



A Sensor Fusion Algorithm for Filtering Pyrometer Measurement Noise in the Czochralski Crystallization Process

M. Komperød¹ J. A. Bones² B. Lie³

¹Faculty of Engineering, Østfold University College, N-1757 Halden, Norway. E-mail: magnus.komperod@hiiof.no

²SINTEF Materials and Chemistry, Department of Metallurgy, N-7465 Trondheim, Norway. E-mail: johnatle.bones@sintef.no

³Faculty of Technology, Telemark University College, N-3901 Porsgrunn, Norway. E-mail: bernt.lie@hit.no

Abstract

The Czochralski (CZ) crystallization process is used to produce monocrystalline silicon for solar cell wafers and electronics. Tight temperature control of the molten silicon is most important for achieving high crystal quality. SINTEF Materials and Chemistry operates a CZ process. During one CZ batch, two pyrometers were used for temperature measurement. The silicon pyrometer measures the temperature of the molten silicon. This pyrometer is assumed to be accurate, but has much high-frequency measurement noise. The graphite pyrometer measures the temperature of a graphite material. This pyrometer has little measurement noise. There is quite a good correlation between the two pyrometer measurements. This paper presents a sensor fusion algorithm that merges the two pyrometer signals for producing a temperature estimate with little measurement noise, while having significantly less phase lag than traditional lowpass-filtering of the silicon pyrometer. The algorithm consists of two sub-algorithms: (i) A dynamic model is used to estimate the silicon temperature based on the graphite pyrometer, and (ii) a lowpass filter and a highpass filter designed as complementary filters. The complementary filters are used to lowpass-filter the silicon pyrometer, highpass-filter the dynamic model output, and merge these filtered signals. Hence, the lowpass filter attenuates noise from the silicon pyrometer, while the graphite pyrometer and the dynamic model estimate those frequency components of the silicon temperature that are lost when lowpass-filtering the silicon pyrometer. The algorithm works well within a limited temperature range. To handle a larger temperature range, more research must be done to understand the process' nonlinear dynamics, and build this into the dynamic model.

Keywords: Complementary filters; Czochralski crystallization process; Measurement noise filtering; Pyrometer temperature measurement; Sensor fusion algorithm.

1 Introduction

The Czochralski (CZ) crystallization process is used to convert multicrystalline materials into monocrystalline materials, i.e. materials that have homogeneous crystal structures. Among the most important applications is

production of monocrystalline silicon. Monocrystalline silicon is used in solar cell wafers and in computers and electronics.

The CZ process is a batch process. During the process, multicrystalline silicon is melted in a crucible. The crucible is heated using a heating element, which

power is manipulated using a triode for alternating current (TRIAC). Tight control of the silicon temperature is most important for achieving high crystal quality. A reliable temperature measurement without too much measurement noise is decisive for achieving good temperature control.

This paper considers the temperature measurement of a real-life CZ process owned and operated by SINTEF Materials and Chemistry in Trondheim, Norway (hereafter referred to as SINTEF). During an experiment two pyrometers were used for temperature measurement. One pyrometer, referred to as the silicon pyrometer, measures the temperature in the molten silicon. The molten silicon temperature is the desired temperature signal for monitoring and control. Unfortunately, the signal from this pyrometer has considerable high-frequency measurement noise. Attenuating the noise using a traditional lowpass filter is an intuitive and feasible solution. However, this solution will cause significant phase lag over the filter, which is unfortunate for the temperature control.

The other pyrometer, referred to as the graphite pyrometer, has little noise. However, this pyrometer has the disadvantage of measuring the temperature of a graphite material encircling the silicon crucible. Hence, the signal of the pyrometer does not represent the actual silicon temperature. Fortunately, there is quite a high correlation between the silicon temperature and the graphite pyrometer signal.

The contribution of this paper is a sensor fusion algorithm that fuses the two pyrometer measurements. The purpose of the algorithm is to estimate the temperature of the molten silicon. For a given cut-off frequency, the algorithm estimate gives the same amount of measurement noise as a traditional lowpass filter, but has significantly less phase lag. The lower the cut-off frequency is chosen, the larger improvement of using the sensor fusion algorithm. On the other hand, if the cut-off frequency is chosen high, there is little phase lag over a traditional lowpass filter, and the algorithm does not give any significant improvement.

The sensor fusion algorithm presented in this paper is implemented as complementary filters. The filters are chosen as a lowpass Butterworth filter and a high-pass Butterworth filter. A statistically optimal temperature estimate can in theory be computed using a Kalman filter or a Wiener filter. However, using Kalman filter or Wiener filter depends on noise descriptions that are not known. Using complementary Butterworth filters, the sensor fusion algorithm has one tuning parameter; the Butterworth filters' cut-off frequency.

The authors have searched for scientific papers issuing pyrometer measurement noise in the CZ processes.

Unfortunately, no relevant results were found. There are several possible reasons for this negative search result: (i) Even though the silicon pyrometer at SINTEF has much measurement noise, this may not be an issue of other pyrometers at other CZ plants. (ii) The noise is attenuated using traditional lowpass filters despite the unfortunate phase lag. (iii) The noise problem is reduced by using controller tunings that attenuate high-frequency noise, for example avoiding derivative action and high gain in PID controllers. (iv) The noise problem is handled in the commercial CZ industry and is not published in scientific papers.

An introduction to complementary filters is given in [Brown and Hwang \(1997\)](#). [Lyons \(2011\)](#) gives a general introduction to digital signal processing. The sensor fusion algorithm also includes an empirical model developed using system identification. A comprehensive introduction to system identification is given in [Ljung \(1999\)](#).

[Lan \(2004\)](#) gives an introduction to crystal growth, including the CZ process, and provides an extensive number of references for further reading. There are many scientific papers covering modeling and control of the CZ process, for example [Irizarry-Rivera and Seider \(1997a,b\)](#) and [Lee et al. \(2005\)](#). However, the authors' literature search indicates that the important topic of sensor technology in the CZ process has received very limited attention.

2 Notation and Definitions

Table 1 presents the notation used in this paper. A variable with subscript k refers to the variable's value at timestep k . For example T_k refers to the silicon temperature at timestep k .

Please note the difference between \hat{T}^g and u : u is the raw signal from the graphite pyrometer in Volt, while \hat{T}^g is an estimate of the *silicon* temperature based on the signal u . These variables are related through the equation $\hat{T}^g = G(z) u$.

To simplify notation, the arguments s and z will be used to specify whether a transfer function H is time continuous or time discrete, respectively. That is, $H(s)$ and $H(z)$ describe the same model or filter, where $H(s)$ is the continuous version and $H(z)$ is the discrete version. This notation is erroneous in a strict mathematical sense, as $H(s)$ and $H(z)$ are different mathematical functions. However, it simplifies the notation without any risk of confusion.

3 The Czochralski Crystallization Process

The Czochralski (CZ) crystallization process is used to convert multicrystalline materials into monocrystalline materials. The plant considered in this paper converts multicrystalline silicon into monocrystalline silicon. Monocrystalline silicon is used in solar cell wafers and in computers and electronics. Solar cells based on monocrystalline silicon have higher efficiency than solar cells based on multicrystalline silicon.

The CZ process is a batch process, which main steps are illustrated in Figure 1. (i) Initially multicrystalline silicon is melted in a crucible. (ii) When the silicon is molten, the tip of a seed crystal is dipped into the melt. The seed crystal is monocrystalline and has the crystal structure that is to be produced. (iii) When the tip of the seed crystal begins to melt, the crystal is slowly elevated. As the crystal is lifted, the molten silicon solidifies on the crystal. During solidification, the crys-

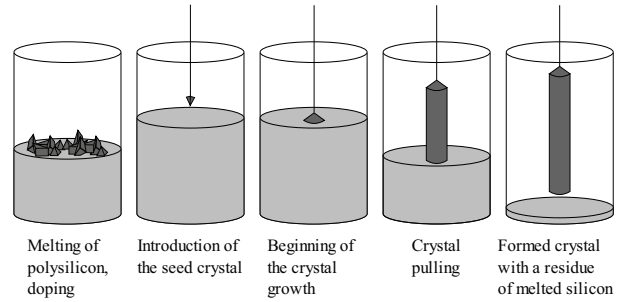


Figure 1: The main batch steps of the CZ process. Illustration from Wikipedia (the illustration is released to public domain by the copyright holder).

tal structure of the seed crystal is extended onto the solidifying silicon. (iv) The crystal grows radially and axially. The produced crystal is referred to as an ingot. The temperature of the molten silicon and the vertical pulling speed are used to control the ingot diameter. Stable growing conditions are essential to produce high crystal quality. (v) As the final ingot length is reached, the crystal growth is terminated by slowly decreasing the crystal diameter to zero. During the entire batch process, the crucible is rotated in one direction, and the seed crystal is rotated in the opposite direction.

SINTEF Materials and Chemistry in Trondheim, Norway, owns and operates a CZ process. Initially there was one pyrometer at this plant. This pyrometer measures the temperature at a graphite material in the CZ process. This pyrometer will be referred to as the graphite pyrometer. The pyrometer has the advantage of little measurement noise, but it does not measure the temperature directly in the molten silicon. Hence, using this pyrometer for temperature control, the temperature of the graphite material is actually controlled, not the temperature of the silicon. This choice of sensor location is then based on the assumption that stable graphite temperature implies stable silicon temperature.

A second pyrometer was installed at the plant. This pyrometer is able to measure the temperature directly in the molten silicon, which is the desired temperature to control. This pyrometer will be referred to as the silicon pyrometer. The authors have logged process data from only one experiment where this pyrometer was tested. The temperature measured by this pyrometer seems reasonable based on the melt temperature of silicon. Also the measured temperature response seems reasonable based on step changes in the heating element power. Unfortunately, the temperature signal of this pyrometer is very noisy. An intuitive and feasible solution is to use a traditional lowpass filter to attenu-

Table 1: Notation used in this paper.

$G(z)$	The dynamic model from the graphite pyrometer to the silicon temperature.
$H(z)$	A lowpass Butterworth filter.
k	An index referring to sample number in the dataset.
T	The true, but unknown, temperature of the molten silicon [°C].
\hat{T}	The temperature of the molten silicon [°C] estimated by the sensor fusion algorithm.
\hat{T}^g	The temperature of the molten silicon [°C] estimated based on the signal from the graphite pyrometer, u .
\hat{T}^s	The temperature of the molten silicon [°C] measured by the silicon pyrometer.
t_s	The sample time [s].
u	The raw signal from the graphite pyrometer [V].
\bar{u}	\bar{u} is u rescaled by a gain and a bias. \bar{u} is only used for comparing u and \hat{T}^s in plots. \bar{u} is not used in the sensor fusion algorithm.
z	The time-shift operator defined by $x_{k+1} = zx_k$ and $x_{k-1} = z^{-1}x_k$.
ω	Frequency [rad/s].
ω_c	The cut-off frequency of Butterworth filters [rad/s].

ate the noise. However, lowpass-filtering will give phase lag over the filter which is unfortunate for temperature control. The more the signal is smoothed in the filter, the more phase lag.

4 Sensor Fusion and Complementary Filters

Online measurements of process variables are decisive for most control systems to operate properly. There are often several sensor technologies available for measuring a specific variable. If no conventional sensor is available, it may be possible to develop a soft sensor. A soft sensor does not measure the desired process variable directly, but relies on other measurements and an algorithm to estimate the desired process variable. In the following text, the terms “sensor” and “sensor technology” will be used for both conventional sensors and soft sensors.

For measuring a specific process variable, different sensor technologies may have different qualities, such as accuracy and amount of measurement noise. A sensor fusion algorithm is an algorithm that combines several sensors for estimating a process variable. The algorithm’s purpose is to achieve an estimate with better qualities than any of the individual sensors provides. The term “sensor fusion” is a general term for using multiple sensors to estimate a variable. A number of algorithms can be used to fuse the sensor signals, including Kalman filter and Bayesian networks.

A special case of the sensor fusion approach is when different sensors have desirable qualities in different frequency ranges. The typical case is when some sensors are accurate at low frequencies, while other sensors are accurate at high frequencies. A commonly used example is estimation of position based on a position sensor and a velocity sensor. The position sensor may not be sufficiently accurate to keep up with smaller position variations. On the contrary, time-integrating the velocity sensor may keep up with smaller position changes, but this estimate is likely to drift over time due to accumulation of small measurement errors. The intuitive solution is to consider the velocity sensor over short time spans, and consider the position sensor over longer time spans. In the frequency domain this translates into lowpass-filtering the position sensor and highpass-filtering the time-integrated velocity (Brown and Hwang, 1997).

Complementary filters is a simple and intuitive approach for fusing sensors which qualities can be discriminated by frequency. Assume that a process variable y is measured by two sensors. The sensor outputs are $\hat{y}_1 = y + v_1$ and $\hat{y}_2 = y + v_2$, respectively, where v_1

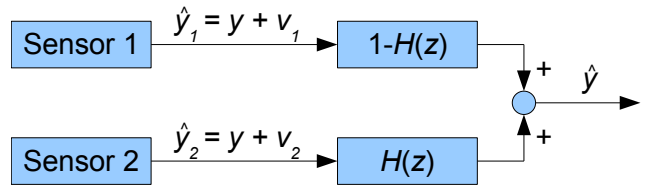


Figure 2: Complementary filters with two inputs. The figure is inspired by Brown and Hwang (1997, Figure 4.9).

and v_2 are measurement noise / measurement errors. Assume that v_1 is low-frequency noise and v_2 is high-frequency noise. Hence, highpass-filtering \hat{y}_1 removes v_1 , but also removes the low-frequency components of the signal y . Similarly, lowpass-filtering \hat{y}_2 removes v_2 , but also removes the high-frequency components of y . Summing the highpass-filtered \hat{y}_1 and the lowpass-filtered \hat{y}_2 will include both the high-frequency and the low-frequency components of y . Assuming the applied lowpass filter is $H(z)$, the highpass filter is chosen as the complementary filter $1 - H(z)$ (Brown and Hwang, 1997). Complementary filters are illustrated in Figure 2.

The filter output \hat{y} can be written as

$$\begin{aligned} \hat{y} &= [1 - H(z)]\hat{y}_1 + H(z)\hat{y}_2 \\ &= [1 - H(z)](y + v_1) + H(z)(y + v_2) \\ &= y + [1 - H(z)]v_1 + H(z)v_2. \end{aligned} \quad (1)$$

Ideally, the lowpass filter $H(z)$ removes v_2 and the highpass filter $1 - H(z)$ removes v_1 . Then the estimated output will be identical to the actual process variable, i.e. $\hat{y} = y$ (Brown and Hwang, 1997).

The choice of the lowpass filter $H(z)$ is important to achieve a good estimate \hat{y} . If the noise characteristics of v_1 and v_2 are known, the statistically optimal $H(z)$ can be computed using a Wiener filter or a Kalman filter (Brown and Hwang, 1997). However, in many real-life applications the noise characteristics are not known.

Although the term complementary filters is not used, Ljung (1999, Section 3.3) uses complementary filters when discussing observers and predictors. Actually, this reference was the most inspiring for the authors when developing the sensor fusion algorithm.

5 Basic Principle

This paper presents a sensor fusion algorithm that aims at attenuating measurement noise of the silicon pyrometer, while giving less phase lag than a traditional lowpass filter. Hence, the output of the algorithm, \hat{T} , is

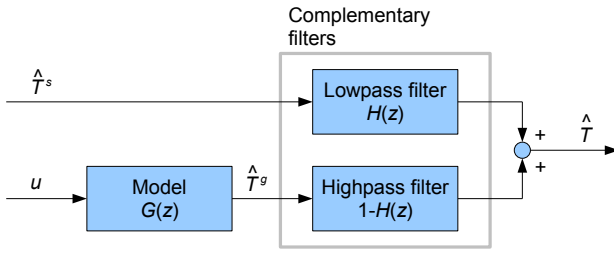


Figure 3: Basic principle of the sensor fusion algorithm.

an estimate of the true, but unknown, silicon temperature, T . This section gives a brief presentation of the algorithm, while the following sections provide a thorough explanation.

The silicon pyrometer measures the temperature of the molten silicon in the CZ process. The pyrometer output signal, \hat{T}^s , seems reasonable based on process knowledge and step-response tests. Unfortunately, the signal is very noisy. The graphite pyrometer measures the temperature of a graphite material. The signal from this pyrometer, u , has little noise. There is quite a good correlation between the signals of the two pyrometers.

The sensor fusion algorithm consists of two sub-algorithms: (i) A dynamic model, $G(z)$, and (ii) complementary filters, $H(z)$ and $1 - H(z)$.

The dynamic model, $G(z)$, estimates the silicon temperature, T , as a function of the graphite pyrometer, u . This estimate is noted \hat{T}^g . Please note that the estimate does not depend on the silicon pyrometer.

The model estimate, \hat{T}^g , is less accurate than the silicon pyrometer, but has the significant advantage of little noise. Due to the measurement noise of the silicon pyrometer, the model estimate is more reliable than the silicon pyrometer, \hat{T}^s , at high frequencies. However, the silicon pyrometer is assumed to be more reliable at low frequencies than the model estimate. The complementary filters take advantage of these qualities by lowpass-filtering the silicon pyrometer through $H(z)$, highpass-filtering the model output through $1 - H(z)$, and sum these two filtered signals. The sensor fusion algorithm is illustrated in Figure 3.

6 Presentation of the Raw Data

The authors have access to data from only one CZ batch where both the silicon pyrometer and the graphite pyrometer were used. Analyses of the data and initial tests of the sensor fusion algorithm conclude that the dynamics from the graphite pyrometer, u , to the silicon temperature, T , (measured by the silicon pyrometer, \hat{T}^s) is nonlinear. The nonlinearity will be demon-

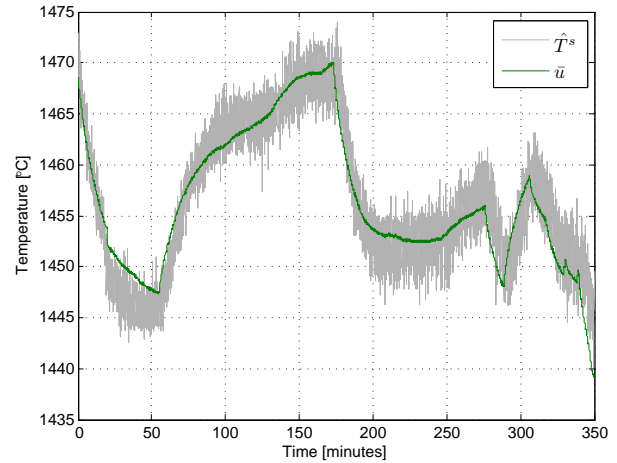


Figure 4: The silicon pyrometer, \hat{T}^s , and the rescaled raw signal of the graphite pyrometer, \bar{u} , plotted over the data section used to develop the sensor fusion algorithm.

strated in Section 10. The present section covers the data used for developing the sensor fusion algorithm.

When deciding which part of the dataset to use for developing the sensor fusion algorithm, the authors were looking for a continuous section where there are significant excitations in the temperature, while the temperature range is not wide enough for nonlinearities to be significant. There is only one section in the dataset that meets these demands. This section covers 5 hours and 50 minutes of the CZ batch, having sampling interval of 2 seconds. That is 10501 samples. The grey curve of Figure 4 shows the temperature measured by the silicon pyrometer, \hat{T}^s , over the chosen data section. The temperature is in the range of approximately 1445°C to 1470°C. For comparison, during the entire CZ batch (melting of the silicon not included) the temperature varies in the range of approximately 1425°C to 1480°C. Hence, the range used to develop the sensor fusion algorithm is approximately 45% of the total temperature range during the CZ batch. Figure 4 shows that the silicon pyrometer gives a very noisy measurement signal. The noise peak-to-peak amplitude is 5 to 10°C.

The green curve of Figure 4 illustrates the raw signal, u , of the graphite pyrometer over the chosen data section. For the purpose of comparing this signal with the silicon pyrometer, \hat{T}^s , the signal has been rescaled to the same range as the silicon pyrometer. The rescaled signal, \bar{u} , shown in the figure is on the form $\bar{u} = p_1 u + p_0$, where p_1 and p_0 are polynomial coefficients. The coefficients are computed as the least squares fit of \bar{u}

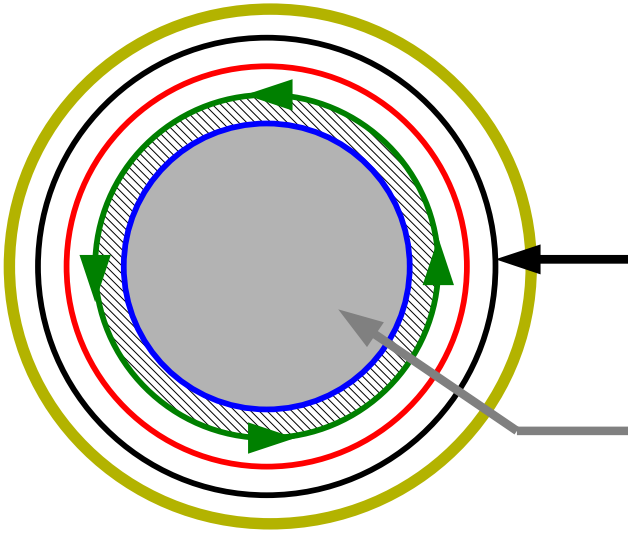


Figure 5: Location of the silicon pyrometer and the graphite pyrometer in the SINTEF CZ process (seen from above). Please refer to the main text for explanation.

to \hat{T}^s . Please note that \bar{u} is the *static* best fit of u to \hat{T}^s . The *dynamic* relationship between u and \hat{T}^s will be developed in Section 8 using system identification. Figure 4 clearly shows that the graphite pyrometer has very little measurement noise compared to the silicon pyrometer. The figure also shows that there is a good correlation between the silicon pyrometer and the graphite pyrometer.

Figure 5 illustrates the location of the silicon pyrometer and the graphite pyrometer in the CZ process (seen from above). The black arrow indicates the measurement location of the graphite pyrometer, and the grey arrow indicates the measurement location of the silicon pyrometer. The grey area in center represents the molten silicon, and the blue circle represents the crucible. The crucible is located in a rotating device (green color) shaped as a cylinder with bottom, and without top. The red color represents the heating element, which power is controlled by a triode for alternating current (TRIAC). The black color represents the graphite ring at which the graphite pyrometer measures temperature. The yellow color represents insulation and the outer wall. The colors in Figure 5 are chosen to simplify the explanation of the figure. The colors do not represent the colors in the process.

7 Deciding the Complementary Filters $H(z)$ and $1 - H(z)$

Figure 3 illustrates the complementary filters to be used in the sensor fusion algorithm. For two complementary filters, there is only one filter characteristic to choose. For the sensor fusion algorithm, the transfer function of the lowpass filter $H(z)$ is to be chosen. The highpass filter is then given by $1 - H(z)$.

The sensor fusion algorithm provides an estimate, \hat{T} , of the silicon temperature. The accuracy of this estimate depends on $H(z)$. $H(z)$ can be computed using a Wiener filter or a Kalman filter (Brown and Hwang, 1997). These approaches give a temperature estimate, \hat{T} , that is optimal in the sense of minimizing the estimation error variance, i.e. $E(T - \hat{T})^2$. However, the Kalman filter depends on covariance matrices that represent the measurement noise of the silicon pyrometer, i.e. $T - \hat{T}^s$, and the estimation error of the model output, i.e. $T - \hat{T}^g$. The Wiener filter depends on the same information presented in other terms. Unfortunately, this information is not known.

A simpler approach is used in this paper. The filter $H(z)$ is chosen as a Butterworth filter. There are now two parameters to be specified: (i) The cut-off frequency, ω_c , and (ii) the filter order. The cut-off frequency will depend on the tolerance for measurement noise. This tolerance depends on the usage of the temperature estimate, \hat{T} . For example, if the estimate is to be used for temperature control, the tolerance for high-frequency noise will depend on the controller tuning. A controller with significant derivative action will have less tolerance than a controller using mainly integral action.

As this paper considers only the sensor fusion algorithm, not its usage, the cut-off frequency, ω_c , will be chosen based on the frequency content of the silicon pyrometer, \hat{T}^s . The cut-off frequency is chosen as the lowest frequency component that in the time domain can not be distinguished from high-frequency measurement noise. That is, the logged data of the silicon pyrometer, \hat{T}^s , are transformed to the frequency domain using the discrete Fourier transform. Then the lowest frequency component is removed, and the remaining components are transformed back to the time domain. If there is any visible signal pattern beyond measurement noise in the time domain, the second lowest frequency component is removed. Then the remaining frequency components are transformed to the time domain. This is repeated until there is no visual pattern in the time domain beyond measurement noise. For the data sequence \hat{T}^s , approximately the 15 lowest frequency components can be distinguished from the high-frequency noise in the time domain.

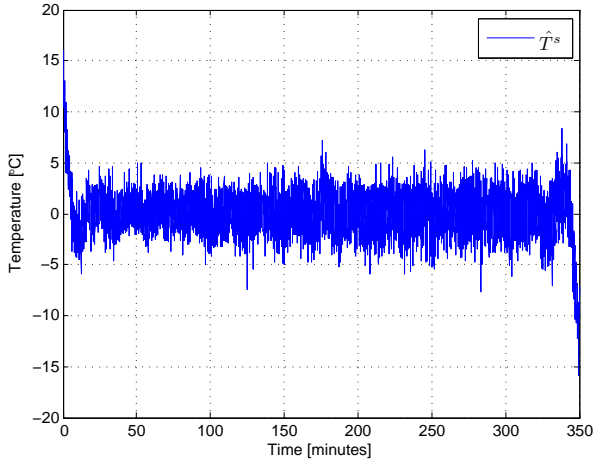


Figure 6: The silicon pyrometer, \hat{T}^s , shown in the time domain, after removing the 15 lowest frequency components.

Figure 6 shows the silicon pyrometer, \hat{T}^s , in the time domain, after removing the 15 lowest frequency components. Hence, the cut-off frequency, ω_c , will be chosen equal to the lowest frequency component presented in Figure 6. In other words: The cut-off frequency will be chosen to attenuate the data shown in Figure 6. The lowest frequency component that is represented in the figure is the 16th component, which corresponds to 4.8×10^{-3} rad/s. Hence, the cut-off frequency is chosen as $\omega_c = 4.8 \times 10^{-3}$ rad/s.

It is emphasized that this approach is only a suggestion for how to choose the cut-off frequency. When using the sensor fusion algorithm in a real-life application, such as temperature control, the application's tolerance to measurement noise will decide the cut-off frequency. Also, the number of frequency components, in this case 15, has been chosen based on human inspection of data plots, and is therefore not precise.

The next issue is to choose the Butterworth filter order. The higher filter order, the sharper cut-off. Considering the sensor fusion algorithm of this paper, the silicon pyrometer, \hat{T}^s , has desirable qualities at low frequencies, and the model estimate, \hat{T}^g , has desirable qualities at high frequencies. However, it seems unlikely that there are sharp frequency limits between the desirable and the undesirable qualities. Gradual changes between the qualities seem more likely. Hence, it seems reasonable to choose a low filter order to get a gradual transition from \hat{T}^s to \hat{T}^g for increasing frequencies. The filter order is therefore chosen to be one.

A continuous time lowpass Butterworth filter with $\omega_c = 4.8 \times 10^{-3}$ rad/s has the transfer function (Hau-

gen, 2004)

$$H(s) = \frac{1}{1 + \frac{s}{\omega_c}}. \quad (2)$$

The dynamic model, $G(z)$, to be developed in Section 8 will be a discrete time model. It is therefore desirable to also have the lowpass filter in discrete time. There are several ways to convert a continuous time transfer function to discrete time. A method referred to as the bilinear transform method will be used here. This method is described in Lyons (2011). The method is to replace s in a continuous transfer function, $H(s)$, with

$$s = \frac{2}{t_s} \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right) \quad (3)$$

to obtain the time discrete transfer function $H(z)$. Here, t_s is the sampling time, which is 2 seconds. Using this method, the discrete time lowpass filter becomes

$$H(z) = \frac{\omega_c t_s + \omega_c t_s z^{-1}}{\omega_c t_s + 2 + (\omega_c t_s - 2)z^{-1}}. \quad (4)$$

The complementary continuous highpass filter is

$$1 - H(s) = \frac{\frac{s}{\omega_c}}{1 + \frac{s}{\omega_c}}. \quad (5)$$

This is a highpass Butterworth filter (Haugen, 2004). In discrete time this becomes

$$1 - H(z) = \frac{2 - 2z^{-1}}{\omega_c t_s + 2 + (\omega_c t_s - 2)z^{-1}}. \quad (6)$$

Figure 7 shows the Bode diagram of the complementary filters $H(z)$ and $1 - H(z)$.

8 Identifying the Dynamic Model

$G(z)$

The sensor fusion algorithm presented in this paper aims at attenuating the measurement noise of the silicon pyrometer, while giving significant less phase lag over the filter than a traditional lowpass filter. Figure 3 illustrates the sensor fusion algorithm. The upper input of the summation point represents a traditional lowpass-filtering of the silicon pyrometer, \hat{T}^s . This signal is believed to give an accurate representation of the silicon temperature, T , but the high-frequency components are removed and the signal is phase lagged in the filter. Hence, the purpose of the lower input of the summation point in Figure 3 is to estimate the high-frequency components of the silicon temperature, T , and give this estimate a positive phase by highpass-filtering it. Therefore, the success of the sensor fusion

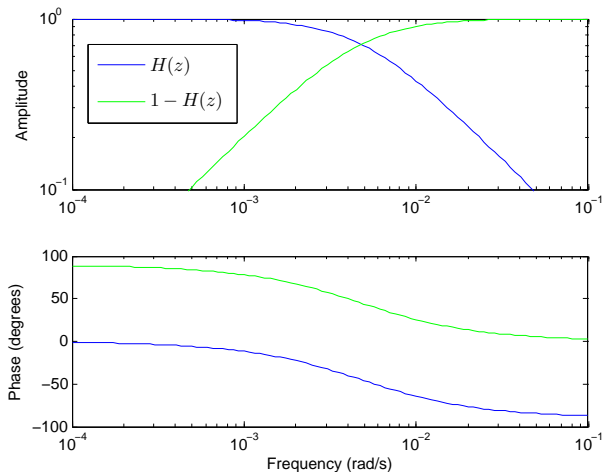


Figure 7: Bode diagram of the complementary filters; the lowpass filter, $H(z)$, and the highpass filter, $1 - H(z)$.

algorithm depends on how accurate the model output, \hat{T}^g , describes the silicon temperature, with emphasis to the higher frequency components. This section discusses how the model, $G(z)$, is developed to meet this demand.

The model, $G(z)$, will be developed using system identification. The model input is the output signal of the graphite pyrometer, u . The desired model output is the silicon temperature, T . As the exact temperature is not known, the best estimate available is the temperature measured by the silicon pyrometer, \hat{T}^s . Unfortunately, the noise at \hat{T}^s is likely to reduce the quality of the model. However, no better options are available.

In the sensor fusion algorithm, the model's output, \hat{T}^g , will be highpass-filtered through the filter $1 - H(z)$ as shown in Figure 3. Hence, when developing the model, $G(z)$, the low-frequency components should be deemphasized in favor of the high-frequency ones. This is achieved by detrending (subtracting sample mean) and highpass-filtering the raw data using the same highpass filter that is used in the complementary filters, i.e. $1 - H(z)$. Figure 8 shows the silicon pyrometer, \hat{T}^s , and the rescaled version of the graphite pyrometer signal, \bar{u} , after being detrended and filtered through the highpass filter $1 - H(z)$. In other words: The data shown in Figure 8 are the data shown in Figure 4 after being detrended and highpass-filtered. When identifying the model, $G(z)$, the input data used is u and the output data used is \hat{T}^s , both after being detrended and highpass-filtered. That is, the data used for identifying the model are the data shown in Figure 8, except that u is used instead of its rescaled version \bar{u} .

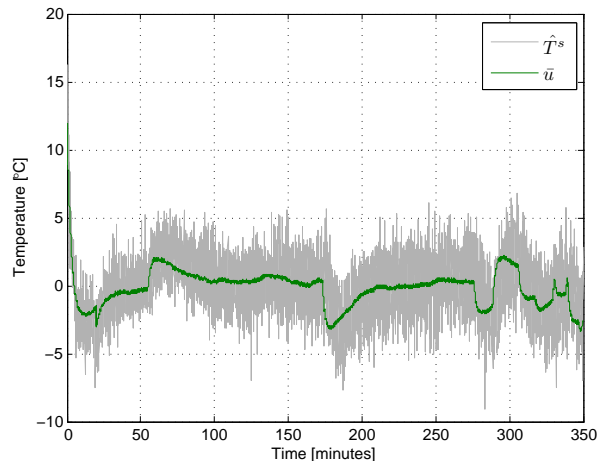


Figure 8: The silicon pyrometer, \hat{T}^s , and the rescaled graphite pyrometer, \bar{u} , after being detrended and highpass-filtered through $1 - H(z)$.

The next issue is to decide a model structure for the model $G(z)$. The dynamics from the graphite pyrometer, u , to the silicon temperature, T , (measured by the silicon pyrometer, \hat{T}^s) seems to be nonlinear. This nonlinearity is demonstrated in Section 10. Flexible, nonlinear black-box model structures usually contain significantly more parameters than linear model structures. As the authors do not have independent data for validating the model, the number of parameters should be limited to avoid overfitting of the model. Therefore, a simple, linear model structure was chosen for the sensor fusion algorithm presented in this paper. Section 11 discusses further work, including this nonlinearity issue.

As pointed out in Ljung (1999), including a noise model in the model structure is equivalent to prefilter the measured data. Hence, a noise model may counteract the data prefiltering done above. Therefore, an output error (OE) model structure will be used. OE models have no noise model, i.e. the noise model is simply 1. The MATLAB System Identification Toolbox includes two linear OE model structures: (i) The process model structure and (ii) the polynomial OE model structure.

The process model is a time continuous transfer function. The human model builder can specify the number of poles (one to three), whether the transfer function should have a zero, and whether the transfer function should have a time delay. Using the maximum number of poles, and a zero and a time delay, the model structure is on the form

$$G(s) = K \frac{1 + t_z s}{(1 + t_1 s)(1 + t_2 s)(1 + t_3 s)} e^{-t_d s}, \quad (7)$$

where the parameters to be identified are K , t_1 , t_2 , t_3 , t_z , and t_d . As pointed out in [Ljung \(2009\)](#), the main advantage of this model structure is that the time delay is estimated. For most other model structures the time delay must be specified by the human model builder. On the other hand, a disadvantage of the process model structure is that the human model builder must specify whether or not the system is underdamped (has complex conjugate poles). If the system is underdamped, the model structure is on the form

$$G(s) = K \frac{1 + t_z s}{(1 + 2\zeta t_\omega s + (t_\omega s)^2)(1 + t_3 s)} e^{-t_d s}. \quad (8)$$

Please refer to [Ljung \(2009\)](#) for further explanation of the process model structure.

The polynomial OE model structure is a model structure in the same family as the more well-known ARX and ARMAX model structures. For single input, single output (SISO) systems, the only difference between these three model structures is the noise models. The OE model has no noise model, i.e. the noise model is simply 1. The SISO polynomial OE model structure is on the form

$$G(z) = \frac{B(z)}{F(z)} z^{-n_d}, \quad (9)$$

where

$$B(z) = \sum_{i=0}^{n_b} b_i z^{-i}, \quad (10)$$

$$F(z) = 1 + \sum_{i=1}^{n_f} f_i z^{-i}. \quad (11)$$

Here, b_i and f_i are polynomial coefficients, and n_d is the time delay in number of samples.

The parameters of the model structures (7) (or (8)) and (9) are identified using the prediction error method (PEM). For an introduction to PEM, please refer to [Ljung \(1999\)](#). Using PEM identification, the model parameters are identified as the solution of a multi-variable optimization problem. In most cases the optimization problem is nonlinear, and the optimization algorithm is in danger of being trapped in a local minimum. Quoting [Ljung \(1999\)](#): "For output error structures, on the other hand, convergence to false local minima is not uncommon."

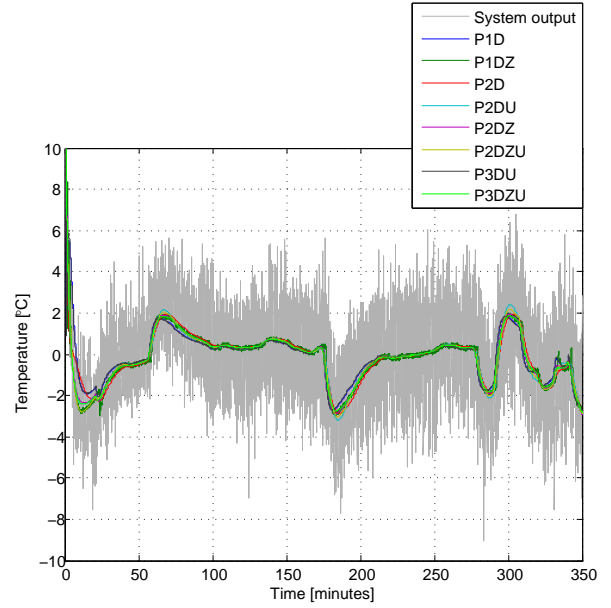


Figure 9: The eight successful process models. 'P1', 'P2', and 'P3' refer to the number of poles (one, two, or three). 'D' means that time delay is estimated. 'Z' means that a zero is estimated. 'U' means that two poles are underdamped (complex conjugate).

The identification work was started by identifying process models ((7) and (8)) using the MATLAB System Identification Toolbox. Ten models with different number of poles, real or complex poles, and with or without zero were identified. The models were then plotted against the system output, i.e. the highpass-filtered \hat{T}^s . Eight of the ten models give good fit to the system output. It is assumed that the failure of the other two models is because the PEM algorithm was trapped in a local minimum. The argument for this conclusion is that simpler model structures, which are subsets of the faulty model structures, did succeed. The eight successful models are shown in Figure 9. There are some differences in the initial values, i.e. the first 20-30 minutes. There are also some smaller differences in the range 325 to 345 minutes (this may be difficult to see in the figure). However, Figure 9 shows that the models are very similar in explaining the system output, and it is very difficult to conclude which model is the better one.

The main reason for beginning the identification work by identifying process models is the process model structure's ability to estimate time delay, i.e. t_d in (7) and (8). Among the eight successful models, t_d ranges from

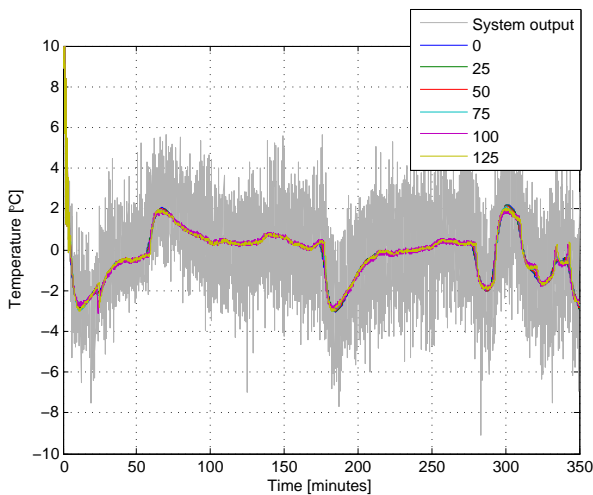


Figure 10: Six polynomial OE models with $n_b = 1$ and $n_f = 2$. The time delay, n_d , ranges from 0 to 125 samples in increments of 25 samples.

0 to 210 seconds. Hence, it must be concluded that using process models for estimating time delay was not very successful for this dataset. However, this is not an unfortunate conclusion: Models with a wide range of time delays give good fit to the system output. Hence, the models seem robust to the choice of time delay. In other words: The poles and the zero seem to be able to compensate a somewhat erroneous time delay. The robustness to the choice of time delay was confirmed by polynomial OE models: Six models with $n_b = 1$ and $n_f = 2$ were identified. The models have different time delays, n_d , ranging from 0 to 125 samples in increments of 25 samples (i.e. from 0 to 250 seconds in increments of 50 seconds). Otherwise these six models were identified under the same conditions. Figure 10 shows the model fit of these six models. Figure 10 confirms the observation of Figure 9: A wide range of time delays give good model fit. Hence, it is concluded that the models are robust to the choice of time delay.

The polynomial OE model with $n_d = 100$ gives an unusual step-response and frequency response (not shown). The model will therefore be ignored in the following discussion.

Zooming in on Figure 10 shows that some models give smooth outputs, while other models give outputs with some high-frequency noise. It turns out that longer time delay gives more noisy output. The reason is to be found in the Bode magnitude diagram of the models. This diagram is shown in Figure 11. The longer time delay, n_d , the higher amplitude at high frequencies. Hence, longer time delay means less lowpass-filtering of the model input signal, u . This explains why longer time delay gives more noisy output.

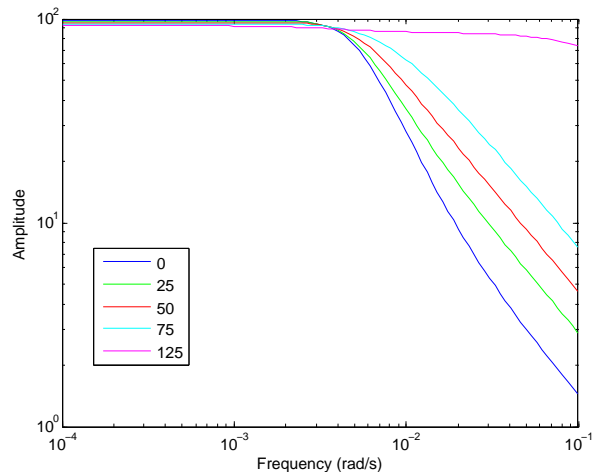


Figure 11: Bode magnitude diagram of the five successful polynomial OE models. The models' time delay, n_d , range from 0 to 125 samples in increments of 25 samples. The model having $n_d = 100$ is excluded.

It remains to be explained *why* longer time delays give less attenuation for increasing frequencies. Visual inspection of Figure 8 strongly indicates that there is some phase lag from u to \hat{T}^s , because the peaks in \bar{u} seem to occur before the corresponding peaks in \hat{T}^s . Long time *delay* gives large phase lag in the frequency domain. Long time *constants* also give large phase lag. Hence, it seems reasonable to conclude that models with long time delay use the time delay to give phase lag, while models with short time delay use long time constants to give phase lag. Long time delays do not change the magnitude of the Bode diagram, while long time constants attenuate high frequencies.

Quoting Ljung (1999): “Our acceptance of models should be guided by ‘usefulness’ rather than ‘truth’.” For the purpose of the sensor fusion algorithm, it is not of main interest to know the exact time delay. The implications of short or long time delay are more important. Hence, it seems reasonable to decide whether or not a lowpass-filtering effect in the model $G(z)$ is desirable, and choose the time delay based on this decision. There is some noise present in the signal from the graphite pyrometer, u . Hence, it is reasonable to require at least some lowpass-filtering. From a physical consideration of the process, it is reasonable to assume that the melted silicon in the crucible heats and cools off slower than the graphite ring. This further favors lowpass-filtering. The time delay is therefore chosen as $n_d = 0$. It is emphasized that this is not an exact scientific conclusion, but a choice that is reasonable based on the need for lowpass-filtering due to some measure-

ment noise in the graphite pyrometer output, u , as well as physical consideration of the process.

The next issue is to choose the polynomial orders n_b and n_f . The models shown in Figure 10 have $n_b = 1$ and $n_f = 2$, i.e. there are two parameters to be identified in each of the nominator and denominator. As the models give good fit, it is reasonable to assume that these polynomial orders are quite good. However, various polynomial orders will be tested to find the optimal ones. Polynomial OE models were identified with all combinations of $n_b \in \{0, 1, 2\}$ and $n_f \in \{1, 2, 3\}$, i.e. in total nine models. For all models the time delay is $n_d = 0$. The fit of these nine models are shown in Figure 12. It may be difficult to separate the nine models in the figure. However, this is not essential. The main point of the figure is that the models form three groups. The first group is all models with $n_f = 1$ (regardless of n_b). The second group is $n_b = 0$ and $n_f \in \{2, 3\}$. The third group is the remaining four models, i.e. $n_b \in \{1, 2\}$ and $n_f \in \{2, 3\}$. Hence, it seems reasonable to conclude that for models with $n_b = 0$ and/or $n_f = 1$, there is some dynamics that is not properly modeled due to too few polynomial parameters. However, the modeled dynamics does not change significantly as n_b increases from 1 to 2 (provided $n_f \geq 2$). Similarly, the dynamics does not change significantly as n_f increases from 2 to 3. It is therefore concluded that $n_b = 1$ and $n_f = 2$ are sufficiently high polynomial orders.

Summing up the identification work: The final model $G(z)$ is chosen as a polynomial OE model, (9), with $n_b = 1$, $n_f = 2$, and $n_d = 0$. Hence, the model is on the form

$$G(z) = \frac{b_0 + b_1 z^{-1}}{1 + f_1 z^{-1} + f_2 z^{-2}}. \quad (12)$$

The model has four parameters to be identified. The final model is shown in Figure 13. The final model turns out to be identical to the model labeled “0” in Figure 11. This figure shows the model’s Bode magnitude diagram.

9 Merging the Sub-Algorithms

As shown in Figure 3, the sensor fusion algorithm consists of two sub-algorithms: (i) The complementary filters, $H(z)$ and $1 - H(z)$, which filter characteristics were developed in Section 7, and (ii) the dynamic model, $G(z)$, which was developed in Section 8. These sub-algorithms are now to be merged into the final sensor fusion algorithm. The output of the sensor fusion algorithm is the silicon temperature estimate, \hat{T} , which is a function of the silicon pyrometer, \hat{T}^s , and the graphite pyrometer, u . The estimate is given by

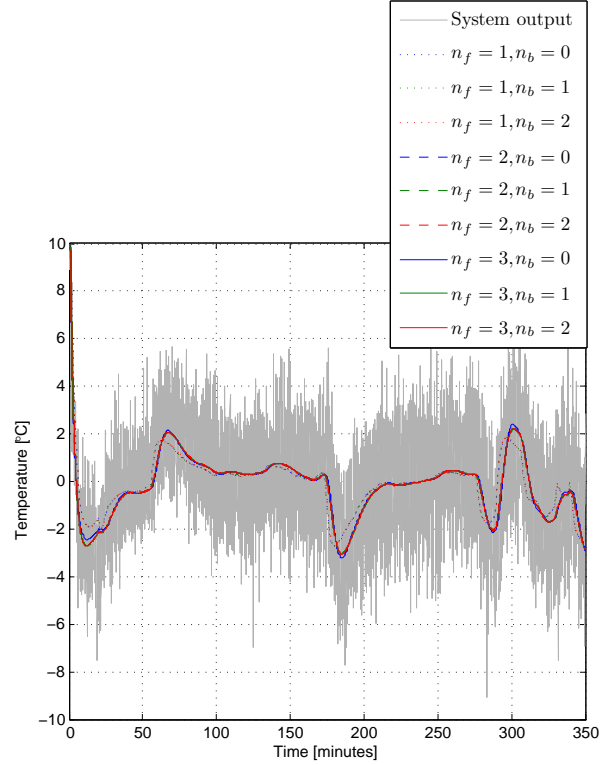


Figure 12: Polynomial OE models with all combinations of $n_b \in \{0, 1, 2\}$ and $n_f \in \{1, 2, 3\}$. The time delay is $n_d = 0$ for all models.

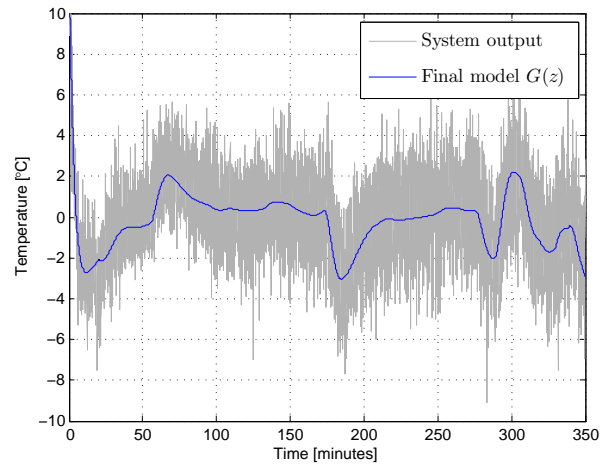
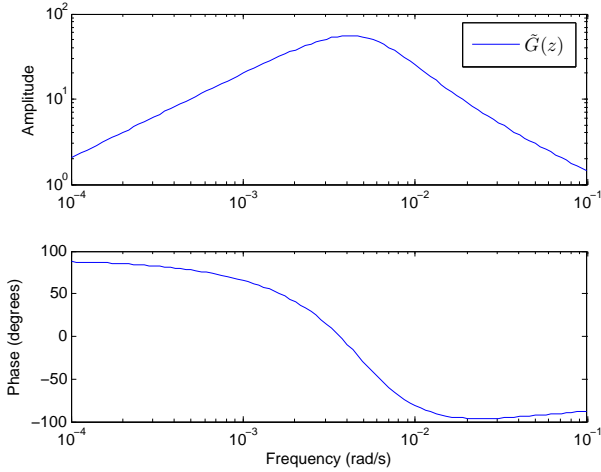


Figure 13: The final model $G(z)$.


 Figure 14: Bode diagram of $\tilde{G}(z)$.

$$\begin{aligned}\hat{T} &= H(z) \hat{T}^s + (1 - H(z)) \hat{T}^g \\ &= H(z) \hat{T}^s + (1 - H(z)) G(z) u.\end{aligned}\quad (13)$$

It is convenient to define

$$\tilde{G}(z) \stackrel{\text{def}}{=} (1 - H(z)) G(z). \quad (14)$$

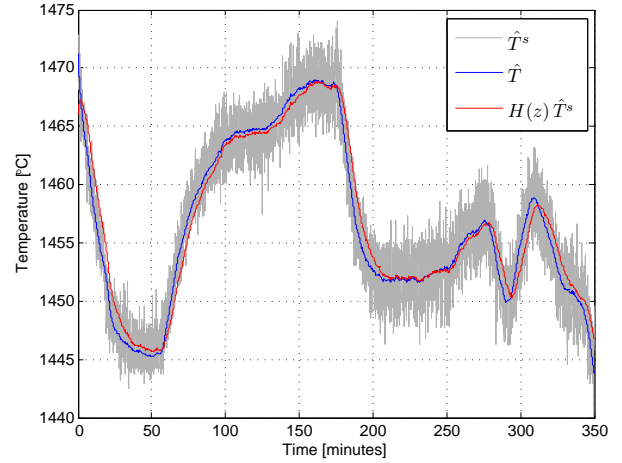
The estimated silicon temperature can then be written

$$\hat{T} = H(z) \hat{T}^s + \tilde{G}(z) u. \quad (15)$$

The transfer function $H(z)$ is a lowpass filter. The Bode diagram of $H(z)$ is shown in Figure 7. The transfer function $\tilde{G}(z)$ is a series connection of the model $G(z)$, which has lowpass characteristics, and the high-pass filter $1 - H(z)$. Hence, $\tilde{G}(z)$ has bandpass characteristics. The Bode diagram of $\tilde{G}(z)$ is shown in Figure 14.

10 Validating the Sensor Fusion Algorithm

The purpose of the sensor fusion algorithm is to filter the measurement noise of the silicon pyrometer, \hat{T}^s , with significant less phase lag than a traditional lowpass filter. It is then natural to compare the output of the sensor fusion algorithm, \hat{T} , with the output of the lowpass filter $H(z)$. That is, the sensor fusion algorithm, $\hat{T} = H(z) \hat{T}^s + \tilde{G}(z) u$, is to be compared with $H(z) \hat{T}^s$. Figure 15 shows \hat{T} and $H(z) \hat{T}^s$ plotted against the unfiltered silicon pyrometer signal, \hat{T}^s . Figure 16 shows the same data as Figure 15, zoomed in at three different areas. The figures show that the


 Figure 15: The output of the sensor fusion algorithm, \hat{T} , and the lowpass-filtered silicon pyrometer, $H(z) \hat{T}^s$, plotted against the unfiltered silicon pyrometer, \hat{T}^s .

sensor fusion algorithm, \hat{T} , follows the unfiltered signal, \hat{T}^s , much closer than the lowpass-filtered value $H(z) \hat{T}^s$ does.

The output of the sensor fusion algorithm, \hat{T} , consists of two contributions: $H(z) \hat{T}^s$ and $\tilde{G}(z) u$. Visual comparison of the two contributions shows that the high-frequency noise at $\tilde{G}(z) u$ is neglectable compared to $H(z) \hat{T}^s$ (this plot is not shown). Hence, \hat{T} has the same amount of high-frequency noise as $H(z) \hat{T}^s$.

Figures 15 and 16 compare \hat{T} and $H(z) \hat{T}^s$ in the time domain. It is also of interest to compare these two signals in the frequency domain. However, it is not straight forward how to do this. The following approach has been chosen here: Two models, $R(z)$ and $S(z)$, are identified. $R(z)$ models the dynamics from \hat{T}^s to \hat{T} , and $S(z)$ models the dynamics from \hat{T}^s to $H(z) \hat{T}^s$. Both models are on the form (9) with $n_b = 1$, $n_f = 2$, and $n_d = 0$. The intention of the models is to get an estimate of which frequency components of the silicon temperature, T , (measured by the silicon pyrometer, \hat{T}^s) that are preserved in \hat{T} and $H(z) \hat{T}^s$, respectively. Figure 17 shows the Bode diagram of the models $R(z)$ and $S(z)$. As expected, the model $S(z)$ is identical to the lowpass filter $H(z)$. The Bode diagram shows that $S(z)$ attenuates at much lower frequencies than $R(z)$. This means that the sensor fusion algorithm preserves higher frequencies of the silicon temperature, T , than the lowpass filter $H(z)$ does, without letting through more noise. The phase diagram shows that the sensor fusion algorithm can handle much higher frequencies before it gives significant phase lag. The magnitude diagram indicates that

the sensor fusion algorithm slightly amplifies some frequency components around $\omega = 10^{-2}$ rad/s. This is an undesirable behavior.

Figures 15 through 17 show that the sensor fusion algorithm is successful when validated on the same measurement data that were used to identify the model $G(z)$. However, the dynamics from the graphite pyrometer output, u , to the silicon temperature, T , (measured by the silicon pyrometer, \hat{T}^s) is nonlinear. The authors still chose to use a simple, linear model with few parameters to avoid overfitting the model. It is now of interest to validate how well the sensor fusion algorithm performs when tested on measurement data in a different temperature range. First the raw data will be presented. Figure 4 shows the 350 minutes of raw data used to identify the model $G(z)$. In this figure, the graphite pyrometer output, u , is replaced by a rescaled version \bar{u} . The relationship between u and \bar{u} is a first order polynomial, i.e. $\bar{u} = p_1 u + p_0$. The poly-

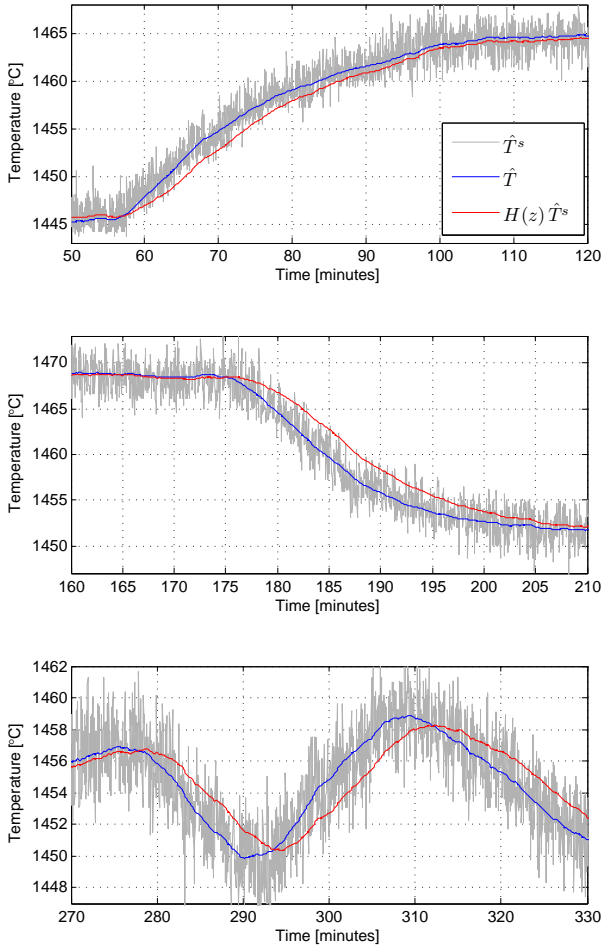


Figure 16: The same data shown in Figure 15, zoomed in at three different areas.

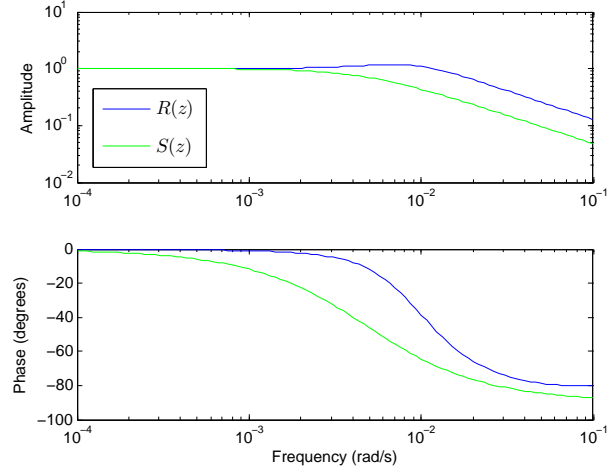


Figure 17: Bode diagram of the models $R(z)$ and $S(z)$.

nomial coefficients p_0 and p_1 were identified as the best (in a least squares sense) static fit between u and the silicon pyrometer, \hat{T}^s , over the data shown in Figure 4. Figure 18 shows the 350 minutes of logged data shown in Figure 4 and the following 150 minutes. While \bar{u} and \hat{T}^s follow closely over the first 350 minutes, i.e. the data range used to identify the polynomial coefficients p_0 and p_1 , \bar{u} and \hat{T}^s deviate significantly over the last 150 minutes. The authors can not see any other explanation of this deviation than that the temperature is significantly lower over the last 150 minutes. If this assumption is correct, the relationship between \bar{u} and \hat{T}^s must be significantly nonlinear.

Based on Figure 18, one can not expect the linear model, $G(z)$, which is developed based on the first 350 minutes, to perform well over the last 150 minutes. Figure 19 compares the output of the sensor fusion algorithm, \hat{T} , and the lowpass-filtered silicon pyrometer, $H(z)\hat{T}^s$, over the last 150 minutes of Figure 18. The sensor fusion algorithm performs poorly until approximately 390 minutes. This is because there is a large temperature drop from 310 minutes to 380 minutes that the linear model, $G(z)$, is not able to handle properly. However, it seems that the sensor fusion algorithm performs quite well after approximately 390 minutes, when the temperature settles at a new, lower level. This is because the highpass filter $1 - H(z)$ of $\tilde{G}(z)$ removes any static or low-frequency error of the temperature estimate \hat{T}^g . As the model, $G(z)$, is not identified for the low temperature range between 390 and 500 minutes, the performance of the sensor fusion algorithm may be poorer than in the temperature range used for identification.

Figures 15 and 16 give a visual impression of how

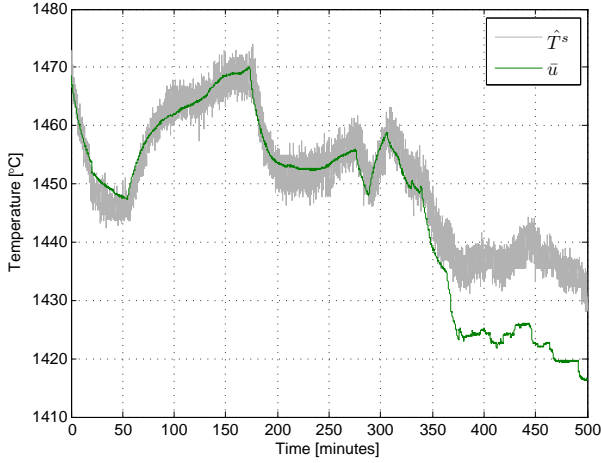


Figure 18: The rescaled graphite pyrometer, \bar{u} , and the silicon pyrometer, \hat{T}^s , over an extended time period.

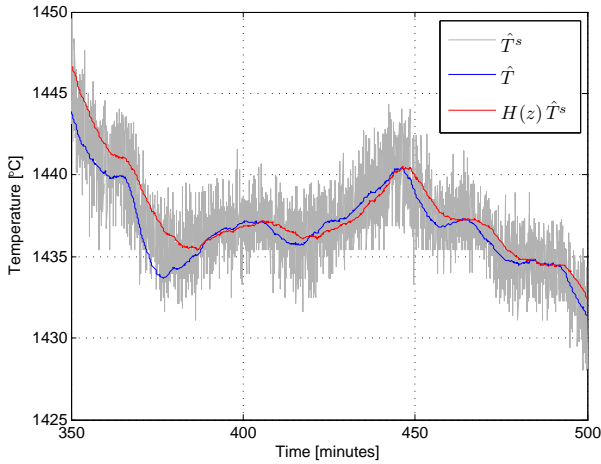


Figure 19: The output of the sensor fusion algorithm, \hat{T} , the lowpass-filtered silicon pyrometer, $H(z)\hat{T}^s$, and the unfiltered silicon pyrometer, \hat{T}^s , over an extended time period.

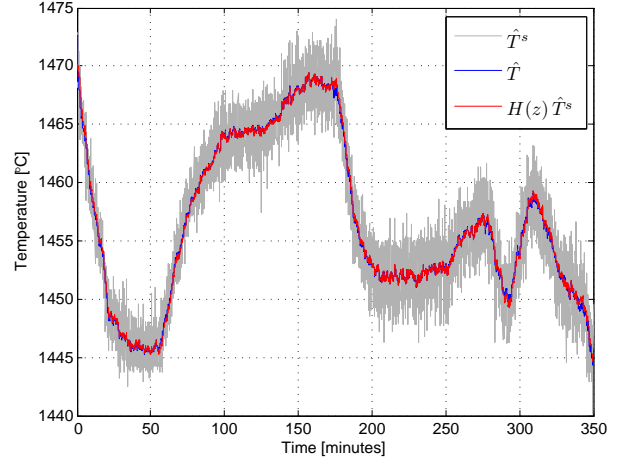


Figure 20: The output of the sensor fusion algorithm, \hat{T} , and the lowpass-filtered silicon pyrometer, $H(z)\hat{T}^s$, for cut-off frequency $\omega_c = 2.4 \times 10^{-2}$ rad/s.

much the sensor fusion algorithm improves the temperature estimate compared to a traditional lowpass filter. However, how much the sensor fusion algorithm improves the temperature estimate highly depends on the cut-off frequency of the complementary filters $H(z)$ and $1 - H(z)$. Figure 20 compares the output of the algorithm, \hat{T} , and the output of the lowpass filter, $H(z)\hat{T}^s$, when the cut-off frequency is increased by a factor of five, i.e. from $\omega_c = 4.8 \times 10^{-3}$ rad/s to $\omega_c = 2.4 \times 10^{-2}$ rad/s. Increasing the cut-off frequency is equivalent to increasing the tolerance for high-frequency noise. Figure 20 shows that the curves for \hat{T} and $H(z)\hat{T}^s$ are almost identical. Hence, there is no significant improvement of using the sensor fusion algorithm over the traditional lowpass filter for this choice of ω_c . Comparing Figures 15 and 20 shows that there is much more high-frequency noise at \hat{T} and $H(z)\hat{T}^s$ in the latter figure.

On the other hand, decreasing the cut-off frequency, i.e. having less tolerance for measurement noise, the improvement of using the sensor fusion algorithm, compared to a traditional lowpass filter, is much larger. Figure 21 compares \hat{T} and $H(z)\hat{T}^s$ when the cut-off frequency is decreased by a factor of five, i.e. from $\omega_c = 4.8 \times 10^{-3}$ rad/s to $\omega_c = 9.6 \times 10^{-4}$ rad/s. For this cut-off frequency the improvement of using the sensor fusion algorithm is very large. There is hardly any visible high-frequency noise at neither \hat{T} nor $H(z)\hat{T}^s$. For the simulations shown in Figures 20 and 21, the raw data were prefiltered with the chosen cut-off frequencies, and the model $G(z)$ was re-identified.

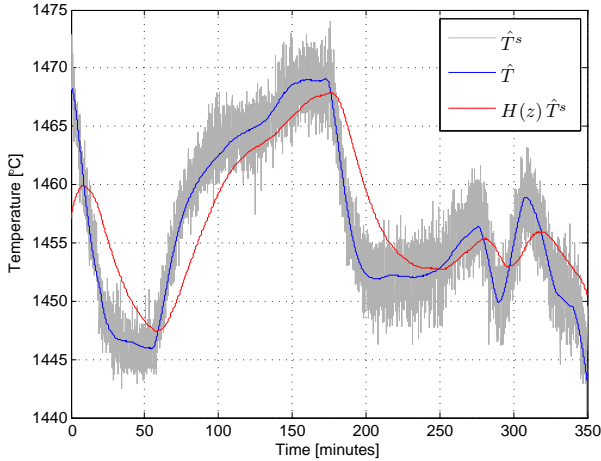


Figure 21: The output of the sensor fusion algorithm, \hat{T} , and the lowpass-filtered silicon pyrometer, $H(z)\hat{T}^s$, for cut-off frequency $\omega_c = 9.6 \times 10^{-4}$ rad/s.

11 Algorithm Weaknesses and Further Work

The sensor fusion algorithm presented in this paper is developed based on logged measurement data from a single CZ batch. Therefore, the algorithm has been validated based on the same data used for developing the algorithm. A simple, linear model with few parameters was chosen to model the dynamics from the graphite pyrometer to the silicon temperature. The model structure was chosen to avoid overfitting, because flexible, nonlinear black-box model structures typically have significantly more parameters. However, the dynamics is known to be nonlinear (see Figure 18). An important part of further work will therefore be to understand this nonlinear dynamics, either physically or empirically, and build this into the model. As the CZ process is a batch process, the dynamics may also be time-varying. In particular, it seems reasonable to assume that the dynamics may vary with the level of molten silicon in the crucible. Preferably, the model should be developed based on data from several CZ batches to make sure the data are representative and sufficiently informative. Also, the algorithm should be validated based on data from several *independent* CZ batches.

According to [Brown and Hwang \(1997\)](#), the complementary filters $H(z)$ and $1 - H(z)$ can be computed to give a statistically optimal temperature estimate, i.e. to minimize the variance of the estimation error. A Kalman filter or a Wiener filter can be used for this

computation. Even though the complementary filters can be chosen optimally in theory, modeling errors and estimation errors in the noise / disturbance covariance matrices are likely to give a sub-optimal temperature estimate. Also, even though the temperature estimate is in fact optimal, it may be too noisy for its desired application. However, even if there are practical issues related to using Kalman filter or Wiener filter, this is an interesting issue to consider for further work.

The lowpass filter removes high-frequency noise from the silicon pyrometer. The graphite pyrometer, in combination with the dynamic model, is used to estimate those frequency components of the silicon temperature that are removed by the lowpass filter. A disadvantage of this approach is that it requires two pyrometers. The heating element heats the entire CZ process, including the crucible and the silicon. It may be possible to replace the graphite pyrometer with the measured heating element power. That is, to develop a model from the heating element power to the silicon temperature, and use this model to estimate the high-frequency components of the silicon temperature that are removed when lowpass-filtering the silicon pyrometer.

12 Conclusions

SINTEF Material and Chemistry operates a Czochralski (CZ) crystallization process. During one CZ batch, two pyrometers were used: The silicon pyrometer measures the temperature of the molten silicon. This pyrometer is assumed to be accurate, but its output signal has much high-frequency noise. The noise can be attenuated using a traditional lowpass filter. However, this approach will give a phase lag that is unfortunate for the temperature control. The graphite pyrometer measures the temperature of a graphite material. Hence, the graphite pyrometer does not give an exact representation of the silicon temperature. However, the graphite pyrometer has little measurement noise. There is quite a good correlation between the silicon pyrometer and the graphite pyrometer.

This paper presents a sensor fusion algorithm that attenuates the measurement noise of the silicon pyrometer, while giving significant less phase lag than a traditional lowpass filter. The algorithm consists of two sub-algorithms: (i) A dynamic model and (ii) complementary filters.

The dynamic model estimates the silicon temperature as a function of the graphite pyrometer. The model is a linear output error (OE) model with four parameters. The parameters were identified using the prediction error method (PEM), where the graphite pyrometer is the system input and the silicon temperature (measured by the silicon pyrometer) is the

system output. A linear model structure was chosen despite the fact that the dynamics is known to be nonlinear. Flexible, nonlinear black-box model structures typically have significantly more parameters than linear model structures. As no independent data were available for model validation, it was desirable to use a model structure with few parameters to avoid model overfitting.

A lowpass filter and a highpass filter are designed as complementary filters. The silicon pyrometer is lowpass-filtered, and the output of the OE model is highpass-filtered. These two filtered signals are then summed. This sum is the output of the sensor fusion algorithm, i.e. the estimated silicon temperature. In other words: The lowpass filter attenuates noise from the silicon pyrometer, while the OE model estimates the high-frequency components of the silicon temperature that are lost in the lowpass-filtering.

Validation of the sensor fusion algorithm shows that it works well on the data that were used to identify the OE model. The algorithm gives significantly less phase lag than traditional lowpass-filtering of the silicon pyrometer. The algorithm performs poorly when there are large, quick temperature changes outside the temperature range used for model identification. This is to be expected, because the linear model can not handle the nonlinear dynamics between the graphite pyrometer and the silicon temperature. The algorithm seems to perform quite well when the temperature is within a limited temperature range, even if this range is outside the temperature range used for model identification. The usefulness of the model depends on the choice of cut-off frequency of the filters. For high cut-off frequencies, the algorithm gives little or no improvement. For low cut-off frequencies, the algorithm gives a large improvement.

Further work on the algorithm should include analysis of the nonlinear dynamics from the graphite pyrometer to the silicon temperature, and build this into the dynamic model. This is likely to improve the algorithm's ability to handle large, quick temperature variations. The suggested analysis and modeling work, as well as more thorough validation of the algorithm, require logged data from more CZ batches.

Acknowledgements

The authors are most grateful to SINTEF Materials and Chemistry in Trondheim, Norway, for giving access to data from the company's CZ process, and for allowing these data to be used in this paper and other publications. The authors are also very grateful to NorSun AS for providing financial support to the CZ experiment which this paper is based on.

The first author is most grateful for the financial support of his PhD study from NorSun AS, Østfold Energi AS, and the Norwegian Research Council. The cooperation with NorSun AS, Prediktor AS, and SINTEF Materials and Chemistry is also very much appreciated.

References

- Brown, R. G. and Hwang, P. Y. C. *Introduction to random signals and applied Kalman filtering - 3rd edition*. John Wiley & Sons Inc., 1997.
- Haugen, F. *Dynamic Systems - Modeling, Analysis and Simulation*. Tapir Academic Press, Trondheim, Norway, 2004.
- Irizarry-Rivera, R. and Seider, W. D. Model-predictive control of the Czochralski crystallization process Part I. Conduction-dominated melt. *Journal of Crystal Growth*, 1997a. 178:593–611. doi:[10.1016/S0022-0248\(97\)00085-7](https://doi.org/10.1016/S0022-0248(97)00085-7).
- Irizarry-Rivera, R. and Seider, W. D. Model-predictive control of the Czochralski crystallization process Part II. Reduced-order convection model. *Journal of Crystal Growth*, 1997b. 178:612–633. doi:[10.1016/S0022-0248\(97\)00086-9](https://doi.org/10.1016/S0022-0248(97)00086-9).
- Lan, C. W. Recent progress of crystal growth modeling and growth control. *Chemical Engineering Science*, 2004. 59:1437–1457. doi:[10.1016/j.ces.2004.01.010](https://doi.org/10.1016/j.ces.2004.01.010).
- Lee, K., Lee, D., Park, J., and Lee, M. MPC based feedforward trajectory for pulling speed tracking control in the commercial Czochralski crystallization process. *International Journal of Control, Automation, and Systems*, 2005. 3:252–257.
- Ljung, L. *System Identification - Theory for the User, 2nd Edition*. Prentice-Hall, Upper Saddle River, New Jersey, 1999.
- Ljung, L. *System Identification Toolbox 7 - User's Guide*. The MathWorks, Inc., 2009.
- Lyons, R. G. *Understanding digital signal processing - 3rd edition*. Prentice Hall, 2011.