MODEL BASED ESTIMATION OF BIOLOGICAL HEAT GENERATION DURING COLD-CHAIN TRANSPORT AND PROCESSING

R. JEDERMANN[(a,b)], W. LANG[(a,b,c)]
[(a)] Institute for Microsensors, –actuators and –systems (IMSAS), University of Bremen, FB1
[(b)] Microsystems Center Bremen (MCB)
Otto-Hahn-Allee, Build. NW 1, D-28359 Bremen, Germany [(a,b)]
Fax: +49 421 218 98 62603, rjedermann@imsas.uni-bremen.de
[(c)] Bremen Research Cluster for Dynamics in Logistics (LogDynamics)
Hochschulring 20, D-28359 Bremen, Germany
Fax: +49 421 218 50154, wlang@imsas.uni-bremen.de

ABSTRACT

Biological processes continue after harvest in most of the fresh fruits. The generated heat, created by such climacteric fruits, must be removed by a cooling unit, but the amount of heat is also a valuable indicator for the current state of the product. In this paper we will present an approach, how the generated heat can be estimated from the measured temperature over time curve as additional – time varying – process state by the recursive and computational efficient algorithm of the Kalman filter. The application of the Kalman filter required a special adaptation. The parameter for heat removal has to be estimated by system identification techniques in the first step. The resulting time-continuous model has to be translated to a discrete state-space description of the process. Noise covariance matrices had to be defined. The required algorithm was implemented as JAVA code on a processing unit mounted directly in our prototype 'Intelligent Container'. In an application example we showed how the ripening process of bananas can be supervised by the suggested system.

1. MOTIVATION AND SCENARIO

Fresh foods and vegetables have arrived at a tremendous market share. Seaborne perishable reefer cargo has increased to 92.4 million tonnes in 2012 (Drewry Shipping Consultants, 2013). In order to guarantee the quality of the products and to detect problems as early as possible, it is necessary to monitor the temperature. Temperature data can be read 'offline' from data loggers packed with the product or from sensors connected to the cooling unit after arrival of the container. New technologies allow transmitting temperature measurements 'online' during transportation (Jedermann et al., 2013). In most cases, the provided data are only checked for maximum or average temperature. But temperature data can tell much more than just threshold checking. In this paper we firstly show, by an example scenario for the transport and ripening of bananas, how a detailed analysis of temperature data can look like. Secondly we demonstrated that this analysis in not only feasible by elaborated mathematical tools on powerful desktop PCs, but also on embedded systems that can be integrated into the container. This so called 'Intelligent Container' can automatically analyse measurement data and generate warning messages, if a problem is detected.

The mathematical models behind our analysis are, of course, very specific to bananas. But our intention is to present our application as a template to motivate the reader to create own mechanism for automated evaluation of temperature data.

1.1. The banana cold chain

Bananas have two distinguishing characteristic features compared to most other fruits. The first one is the fact that most parts of the ripening process with conversion from starch to sugar takes place after harvest. This leads to the second feature, their high respiration activity. The conversion creates large amounts of CO_2 and thermal energy. The heat production can vary over a wide range from 20 W/ton for green bananas at optimal storage conditions (13°C, CO_2=5%, O_2<5%) to 800 W/ton (Kerbel 1986) for yellow bananas at 20°C in normal atmosphere.

In our field studies, the bananas were packed to pallets and stowed into our test container in Costa Rica with a temperature of about 25°C. During transport they were cooled to about 14°C. The container was shipped to Antwerp (2 weeks) and then trucked to Germany (1 day), where ripening was initiated by ethylene treatment (1 day). After 5 days of ripening, the pallets were handed over to a wholesale trader.

Banana packing to boxes and stowage of pallets in the container is optimized to make best use of the available space. Due to the dense packing, only a reduced share of the cooling air from the reefer unit actually arrives at the bananas inside the boxes. The airflow might be completely blocked by packing mistakes. The high respiration activity can lead to a hot spot, in which the amount of generated heat is larger than the heat that is removed by the airflow.

1.2. Modelling approach

The temperature should be monitored permanently, in order to detect such risks as early as possible. However, a simple threshold checking does not make sense because of the temperature change from 25° to 14°C during cooling. In our approach we approximate the temperature curve by a parameterised model. Threshold checking is applied on model parameters, not on the actual temperature values.

The different behaviour of green and yellow bananas made it necessary to apply two variants of our model. The first variant uses time constant parameters to describe temperature changes of green bananas during transport. The second variant has to take the varying respiration activity during ripening into account and describe it by an unknown time varying function.

The first model for transportation has been described in an earlier publication (Jedermann et al., 2013) in detail and will only be shortly summarized in section 2. The main focus of this paper is to extend this model for estimation of the time varying respiration activity during ripening (section 3).

The required model parameters, which have to be estimated individually for each sensor or pallet, respectively, are summarized in table 1. During transport only two parameters or proportional factors are required: k_M describes the cooling effect per box or pallet. It is used to verify cooling and correct packaging and stowage. k_P scales the respiration activity, which is reduced by controlled atmosphere conditions (3% CO_2 and 2-5% O_2). A high value indicates pre-mature start of the ripening process. After ethylene treatment, k_P has to be replaced by the time function *r(t)*.

Table 1. Properties for remote monitoring of cooling in the banana supply chain

Parameter	Unit	Type	Related to	Description	Warning conditions
k_M	-	Constant	Transport, Ripening	Factor for cooling effect on box/pallet	Low k_M → poor cooling by packing / stowage mistakes
k_P	°C/hour	Constant	Transport	Factor for respiration heat	High k_P → pre-mature ripening
r(t)	W/ton	Time function	Ripening	Respiration heat as time function	Low *r(t)* → Ripening not fully started High *r(t)* → Increased cooling required

1.3. Hardware platform

Our intelligent container (Jedermann et al., 2014) consists of a set of wireless sensors based on the TelosB platform (Crossbow Technology Inc., California). Additional measurements were taken by iButton data loggers (DS1922L, Maxim Integrated, California). The measurements from the wireless sensors were collected and processed by an embedded computer (VTC6100 from Nexcom, UK with Intel Atom N270 single core processor @ 1.6GHz), the so called Freight Supervision Unit (FSU). The FSU also provides an interface to read out and adjust the set point of the cooling unit. Either full measurement data or only warning messages on critical parameters values can be sent by a telematics unit (Iridium based by OHB, Germany) over a satellite network to a remote server. If the container is ashore, the external communication can be switched to more cost efficient cellular / GSM networks.

2. AUTOMATED TEMPERATURE DATA EVALUATION DURING TRANSPORT

After initial tests with a general second order model, we found the model in figure 1 most useful to approximate the measured temperature curves and to describe them with a minimal set of parameters. The model predicts changes of the temperature in the centre of a banana box *y(t)* with the measured supply temperature over time function *u(t)* as input. The model structure in figure 1 has to major advantages a) it enables to estimate index values for the generated and the removed heat as two separate parameters, and b) it also describes the non-linear relation between respiration heat and temperature.

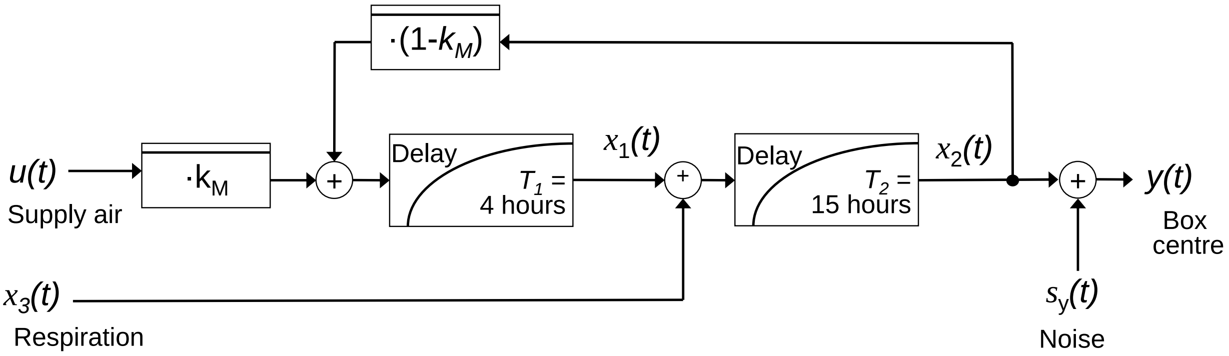

Figure 1. Block diagram of dynamic system model for the temperature *y(t)* in box centre.

The temperature change caused by respiration activity $x_3(t)$ depends on the current biological state of the bananas (k_P), but also increases exponentially with the current temperature according to eq. (1) with $Q_{10} = 3$.

$$x_3(t) = k_P \cdot e^{\ln(Q10)\cdot(y(t)-13°\mathrm{C})/10°\mathrm{C}} \tag{1}$$

The model basically consists of two delay elements with time constants of 4 and 15 hours. The constants T_1, T_2 and Q_{10} were set according to earlier tests in order to enable adaptation of the model to a wide range of fast and slow cooling curves (Jedermann et al., 2013). The fact that 3 out of 5 model parameters can be considered as constant is a further advantage of the specific model structure in figure 1 compared with a general second order dynamic model.

The 2 variable parameters k_M and k_P are estimated with help of the Matlab System Identification Toolbox (The MathWorks, Inc., Massachusetts). $x_1(t)$ and $x_2(t)$ are the internal states of the model. The influence of noise, which creates a difference between the model state $x_2(t)$ and the observed output value *y(t)* is discussed in more detail in the next section, see eq. (4).

Figure 2a shows the measured temperature curves during one transport in April 2013. The sensors B144 (centre of pallet) and B138 (close to gap between pallets) were selected from the data set to represent typical curves for slow and fast cooling. The model provided a good approximation with a root-mean-square error (RMSE) below 0.1 K (Kelvin, average RMSE of 168 sensors/model instances = 0.044 ± 0.017 K during two separate test transports). The identified model parameters are given in the table 2. The respiration activity was re-calculated based on the model and converted to the unit W/ton (figure 2b) according to eq. (2).

$$r(t) = c_R \cdot x_3(t) \qquad c_R = 930.5\frac{\mathrm{W}\cdot\mathrm{s}}{\mathrm{K}\cdot\mathrm{ton}} \tag{2}$$

At the end of the transport *r(t)* arrives at values between 15 and 30 W/ton.

The model in figure 1 describes a non-linear system for prediction of the box temperature as output signal, due to the exponential function. However, for the purpose of parameter identification, both system input and output are known in advance. The model equations can be re-arranged to include only linear dependency from the unknown parameters k_m and k_p.

The term $e^{\ln(Q10)\cdot(y(t)-13°\mathrm{C})/10°\mathrm{C}}$ can be calculated in advance for constant Q_{10} and thus making eq. (1) linear. The input of the first delay element can be re-arranged to

$$g(t) = y(t) + k_m \cdot \big(u(t) - y(t)\big). \tag{3}$$

The two parameters k_m and k_p both appear only once in the equations. If T_1 and T_2 are considered as constant, the task of system identification is simplified to a linear problem. Our implementation of the FSU solves this problem by an incremental algorithm requiring only the update of a 4 by 4 matrix after each measurement.

3. AUTOMATED TEMPERATURE DATA EVALUATION DURING RIPENING

If the biological state of the bananas changes over time, as it is the case after ethylene treatment, the respiration activity $x_3(t)$ can no longer be calculated by eq. (1) with a constant parameter k_P. The more general case of estimating $x_3(t)$ as arbitrary time function needs a more elaborated approach, which is described for the first time in this paper.

The Kalman filter has been used for estimation of system states since several decades. It is predestined for embedded applications because of its simple recursive formulation. Current applications include navigations systems for pedestrians by outdoor GPS and an RFID based location system for indoor use (Kourogi et al., 2006), and measurement of performance parameters of aircraft engines (Kobayashi et al., 2005).

The general idea of the Kalman filter can be illustrated by a simple example. A pendulum has two states: its position and velocity. If only the position can be directly observed, the velocity can be calculated as differentiation of position. This simple, straightforward approach has the disadvantage of being very sensitive towards noise. The Kalman filter avoids numerical differentiation and thus provides more accurate estimation. Instead, the Kalman filter calculates the system states by simulation of the system model based on known inputs such as forces affecting the pendulum. The simulation is corrected according to the residual between predicted and observed output signals, e.g. the position. The residual is multiplied with a Kalman gain, depending on the statistical properties of system noise. The Kalman filter is mathematically derived from the condition that the squared error between estimated and actual system states becomes minimal.

Because of limited space, we will give only a short summary of the mathematical formulation of the Kalman filter in the following. A more detailed introduction can be found in standard textbooks, e.g. Brown and Hwang, 2012, or internet tutorials, e.g. Welch and Bishop, 2006.

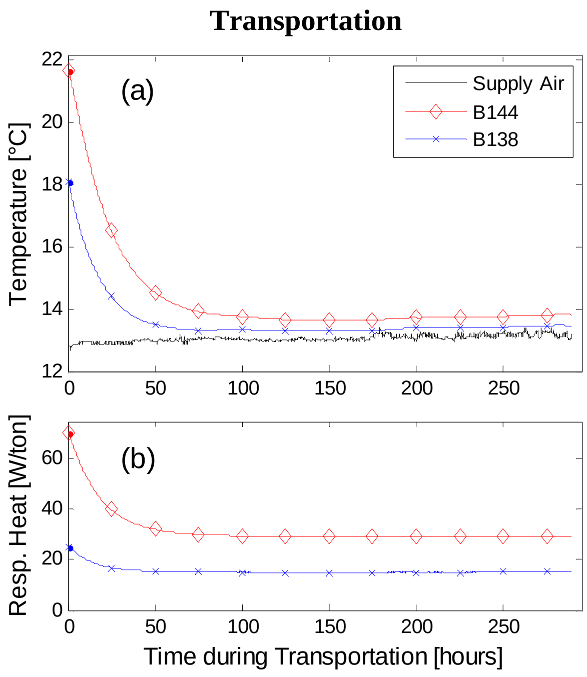

Figure 2. Measured temperature curves during transportation (a) and calculated respiration heat (b) for two example sensor positions B144 and B138.

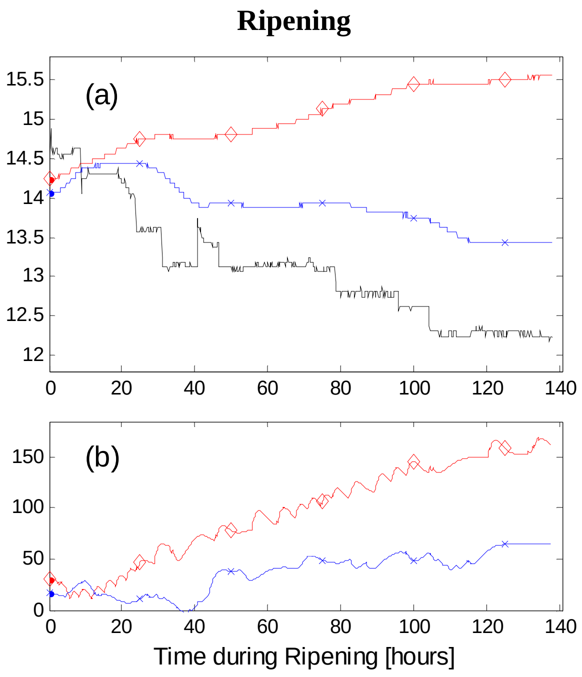

Figure 3. Measured temperature during ripening (a) and calculated respiration heat (b) according to Kalman filter.

Table 2. Identified k_M and k_P parameters, respiration activity at the end of transport/ripening and the root-mean-square-error between measured values and model for the curves in figure 2a and 3a.

Sensor Number	k_M	k_P	RMSE Model	Resp. Heat Transport	Resp. Heat Ripening	RMSE Kalman
B144	0.768	0.0292 K/h	0.033 K	29.5 W/ton	159.3 W/ton	0.032 K
B138	0.904	0.0157 K/h	0.042 K	15.3 W/ton	64.3 W/ton	0.026 K

3.1. General state space system description

A time continuous system can be described in the state space representation in eq. (4), with **x**(t) as state vector, **A** as system matrix, **B** as input and **C** as output matrix. For simplicity we consider only systems with a single input and output. However, eq. (4) can be extended to multi input or output systems by replacing the scalars *u(t)* and *y(t)* with vector functions. Matrix **Q** describes amplitude of process noise that directly affects the system states. If this effect is statistically independent for different system states, only the elements on the diagonal of **Q** have to be set. **R** describes the measurement noise at the output of the system. For a single-output system **R** becomes a scalar. $\mathbf{s}_x(t)$ and $\mathbf{s}_y(t)$ are two vector random variables with a unit normal distribution (σ =1). Matrices that are specific to the time-continuous model are marked with a single apostrophe.

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}' \cdot \mathbf{x}(t) + \mathbf{B}' \cdot u(t) + \mathbf{Q} \cdot \mathbf{s}_{\mathbf{x}}(t) \\ y &= \mathbf{C} \cdot \mathbf{x}(t) + \mathbf{R} \cdot \mathbf{s}_y(t) \end{aligned} \tag{4}$$

3.2. Discretisation

In general, the time discrete domain is preferred for numerical solution of the system equations. But in our work we decided to use the time continuous domain for system identification as shown in section 2 with the advantage that it directly estimates meaningful parameters for generated and removed heat. However, the Kalman filter is easier to solve in the time discrete form. Before application of the Kalman filter, the system matrices have to be translated to their time discrete equivalent marked with a double apostrophe. This process is called discretisation. Matlab provides powerful functions for this task such as ‘c2d’ in the ‘Control System Toolbox’. A general description of the underlying algorithm can be found in Middleton and Goodwin (1990). Because Matlab is rather impractical for implementation on an embedded data evaluation unit, the discretisation algorithm in eq. (5) has to be translated to a standard programming language.

$$\mathbf{A}'' = e^{\mathbf{A}' \cdot T_S} \qquad \mathbf{B}'' = \mathbf{A}'^{-1} \cdot \left(\mathbf{A}'' - \mathbf{I}\right)\mathbf{B}' \tag{5}$$

There are several ways to calculate the matrix exponential (Moler and Van Loan, 1978) as required for eq. (5). We decided to use the Taylor series approach because of its simple implementation, although it was criticized by Moler and Van Loan (1978) for its computational inefficiency. However, the computation time on our target system was still acceptable.

3.3 Application of the Kalman filter

After having translated the system description to a discrete form, the system states can be estimated by an iterative process. The equation set (6) has to be updated after each measurement interval. The two equations on the left formulate the predictor for the estimated system state and the covariance of the estimation error **P**. The 3 equations on the right formulate the correction of the Kalman filter after each step, for the Kalman gain **K**, the estimated system state and the error covariance matrix. The superscript ‘minus’ marks matrices before correction, which is changed to ‘plus’ after correction. A matrix inversion is only necessary for multi-output systems; otherwise it is reduced to a scalar division. This recursive formulation makes the Kalman filter computationally very efficient. For a detailed description see Brown and Hwang, 2012.

$$\begin{aligned} \hat{\mathbf{x}}_k^- &= \mathbf{A}'' \cdot \hat{\mathbf{x}}_{k-1}^+ + \mathbf{B}'' \cdot u_{k-1} \\ \mathbf{P}_k^- &= \mathbf{A}'' \cdot \mathbf{P}_{k-1}^+ \cdot \mathbf{A}''^T + \mathbf{Q} \end{aligned} \qquad \begin{aligned} \mathbf{K}_k &= \mathbf{P}_k^- \cdot \mathbf{C}^T \cdot \left(\mathbf{C} \cdot \mathbf{P}_k^- \cdot \mathbf{C}^T + \mathbf{R}\right)^{-1} \\ \hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k \cdot \left(y_k - \mathbf{C} \cdot \hat{\mathbf{x}}_k^-\right) \\ \mathbf{P}_k^+ &= \left(\mathbf{I} - \mathbf{K}_k \cdot \mathbf{C}\right) \cdot \mathbf{P}_k^- \end{aligned} \tag{6}$$

3.4. System description for the case of ripening

The Kalman filter only estimates system states, not inputs. The current respiration activity is related to the concentration of active enzymes converting starch to sugar. Because enzyme concentrations change only slowly over time, we can consider the respiration activity as an additional system state. The speed of the underlying biological processes depends on several – not measured – influence factors such as O_2, CO_2 and ethylene gas concentrations. Therefore it is hardly possible to calculate respiration activity by a biological model under the conditions of container transportation. Instead, we assumed that changes of the respiration activity over time $\dot{x}_3(t)$ is given by the random variable $\mathbf{s}_{x,3}$.

The simplified model described in section 2 is valid during sea and road transportation (phase 1) until ethylene treatment starts. At this point of time we switch to the extended model with an additional state variable and start the iteration of the Kalman filter (phase 2). The matrices for the time continuous description of the resulting system were set according to the model given in figure 1. Both delay elements with input $u_n(t)$ were translated to a differential equation according to eq. (7). The 3 differential equations were combined in the matrix form according to eq. (4). For k_M, the value was used that was estimated during transportation.

$$\dot{x}_n(t) = -\frac{x_n(t)}{T_n} + \frac{u_n(t)}{T_n} \tag{7}$$

$$\mathbf{A}' = \begin{bmatrix} \frac{-1}{T_1} & \frac{1-k_M}{T_1} & 0 \\ \frac{1}{T_2} & \frac{-1}{T_2} & 1 \\ 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{B}' = \begin{bmatrix} \frac{k_M}{T_1} \\ 0 \\ 0 \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \tag{8}$$

3.5 Initialisation of the Kalman filter

The initialisation of the system states and the noise covariance matrices is the trickiest task during the application of the Kalman filter. A useful setting of the initial values of the system state vector $\mathbf{x}_0$ and of the error covariance matrix $\mathbf{P}_0$ speeds up the conversion process. The initial system state $\mathbf{x}_0$ of phase 2 can be calculated according the model of phase 1 by eq. (9) under the assumption that the cooling process has completed and both $u(t)$ and $y(t)$ have arrived at a constant value.

$$\mathbf{x}_0 = \begin{bmatrix} k_M \cdot u_0 + (1-k_M) \cdot y_0 \\ y_0 \\ k_p \cdot e^{\ln(Q10)\cdot(y_0 - 13°C)/10°C} \end{bmatrix} \tag{9}$$

The noise of the observed discrete output signal y_k depends basically on quantisation effects caused by the limited resolution of the applied sensor of 1/16 K (iButton, Maxim Integrated). The covariance matrix $\mathbf{R}$ for the output noise is set to the variance of the quantisation error v_s.

$$v_s = \left(0.25 \cdot \frac{1}{16}\mathrm{K}\right)^2 = \frac{1}{4096}\mathrm{K}^2 \qquad \mathbf{R} = v_S \tag{10}$$

The measured discrete input signal u_k is disturbed by quantisation of the same amplitude. The noise of u_k has a direct effect on the first system state $\mathbf{x}_{1\mathrm{k}}$ in relation to first element of the input matrix **B”**. The change of respiration activity $\mathbf{x}_{3\mathrm{k}}$ over time is described as a random process. We assume that the average change of $\mathbf{x}_{3\mathrm{k}}$ per sample interval of 10 minutes is equivalent to 3% of its initial value $\mathbf{x}_{30}$ before ethylene treatment, leading to a variance v_r given in eq. (11). The elements of the covariance matrix of the process noise were set accordingly. The initial error of the estimation $\mathbf{P}_0$ for the first two system states was set in relation to the quantisation noise v_s. We assume a smaller error than v_r for the third state, since the initial value for $\mathbf{x}_{30}$ was directly calculated by eq. (9).

$$v_R = \left(0.03 \cdot x_{3_0}\right)^2 \qquad \mathbf{Q} = \begin{bmatrix} B_1'' \cdot v_S & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & v_R \end{bmatrix} \qquad \mathbf{P}_0^+ = \begin{bmatrix} v_S & 0 & 0 \\ 0 & v_S & 0 \\ 0 & 0 & 0.25 \cdot v_R \end{bmatrix} \tag{11}$$

Although the setting of the matrices **Q**, **R** and $\mathbf{P_0}$ has some inaccuracies in our case, the Kalman filter turned out to provide a robust and fast iteration. In practice, the noise covariance matrices **Q** and **R** are often used to 'tune' the Kalman filter (Welch and Bishop, 2006). Higher values of v_r lead to faster iteration, but also to a more noisy estimation of $\mathbf{x}_3$.

3.6. Application of the continuous to discrete transformation

Because k_m is estimated anew for each transport and sensor, the discretisation of the system matrix cannot be done in advance. We used the Taylor series approximation for the matrix exponential in our implementation and stopped after the element $\mathbf{A}'^5$. The inversion of the system matrix $\mathbf{A}'$ as required for eq. (5) can be simplified. In this special case, it is only necessary to invert the 2 by 2 sub-matrix in top-left position because the system state $\mathbf{x}_3$ depends neither on other states nor on the input signal *u(t)*. A comparison of the calculated matrices **A**" and **B**" for our implementation for the continues to discrete transformation and the original *c2d* Matlab function showed only a very small error of maximum 14×10^{-12} per matrix element.

3.7 Application example

During 3 field tests, bananas were ripened inside our test container. Figure 3a shows two example temperature curves measured during ripening. The set point of the container was stepwise reduced to compensate for the increasing respiration activity over a remote interface. The generated heat in figure 3b was calculated by the Kalman filter. The calculation is based on the assumptions that a) the heat removal by cooling can be modelled by a second order system ─ either in general form or by the structure in figure 1 ─ and b) the related model parameters can be estimated separately from the respiration heat by the temperature curve of the cooling process during transportation. For verification of the implementation of the Kalman filter, we simulated the model in figure 1 by setting the input $x_3(t)$ to the estimated respiration activity. The measured and predicated system output showed only a small deviation with an RMSE of about 0.05 K.

At the end of the ripening process, the cooling unit has to be able to remove about 5 times more thermal energy than before ethylene treatment. The temperature curves were still increasing at the end of our test, which means that the biological activity had not arrived at its maximum. An evaluation of skin colour also indicated that the ripening process was not completed. The bananas were removed from the container in a colour stage between 3 (more green than yellow) and 4 (more yellow than green). Full ripeness should only be achieved when the bananas arrive at the retail store, in order to provide maximum freshness for the customers. The maximal respiration activity of 159.3 W/ton in our test is about 42% of the value given by Kerbel (1986) of 376 W/ton at 15°C in normal atmosphere. The lower value can be ascribed to two factors: a) the higher CO_2 concentration of ~4.5% in our test and b) the peak of heat generation had not been reached.

Although the embedded implementation of the Kalman filter had not been finished in time for the field test, we could show in later tests with recorded temperature curves that it is fully feasible to run the algorithm on the Freight supervision unit (FSU) of the container. We selected the programming language Java to write the software implementation. Due to it dynamic nature, Java provides built-in solutions to update software components on the FSU (Dannies et al., 2013), for example, if a new type of fruit requires a specific model.

Additional libraries such as Jama (http://math.nist.gov/javanumerics/jama/) provide high accuracy matrix operations. The Java software bundle required 700 ms to process the data of 827 measurement intervals on an Intel ATOM processor running at 1.6 GHz, which is less than 1 ms per interval. The discretisation and initialisation of the matrices required 19 ms. The software should therefore be suited for smaller µControllers with floating point support, eg. ARM based systems.

4. SUMMARY AND DISCUSSION

We could show by our case study that additional information can be extracted from measured temperature curves during transportation, including parameters about cooling efficiency, respiration activity of fruits and status of the ripening process. For the case that system properties cannot be described by time constant parameters, the Kalman filter is a very useful approach to estimate the required information in the form of time-varying system states. We showed by our application example for the transport of packed bananas, that an embedded implementation of the required algorithms is feasible. An 'Intelligent Container' or even intelligent sensors can autonomously analyse temperature data and thereby provide a more accurate monitoring of fruit transportation and ripening. The translation of our approach to another type of product or packing requires an analysis of the thermal relations between product, packing and cooling for the specific product, but should be worthwhile to undertake the necessary effort.

NOMENCLATURE

Scalar		Unit
t	Time	h
$u(t)$	Supply air temperature / system input	°C
$y(t)$	Box centre temperature / system output	°C
$x_{1,2}(t)$	State of delay elements	°C
$x_3(t)$	Temperature change caused by respiration	$K\cdot h^{-1}$
$r(t)$	Respiration heat	$W\cdot t^{-1}$

Constants / Parameters		Unit
k_M	Cooling efficiency	-
k_P	Proportional factor respiration heat	$°C\cdot h^{-1}$
c_R	Conversion factor for respiration heat	930.5 $Ws\cdot K^{-1}\cdot t^{-1}$
T_1,T_2	Time constants	4 h / 15 h
T_S	Sample interval	0.166 h
n	Number of states	3
v_S	Variance of sensor noise	K^2
v_R	Variance of measurement noise	K^2

Matrix/Vector		Size
x	System state vector	$n\times 1$
A	System matrix	$n\times n$
B	Input matrix	$n\times 1$
C	Output matrix	$1\times n$
K	Kalman gain	$n\times 1$
I	Unity matrix	$n\times n$
$\mathbf{s_x}$	White noise (input)	$n\times 1$
$\mathbf{s_y}$	White noise (output)	1×1

Covariance Matrixes		Size
Q	Process noise	$n\times n$
R	Measurement noise	1×1
P	Estimation error	$n\times n$

Matrix size related to single input – single output system.

Absolute temperature given in °C, relative temperatures in K (Kelvin).

Unit ‘t’ = ton.

Subscripts	
k	Time / iteration step
0	Initial state / before ethylene treatment

Superscripts	
T	Matrix transpose
$'$	Time continuous description
$''$	Time discrete description
-	Before Kalman correction (a priori)
+	After Kalman correction (a posteriori)
-1	Matrix inversion
^	Estimated value

REFERENCES

1. Drewry Shipping Consultants, 2013, Reefer Shipping Market Annual Review and Forecast 2013/14, http://www.drewry.co.uk/news.php?id=247
2. Jedermann R, Geyer M, Praeger U, Lang W, 2013, Sea transport of bananas in containers - Parameter identification for a temperature model. *Journal of Food Engineering* **115**(3), 330-338.
3. Kerbel E, 1986, revised 2004, Banana and Plantain. In *The Commercial Storage of Fruits, Vegetables, and Florist and Nursery Stocks* (eds. Hardenburg RE, Watada AE, Wang CY). www.ba.ars.usda.gov/hb66/contents.html
4. Jedermann R, Lloyd C, Pötsch T, 2014, Communication techniques and challenges for wireless food quality monitoring. *Philosophical Transactions of the Royal Society A*, **372**(2017), 20130304. (doi: 10.1098/rsta.2013.0304).
5. Kourogi M, Sakata N, Okuma T, Kurata T, 2006, Indoor/outdoor pedestrian navigation with an embedded GPS/RFID/self-contained sensor system. In *Advances in Artificial Reality and Tele-Existence* pp. 1310-1321, Springer.
6. Kobayashi T, Simon DL, Litt JS, 2005, Application of a constant gain extended Kalman filter for in-flight estimation of aircraft engine performance parameters. In *ASME Turbo Expo 2005: Power for Land, Sea, and Air* (pp. 617-628, American Society of Mechanical Engineers.
7. Brown RG, Hwang PYC, 2012, *Introduction to random signals and applied Kalman filtering: with Matlab exercises*. 4. ed ed. Hoboken, NJ, Wiley.
8. Welch G, Bishop G, 2006, An Introduction to the Kalman Filter, http://www.cs.unc.edu/~welch/kalman/kalmanIntro.html
9. Middleton RH, Goodwin GC, 1990, *Digital control and estimation: a unified approach*. Englewood Cliffs, Prentice Hall.
10. Moler C, Van Loan C, 1978, Nineteen dubious ways to compute the exponential of a matrix. *SIAM review* **20**(4), 801-836. (doi:10.1137/1020098).
11. Dannies A, Palafox-Albarrán J, Lang W, 2013, Smart dynamic software components enabling decision support in Machine- to -machine networks *IJCSI Int. J. of Computer Science Issues* **10**(1,3), 540-550.