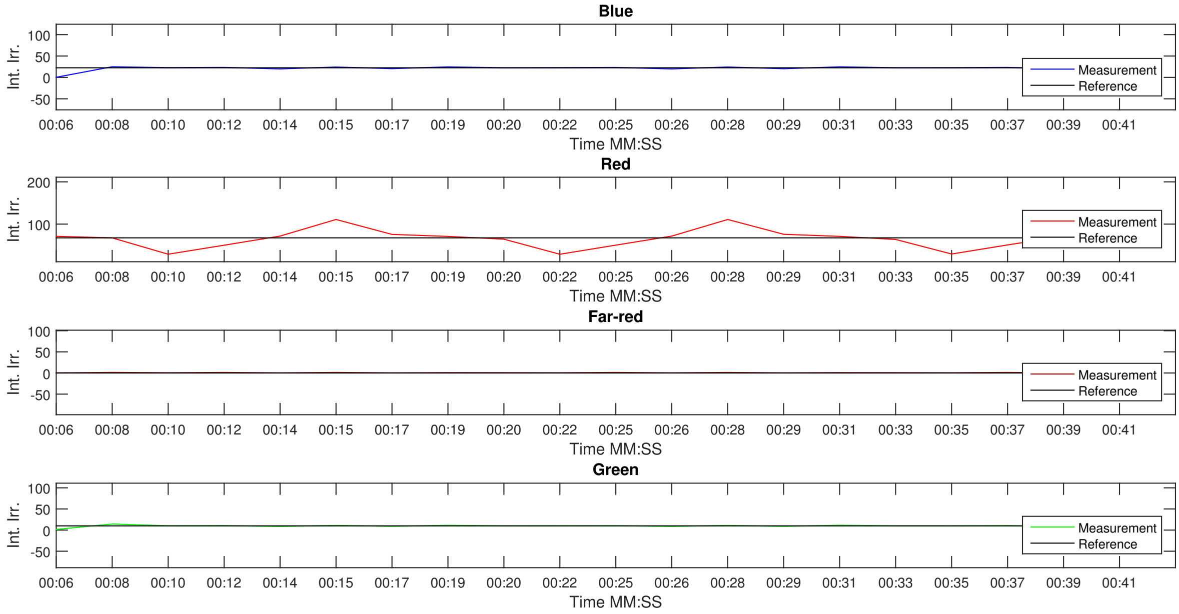


Figure 5.5: Test run where the reference is close to the PWM/DC switch for the red color.

The figure shows a case where the red colors reference is close to the switch threshold. The reference is reached when the input of the red lamp is set to 87. The switch from PWM to DC occurs at 90. When the lamp is in DC (measurement is above the reference) the intensity is slowly decreased. When it is close to the reference it will switch to PWM. Then the calculated input to the decoupling network will use a new model, i.e. the PWM model for the red lamp. As a result the intensity is set lower. Because the inverse of the PWM model has much smaller elements than the DC one. Vice versa the switch from PWM to DC will cause the intensity to be set too high. The switch from PWM to DC will occur since the controller overshoots the DC level and therefore cause the switch. Reducing K_i solves this problem. As can be seen in Figure 5.6, a K_i of 0.6 will result in a stable control.

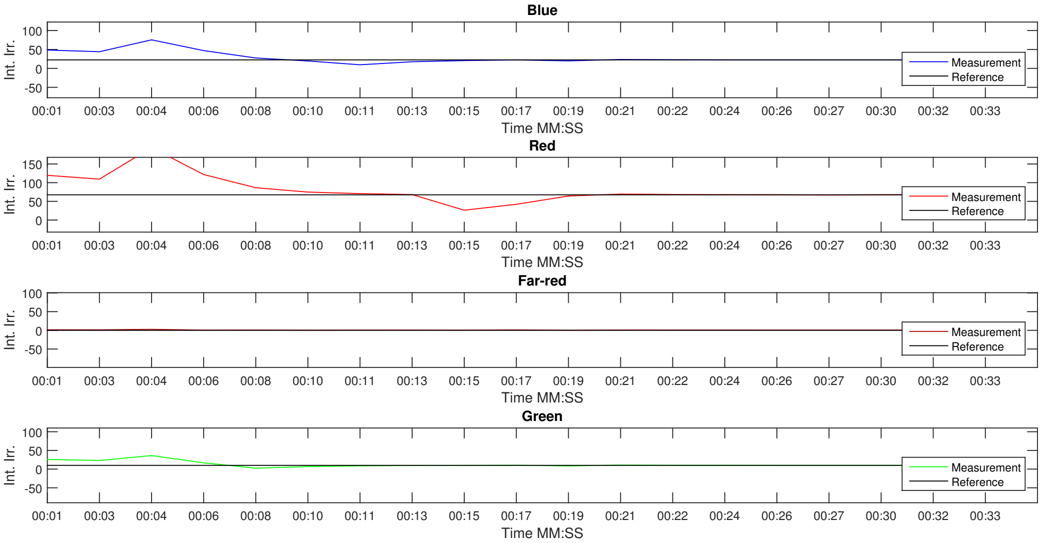


Figure 5.6: Test run where the reference is close to the PWM/DC switch for the red color, now with lower integral action.

The overshoot is smaller for the LQI controller, as can be seen in Figure 5.7

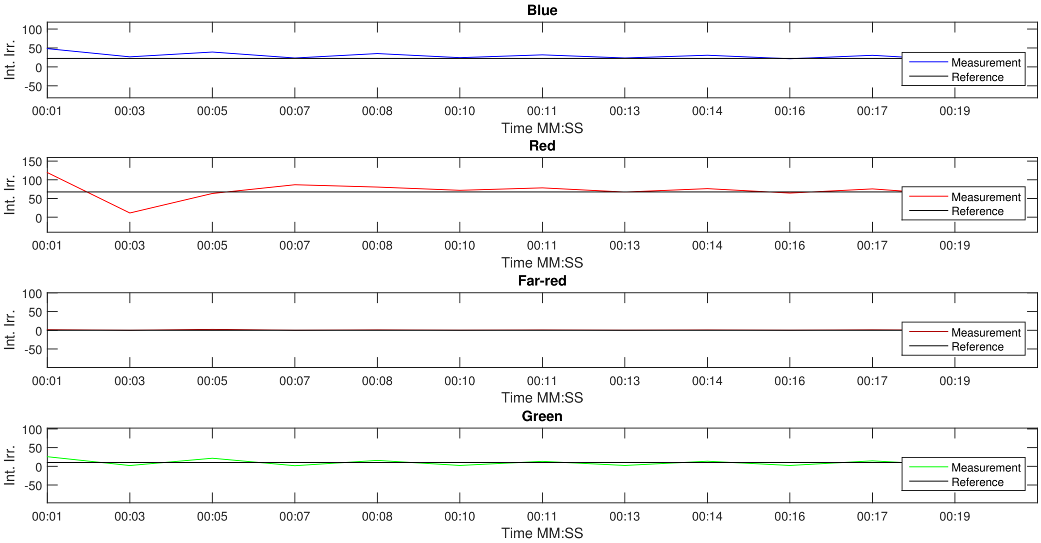


Figure 5.7: Test run where the reference is close to the PWM/DC switch for the red color using the LQI controller.

However since two models need to be considered when tuning the controller, stability is hard to obtain for the LQI controller. The used weights stated in Equation (5.13) and (5.14) are not sufficiently stabilizing the system in this situation. When the switch occurs the LQI controllers state space model changes and a new feedback gain matrix is calculated. Similarly, for the integral action controller the decoupling network is changed. Since each LED group can have different input (PWM or DC) simultaneously each column of the K_λ-matrix needs to be changed individually.

5.5 PAR control

When controlling the lamps in an environment with external light sources, such as a greenhouse or an indoor growth chamber with additional lamps affecting the measured spectrum, it is possible to end up in states where the reference cannot correctly be followed due to hardware limitations. Say, for example, that the desired integrated spectrum of one color is rather low, if an external light source provides a higher light intensity than that, the reference will be impossible to reach since the only possible action at that point is to completely shut off the lamp and even that will not be enough. A vice-versa scenario occurs if a reference is set too high to be reached even with the highest intensity input. An example of this is a rather high reference that holds during the day in a greenhouse. During night time when the external light is low or non-existing it might no longer be possible to follow the references of certain color partitions. Using references like these are, however, not recommended.

When either of those scenarios occur the system will control the total PAR irradiance. The PAR control loop can be seen in Figure 5.8.

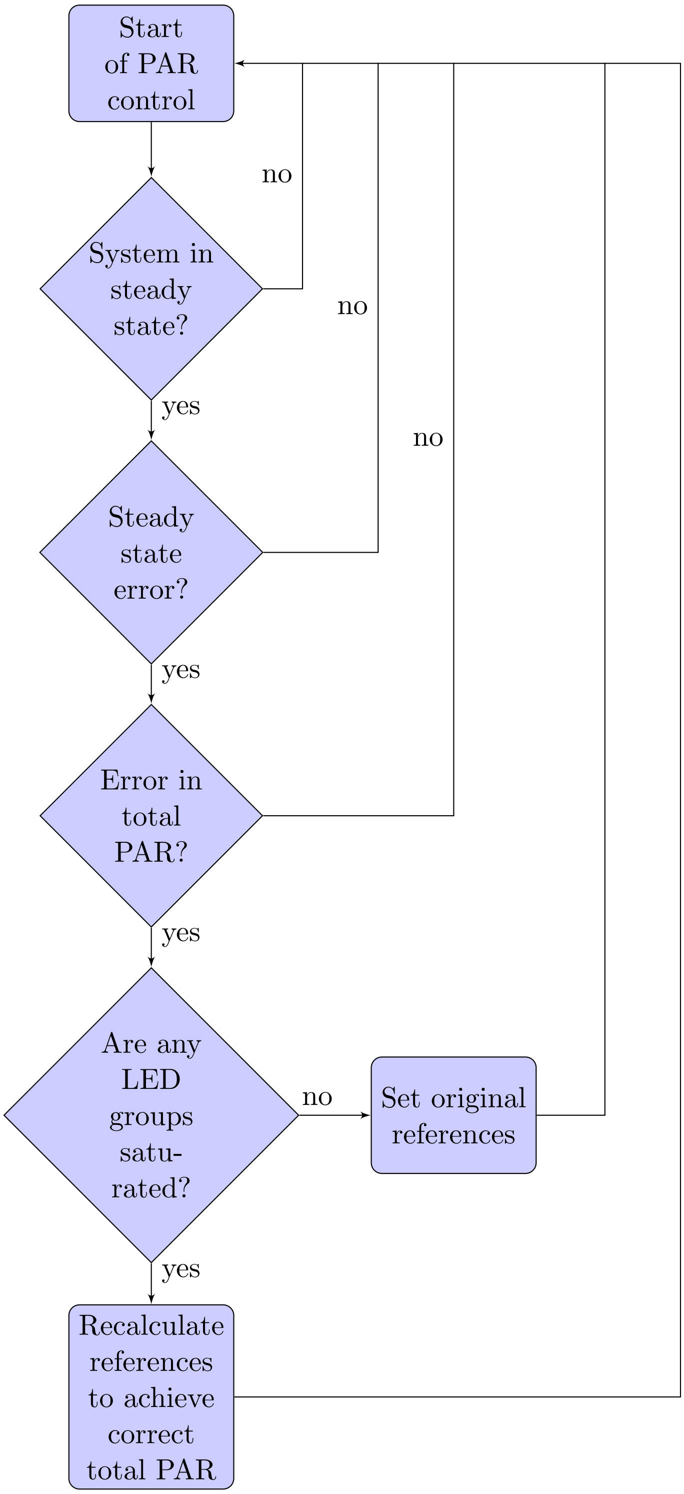


Figure 5.8: PAR loop.

The PAR control loop starts by determining if the system is in steady state or not. If that is the case, calculations are done in order to determine if the system has steady state errors. A steady state error present in the system in combination with the most influential corresponding LED group being saturated means that a particular integrated spectrum partition reference is impossible to reach. If this happens the references are recalculated in order to uphold the total PAR determined by the original references. More specifically this is done by equally adjusting the references of all the lamps that are not saturated such that they add up to the total PAR stated by the original reference. This way the main control scheme can still be used but with adjusted references. If and when external light conditions are changed in a way such that the original reference is possible reach again, the system will simply switch back to the original references.

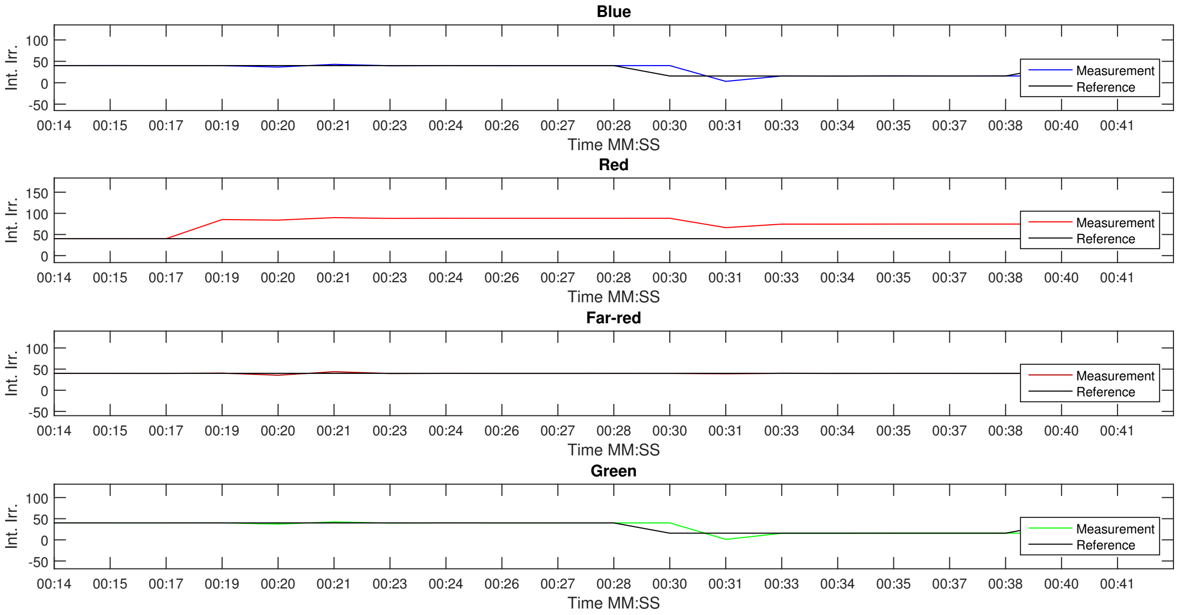


Figure 5.9: PAR control running on the integral control scheme.

Figure 5.9 shows an example of how the PAR control works. A disturbance occurs at time 19 seconds resulting in a steady state error above the reference value. Since the reference value of the red spectrum is relatively low the control signal is set to zero for the red lamps but the steady state error remains due to the strong disturbance. The total irradiance within PAR wavelengths is now too high. When the system is in steady state, i.e. no changes have been made in the system, in this case at time 28 seconds (this is decided by a threshold variable set high at this time to ensure that steady state is achieved) the references of the blue and green colors are lowered such that the desired PAR is upheld. The decrease in blue and green is equal and together they sum up to the steady state error of the red color. The far-red spectrum partition does not belong to PAR and is therefore unchanged.

5.6 Saturation

The highest input setting most of the lamps can use is 1000. However, since many LEDs affect adjacent LED spectra, a limit on the input is not enough. Since the controller is not constrained it operates under the assumption that inputs over 1000 are valid. It will therefore sometimes lower the intensity setting of a LED group under the assumption that a color partition gets enough light from another LED group, which is not the case if that one is saturated. For example if two LED groups affect the same spectrum partition and the controller demands an output over 1000 from the first LED group, the second LED groups intensity is improperly based on the first LED groups unsaturated setting. The consequence of this is that the spectrum partition will get too little light even though the second LED group might not be saturated. To solve this, the saturation is placed on the integrated irradiance instead of the intensity. This limit is calculated by multiplying K_λ with the maximum intensity, namely 1000. This saturation will, however, have some errors due to model inaccuracies and must therefore be saturated based on the intensity as well, otherwise the final control signal that is sent to the lamp might be above 1000 or below 0 which causes a system crash.

For the integral controller this saturation is easily added before the decoupling network. However, for the LQI controller the same problem arises but the limit can not be added to the system in a similar way. The LQI controller does not calculate the integrated irradiance but instead calculates the input signal directly. To be able to limit the integrated irradiance it has to be calculated based on this input. This is done by multiplying the output with K_λ. The integrated irradiance is then obtained and can be saturated. The saturated values then need to be calculated into output again with the inverse of K_λ. This will give new outputs that are saturated in a correct way. Since the integral is used for both controllers the saturation also leads to windup. For the integral controller this is easily handled since the control signal is calculated as an increment. Each integrated irradiance measurement is saturated before it is saved for the next iteration which removes the windup.

The windup in the LQI controller is caused by the use of the accumulated error for integral action. The LQI controller removes windup using a standard anti-windup algorithm (Lennartson (2002) [13])

$$
\begin{aligned}
&if(u >= satMax) \\
&\quad v = satMax - u \\
&\quad u = satMax \\
\\
&if(u <= satMin) \\
&\quad v = satMin - u \\
&\quad u = satMin \\
\\
&\quad u = -Lx + v \cdot Tt,
\end{aligned}
\tag{5.15}
$$

where v is the excess control signal and Tt is an anti-windup tuning parameter.

5.7 Choice of controller

Two different control types are presented, an integral controller using decoupling of the system and an LQI controller. Comparing Figures 5.2 and 5.4 it can be seen that the integral controller have a better stability. The weightings on the LQI could however be improved to get a similar result. This is tedious and since the PWM and DC have different gains, poles will not be the same for the PWM as for the DC. This is not the case for the integral controller, which makes it perform better. The poles will also change with new setups for the LQI controller. Which requires the LQI to recalculate its gain matrix for each change in setup. This is not required for the integral controller. Considering the RPI implementation written in C the integral controller is also far less complex. This is mainly due to the fact that the LQI controller uses Matlabs 'dlqr' command. This command calculates the optimal gain matrix L by minimizing a quadratic cost function. Doing these equations is rather problematic to implement in C.

Considering this, the decoupled integral controller is the best choice of controller for the system. From this point every mention of controller refers to the integral controller.

6

Light references

This chapter is about the different ways to define a reference spectrum and how this is handled in the control system. The control system always uses the integrated irradiance over specified wavelength intervals. However, some modifications must be made for different input references in order to produce a usable reference.

6.1 References used in research setups

In plant research the light spectrum is often defined based on total integrated irradiance within PAR wavelengths and ratios between different colors or wavebands. Since PAR is the wavelength interval 400-700 nm only red, green, white and blue colored LEDs are used with an occasional far-red component.

All of Heliospectras lamps have a pure blue and red group of LEDs covering the wavelengths 400-500 nm and 600-700 nm. The green wavelength interval 500-600 is produced either with a green LED or a white one for those lamps that do not have a green LED. Most of the lamps also have additional LEDs that have a range completely or partly outside the PAR wavelength range. These LEDs may or may not be used to create this type of reference depending on the operators preferences. By using lamps with more LED groups it is possible to create a more evenly spread spectrum inside PAR but at the cost of system complexity. When using additional LEDs the PAR ratio can either be split equally amongst the LEDs covering each waveband or divided according to the capabilities of each LED group, which is determined by the K_λ-matrix.

6.2 Dynamic references

The system is able to handle dynamic or moving references that enables the simulation of circadian cycles. Use of such a reference is enabled by input of a matrix of size $1024xNumberOfReferences$ which then specifies the entire spectrum over a series of samples. The sample time is also specified at system start and decides how often the reference is changed. This could for instance be every hour, but also every minute if a higher resolution is needed. Note that this interval must be the same as for a sunlight data collection algorithm or the sunlight reference programs sample time in order to produce a reasonable circadian reference.

6.2.1 Measuring sunlight

By measuring real sunlight it is possible to achieve a reference spectrum that mirrors real outdoors conditions. It will also provide understanding of some of the difficulties the controller faces when placed in a greenhouse. Difficulties are for example, moving clouds and major differences in light intensity at different time of day due to shadows and weather conditions. Sunlight measurements are taken with the STS-VIS spectrometer connected to the RPI to enable a flexible placement of the measurement unit.

6.2.2 Calculating sunlight

There are programs available that can calculate the spectrum of the sun at a specified date, time, longitude, latitude and above sea level. These programs use meteorological data to calculate the amount of light that reaches the earths surface. The calculations only consider the atmosphere and the angle of the sun. Disturbances from weather is therefore not taken into account. One such program is Solar CalQ which is used in this system to create sunlight references [14]. By specifying the above mentioned parameters Solar CalQ produces an Excel spreadsheet containing an array of spectra. This spreadsheet can easily be used in Matlab or other programs to create a usable reference for the system.

7

Raspberry Pi implementation

Since it is desirable for Heliospectra to use the developed control system as a complete product it is suitable to implement the system on a platform other than a PC running a Matlab script, in this case a Raspberry Pi single-board computer. This is done partly in order to simplify the use of the STS since it will then no longer need to be connected to a PC, but also to circumvent the use of Matlab. The RPI (Figure 7.1) used in this application is running a Pidora operating system [15]. Initially the RPI was to be used simply as an extension of the spectrometer, replacing a USB connection with Wi-fi and run the control system as a desktop application (not Matlab). However an implementation such as that requires continuous connection between PC and RPI and in the event of a network failure the control system will be shut down. By instead placing the controller locally on the RPI it can run offline and connection only needs to occur when observation of the system is required. Since the system will typically run in sessions of several days it is not of interest to continuously observe the system.

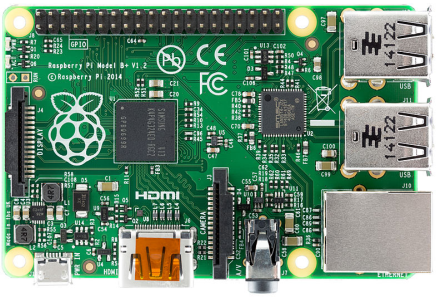


Figure 7.1: The Raspberry Pi single-board computer. Photo courtesy of Lucasbosch. [16]

Spectrometer manufacturer OceanOptics provide a C library called SeaBreeze which includes functions for operating the STS-VIS. Functions include for example setting the integration time, reading a spectrum and taking a dark reference.

Employees at Heliospectra have previously created a daemon program using this library as a basis. Said program runs as an application on the RPI and starts at boot, an endless event loop is then started which continuously asks for commands. Connecting the RPI via Wi-fi to a PC using a telnet protocol enables transmissions of such commands and the result can be read using the same telnet port.

The RPI implementation is a scaled down version of the Matlab implementation, it does not include the PAR controller or the handling of dynamic references.

7.1 Setting the lamp intensity using C

In order to control the lamps solely with the RPI a connection between the lamp and the RPI needs to be set up.

The software architecture of Heliospectras lamps is constructed so that a packet describing lamp settings can be sent via ethernet using a basic UDP protocol. Packets must include the following elements in order to successfully operate the lamp:

- A timestamp which increments at each sent packet.
- A calculated checksum of the packet.
- Length of the packet.
- Command type, for example set a light intensity.
- Command data, describing the number of different LED colors in receiving lamp.
- Multicast IP adress.
- Pairs of LED wavelengths and desired setting.

Each of these elements speak for themselves except for the multicast IP adress. This is used as a mean to enable control of several lamps at once. By setting up a lamp to run in a slave mode on a defined IP it will follow all settings that are done for the master lamp which holds the multicast IP. This is used by simulating a master lamp on the Raspberry Pi but can also be used to run several lamps with the same control scheme and spectrometer.

The UDP protocol has a very simple structure. It uses no safety techniques to ensure the data transfer. Which means that the recipient (the lamp) cannot know if it has received every package or not. This could potentially lead to some package loss. However, this will only affect the control system in the sense that a sample is lost in which case the system will keep its previous settings and wait for the next measurement.

7.2 Raspberry Pi control system

The integral action control system with decoupling designed in Matlab is implemented on the RPI by translating the code to C. A thread is created that runs the control system in parallel with the previously mentioned command triggered event loop, this thread is also started from said event loop. The parallelism makes it possible to manually override reference values at runtime. It also makes it possible to read measurements and other variables from an external program in order to visually observe the system. The control system is installed and run as a daemon, which means that the system is started when the RPI boots. That way it is not necessary to start up the interface of the operating system on the RPI or even log in. This circumvents the need for a monitor, keyboard and mouse. The RPI simply needs to be powered in order to remotely run the control system.

8

Results

In this chapter the results of tests performed on the complete final system is presented and reviewed. Plots depicting the measured and reference spectra show a large difference. This is because the system does not strive to produce the exact form of the spectrum but rather the irradiance content of it, which is shown in detail in the corresponding spectrum partition plots.

8.1 Plant lab conditions

This section presents results from tests performed on the system in a typical plant lab setup. This means that the environment is controlled and there are little or no disturbances from external light sources.

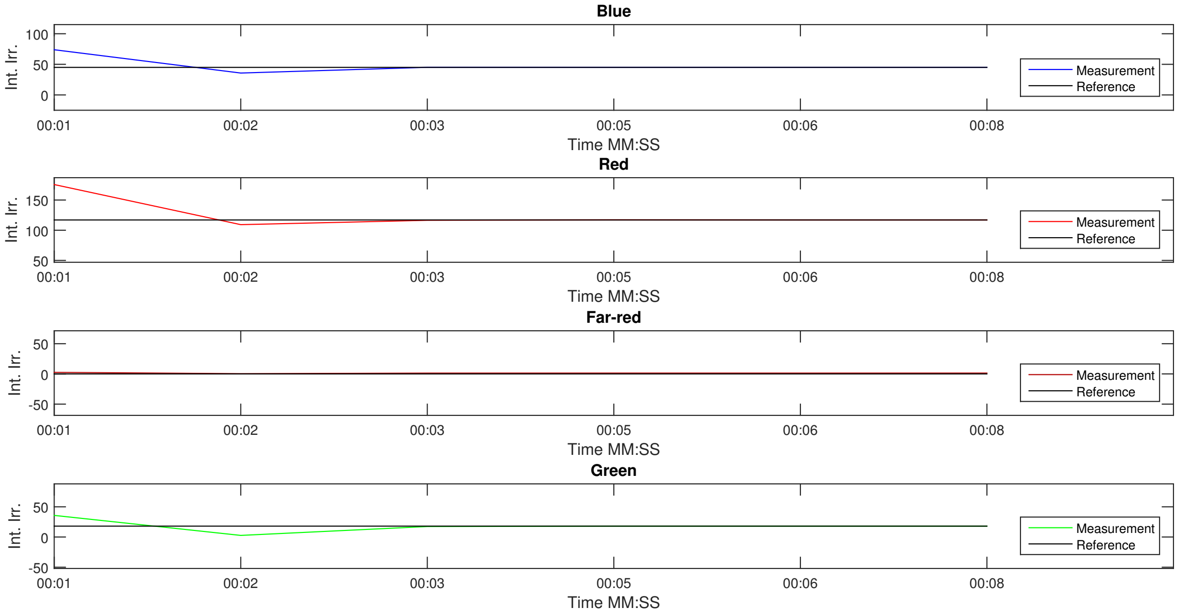


Figure 8.1: 4 LED lamp run in plant lab environment, $K_i = 1$.

Figure 8.1 shows the behavior of the control system with arbitrary references within the PAR waveband. More specifically the references are 40 integrated irradiance for the blue parition, 120 for the red, and 20 for the green. As seen in the figure the reference values are reached in two samples. Here, the integral action gain K_i is set to 1. However, since there is no real requirement for high speed in the control system in terms of minimizing the error, a lower integral action could also be used to increase stability margins. When the integral action is set to 1 as in Figure 8.1 the overshoot is big enough for the system to instantly switch to PWM level the first sample. It will then slowly get to the reference and never go over it, making so that it stays on PWM. With a lower K_i the system will in this case slowly reach the PWM level and then overshoot. However with a low enough K_i it will stabilize after the switch.

Figure 8.2 shows the behavior of the same system with the same references but with a disturbance generated by the 9 LED lamp. The 9 LED lamp is placed next to the 4 LED lamp with every color intensity set to 500.

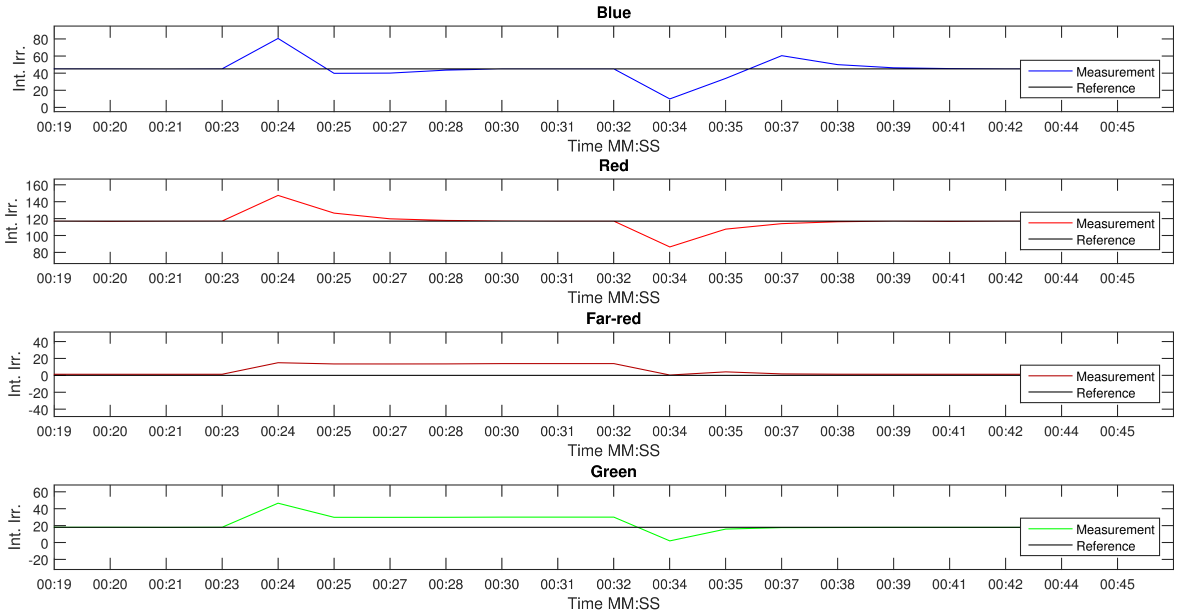


Figure 8.2: 4 LED lamp with par references interfered with 9 LED lamp with 500 intensity setting on all colors.

It can be seen that peaks are generated in the spectrum partitions after 24 s when the disturbance is applied but the controller manages to compensate and return to reference on the blue and red partitions. The far-red and green partitions get a steady state error since the disturbance is high enough to cause a complete shutdown of those LEDs and even that is not enough to get to the reference. At 34 s the disturbance is shut off and the system goes back to normal. This test is run with K_i set to 1 since this setting is the most sensitive used. The PAR controller is turned off in this test.

8.2 Natural varying sunlight

The RX30 (9 LED) lamp has the widest spectrum range of all Heliospectras lamps. It is therefore the most likely to best follow a sunlight reference. The reference light spectrum used for this test is calculated with Solar CalQ and simulates a 24 hour period at the spring equinox March 20th 2015 in Gothenburg. The sampling time of the calculated sunlight reference is one hour.

Figures 8.3 and 8.4 shows the integrated spectrum partitions and complete spectrum respectively at a sample corresponding to 8:00 in the morning. As seen in the figure references are followed correctly.

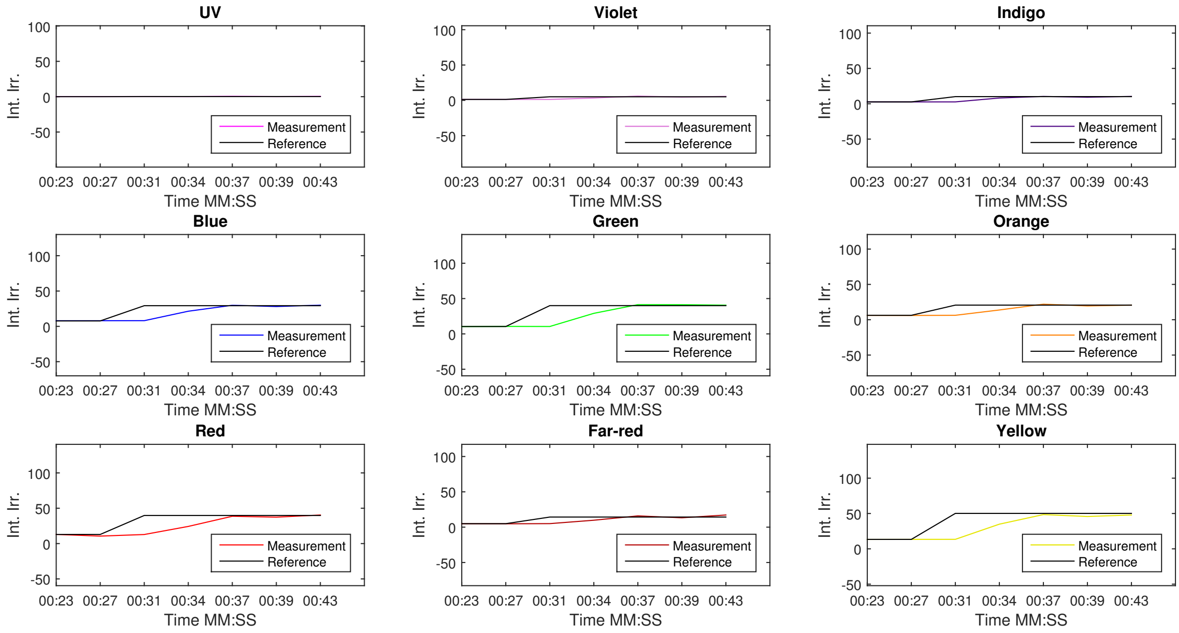


Figure 8.3: Integrated spectrum partitions at 8:00.

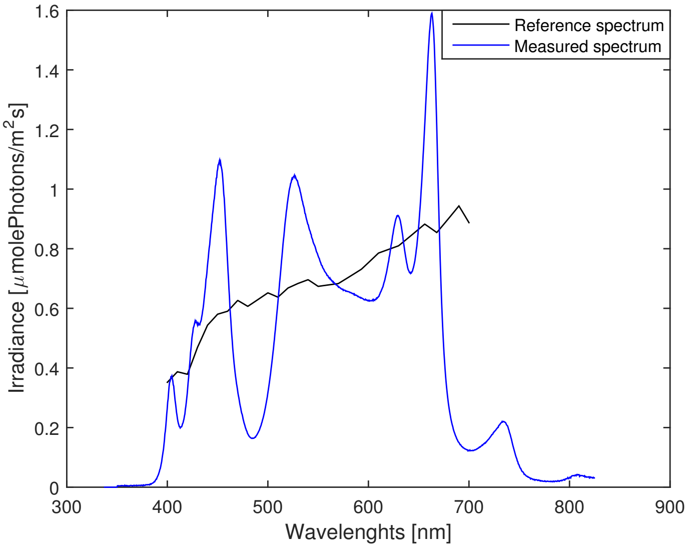


Figure 8.4: Measured and reference spectrum at 8:00.

Figures 8.5 and 8.6 shows the integrated spectrum partitions and complete spectrum respectively at a sample corresponding to 10:00 in the morning. As seen in the spectrum partitions plot there is a steady state error present in many of them. This is due to hardware restrictions of the lamp which simply cannot produce enough light to reach the reference values.

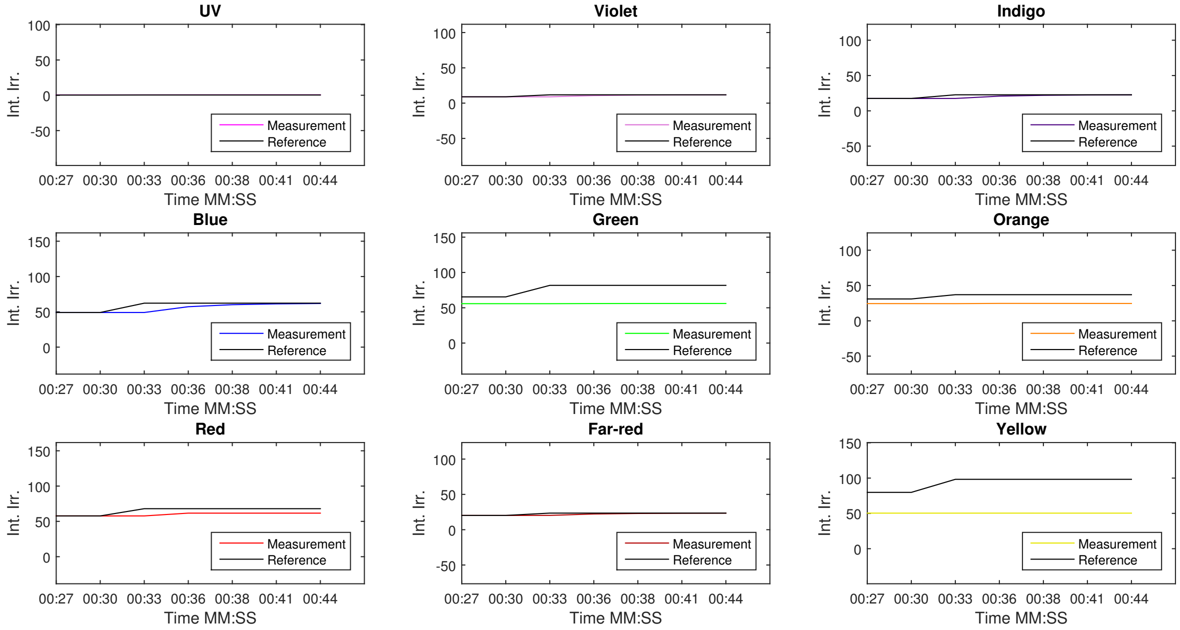


Figure 8.5: Integrated spectrum partitions at 10:00.

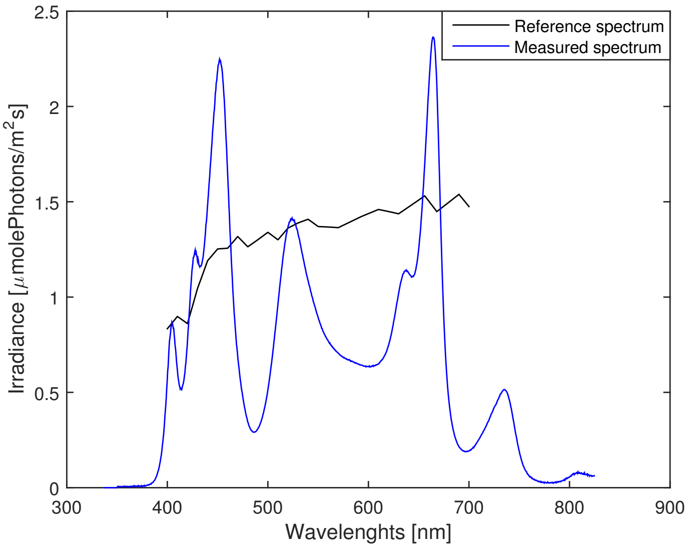


Figure 8.6: Measured and reference spectrum at 10:00.

Figures 8.7 and 8.8 show the same thing as Figures 8.5 and 8.6 but with the references set to 50 % of the original. It can now be seen that the references are followed correctly. This test shows that it is possible to correctly produce varying natural sunlight by using additional lamps since the irradiance produced is additive. The reason the test is done this way rather than by simply using an additional lamp is because no more lamps were available at the time of the test.

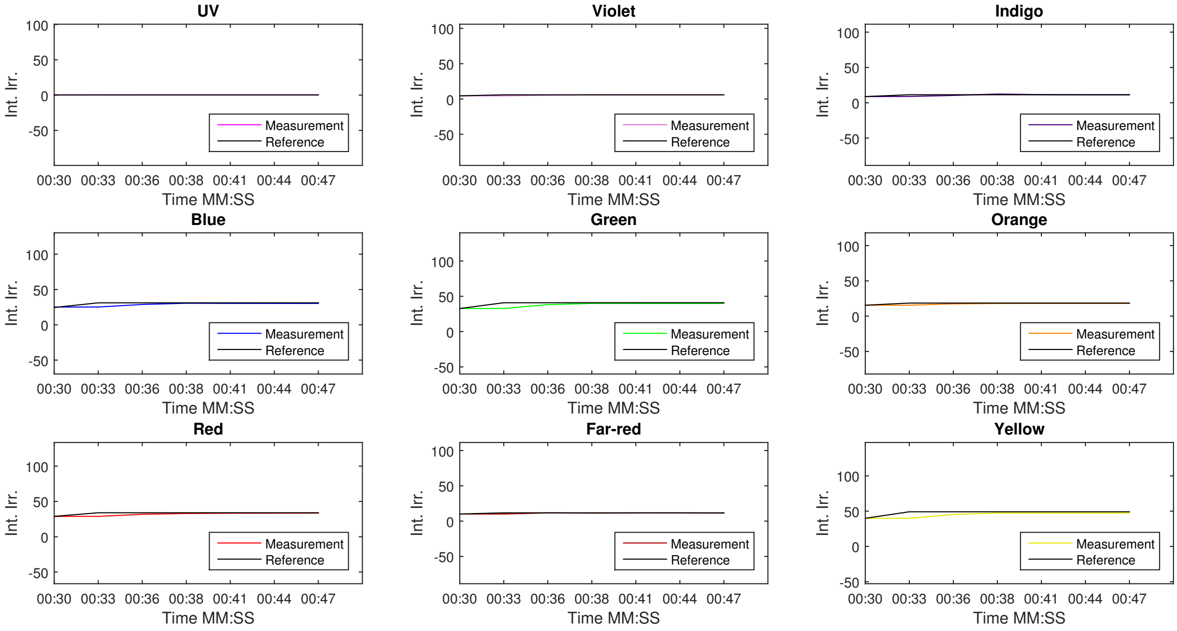


Figure 8.7: Integrated spectrum partitions at 10:00 using 50 % of the reference in Figure 8.5.

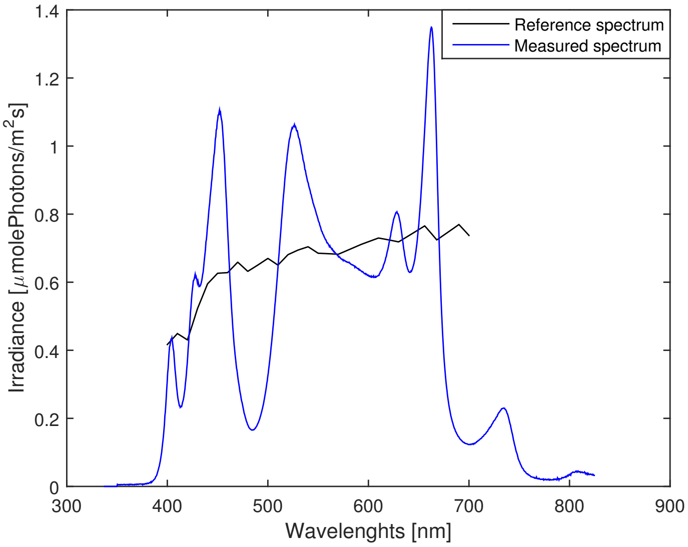


Figure 8.8: Measured and reference spectrum at 10:00 using 50 % of the reference in Figure 8.6.

8.3 Greenhouse conditions

This section presents results from tests performed in a simulated greenhouse environment. No greenhouse was available at the time tests were made so in order to test the system another Heliospectra lamp was used as a disturbance source. The disturbance lamp is a 9 LED lamp which is set to run saved control signal values from the test in Section 8.2 (sunlight from 8 am) in open loop. Note that hardware limitations of the lamp prevent a complete reference tracking of the sunlight reference, which means that the disturbance would be even higher in a real greenhouse scenario. Figure 8.9 shows the behavior in this test.

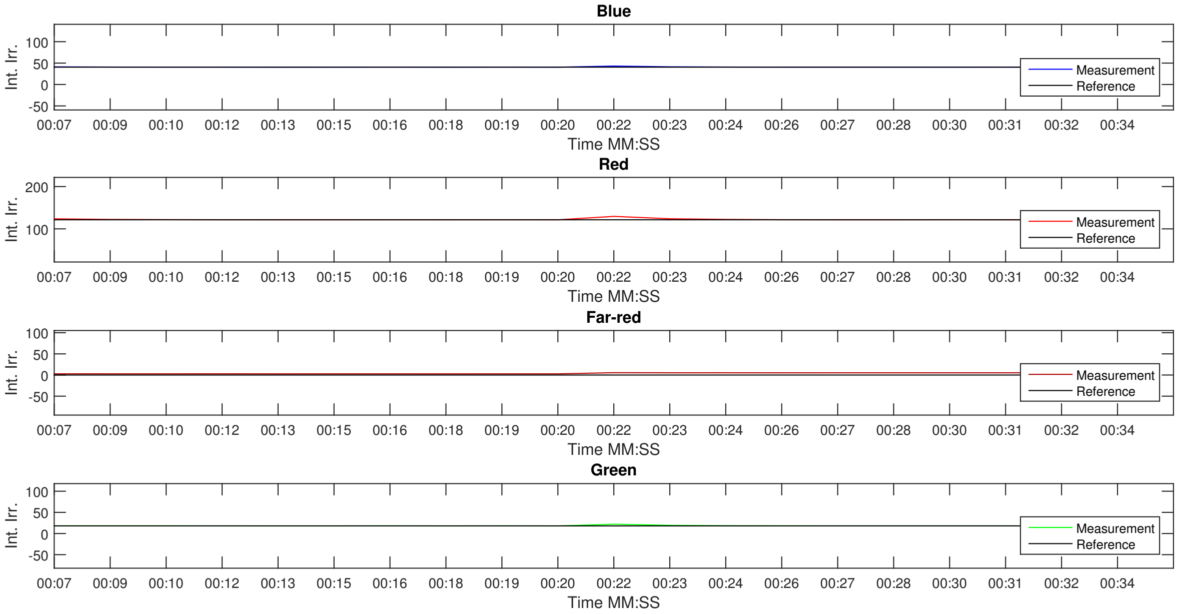


Figure 8.9: 4 LED lamp with plant lab references disturbed by 9 LED lamp producing a daylight cycle.

As seen in the figure the sunlight disturbance has little effect on the systems ability to follow the reference values. This is expected since a step from 0 to 500 intensity as seen in Figure 8.2 in Section 8.1 also had little effect and this time around the disturbance is slowly stepwise incremented instead.

9

Discussion

In this chapter, discussions around the final implementation of the system are presented as well as problems that have arisen during the work. The chapter also contains some recommendations for future work with the system.

9.1 Consequences of PWM/DC alternating input

The alternation of the input between PWM and DC proved to be problematic for the system. This is because a change in input setting requires a change of the K_λ matrix. To be able to ensure stability this increases the complexity of the system. It also becomes a problem when a reference is used that requires an input setting that lies close to the threshold between the two input types. The system is then prone to switch back and forth with oscillations as a consequence. The difference in the gain matrices between PWM and DC are rather large which is why these consequences become rather substantial. The used method of lowering the integral action so that there is little or no overshoot was seen to improve the system. However, situations could arise when this is not enough, such as when more disturbances are added or the references are set to exactly where the switch occurs. A solution to this problem could be to smooth the transition between the two input types by changing the software in the lamps so that the PWM and DC inputs have approximately the same gains. This is something that is outside the scope of this project but could be beneficial for the system. Another solution could be to use a nonlinear model of the lamps in order to smoothen the transition. However, that would make to system vastly more complex.

9.2 Hardware limitations and additional lamps

The lamps produced by Heliospectra have a rather high output but a single lamp is not able to recreate the full effect of natural sunlight. However, if more lamps that cover the same area are added to the system it is possible to achieve a better natural sunlight since the light have an additive effect. The increase of the light approximately follows the inverse square law as stated in Section 4.5. Placing the spectrometer and plants closer to the lamp will therefore also increase the intensity of the light but it will also decrease the spread of the light.

Another limitation of the lamps is the spectrum range produced by the individual LEDs. As seen in the spectrum plots in Section 8.2 the reference spectrum looks very different from the measured one. With more LEDs covering other wavelengths it is possible to create a spectrum that more closely follows that of the sun.

The Matlab version of this system is not designed to run several lamps simultaneously. It could however be changed to use the same implementation as the RPI which uses a master-slave mode to run the lamps. This means that several lamps could mirror the settings of one lamp that runs the control system. That way an entire greenhouse of lamps could be controlled using a single spectrometer as long as the conditions for all the lamps are approximately the same.

9.3 Spectrometer limitations

The reason the STS-VIS is used is because of its small size and compatibility with the RPI. This enables the setup to easily be placed in a greenhouse and configured to run the control system. The STS-VIS is also relatively cheap compared to bigger and more advanced spectrometers. However, with a better spectrometer a more precise measurement could be made. For example the dark spectrum can be accounted for during runs with some spectrometers using an active shutter mechanism. This would make the dark spectrum correction more accurate since the exact integration time and temperature would be used instead of a lookup table with a limited amount of samples. Additionally the speed of the sampling, the integration time, would also be much faster with a better spectrometer. This would make it possible to use a smaller slit size which would also increase the optical resolution. All these improvements would make the system even more stable.

9.4 Further work

Possible features that could be added to the RPI implementation is the PAR controller and an interface for a PC that can plot and change settings remotely. As of now the RPI system can be launched via Matlab and it is then also possible to observe the system using the same plot functions as in the Matlab version of the system.

Both the Matlab- and RPI-system are also rather difficult to use for inexperienced personnel. Recommendations are to make both systems more user friendly before a possible release to customers. For the Matlab system this would first and foremost be to generalize the use of dynamic references. As it is now, the system requires knowledge of Matlab coding and processing of sunlight data in order to use it correctly. To increase the user friendliness of the RPI implementation it is recommended that some kind of user interface is implemented to circumvent the need to connect and observe via a console/terminal.

10

Conclusions

In this thesis a feedback controller for LED-lamps has been presented. The controller uses a PID-structure with only integral action and a decoupling network to be able to handle the MIMO system. Both a Matlab code and a C code, for use in a Raspberry Pi, have been developed. The presented controller has shown good results with high robustness and a fast settling time. It has not yet been tested in greenhouses where real sunlight interferes. However tests with calculated sunlight show that the controller should be able to handle the sunlight, given that the sunlight does not overpower the light from the lamp. Under the right conditions, it is also deemed that the controller is able to successfully increase the spectral quality of natural sunlight present in a greenhouse. The final implementation of the Matlab version enables plant researchers to closely reproduce sunlight in a closed controlled environment using references acquired either by measuring or by calculating. The control system will, in a near future, be used to study the responses of plants when subjected to dynamic light conditions.

Some oscillations are caused by the switch between PWM and DC. This has, however, not made the system severely unstable. Some extreme cases where the reference is exactly on the switch could, however, make the system unstable. This problem could be solved in a couple of different ways (see Chapter 9).

By implementing the controller on the RPI, a system that could be used as a stand alone product is created. This system is, however, not entirely completed due to lack of time. Since the Matlab version of the system was used as a basis for the RPI implementation that version has the most functionality but will require some basic knowledge of Matlab in order to run it successfully.

In conclusion the aim of controlling the lamps as proposed is possible under the right conditions.

Bibliography

[1] B. F. Lucker, C. C. Hall, R. Zegarac, and D. M. Kramer, “The environmental photobioreactor (epbr): An algal culturing platform for simulating dynamic natural environments,” *Algal Research*, vol. 6, Part B, pp. 242–249, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.algal.2013.12.007

[2] S. Muthu, F. J. P. Schuurmans, and M. D. Pashley, “Red, green, and blue leds for white light illumination,” *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 8, no. 2, pp. 333–338, 2002. [Online]. Available: http://dx.doi.org/10.1109/2944.999188

[3] J. C. Guerrero Lozada, “Development of a light emission control system through plant reflectance,” Master Thesis, Chalmers university of technology, Gothenburg, 2012.

[4] D. Bånkestad, “Optimisation of led lighting in greenhouses by remote sensing of plant reflectance and fluorescence,” Master Thesis, Chalmers university of technology, Gothenburg, 2013.

[5] Ocean Optics, “STS-VIS,” [Accessed: 2015-06-05]. [Online]. Available: http://oceanoptics.com/product/sts-vis-microspectrometer/

[6] Ocean Optics, “Cosine corrector,” [Accessed: 2015-06-29]. [Online]. Available: http://oceanoptics.com/product/cosine-correctors/

[7] Ocean Optics, “How can I reduce the noise in my spectra?” [Accessed: 2015-06-05]. [Online]. Available: http://oceanoptics.com/faq/can-reduce-noise-spectra/

[8] Ocean Optics, “Data sheet,” [Accessed: 2015-06-05]. [Online]. Available: http://oceanoptics.com//wp-content/uploads/STS-Data-Sheet.pdf

[9] Ocean Optics, “Vis-nir radiometric calibration sources,” [Accessed: 2015-06-05]. [Online]. Available: http://oceanoptics.com/product/hl-3-cal-hl-3plus-cal-family/

[10] A. S. Boksenbom and R. Hood, *General algebraic method applied to control analysis*

of complex engine types. Washington, D.C.: National Advisory Committee for Aeronautics, 1949.

[11] C. H. Liu, Ed., *General Decoupling Theory of Multivariable Process Control Systems.* Berlin, Heidelberg: Springer-Verlag, 1983, vol. 53. [Online]. Available: http://dx.doi.org/10.1007/BFb0041217

[12] I. Ahmed, “Implementation of PID and Deadbeat Controllers with the TMS320 Family,” Texas Instruments, Tech. Rep., 1997.

[13] B. Lennartson, *Reglerteknikens grunder.* Malmö, Sweden: Studentlitteratur AB, 2002.

[14] Agricultural Research Service (ARS), “Solarcalq – version 1.0,” [Accessed: 2015-06-05]. [Online]. Available: http://www.ars.usda.gov/services/software/download.htm?softwareid=157

[15] Red Hat, “Pidora - fedora remix,” [Accessed: 2015-06-05]. [Online]. Available: http://pidora.ca/

[16] Lucasbosch, “Top face of the raspberry pi model b+,” [Accessed: 2015-06-29]. [Online]. Available: https://commons.wikimedia.org/wiki/File:Raspberry_Pi_B%2B_top.jpg

A

Appendix 1

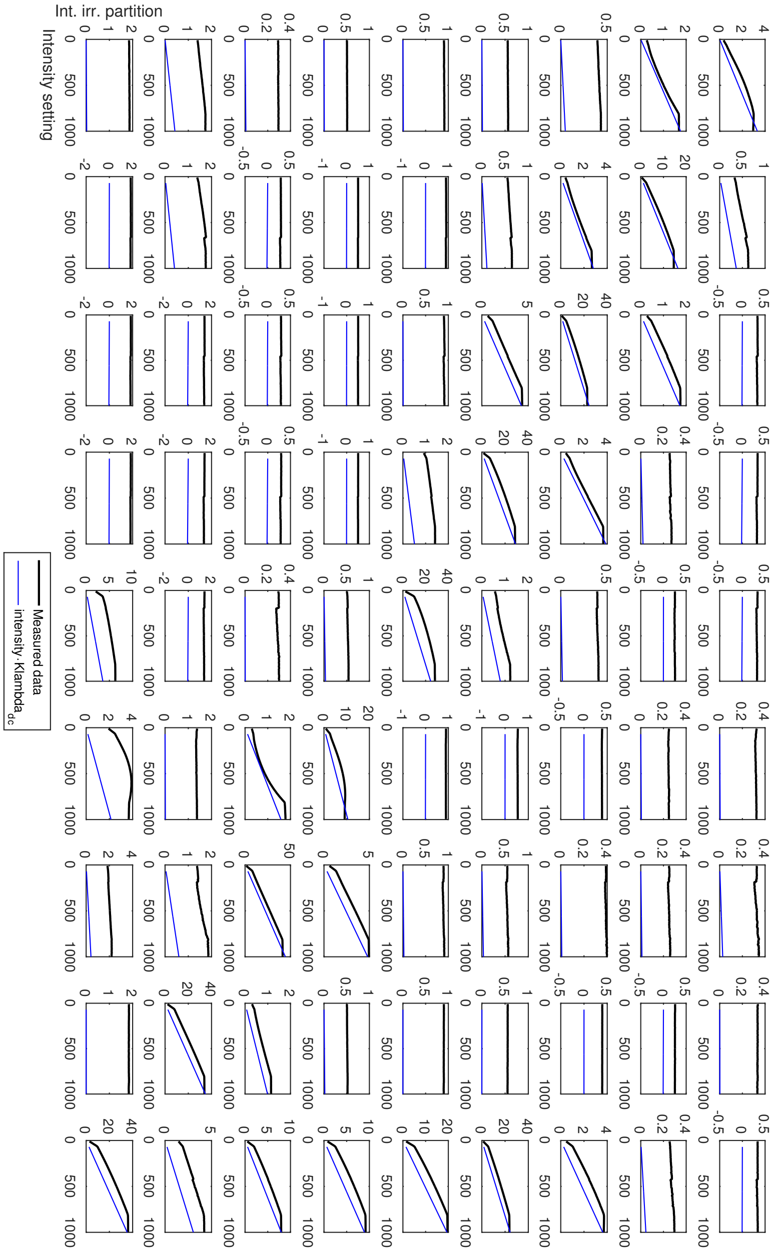


Figure A.1: The input output relations of a lamp with 9 LEDs: uv, violet, indigo, blue, green, orange, red, far-red and yellow. Plot only shows the DC version of the K_λ-matrix due to size issues.