

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Kalman Filter Estimation and Its Implementation

Erick Ulin-Avila and Juan Ponce-Hernandez

Abstract

In this chapter, we use the Kalman filter to estimate the future state of a system. We present the theory, design, simulation, and implementation of the Kalman filter. We use as a case example the estimation of temperature using a Resistance Temperature Detector (RTD), which has not been reported before. After a brief literature review, the theoretical analysis of a Kalman filter is presented along with that of the RTD. The dynamics of the RTD system are analytically derived and identified using Matlab. Then, the design of a time-varying Kalman filter using Matlab is presented. The solution to the Riccati equation is used to estimate the future state. Then, we implement the design using C-code for a microprocessor ATMega328. We show under what conditions the system may be simplified. In our case, we reduced the order of the system to that of a system having a 1st order response, that of an RC system, giving us satisfactory results. Furthermore, we can find two first order systems whose response defines two boundaries inside which the evolution of a second order system remains.

Keywords: Kalman filter, prediction, Riccati equation

1. Introduction

A deterministic system is a system whose governing physical laws are specified so that if the state of the system at some time is known, then one can precisely predict the state at a later time. Nondeterministic systems are divided into two categories: stochastic and random. A stochastic system has governing physical laws that even if the state at some point in time is known precisely, it is impossible to determine the state of the system at a later time precisely. It is possible to determine the probability of a state, rather than the state itself. A random system is one which has no apparent governing physical laws. Practically, we treat all unpredictable systems, stochastic or random as stochastic systems, since we employ the same methods to study them. While we are unable to predict the state of a random process, we can evolve a strategy to deal with such processes. Such a strategy is based on a branch of mathematics dealing with unpredictable systems, called statistics.

Estimation is the process of extracting information from data which can be used to predict the behavior of state variables in a system. The estimation uses statistical criteria to infer the actual value of unknown variables. Estimation models are used to process noisy measurements, filter them, and detect inaccuracies. When random signals are passed through a deterministic system, their statistical properties are

modified. A deterministic system to which random signals are input, so that the output is a random signal with desired statistical properties is called a filter. Filters can be linear or nonlinear, time-invariant or time varying. However, for simplicity we will usually consider linear, time-invariant filters. Linear, time-invariant filters are commonly employed in control systems to reduce the effect of measurement noise on the control system. In such systems, the output is usually a superposition of a deterministic signal and a random measurement noise.

The output of a filter not only has a frequency content different from the input signal, but also certain other characteristics of the filter, such as a phase-shift or a change in magnitude. In other words, the signal passing through a filter is also distorted by the filter, which is undesirable. A filter would produce an output signal based upon its characteristics, described by the transfer-function, frequency or impulse response, or a state-space representation of the filter. However, a filter can be designed to achieve a desired set of performance objectives, i.e. the numerator and denominator polynomials of the filter's transfer function, or coefficient matrices of the filter's state-space model, can be selected by a design process to achieve the conflicting requirements of maximum noise attenuation and minimum signal distortion.

There are several prediction models to infer the system state, although, it can be shown that of all estimation tools Kalman Filter (KF) is the one that minimizes the variance of the estimation error which enables accurate estimation of the process.

1.1 Literature review

The first application of state estimation was in the aerospace field to solve problems related to the prediction of position in aerospace vehicles. Nowadays, estimation has been applied in several fields of engineering and control systems. One common application is in data acquisition, to solve the problem of predicting the state of a system that cannot be measured directly due to the characteristics and complexity of the environment.

KF is an estimator proposed by Rudolph E. Kalman in 1960. It is an algorithm to estimate the evolution of a dynamic system, especially when data has a lot of noise. The principle of the filter is to find the probability of the hypothesis of predicted state and using the data from the measurement to correct it and improve the future estimation at each time. It is a suitable algorithm to apply in dynamic systems, linking real-time measurements and predicting the state of system parameters through time approaches. KF has been implemented in several fields, such as in navigation systems [1–4], financial models [5–7], tracking vehicles [8, 9] and image processing [10–12]; only to mention some of them. Nevertheless, this statistical tool is useful for two main purposes: estimation and performance analysis of estimators.

In the field of IC technology, it has been implemented for thermal estimation. Multicore processors use a dynamic thermal management mechanism that use embedded thermal sensors for monitoring the real-time thermal behavior of the processor, this kind of sensors are susceptible to a variety of source of noise and this causes the discrepancies between actual temperatures observed by on-chip thermal sensors. Therefore, to fix the discrepancies in sensing, Kalman's prediction is used to estimate real values from noisy sensor readings [13]. Another novel application of KF is in the electric vehicle industry, the estimation of the charge state of lithium-ion battery is an important parameter in order to guarantee a safe operation of them. The battery performance is influenced by aging; this fact makes difficult to predict the battery state, to overcome this issue the application of KF in combination with other methods is a suitable methodology [14–17].

Recently, KF has been applied in several industrial applications. With the development of manufacturing process, welding automation emerges as one important tool to speed up the production rate in the assembly line in stronger and high-quality welds. Nevertheless, there are several factors that could influence the welding quality and the most important is the arc length, which could be influenced by the irregular surface of the workpiece and the loss of the tungsten electrode. To enhance the quality during the Gas-Tungsten Arc Welding (GTAW) process, KF is applied in order to keep the arc length stable and minimize the external noise [18]. In the field of sensorless control, KF have been used in intelligence electrical drives. To control induction motor drives without mechanical speed sensor at the motor shaft allows reduced hardware complexity, and low costs. Additionally, the use of induction motors without position sensor is useful for applications with abrasive and hard surface. Thereby, the application of an estimation method it's necessary in order to predict the position and velocity of the shaft [19–21].

In applications related with radio astronomy, KF has been applied for the analysis of Very-Long-Baseline Interferometry (VLBI) data, in order to analyze parameters such as base line lengths, earth orientation parameters, radio source coordinates and tropospheric delays. Nowadays, modern antennas are being constructed and equipped with highly accurate broadband receiving systems. Besides the accurate observations gotten by astronomic instruments, it is necessary to implement estimation methods in order to optimize the models applied in data analysis [22, 23]. In power systems, one of the main difficulties is power quality due to total harmonics distortion (THD) that is mainly caused by nonlinear loads. THD effects are strongly correlated with issues as device heating, break down electronic components, network interference, etc. Several filters have been performed to decrease the effect of harmonics; nevertheless, the application of KF has shown an important reduction in the effect of harmonics [24–26]. In the field of biomedicine KF is widely used over other estimation methodologies to overcome the different sources of noise. Specifically, KF has been used to smooth and predict signals from Electroencephalogram and Electrocardiogram signals [27, 28]. Recently in the literature there are reports on a new methodology to protect the confidentiality of the transmitted data based on a Kalman filter. This strategy proposes the implementation of encrypted algorithm using KF, and is suggested to be used in Industrial cyber-physical systems (ICPSs) to protected data privacy [29, 30].

As it has been mentioned above, KF has been used in diverse fields of science and technology to predict specific parameters of interest according to the application. Temperature evolution is an important parameter to measure and predict, in order to study or control the temperature in an environment [31, 32], device [13, 33, 34] and process [18]. It is well known that RTDs are commercial devices very useful to monitor the temperature due their stability and accuracy. However, RTDs are self-heating causing noisy readings making the RTD a suitable example to implement KF for temperature estimation. Importantly, we searched in the literature and found no evidence of previous work reporting the use of a KF to filter the noise and predict the temperature behavior from RTD readings.

2. Theoretical analysis of a Kalman Filter

The final objective of this study is to obtain the specification of a linear dynamic system (Wiener filter [35]) which accomplishes the prediction, separation, or detection of a random signal [36]. With the state-transition method, a single derivation covers a large variety of problems: growing and infinite memory filters, stationary and non-stationary statistics, etc. Having guessed the “state” of the

estimation (i.e., filtering or prediction) problem correctly, one is led to a nonlinear difference (or differential) equation for the covariance matrix of the optimal estimation error. From the solution of the equation for the covariance matrix we obtain the coefficients characterizing the optimal linear filter [36]. The following is a simplified derivation described previously in the references [37, 38].

2.1 Defining statistical quantities of use

The initial state, $\mathbf{x}(0)$, of a stochastic system is insufficient to determine its future state, $\mathbf{x}(t)$. Thus, based upon a statistical analysis of similar systems, and taking the average of their future states at a given time, t , we can calculate the mean state-vector as follows:

$$\mathbf{x}_m(t) = (1/N) \sum_{i=1}^N \mathbf{x}_i(t) \quad (1)$$

Thus $\mathbf{x}_m(t)$ is the expected state vector after studying N systems. It is also called the expected value of the state-vector, $\mathbf{x}_m(t) = E[\mathbf{x}(t)]$. Another statistical quantity of use is the correlation matrix of the state-vector:

$$\mathbf{P}_x(t, \tau) = (1/N) \sum_{i=1}^N \mathbf{x}_i(t) \mathbf{x}_i^T(\tau) \quad (2)$$

The correlation matrix, $\mathbf{P}_x(t, \tau)$, is a measure of correlation, a statistical property among the different state variables, and between the same state variable at two different times. Two scalar variables, $x_1(t)$ and $x_2(t)$, are said to be uncorrelated if the expected value of $x_1(t)x_2(\tau)$, i.e. $E[x_1(t)x_2(\tau)] = 0$, where τ is different from t .

The correlation matrix is the expected value of the matrix $\mathbf{x}_i(t)\mathbf{x}_i^T(\tau)$, or $\mathbf{P}_x(t, \tau) = E[\mathbf{x}_i(t)\mathbf{x}_i^T(\tau)]$. When $t = \tau$, the correlation matrix $\mathbf{P}_x(t, t) = E[\mathbf{x}_i(t)\mathbf{x}_i^T(t)]$, is called the covariance matrix. The covariance matrix, $\mathbf{P}_x(t, t)$, is symmetric. If $\mathbf{P}_x(t, \tau)$ is a diagonal matrix i.e. $E[x_i(t)x_j(\tau)] = 0$, where $i \neq j$, it implies that all the state variables are uncorrelated.

2.2 Defining the filter in state space - discrete domain

Consider a plant which we cannot model accurately using only a deterministic model, because of the presence of uncertainties called process noise and measurement noise:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{w}_k \quad (3)$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k \quad (4)$$

In the linear, time-varying state-space representation above, \mathbf{w} is the process noise vector which may arise due to modeling errors such as neglecting nonlinear dynamics, and \mathbf{v} is the measurement noise vector. The random noises, \mathbf{w} and \mathbf{v} , are assumed to be stationary white noises. The covariance matrices of stationary white noises, \mathbf{w} and \mathbf{v} , can be expressed as follows:

$$\mathbf{Q} = E[\mathbf{w}_k \mathbf{w}_k^T] \quad (5)$$

$$\mathbf{R} = E[\mathbf{v}_k \mathbf{v}_k^T] \quad (6)$$

Since we cannot predict the state-vector, \mathbf{x} of a stochastic plant, an observer is required for estimating the state-vector, based upon a measurement of the output, \mathbf{y} and a known input, \mathbf{u} . We need an observer that calculates the estimated state-vector, $\hat{\mathbf{x}}$, optimally, based upon statistical description of the vector output and plant state. Such an observer is the Kalman Filter, which minimizes a statistical measure of the estimation error, $\mathbf{e}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k$. This statistical measure is the covariance of the estimation error:

$$\mathbf{P}_k = E[\mathbf{e}_k \mathbf{e}_k^T] = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T] \quad (7)$$

Since the state-vector, \mathbf{x} , is a random vector and the estimated state $\hat{\mathbf{x}}$, is based on the measurement of the output, \mathbf{y} , for a finite time, say T , where $t \geq T$ then a true statistical average of \mathbf{x} would require measuring the output for an infinite time.

If $T < t$, this is a data-smoothing (interpolation) problem. If $T = t$, this is called filtering. If $T > t$, we have a prediction problem. Since the original treatment is general enough, the collective term estimation is used [36].

Hence, the best estimate to obtain for \mathbf{x} is not the true mean, but a conditional mean, \mathbf{x}_m , based on only a finite time record of the output, \mathbf{y} :

$$\mathbf{x}_m = E[\mathbf{x} : \mathbf{y}, T \leq t] \quad (8)$$

Taking in consideration the deviation of the estimated state- vector, $\hat{\mathbf{x}}$, from the conditional mean, \mathbf{x}_m , we can write the estimated state- vector as:

$$\hat{\mathbf{x}} = \mathbf{x}_m + \Delta \mathbf{x} \quad (9)$$

$\Delta \mathbf{x}$ is the deviation from the conditional mean. The conditional covariance matrix of the estimation error based on a finite record of the output is then:

$$\mathbf{P}_k = E[\mathbf{e}_k \mathbf{e}_k^T : \mathbf{y}, T \leq t] = E[\mathbf{x} \mathbf{x}^T] - \mathbf{x}_m \mathbf{x}_m^T + \Delta \mathbf{x} \Delta \mathbf{x}^T \quad (10)$$

The best estimate of state-vector happens if $\Delta \mathbf{x} = 0$, or $\hat{\mathbf{x}} = \mathbf{x}_m$, and would result in a minimization of the conditional covariance matrix, or error covariance matrix, \mathbf{P}_k . In other words, minimization of \mathbf{P}_k yields the optimal observer, which is the Kalman filter.

2.3 Defining the Kalman gain

The state-equation of the Kalman filter is that of a time-varying observer, and can be written as follows:

$$\hat{\mathbf{x}}_{k+1} = \mathbf{A} \hat{\mathbf{x}}_k + \mathbf{B} \mathbf{u}_k + \mathbf{K}_k [\mathbf{y}_k - \mathbf{C} \hat{\mathbf{x}}_k] \quad (11)$$

\mathbf{K}_k is the gain matrix of the Kalman filter. Assuming the prior estimate of $\hat{\mathbf{x}}_k$ is called $\hat{\mathbf{x}}'_k$, gained by knowledge of the system. We write an update equation for the new estimate, combining the old estimate with measurement data, $\hat{\mathbf{x}}'_k = \mathbf{A} \hat{\mathbf{x}}_k + \mathbf{B} \mathbf{u}_k$ and:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}'_k + \mathbf{K}_k [\mathbf{y}_k - \mathbf{C} \hat{\mathbf{x}}'_k] \quad (12)$$

If we substitute Eq. (4) into Eq. (12) we get:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}'_k + \mathbf{K}_k [\mathbf{C}\mathbf{x}_k + \mathbf{v}_k - \mathbf{C}\hat{\mathbf{x}}'_k] \quad (13)$$

Substituting Eq. (13) into Eq. (7)

$$\mathbf{P}_k = E \left[((\mathbf{I} - \mathbf{K}_k \mathbf{C})(\mathbf{x}_k - \hat{\mathbf{x}}'_k) - \mathbf{K}_k \mathbf{v}_k) ((\mathbf{I} - \mathbf{K}_k \mathbf{C})(\mathbf{x}_k - \hat{\mathbf{x}}'_k) - \mathbf{K}_k \mathbf{v}_k)^T \right] \quad (14)$$

Here $(\mathbf{x}_k - \hat{\mathbf{x}}'_k)$ is the error of the prior estimate. Since there is no correlation among the input, process noise and measurement noise, then the expectation may be re-written as;

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{C}) E \left[(\mathbf{x}_k - \hat{\mathbf{x}}'_k) (\mathbf{x}_k - \hat{\mathbf{x}}'_k)^T \right] (\mathbf{I} - \mathbf{K}_k \mathbf{C})^T + \mathbf{K}_k E [\mathbf{v}_k \mathbf{v}_k^T] \mathbf{K}_k^T \quad (15)$$

Using Eqs. (6) and (7), we obtain:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{C}) \mathbf{P}'_k (\mathbf{I} - \mathbf{K}_k \mathbf{C})^T + \mathbf{K}_k \mathbf{R} \mathbf{K}_k^T \quad (16)$$

Eq. (16) is the error covariance update equation, where \mathbf{P}'_k is the prior estimate of \mathbf{P}_k .

The trace of the error covariance matrix is the sum of the mean squared errors. The mean squared error may be reduced by minimizing the trace of \mathbf{P}_k . This requires to differentiate the trace of \mathbf{P}_k with respect to \mathbf{K}_k , then the result set to zero to find \mathbf{K}_k that minimizes the trace of \mathbf{P}_k .

We rewrite Eq. (16);

$$\mathbf{P}_k = \mathbf{P}'_k - \mathbf{P}'_k \mathbf{C}^T \mathbf{K}_k^T - \mathbf{K}_k \mathbf{C} \mathbf{P}'_k + \mathbf{K}_k \mathbf{C} \mathbf{P}'_k \mathbf{C}^T \mathbf{K}_k^T + \mathbf{K}_k \mathbf{R} \mathbf{K}_k^T \quad (17)$$

Taking the trace of this expression gives:

$$\mathbf{T}[\mathbf{P}_k] = \mathbf{T}[\mathbf{P}'_k] - 2\mathbf{T}[\mathbf{K}_k \mathbf{C} \mathbf{P}'_k] + \mathbf{T}(\mathbf{K}_k (\mathbf{C} \mathbf{P}'_k \mathbf{C}^T + \mathbf{R}) \mathbf{K}_k^T) \quad (18)$$

Then, we differentiate with respect to \mathbf{K}_k ;

$$\frac{d\mathbf{T}[\mathbf{P}_k]}{d\mathbf{K}_k} = -2\mathbf{T}[\mathbf{C} \mathbf{P}'_k] + 2\mathbf{T}(\mathbf{K}_k (\mathbf{C} \mathbf{P}'_k \mathbf{C}^T + \mathbf{R})) \quad (19)$$

Setting to zero and solving for \mathbf{K}_k we obtain the Kalman gain equation:

$$\mathbf{K}_k = \mathbf{P}'_k \mathbf{C}^T (\mathbf{C} \mathbf{P}'_k \mathbf{C}^T + \mathbf{R})^{-1} \quad (20)$$

Substitution of Eq. (20) into [17], gives:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{C}) \mathbf{P}'_k \quad (21)$$

Eq. (21) is the update equation for the error covariance matrix with optimal gain.

State projection is derived using;

$$\hat{\mathbf{x}}'_{k+1} = \mathbf{A} \hat{\mathbf{x}}'_k + \mathbf{w}_k \quad (22)$$

To project the error covariance matrix into the next time interval, $k + 1$ we first find an expression for the error based on the prior error;

$$\begin{aligned} \mathbf{e}'_{k+1} &= \mathbf{x}_{k+1} - \hat{\mathbf{x}}'_{k+1} \\ &= \mathbf{A}\mathbf{x}_k + \mathbf{w}_k - \mathbf{A}\hat{\mathbf{x}}'_k \\ &= \mathbf{A}\mathbf{e}_k + \mathbf{w}_k \end{aligned} \quad (23)$$

Eq. (7) in time $k + 1$ is;

$$\mathbf{P}'_{k+1} = E[\mathbf{e}'_{k+1}\mathbf{e}'_{k+1}{}^T] = E[(\mathbf{A}\mathbf{e}_k + \mathbf{w}_k)(\mathbf{A}\mathbf{e}_k + \mathbf{w}_k){}^T] \quad (24)$$

Assuming that \mathbf{e}_k and \mathbf{w}_k have zero cross-correlation.

$$\begin{aligned} \mathbf{P}'_{k+1} &= E[\mathbf{e}'_{k+1}\mathbf{e}'_{k+1}{}^T] \\ &= E[(\mathbf{A}\mathbf{e}_k\mathbf{e}_k{}^T\mathbf{A}^T + \mathbf{w}_k\mathbf{w}_k{}^T)] \\ &= \mathbf{A}\mathbf{P}_k\mathbf{A}^T + \mathbf{Q} \end{aligned} \quad (25)$$

This completes the description of the filter.

2.4 Algorithm loop

An algorithm loop is required to make the program in MATLAB and in C-code for the microprocessor. The loop is summarized in the **Figure 1**.

The KF assumes that the system model is linear and known, the system and measurement noises are white, and the states have initial conditions with known means and variances. The power spectral densities used can be treated as tuning parameters to design an observer with excellent performance and robustness. The linear Kalman filter can also be used to design observers for nonlinear plants, by treating nonlinearities as process noise with appropriate power spectral density matrix.

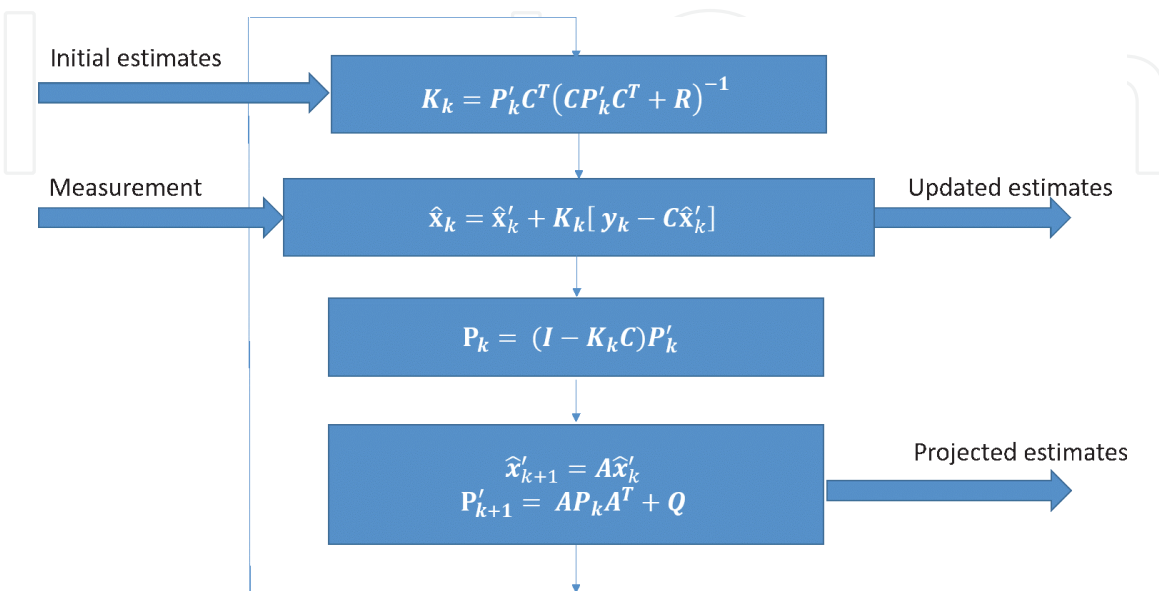


Figure 1.
 Recursive algorithm for the Kalman filter.

2.5 Derivation of the Riccati equation

Since the Kalman filter is an optimal observer the appearance of matrix Riccati equation is not surprising. We are interested in a steady Kalman filter, i.e. the Kalman filter for which the covariance matrix converges to a constant in the limit $t \rightarrow \infty$. This happens when the plant is time invariant. The derivation goes as follows [39, 40]:

From the projections into ∞ we get:

$$\hat{\mathbf{x}}'_{\infty+1} = \mathbf{A}\hat{\mathbf{x}}_{\infty} \quad (26)$$

$$\mathbf{P}_{\infty+1} = \mathbf{A}\mathbf{P}_{\infty}\mathbf{A}^T + \mathbf{Q} \quad (27)$$

$$\mathbf{P}_{\infty} = (\mathbf{I} - \mathbf{K}_{\infty}\mathbf{C})\mathbf{P}'_{\infty} \quad (28)$$

$$\hat{\mathbf{x}}_{\infty} = \hat{\mathbf{x}}'_{\infty} + \mathbf{K}_{\infty}[\mathbf{y}_{\infty} - \mathbf{C}\hat{\mathbf{x}}'_{\infty}] \quad (29)$$

$$\mathbf{K}_{\infty} = \mathbf{P}'_{\infty}\mathbf{C}^T(\mathbf{C}\mathbf{P}'_{\infty}\mathbf{C}^T + \mathbf{R})^{-1} \quad (30)$$

Using the Eqs. 26–29 we get:

$$\hat{\mathbf{x}}'_{\infty+1} = \mathbf{A}\hat{\mathbf{x}}'_{\infty} + \mathbf{A}\mathbf{K}_{\infty}[\mathbf{y}_{\infty} - \mathbf{C}\hat{\mathbf{x}}'_{\infty}] \quad (31)$$

$$\mathbf{P}_{\infty+1} = \mathbf{A}(\mathbf{I} - \mathbf{K}_{\infty}\mathbf{C})\mathbf{P}'_{\infty}\mathbf{A}^T + \mathbf{Q} \quad (32)$$

Using Eq. 30 in Eqs. 31 and 32 we get:

$$\hat{\mathbf{x}}'_{\infty+1} = \mathbf{A}\hat{\mathbf{x}}'_{\infty} + \mathbf{A}\mathbf{P}'_{\infty}\mathbf{C}^T(\mathbf{C}\mathbf{P}'_{\infty}\mathbf{C}^T + \mathbf{R})^{-1}[\mathbf{y}_{\infty} - \mathbf{C}\hat{\mathbf{x}}'_{\infty}] \quad (33)$$

$$\mathbf{P}_{\infty+1} = \mathbf{A}\left(\mathbf{I} - \mathbf{P}'_{\infty}\mathbf{C}^T(\mathbf{C}\mathbf{P}'_{\infty}\mathbf{C}^T + \mathbf{R})^{-1}\mathbf{C}\right)\mathbf{P}'_{\infty}\mathbf{A}^T + \mathbf{Q} \quad (34)$$

Rewriting Eq. 34 we get:

$$\mathbf{P}_{\infty+1} = \mathbf{A}\mathbf{P}'_{\infty}\mathbf{A}^T - \mathbf{A}\mathbf{P}'_{\infty}\mathbf{C}^T(\mathbf{C}\mathbf{P}'_{\infty}\mathbf{C}^T + \mathbf{R})^{-1}\mathbf{C}\mathbf{P}'_{\infty}\mathbf{A}^T + \mathbf{Q} \quad (35)$$

When in steady state:

$$\mathbf{P}_{\infty+1} = \mathbf{P}'_{\infty} = \mathbf{P}_{\infty} \quad (36)$$

Then we arrive at the Riccati equation:

$$\mathbf{A}\mathbf{P}_{\infty}\mathbf{A}^T - \mathbf{A}\mathbf{P}_{\infty}\mathbf{C}^T(\mathbf{C}\mathbf{P}_{\infty}\mathbf{C}^T + \mathbf{R})^{-1}\mathbf{C}\mathbf{P}_{\infty}\mathbf{A}^T - \mathbf{P}_{\infty} + \mathbf{Q} = \mathbf{0} \quad (37)$$

The iterative solution of the Riccati equation is not required in real time. The observer gain is calculated off-line for predictive control applications [40]. Riccati equations are mainly used to control large scale systems, estimation, and, detection processes.

2.6 Solution to the Riccati equation using MATLAB

In this work the discrete-time algebraic Riccati equation (DARE) was solved to obtain the covariance matrix \mathbf{P} of the Kalman gain. The discrete-time algebraic Riccati equation is represented by the next form [41]:

$$X = A^T X A + Q - (A^T X B)(R + B^T X B)^{-1}(B^T X A) \quad (38)$$

Where $A, X, Q = A^T \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $R \in \mathbb{R}^{m \times m}$ ($m \leq n$), and $R = B^T > 0$.
 Eq. (38) can be written in the short form:

$$A^T X (I + SX)^{-1} A - X + Q = 0 \quad (39)$$

Where:

$$S = BR^{-1}B^T \quad (40)$$

The application of the Kalman filter implies solving the DARE, which can be solved by several solution methods. Computational methods to solve Riccati equations can be categorized into three classes: invariant subspace methods, deflating subspace methods, and Newton's methods. The generalized Schur method that is classified as a deflating subspace method is used to solve DARE. The generalized Schur algorithm is a strong algebraic tool that allows computing classical decompositions of matrices, such as the QR and LU factorizations [42]. The next algorithm was used to solve DARE [43]:

Input arguments:

A – An $n \times n$ matrix

B – An $n \times m$ matrix

Q – An $n \times n$ symmetric matrix

R – An $m \times m$ symmetric matrix

Output arguments: X – DARE solution

1. Form the pencil $P_{DARE} - \lambda N_{DARE}$, where

$$P_{DARE} = \begin{pmatrix} A & 0 \\ -Q & I \end{pmatrix}, \quad (41)$$

$$N_{DARE} = \begin{pmatrix} I & S \\ 0 & A^T \end{pmatrix} \quad (42)$$

2. Transform the pencil $P_{DARE} - \lambda N_{DARE}$ to the generalized real Schur form apply QZ algorithm, that is, find orthogonal matrices Q_1 and Z_1 such that:

$$Q_1 P_{DARE} Z_1 = P_1 = \begin{pmatrix} P_{11} & P_{12} \\ 0 & P_{22} \end{pmatrix}, \quad (43)$$

$$Q_1 N_{DARE} Z_1 = N_1 = \begin{pmatrix} N_{11} & N_{12} \\ 0 & N_{22} \end{pmatrix} \quad (44)$$

3. Using an orthogonal transformation and reorder the generalized real Schur form. So that all the pencil $P_{11} - \lambda N_{11}$ has all the eigenvalues with moduli less than 1. Find Q_z and Z_2 orthogonal matrices, such that:

$$Q_2 Q_1 P_{DARE} Z_1 Z_2 = \text{quasi} - \text{upper triangular} \quad (45)$$

$$Q_2 Q_1 N_{DARE} Z_1 Z_2 = \text{upper triangular} \quad (46)$$

4. Form the matrix:

$$Z = Z_1 Z_2 = \begin{pmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{pmatrix} \quad (47)$$

5. Compute $X = Z_{21} Z_{11}^{-1}$

3. Application example: resistive temperature detectors (RTD)

Resistive temperature detectors (RTD) have attracted attention to be employed as thermal health monitors. As clinical thermometers they are stable and reliable presenting high accuracy and resolution [44]. One of the most widely used RTD is the emerging thin-film resistor which has minimal impact on complex circuits due to its small size and due to their negligible mass.

The basic function of the sensor is determined by a proportional increment of resistance when temperature is applied. RTDs can be employed on a rigid or flexible substrate [45–47], the metal combination with a flexible or rigid substrate can cover conformal applications. RTD fabrication can be done by metals like Pt [48–50], Cu [51], Ag [52], and Ni [53], among other materials. Nickel presents a suitable option for RTD fabrication due to its wide temperature linear range of operation and its relatively low price.

Clinical thermometers require a high definition and reliability because less than 1°C difference can indicate a health problem. The thermometer signal can be amplified by electronic means, but it is desirable to filter such readings. This work is focused to the filtering and prediction of an highly sensitive Nickel based thin film RTD (range, 273–325 K), to be incorporated to complex circuits [54], we present the theoretical analysis about the relation sensibility-resistance that matches with experimental results.

3.1 Theoretical analysis of an RTD

All metals produce an increase in its resistance to an increase in specific temperature, which means that resistance is linearly proportional to temperature change. This dependence between electrical resistance and temperature is the principle of operation used by a resistance temperature detector (RTD). The relation between temperature-resistance for Pt wire (RTD), is described by the equation known as the Calendar-Van Dusen, Eq. 41) [50].

$$R_{(T)} = R_{0^\circ C} (1 + \alpha T + \beta T^2) \quad (48)$$

Where $R_{0^\circ C}$ is the resistance at 0°C, α and β are temperature coefficients and T is temperature, the temperature coefficients depend only on material properties. In addition, the RTD resistance depends on its geometrical design, according to Eq. 42.

$$R = \frac{\sigma * L}{A} = \frac{\sigma * L}{w * t} \quad (49)$$

Where “ σ ” is the resistivity, “L” length, “A” lateral area, “w” channel width, and “t” channel height. Only by increasing the length “L” or decreasing the area “A” that means reducing the “t” film thickness or the “w” channel wide, the resistance can increase.

3.1.1 State-space description of an RTD

The estimation of the thermal system is represented by the linear stochastic state-space description $x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$ and $y_k = Cx_k + v_k$. Where A is an nxn state transition matrix applied to the previous state vector x_{k-1} , B is the control-input matrix applied to the control vector u_{k-1} , and w_{k-1} is the process noise vector. The linear combination of the measurement noise and the signal value is represented by y_k , where C is the measurement matrix, and v_k is the measurement noise vector with covariances matrices represented by Q and R. The covariances are assumed to be independent and are given by $Q = E[w_k w_k^T]$ and $R = E[v_k v_k^T]$.

Generally, the RTD system is modeled as an RLC circuit, which consists of a resistor a capacitor and an inductor in series with an input voltage. The output that we analyzed is the voltage across the resistor which is related to temperature change. The RLC circuit is represented by a second-order differential equation $L \frac{d^2 i(t)}{dt^2} + R \frac{di(t)}{dt} + \frac{1}{C} i(t) = 0$. To solve the above equation we implement the next matrix system:

$$\begin{aligned} A &= \begin{pmatrix} 0 & 1 \\ -1/L \cdot C & -R/L \end{pmatrix} & B &= \begin{pmatrix} 0 \\ 1/L \cdot C \end{pmatrix} \\ C &= (1 \ 0) & D &= 0 \end{aligned} \quad (50)$$

Also, we may simplify the response of the system to that of a first-order RC circuit. This implies to solve a first-order ordinary differential equation: $RC \frac{dq}{dt} + q = VC$. The dynamic model is defined by the following system:

$$\begin{aligned} A &= \begin{pmatrix} -\frac{1}{R \cdot C} \end{pmatrix} & B &= \begin{pmatrix} \frac{1}{R \cdot C} \end{pmatrix} \\ C &= 1 & D &= 0 \end{aligned} \quad (51)$$

4. Design and simulations

4.1 Kalman filter in resistance thermal detectors (RTD)

In this work, a Kalman Filter is proposed to decrease the time response to improve the speed feedback and filtering of the perturbations by signal noise from physical signals as thermal detectors. In some instances, a reduced model is advisable to use in an embedded system due to easy implementation and low computational complexity [2].

Kalman filter can be embedded in a temperature system made by Resistance Thermal Detectors (RTD). RTD's are robust elements that require relatively easy measurement, as a consequence are a useful thermal sensor for industry and medical applications. Nevertheless, these devices are exposing to vibration, electrical noise, and measurement errors generated by the thermoelectric effect caused by the temperature difference between electrical contacts, which affects the response time of the sensor. The implementation of the Kalman filter in a temperature system

produces an optimal estimative of thermal behavior and decreases the uncertainties about the prediction of the temperature.

In order to describe the system in the state space, it is necessary to apply system identification methods using MATLAB. Then, after obtaining the system's state space model we are able to use the Kalman filter algorithm to estimate the future output of the system.

To study the dynamics of our system, we used MATLAB functions **etfe** and **spa** to firstly estimate the empirical transfer functions and then estimate the frequency response with fixed frequency resolution using spectral analysis. The continuous time-identified transfer function obtained is:

$$\frac{2.278 s + 0.1711}{s^2 + 2.488 s + 0.1695} \quad (52)$$

Using MATLAB we are able to acquire the Discrete-time identified state-space model:

$$\begin{aligned} \mathbf{x}(t + T_s) &= \mathbf{A} \mathbf{x}(t) + \mathbf{B} u(t) + \mathbf{K} e(t) \\ y(t) &= \mathbf{C} \mathbf{x}(t) + \mathbf{D} u(t) + e(t) \end{aligned} \quad (53)$$

with:

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 0.8342 & -0.08908 \\ 0.0942 & -0.9716 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0.01966 \\ -0.03341 \end{bmatrix}, \\ \mathbf{C} &= [7.966 \quad 0.3005], \mathbf{D} = 0, \mathbf{K} = \begin{bmatrix} 0.006289 \\ -0.2834 \end{bmatrix} \end{aligned} \quad (54)$$

Estimated using **N4SID** on time domain data. Fit to estimation data: 90.27% (prediction focus) with FPE: 0.4532 and MSE: 0.2292. **Figure 2** shows the

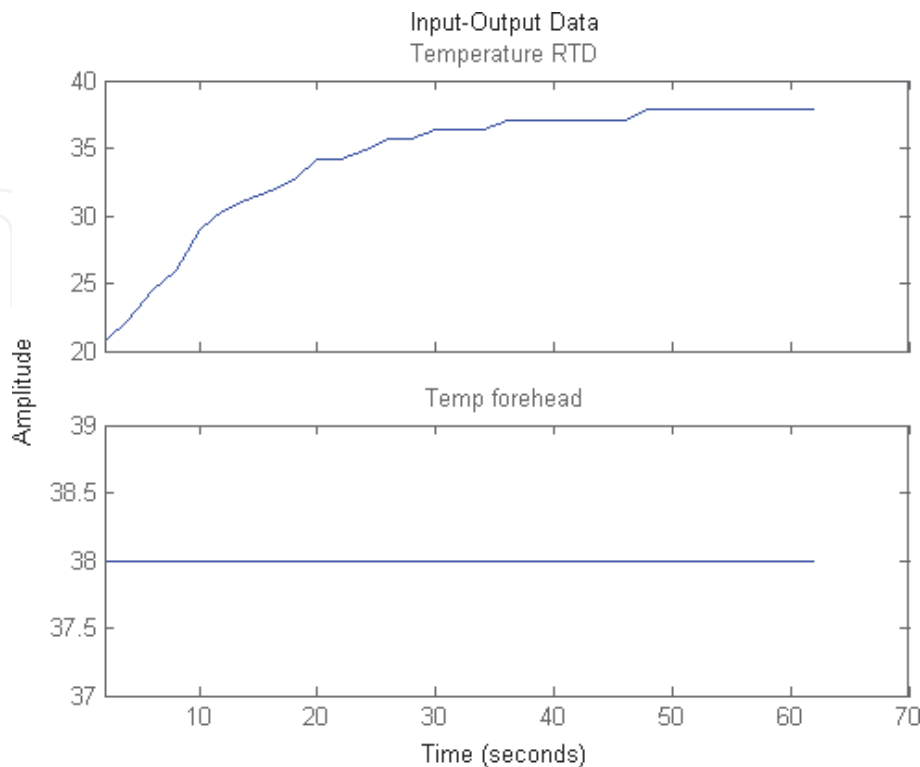


Figure 2.
System ID using MATLAB. Input output model for a step response defined problem.

Input–output model for which the input was set to a constant value of 38°C. The output, the step response, is that of the second order system as can be seen in the bode plot shown in **Figure 3**. **Figure 4** shows the evolution of the measured versus the modeled step responses.

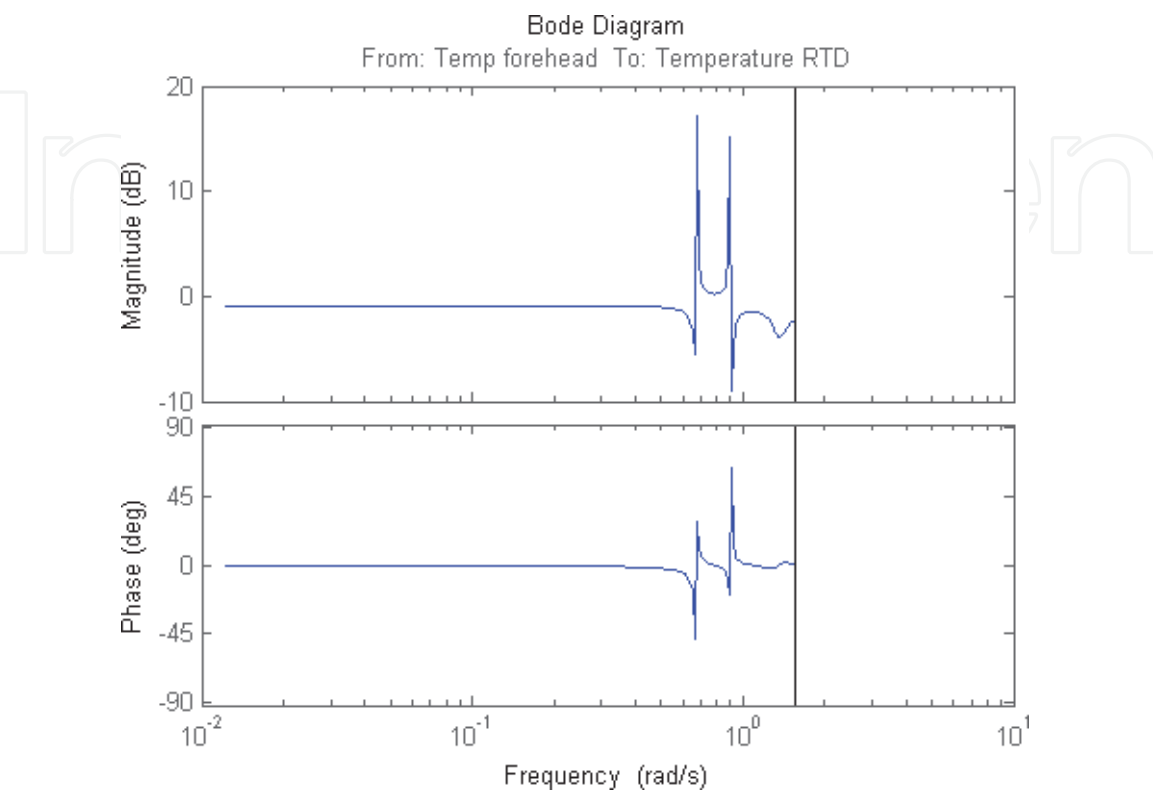


Figure 3.
Bode diagram indicating the system is a second order system as described by the system transfer function.

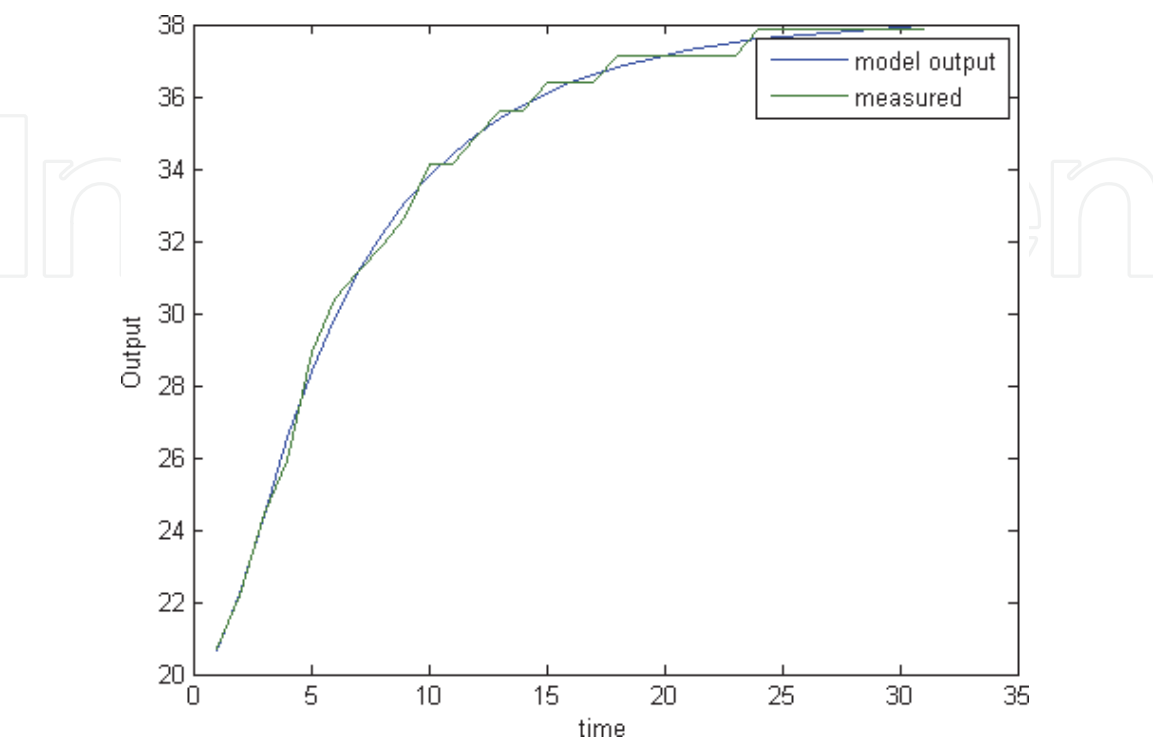


Figure 4.
Systems model and measured evolutions in time. Fit to estimation data: 90.27%.

4.2 Kalman filter

We modify the MATLAB example for the time-varying case found in [55] and we code our own function to solve the Discrete Algebraic Riccati Equation. MATLAB functions like predict or forecast were found useful to understand the problem at hand, however they were not used in the code we present here.

4.2.1 Time-varying Kalman filter using MATLAB

```
w(1:n) = sqrt(Q)*randn(n,1);
v(1:n) = sqrt(R)*randn(n,1);
systv = ss(A,B,C,0,Ts);
ytv(1:n) = lsim(systv,U(1:n) + w(1:n)).
yvtv(1:n) = ytv(1:n) + v(1:n);
Ptv(:, :) = B(:, :) * Q * B(:, :)'; % Initial error covariance.
x = zeros(order,1); % Initial condition on the state.
order = 2;
yetv(1:n) = zeros(n,1);
ycov(1:n) = zeros(n,1);
for i = 1:n.
% Measurement update.
Mn(:, :) = Ptv(:, :) * C(:, :)' / (C(:, :) * Ptv(:, :) * C(:, :)' + R);
x = x + Mn(:, :) * (yvtv(i) - C(:, :) * x); % x[n|n].
Ptv(:, :) = (eye(order) - Mn(:, :) * C(:, :)) * Ptv(:, :); % P[n|n].
yetv(i) = C(:, :) * x;
errcov(i) = C(:, :) * Ptv(:, :) * C(:, :)';
% Time update.
x = A(:, :) * x + B(:, :) * U(i); % x[n + 1|n].
Ptv(:, :) = A(:, :) * Ptv(:, :) * A(:, :)' + B(:, :) * Q * B(:, :)'; P[n + 1|n].
end
%% DARE. We coded our own dare function [X,L,G] = sdare(A,B,Q,R).
[P_inf,L,M_inf] = sdare(atv,ctv',Q,R);
for i = 1:p
% Measurement update.
x = x + M_inf * (yvtv(i) - ctv * x); % x[n|n].
yetv_inf(i) = ctv * x;
errcov_inf(i) = ctv * P_inf * ctv';
% Time update.
x = atv * x + btv * U(i); % x[n + 1|n].
P_inf = atv * P_inf * atv' + btv * Q * btv'; % P[n + 1|n].
end

function [SD] = sdare(A,B,Q,R).
At = transpose(A);
Bt = transpose(B);
S1 = size(A);
E = eye(S1);
Z = zeros(S1);
Ri = inv.(R);
S = B * Ri * Bt;
Pdare = [A Z; -Q E];
Ndare = [E S; Z At];
```

```
[AA,BB,L,Z] = qz(Pdare,Ndare);
[AAS1,BBS1,QS1,ZS1] = ordqz(AA,BB,L,Z,'udi');
O = ZS1(1:2,1:2);
P = ZS1(3:4,1:2);
H = inv(O);
SD = P*H;
end
```

5. Simulations

Matlab was used to simulate the response of an RTD modelled as a second order system. In **Figure 5(A)** we show the plot of the true response y (cyan line) and the filtered response (red line). In **Figure 5(B)** the plot compares the measurement error with the estimation error. As can be seen in **Figure 5(C)** the time-varying filter also estimates the covariance errcov of the estimation error at each sample which shows when the filter reached steady state. As it can be seen, we have the possibility to predict the state after approximately 8 seconds. Also, we show the evolution of the estimated temperature response showing an error of -0.0948°C in the best of the cases and less than 1°C in the worst of the cases after 45 seconds.

6. Implementation

The unit step response depends on the roots of the characteristic equation. If both roots are real-valued, the second-order system behaves like a chain of two first-order systems, and the step response has two exponential components. If the roots are complex, the step response is a harmonic oscillation with an exponentially decaying amplitude [56]. In our case, the roots of the characteristic polynomial: $s^2 + 2.488s + 0.1695$ are -2.4179 and -0.0701 . Thus our system behaves like two first order systems in series.

The state description for an RC system is described above. From there, we know that the dynamics are dependent only on the RC constant. In addition, there is an amplifier in the system electronics that has a gain of 260. To solve for the RC constant of the system we use the least-squares method (Chi square minimization). The system has a solution of the form $y = e^{Bx}$ and we take n data points to form the vectors x_i and Y_i . The problem is to minimize the error function, $\text{err} = \sum_{i=1}^n (Y_i - Ae^{Bx_i})^2$. The trick on the algorithm goes as follows:

$$Yn_i = \ln Y_i = \ln (Ae^{Bx_i}) = \ln A + \ln (e^{Bx_i}) = C + Bx_i \quad (55)$$

Which is a linear equation. Using a linear fitting program:

$$B = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2} \text{ and } c = \frac{\sum x_i^2 \sum y_i - \sum x_i \sum x_i y_i}{n \sum x_i^2 - (\sum x_i)^2} \quad (56)$$

We obtain B and $A = \exp(c)$. We coded two Kalman filters with different calibration parameters (Q and R) as is written below. The KF was implemented in an ATMEGA328 microprocessor. Code for an Arduino was generated using C language which is available in the following section. Only the relevant portion is written.

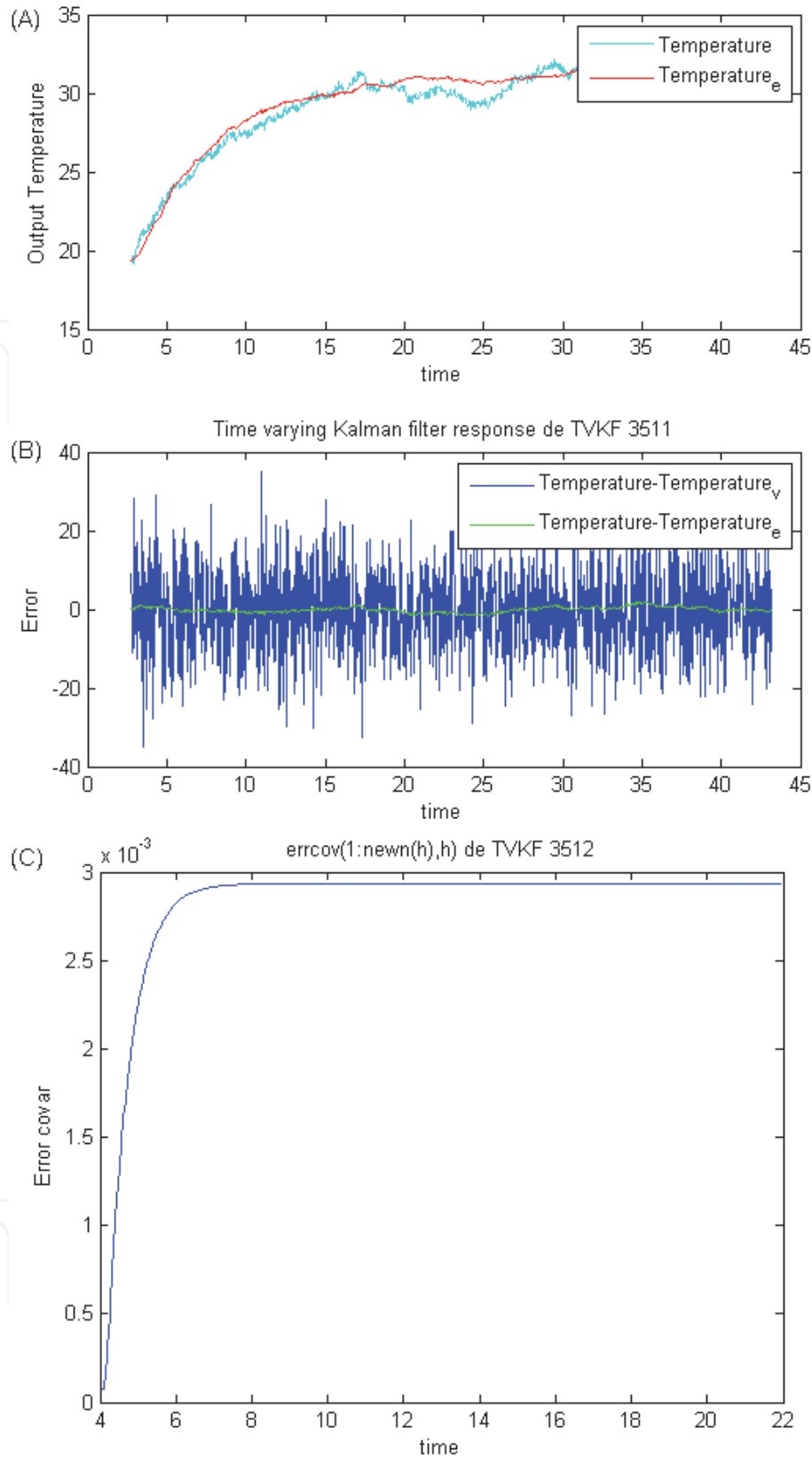


Figure 5.

(A) Evolution of the estimated temperature response showing an error of -0.0948°C in the best of the cases and less than 1°C in the worst of the cases. (B) Evolution of the measurement and estimation errors. (C) Evolution of the covariance of the error showing the possibility to predict the state after approximately 8 seconds.

6.1 Arduino code

```
readings[readIndex] = analogRead(inputPin); // read from the sensor.
total = total + readings[readIndex]; // add the reading to the total.
readIndex = readIndex + 1; // advance to the next position in the array.
```

```

time_equis_readings[time_equis_readIndex] = time_equis_readIndex;
time_equis_readIndex = time_equis_readIndex + 1;
if (readIndex >= numReadings) // if we're at the end of the array.
{
    for(i = 0; i <= numReadings-1; i++).
    {
        Y[i] = log(readings[i]);
        time1[i] = time_equis_readings[i];
        sumx = (sumx + time_equis_readings[i]);
        sumx2 = (sumx2 + time_equis_readings[i]*time_equis_readings[i]);
        sumy = (sumy + Y[i]);
        sumxy = (sumxy + time_equis_readings[i]*Y[i]);
    }
    den = (numReadings*sumx2-sumx*sumx);
    a = (sumx2*sumy - sumx*sumxy)/den;
    Bc = (n*sumxy-sumx*sumy)/den;
    // State description.
    A = -Bc; B=Bc; C = 260; D = 0;
    //wrap around to the beginning:
    readIndex = 0; time_equis_readIndex = 0;
}
// KALMAN.
errcov = C*P*C;
for(i = 0; i <= numReadings-1; i++).
{
    Mn = P*C/((C*P*C + R)); // initial estimate.
    X = X + Mn*(readings[i]-C*X); // update estimate Average_readings[i].
    P = (1-Mn*C)*P; // update covariance.
    y_e[i] = C*X;
    errcov = C*P*C;
    X = A*X + B*U; // project into k + 1.
    P = A*P*A + B*Q*B; // project into k + 1.
}
timer0_millis = millis();
// Solution to the Riccati equation.
F = -Bc; H = 260;
SQ = sqrt((H*H*Q*R) + (F*F*R*R));
SR = F * R;
P_inf = (SQ + SR)/(H*H);
M_inf = P_inf*C/(C*P_inf*C + R);
for(i = 0; i <= numReadings-1; i++).
{
    // Measurement update.
    // M_inf;
    x_inf = x_inf + M_inf*(readings[i]-C*x_inf); // % x[n|n].
    //P_inf; % P[n|n].
    y_e_inf = C*x_inf;
    errcov_inf = C*P_inf*C;
    // Time update.
    x_inf = A*x_inf + B*U;
    P_inf = A*P_inf*A + B*Q*B;
}
}

```


7. Results

The experiments were performed in a thermal bath giving step responses to the desired setup temperature. **Figure 6** depicts the upward and downward evolution of the temperature, the Kalman filter and the two predictors (using two different Q and R settings). As can be seen the predictors follow the Temperature of the sensor closely, especially for the upward way, while the Kalman filter lags behind.

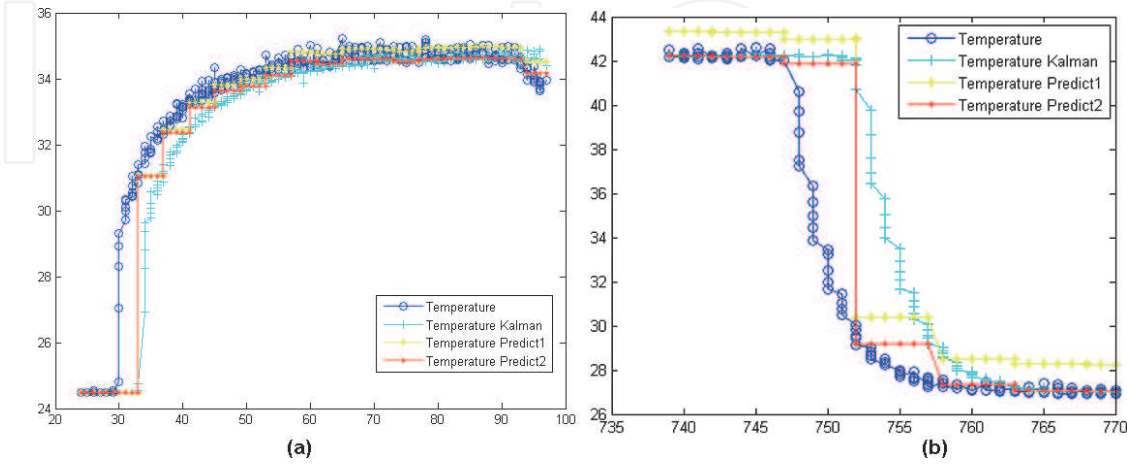


Figure 6.

Implemented system. Step response for the upwards and downwards evolution. Two different Kalman filters were used to predict (by solving the DARE equation) the evolution of the future state with different Q and R to calibrate the desired response. In blue the evolution of the RTD sensor analog input, in Yellow and red the two Kalman predictors and the Kalman estimation in cyan color.

8. Boundary layer

Sliding control [57] is an additional tool to predict the behavior of a second order system basically smoothing the system by boundary layers. The prediction of the system state trajectory is given using an uncertain model of the system. The subspace which represents the quantity of uncertainties in the prediction process, forces the estimate state trajectory to switching gain to converge the estimates to within a boundary of the real state values. To predict the state trajectory of our RLC system it's possible to switch its gain by the subspace represented by a first-order RC model. The estimated state trajectory is forced to keep a switch back and forth

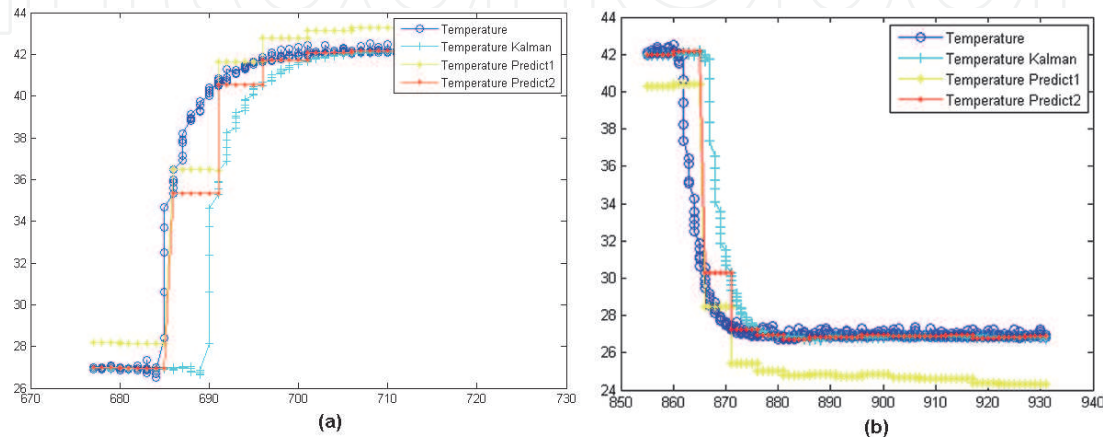


Figure 7.

Ascending and descending step responses of the Kalman filter and two Predictors which function in real time. In blue the RTD sensor response, in cyan the estimator response, in yellow and in red the two differently calibrated Kalman predictors.

within the boundary layer represented in our case by a RC model. By creating a boundary layer, the system is further constrained to have a solution existing in between two RC model solutions.

In **Figure 7** it can be clearly seen that the use of two estimators may help predict the behavior of the RTD in a much better way. The system needs to be calibrated first in order to have the two Kalman filters enveloping the required solution. As can be seen in the upward direction, both predictors (yellow and red) envelope the desired response (blue), that of the RTD sensor improving the response of the Kalman filter without boundaries (cyan). Unfortunately, this is not the case in the downward evolution. From the nonlinear control systems point of view these two evolutions demark a region where the RTD stands thus making possible to program a better estimator. It is left as an outlook to program a third estimator using this boundary layer in order to have a better predictor, especially for the downward evolution.

9. Conclusions


As it can be shown the implementation of the Kalman filter brings the opportunity to estimate the forecast in real time of a second order system using first, MATLAB and second that of two first order systems using a simple RC system coded in C-language for a microprocessor. It has been shown that the program is able to predict the evolution of temperature for a RTD system. Even if the system is implemented using a first order system we can find evolving solutions for our estimation and prediction to be good enough. We predict the state after approximately 8 seconds showing an error of -0.0948°C in the best of the cases. In addition, a boundary layer may be programmed using two first order Kalman predictors which may be tuned by setting Q and R properly. We believe this is the first report on the use of a Kalman filter to predict the evolution of temperature from a RTD.

Author details

Erick Ulin-Avila* and Juan Ponce-Hernandez
Center for Engineering and Industrial Development, Querétaro, México

*Address all correspondence to: ulinavilaerick@gmail.com

IntechOpen

© 2021 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Xincun, Y.; Yongzhong, O.; Fuping, S.; Hui, F. Kalman Filter Applied in Underwater Integrated Navigation System Underwater Integrated Navigation System Profile Kalman Filter 4 Kalman Rllter Integrated Navigation System of Underwater. *Geod. Geodyn.* **2013**, *4* (1), 46–50. <https://doi.org/10.3724/SP.J.1246.2013.01046>.
- [2] Popov, I.; Koschorrek, P.; Haghani, A.; Jeinsch, T. Adaptive Kalman Filtering for Dynamic Positioning of Marine Vessels. *IFAC-PapersOnLine* **2017**, *50* (1), 1121–1126. <https://doi.org/10.1016/j.ifacol.2017.08.394>.
- [3] Zhao, Y. Performance Evaluation of Cubature Kalman Filter in a GPS/IMU Tightly-Coupled Navigation System. *Signal Processing* **2016**, *119*, 67–79. <https://doi.org/10.1016/j.sigpro.2015.07.014>.
- [4] Allotta, B.; Caiti, A.; Costanzi, R.; Fanelli, F.; Fenucci, D.; Meli, E.; Ridolfi, A. A New AUV Navigation System Exploiting Unscented Kalman Filter. *Ocean Eng.* **2016**, *113*, 121–132. <https://doi.org/10.1016/j.oceaneng.2015.12.058>.
- [5] Khan, N.; Bacha, S. A.; Khan, S. A. Improvement of Compensated Closed-Loop Kalman Filtering Using Autoregressive Moving Average Model. *Measurement* **2019**, *134*, 266–279. <https://doi.org/10.1016/j.measurement.2018.10.063>.
- [6] Yuan, J.; Wang, Y.; Ji, Z. A Differentially Private Square Root Unscented Kalman Filter for Protecting Process Parameters in ICPSs. *ISA Trans.* **2020**, *104*, 44–52. <https://doi.org/10.1016/j.isatra.2019.12.010>.
- [7] Hamiche, K.; Abouaïssa, H.; Goncalves, G.; Hsu, T. A Robust and Easy Approach for Demand Forecasting in Supply Chains. *IFAC-PapersOnLine* **2018**, *51* (11), 1732–1737. <https://doi.org/10.1016/j.ifacol.2018.08.206>.
- [8] Baradaran Khalkhali, M.; Vahedian, A.; Sadoghi Yazdi, H. Vehicle Tracking with Kalman Filter Using Online Situation Assessment. *Rob. Auton. Syst.* **2020**, *131*, 103596. <https://doi.org/10.1016/j.robot.2020.103596>.
- [9] Farahi, F.; Yazdi, H. S. Probabilistic Kalman Filter for Moving Object Tracking. *Signal Process. Image Commun.* **2020**, *82* (December 2019), 115751. <https://doi.org/10.1016/j.image.2019.115751>.
- [10] Piovoso, M.; Laplante, P. A. Kalman Filter Recipes for Real-Time Image Processing. *Real-Time Imaging* **2003**, *9* (6), 433–439. <https://doi.org/10.1016/j.rti.2003.09.005>.
- [11] Hamuda, E.; Mc Ginley, B.; Glavin, M.; Jones, E. Improved Image Processing-Based Crop Detection Using Kalman Filtering and the Hungarian Algorithm. *Comput. Electron. Agric.* **2018**, *148* (February), 37–44. <https://doi.org/10.1016/j.compag.2018.02.027>.
- [12] Wang, L.; Loffeld, O.; Ma, K.; Qian, Y. Sparse ISAR Imaging Using a Greedy Kalman Filtering Approach. *Signal Processing* **2017**, *138*, 1–10. <https://doi.org/10.1016/j.sigpro.2017.03.002>.
- [13] Predictor, K. On-Line Temperature Estimation for Noisy Thermal Sensors Using a Smoothing Filter-Based. **2018**. <https://doi.org/10.3390/s18020433>.
- [14] Shrivastava, P.; Soon, T. K.; Idris, M. Y. I. Bin; Mekhilef, S. Overview of Model-Based Online State-of-Charge Estimation Using Kalman Filter Family for Lithium-Ion Batteries. *Renew. Sustain. Energy Rev.* **2019**, *113* (December 2018), 109233. <https://doi.org/10.1016/j.rser.2019.06.040>.

- [15] Linghu, J.; Kang, L.; Liu, M.; Luo, X.; Feng, Y.; Lu, C. Estimation for State-of-Charge of Lithium-Ion Battery Based on an Adaptive High-Degree Cubature Kalman Filter. *Energy* **2019**, 189 (xxxx), 116204. <https://doi.org/10.1016/j.energy.2019.116204>.
- [16] Zhang, S.; Guo, X.; Zhang, X. An Improved Adaptive Unscented Kalman Filtering for State of Charge Online Estimation of Lithium-Ion Battery. *J. Energy Storage* **2020**, 32 (September), 101980. <https://doi.org/10.1016/j.est.2020.101980>.
- [17] Sassi, H. Ben; Errahimi, F.; Es-sbai, N. State of Charge Estimation by Multi-Innovation Unscented Kalman Filter for Vehicular Applications. *J. Energy Storage* **2020**, 32 (October), 101978. <https://doi.org/10.1016/j.est.2020.101978>.
- [18] Wang, H.; Lei, T.; Rong, Y.; Shao, W.; Huang, Y. Arc Length Stable Method of GTAW Based on Adaptive Kalman Filter. *J. Manuf. Process.* **2020**, No. December 2019, 0–1. <https://doi.org/10.1016/j.jmapro.2020.01.029>.
- [19] Holtz, J. Sensorless Control of Induction Machines – with or without Signal Injection ? **2019**, No. July. <https://doi.org/10.1109/TIE.2005.862324>.
- [20] Ameid, T.; Menacer, A.; Talhaoui, H.; Harzelli, I. Rotor Resistance Estimation Using Extended Kalman Filter and Spectral Analysis for Rotor Bar Fault Diagnosis of Sensorless Vector Control Induction Motor. *Meas. J. Int. Meas. Confed.* **2017**, 111, 243–259. <https://doi.org/10.1016/j.measurement.2017.07.039>.
- [21] Chen, Z.; Wang, L.; Liu, X. *Sensorless Direct Torque Control of PMSM Using Unscented Kalman Filter*; IFAC, 2011; Vol. 44. <https://doi.org/10.3182/20110828-6-IT-1002.02515>.
- [22] Nilsson, T.; Soja, B.; Karbon, M.; Heinkelmann, R.; Schuh, H. Application of Kalman Filtering in VLBI Data Analysis. *Earth, Planets Sp.* **2015**. <https://doi.org/10.1186/s40623-015-0307-y>.
- [23] Karbon, M.; Soja, B.; Nilsson, T.; Deng, Z.; Heinkelmann, R.; Schuh, H. Earth Orientation Parameters from VLBI Determined with a Kalman Filter. *Geod. Geodyn.* **2017**, 8 (6), 396–407. <https://doi.org/10.1016/j.geog.2017.05.006>.
- [24] Teh, L. A. and J. Kalman Filter for Reducing Total Harmonics Distortion in Stand-Alone PV System. *2020 Glob. Congr. Electr. Eng. (GC- ElecEng)* **2020**, 81–87. <https://doi.org/10.23919/GC-ElecEng48342.2020.9286275.Abstract>.
- [25] Docimo, D. J.; Ghanaatpishe, M.; Mamun, A. Extended Kalman Filtering to Estimate Temperature and Irradiation for Maximum Power Point Tracking of a Photovoltaic Module. *Energy* **2017**, 120, 47–57. <https://doi.org/10.1016/j.energy.2016.12.089>.
- [26] Monteiro, R. V. A.; Guimarães, G. C.; Moura, F. A. M.; Albertini, M. R. M. C.; Albertini, M. K. Estimating Photovoltaic Power Generation: Performance Analysis of Artificial Neural Networks, Support Vector Machine and Kalman Filter. *Electr. Power Syst. Res.* **2017**, 143, 643–656. <https://doi.org/10.1016/j.epsr.2016.10.050>.
- [27] Madhukar, P. S. S. M. Overview. **2020**, No. Icosec, 1268–1272.
- [28] Belkhatir, Z.; Mechhoud, S.; Laleg-Kirati, T. M. Kalman Filter Based Estimation Algorithm for the Characterization of the Spatiotemporal Hemodynamic Response in the Brain. *Control Eng. Pract.* **2019**, 89 (May), 180–189. <https://doi.org/10.1016/j.conengprac.2019.05.017>.
- [29] Alguliyev, R.; Imamverdiyev, Y.; Sukhostat, L. Cyber-Physical Systems

- and Their Security Issues. *Comput. Ind.* **2018**, *100* (July 2017), 212–223. <https://doi.org/10.1016/j.compind.2018.04.017>.
- [30] Wang, J.; Luo, J.; Liu, X.; Li, Y.; Liu, S.; Zhu, R. Improved Kalman Filter Based Differentially Private Streaming Data Release in Cognitive Computing. *Futur. Gener. Comput. Syst.* **2019**, *98*, 541–549. <https://doi.org/10.1016/j.future.2019.03.050>.
- [31] Zhang, Y.; Wang, R.; Li, S.; Qi, S. Temperature Sensor Denoising Algorithm Based on Curve Fitting and Compound Kalman Filtering. *Sensors (Switzerland)* **2020**, *20* (7), 1–13. <https://doi.org/10.3390/s20071959>.
- [32] Mouzinho, L. F.; Fonsecaneto, J. V.; Luciano, B. A.; Freire, R. C. S. Indirect Measurement of the Temperature via Kalman Filter. *18th IMEKO World Congr. 2006 Metrol. a Sustain. Dev.* **2006**, *1*, 818–823.
- [33] Ma, Y.; Cui, Y.; Mou, H.; Gao, J.; Chen, H. Core Temperature Estimation of Lithium-Ion Battery for EVs Using Kalman Filter. *Appl. Therm. Eng.* **2020**, *168* (April 2019), 114816. <https://doi.org/10.1016/j.applthermaleng.2019.114816>.
- [34] Eleffendi, M. A.; Johnson, C. M. Application of Kalman Filter to Estimate Junction Temperature in IGBT Power Modules. *IEEE Trans. Power Electron.* **2016**, *31* (2), 1576–1587. <https://doi.org/10.1109/TPEL.2015.2418711>.
- [35] Wiener, N. *The Extrapolation, Interpolation and Smoothing of Stationary Time Series with Engineering Applications*; John Wiley & Sons: New York, 1949.
- [36] Kalman, R. E. A New Approach to Linear Filtering and Prediction Problems. *J. Fluids Eng. Trans. ASME* **1960**, *82* (1), 35–45. <https://doi.org/10.1115/1.3662552>.
- [37] Tewari, A. Modern Control Design With MATLAB and SIMULINK. **2002**, 518.
- [38] Lacey, T. Tutorial : The Kalman Filter. 133–140.
- [39] Grewal, M. S.; Andrews, A. P. *Kalman Filtering: Theory and Practice Using MATLAB®: Third Edition*; 2008. <https://doi.org/10.1002/9780470377819>.
- [40] Wang, L. *Model Predictive Control System Design and Implementation Using MATLAB*; 2009; Vol. 53.
- [41] Lu, L. Z.; Lin, W. W. An Iterative Algorithm for the Solution of the Discrete-Time Algebraic Riccati Equation. *Linear Algebra Appl.* **1993**, *188–189* (C), 465–488. [https://doi.org/10.1016/0024-3795\(93\)90476-5](https://doi.org/10.1016/0024-3795(93)90476-5).
- [42] Laudadio, T.; Mastronardi, N.; Van Dooren, P. The Generalized Schur Algorithm and Some Applications. *Axioms* **2018**, *7* (4), 1–18. <https://doi.org/10.3390/axioms7040081>.
- [43] DATTA, B. N. Numerical Solutions and Conditioning of Algebraic Riccati Equations. *Numer. Methods Linear Control Syst.* **2004**, 519–599. <https://doi.org/10.1016/b978-012203590-6/50017-3>.
- [44] Chauhan, J.; Neelakantan, U. An Experimental Approach for Precise Temperature Measurement Using Platinum RTD PT1000. *Int. Conf. Electr. Electron. Optim. Tech. ICEEOT 2016* **2016**, 3213–3215. <https://doi.org/10.1109/ICEEOT.2016.7755297>.
- [45] Trung, T. Q.; Ramasundaram, S.; Hwang, B. U.; Lee, N. E. An All-Elastomeric Transparent and Stretchable Temperature Sensor for Body-Attachable Wearable Electronics. *Adv. Mater.* **2016**, *28* (3), 502–509. <https://doi.org/10.1002/adma.201504441>.

- [46] Chen, Y.; Lu, B.; Chen, Y.; Feng, X. Breathable and Stretchable Temperature Sensors Inspired by Skin. *Sci. Rep.* **2015**, *5*, 1–11. <https://doi.org/10.1038/srep11505>.
- [47] Wang, Z.; Gao, W.; Zhang, Q.; Zheng, K.; Xu, J.; Xu, W.; Shang, E.; Jiang, J.; Zhang, J.; Liu, Y. 3D-Printed Graphene/Polydimethylsiloxane Composites for Stretchable and Strain-Insensitive Temperature Sensors. *ACS Appl. Mater. Interfaces* **2019**, *11* (1), 1344–1352. <https://doi.org/10.1021/acsami.8b16139>.
- [48] Kim, J.; Kim, J.; Shin, Y.; Yoon, Y. A Study on the Fabrication of an RTD (Resistance Temperature Detector) by Using Pt Thin Film. *Korean J. Chem. Eng.* **2001**, *18* (1), 61–66. <https://doi.org/10.1007/BF02707199>.
- [49] Noh, J.; Park, S.; Boo, H.; Kim, H. C.; Chung, T. D. Nanoporous Platinum Solid-State Reference Electrode with Layer-by-Layer Polyelectrolyte Junction for PH Sensing Chip. *Lab Chip* **2011**, *11* (4), 664–671. <https://doi.org/10.1039/c0lc00293c>.
- [50] Hassan, A. S.; Juliet, V.; Joshua Amrith Raj, C. MEMS Based Humidity Sensor with Integration of Temperature Sensor. *Mater. Today Proc.* **2018**, *5* (4), 10728–10737. <https://doi.org/10.1016/j.matpr.2017.12.356>.
- [51] Imran, M.; Bhattacharyya, A. Thermal Response of an On-Chip Assembly of RTD Heaters, Sputtered Sample and Microthermocouples. *Sensors Actuators, A Phys.* **2005**, *121* (2), 306–320. <https://doi.org/10.1016/j.sna.2005.02.019>.
- [52] Kang, L.; Shi, Y.; Zhang, J.; Huang, C.; Zhang, N.; He, Y.; Li, W.; Wang, C.; Wu, X.; Zhou, X. A Flexible Resistive Temperature Detector (RTD) Based on in-Situ Growth of Patterned Ag Film on Polyimide without Lithography. *Microelectron. Eng.* **2019**, *216* (July), 111052. <https://doi.org/10.1016/j.mee.2019.111052>.
- [53] Cui, J.; Liu, H.; Li, X.; Jiang, S.; Zhang, B.; Song, Y.; Zhang, W. Fabrication and Characterization of Nickel Thin Film as Resistance Temperature Detector. *Vacuum* **2020**, *176*, 109288. <https://doi.org/10.1016/j.vacuum.2020.109288>.
- [54] Lee, Y.; Cheng, S.; Fang, W. MONOLITHIC INTEGRATED CMOS-MEMS FLUORESCENCE QUENCHING GAS SENSOR AND RESISTIVE TEMPERATURE DETECTOR (RTD) FOR TEMPERATURE COMPENSATION. *2019 20th Int. Conf. Solid-State Sensors, Actuators Microsystems Eurosensors XXXIII (TRANSDUCERS EUROSensors XXXIII)* **2019**, No. June, 1293–1296.
- [55] Mathworks. Kalman filtering.
- [56] Haidekker, M. A. Solving Differential Equations in the Laplace Domain. *Linear Feed. Control.* **2013**, 27–56. <https://doi.org/10.1016/b978-0-12-405875-0.00003-6>.
- [57] Gadsden, S. A.; Eng, B. M. Smooth Variable Structure Filtering: Theory and Applications. *Thesis* **2011**.