Environmental Modelling and Software 139 (2021) 105031

Contents lists available at ScienceDirect

Environmental Modelling and Software

journal homepage: http://www.elsevier.com/locate/envsoft

Introductory overview: Systems and control methods for operational management support in agricultural production systems

Simon van Mourik [a,*], Rik van der Tol [a], Raphael Linker [b], Daniel Reyes-Lastiri [a], Gert Kootstra [a], Peter Groot Koerkamp [a], Eldert J. van Henten [a]

[a] Wageningen University, Farm Technology Group, PO Box 16, 6700 AH Wageningen, Netherlands
[b] Technion-Israel Institute of Technology, Faculty of Civil and Environmental Engineering, Haifa, 32000, Israel

ARTICLE INFO

Keywords:
Modelling
Observation
Control
Precision farming
Disturbances
Errors

ABSTRACT

A challenge in modern farming is to find a sustainable way of achieving sufficient production. Precision in dosage, timing and allocation of water, biocides, fertilizer and other inputs is essential, as are such management actions as harvesting, pruning and weeding. Despite the increasing availability of sensor and actuator technologies, decision-making is still largely left to the farmer. This is creating a strong demand for support in operational management.

This paper presents an overview of methods involving the use of technology and data to develop model-based management support and automation for productive and input-efficient farming. For each method, the main advantages and drawbacks relating to typical farm characteristics are discussed and summarized. Three case studies are presented, to illustrate the design steps involved in developing a model, observer and controller. The overall design procedure is summarized in a flowchart, and serves as a basic guide for method selection and model development.

1. Introduction

During the 1950s and 1960s, farming was intensified in order to meet the growing demand for food in the developing world after the Second World War. The main mission of this 'Green Revolution' was food security, and its objective was to achieve sufficient production rates at affordable costs. The intensification of food production was made possible by extensive research and development on high-yielding crop varieties, agrochemicals (biocides, chemical fertilizer), mechanization, breeding and controlled water supply. The Green Revolution arguably saved billions of people from starvation, while increasing human prosperity tremendously. At the same time, however, it was accompanied by serious adverse consequences for human health and the environment (Mason 2003; Innes 2013), due to the large-scale use of fossil fuels, fresh water, fertilizer, pesticides and antibiotics. Moreover, there has been a decrease in the availability of resources, including arable land (Oliver et al., 2013a,b), oil reserves, phosphates (for fertilizer) and fresh water. Another resource that is becoming increasingly scarce in the primary food production sector is human labour (Development 2019). Furthermore, the intensification of livestock farming using inexpensive, affordable production systems is often in conflict with requirements relating to animal welfare and health (Berckmans, 2014). Combined with a rapidly growing world population (which is projected to increase from 7 billion in 2011 to an estimated 9 billion by 2050), modern farming in its current form is unsustainable in terms of resource availability, human health, animal health and the ecological carrying capacity of the planet (Pimentel and Giampietro 1994). A green engineering approach that can help to minimize labour input, pollution, waste, animal discomfort and the depletion of resources (Dorf and Bishop 2011) is needed. 'Precision farming' (also known as 'intelligent farming', 'site-specific farming' or 'smart farming') is one such approach.

1.1. Precision farming

Modern, large-scale farms are characterized by the uniform application of inputs. For example, in the field, it is common practice to distribute water, fertilizer and pesticides uniformly, regardless of variations in soil properties, crop density or crop needs. In greenhouse cultivation, it is common practice to maintain constant indoor

* Corresponding author.
E-mail address: simon.vanmourik@wur.nl (S. van Mourik).

https://doi.org/10.1016/j.envsoft.2021.105031
Accepted 9 March 2021
Available online 12 March 2021

temperature levels day and night. In dairy farming, cows are usually fed rations based on an average cow and a specific average daily production, and they are milked at fixed time intervals.

Precision farming is a means of primary food production through precisely controlled input and management actions, along with precise monitoring (also known as state observation), which is enabled by technological solutions (Auernhammer 2001; Day 2005). All precision-farming operations (input, actions, monitoring) incorporate one or more levels of precision, in terms of dosage, timing or spatial allocation (e.g. by location, group or individual). Within the wide range of available precision-farming applications, three main domains of application can be distinguished. These domains are described below.

Precision agriculture (PA). This concerns open-field systems for crops like maize, grain and potatoes. The objective within this domain is to increase resource efficiency in arable farming by applying inputs according to site-specific crop demands (Oliver et al., 2013a,b). This is accomplished through the precise management of inputs (e.g. water, fertilizer, pesticides) and actions (e.g. weeding, harvesting). Examples of PA include controlled-drip irrigation (Prathyusha and Suman 2012), automated seeding, the precise application of manure and pesticides, and robotic weed control (Slaughter et al., 2008).

Precision horticulture (PH). This concerns open systems (e.g. orchards) and protected systems (e.g. greenhouses, vertical farms). The objective of PH is to exercise precise control over inputs for crops (e.g. climate and light conditioning, water, fertilizer). Actions (e.g. pruning, harvesting) are performed according to the status of individual plants or fruits (e.g. amount of leaves, ripeness) through technological solutions, like robotic harvesting (Hemming et al., 2014).

Precision livestock farming (PLF). This concerns production systems for animals (e.g. cattle, poultry, fish (Føre et al., 2017)) and algae. The objective of PLF is optimize resource efficiency and animal welfare (Berckmans, 2017). Inputs (e.g. feed, antibiotics, veterinary treatments) are applied according to the needs of individual animals. Examples of PLF technologies include robotic cow milking, livestock health monitoring (Berckmans, 2014), automated monitoring and control of broiler growth (Aerts et al., 2003), the feeding of individual cows (Halachmi et al., 1998), indoor climate conditioning, automated incubators for egg hatching and the camera-based estimation of animal weight (Song et al., 2018).

The fine-tuning of inputs based on the time-dependent, location-specific and individual needs of crops and animals can yield considerable savings in resources. The following are several examples:

- PA: Precise dosage control based on crop monitoring resulted in fertilizer savings of 23%, while improving grain yield by 4% (Zhang et al., 2002).
- PH: In greenhouse cultivation, time-dependent climate control based on fluctuations in outdoor climate has been associated with savings of up to 47% in heating energy (van Beveren et al., 2015a,b).
- PLF: The optimization of milking intervals for individual cows has been associated with increases of up to 25% in milk production (André et al., 2010).

One functionality that is common to all domains of precision farming is pest management. In an extension to primary food production, post-harvest management, precision technology is employed in such operations as automated quality monitoring, selection, packaging and air conditioning during storage and transport of produce.

In addition to uniform input, risk avoidance can lead to the over-application of inputs. Variable circumstances (e.g. weather, soil properties, disease load and the physical condition of animals and crops) make it challenging to make precise estimates of how much input is needed and how the amount of input is linked to the risk of under-application. The easiest way to decrease the risk of production loss due to pests or malnutrition is to make slight increases in the minimal dosages of pesticides, fertilizer and other inputs (Stuart et al., 2014). Given that risk mitigation is rarely balanced rationally (Kahneman 2003), however, farmers tend to have excessive risk-avoidance attitudes (Anderson and Dillon 1992). Uncertainty about how much input is actually needed, combined with the tendency to avoid production-loss risks, can lead to severe over-application. For example, worldwide irrigation efficiency is estimated to be only 37%, thus implying a loss of 63% due to runoff and drainage (Wallace 2000). Possible ecological consequences of such inefficiency include the massive use of fresh water from rivers and aquifers, and the leaching of chemicals and nutrients into groundwater due to drainage.

1.2. Automation

One way to improve labour efficiency is to automate the operational management loop (also known as the cyber-physical management cycle (Verdouw et al., 2013)). Automation can also help to avoid unnecessary risk mitigation by human operators, and it can serve as a reliable means of precision management, making it possible to attain a predefined level of product quality and harvest timing. As illustrated in Fig. 1, four stages of automation have been identified: traditional management by hand, manual control, supervised control, fully automatic control. As the management loop advances towards automatic control, farmers receive more support from and become more reliant on technology and automation. In traditional farming systems, farmers perform all management tasks themselves, both physically (in terms of perception and actuation) and mentally (e.g. making decisions on input management). In manually controlled system, farmers perform all of the decision-making themselves, assisted by actuation and, possibly, sensing technology (e.g. soil-moisture sensors, irrigation devices). This type of control is typical of developing countries. In supervised control systems, farmers are assisted by low-level controllers that operate autonomously according to the settings and set-points specified by the farmers (e.g. water dispensers that keep soil water near the desired level). These systems also allow farmers to utilize sensor information on system states of interest (e.g. soil water content) and to anticipate future events (e.g. precipitation) based on forecasts. This type of control is typical of high-tech agricultural systems in developed countries. In automatically controlled systems, farmers are no longer involved in the management loop. Examples of such systems include milking robots, feeding systems and autonomous climate-control systems.

It must be noted that supervised and automatic control are not always clearly separate classes. For example, in many automatically controlled systems, farmers are still able to overrule automated management actions or change system settings or set-points. Furthermore, automatic control is not always preferred over supervised control, for at least two reasons: 1) farmers prefer to remain involved in high-level management, and 2) for some applications, control algorithms that outperform farmers have yet to be developed.

1.3. Decision support

Decision support can be provided in three different forms.

- **Observation** of the current state of the system, to help farmers take appropriate action. Computers may be used to help provide diagnoses by observing the current state of particular systems (e.g. the climate inside a greenhouse, the water level in the soil, the health status of an animal).
- **Prediction** of how the process will respond to input, and of the corresponding performance. For example, predictions concerning yield or energy requirements can assist in the selection of appropriate climate set-points in greenhouse management.
- **Control** algorithms that schedule input under specific management objectives (e.g. high yield, low resource use). They can be used to achieve fully automated processes or to provide advice.

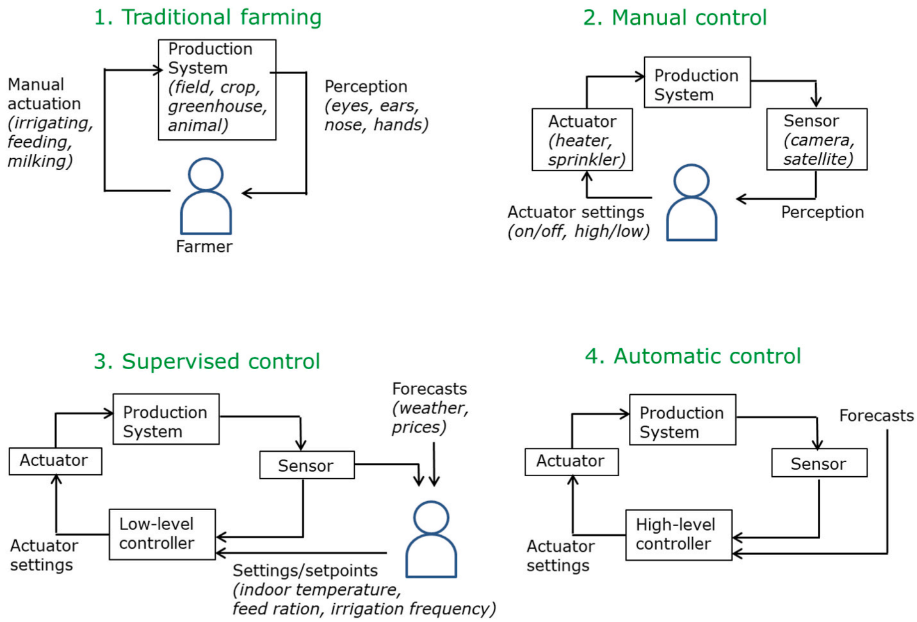

Fig. 1. Illustrative flow diagrams of four stages of automation in operational farm management. 1. Traditional farming: Farmers perform all physical tasks manually and make actuation decisions, based on their own perceptions. 2. Manual control: Farmers are assisted by actuation technology and, possibly, sensor technology. The actuation devices are operated manually. 3. Supervised control: Low-level control is performed automatically, with farmers making high-level control decisions on settings and set-points. Farmers may be informed by sensor information on current system states (e.g. soil-water content) and, possibly, by forecasts, in order to anticipate future events (e.g. weather changes). 4. Automatic control: A closed control loop is formed by connecting actuation and sensing technologies through a high-level controller. This configuration does not necessarily require any farmer involvement.

In some respects, these types of support are cumulative. Observations of current system states (e.g. indoor greenhouse climate, animal health status) can provide the initial conditions for model predictions concerning how those states would develop under different forms of input management. Subsequently, control algorithms employ model predictions to optimize input management. Fig. 2 provides a closer view of the supervised control scheme depicted in Fig. 1. More specifically, it illustrates the role of prediction, observation and control advice as a form of decision support. The diagram shows two feedback loops: a low-level management loop with input consisting of the machine settings and set-points to be tracked (as provided by users), and a high-level management loop, in which users are supported. In the high-level management loop, user support consists of automatically generated state observations, predictions of system response or control advice concerning input management (e.g. advice on climate set-points). Decision-support systems can use sensor information about current states (e.g. the climate inside and outside a greenhouse) or forecasts (e.g. weather or market prices) in order to optimize input scheduling in anticipation of future events.

Within this framework, this paper discusses methods of observation, prediction and control. The methods are based on a systems approach, using systems models. Various methods are addressed separately in the following sections. Case studies on three different farm management applications were conducted, one for each section. The actual case studies are included in the Supplementary Material.

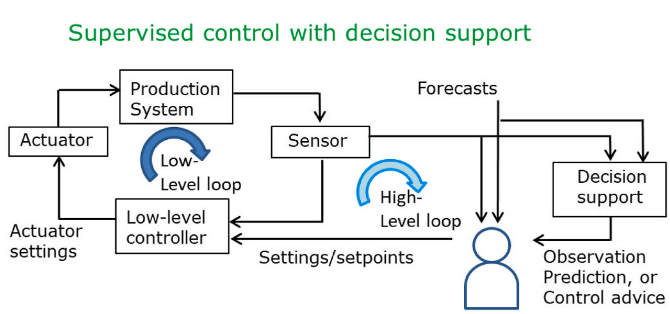

Fig. 2. Illustration of decision support for operational management with supervised control. Two feedback loops are shown: a low-level management loop (with a low-level controller) with input consisting of the set-points provided by users, and a high-level management loop, in which users make decisions on settings and set-points based on sensor information, forecasts and decision support (observation, prediction or control advice). The decision-support system may use both sensor information and forecasts.

2. Building a systems model

Farming systems generally involve many components in complex, dynamic interaction with each other, thereby forming intricate interaction networks. These components are subject to the influence of multiple inputs, some of which are controlled, while others are not. A systems model creates structure by describing the components as process variables and by describing their dynamic responses to changes in input. As described in Section 2.1, a predictive-systems model can be designed to improve input scheduling in scenario studies, in which the insight obtained into the system's input response contributes to the development of practical management guidelines. For example, this has been done within the context of greenhouse crop cultivation (Vanthoor et al., 2011), pest management (Mul et al., 2017) and irrigation scheduling (Mondaca-Duarte et al., 2020). As described in Section 2.2, the structure of a systems model can be extended with additional equations that describe relationships between states and that measure output, input and state constraints, as well as performance and systematic disturbances. This extended structure forms a foundation for state-estimation methods and control methods, as described in Sections 3 and 4.

2.1. Predictive model

This section provides a brief overview of the key elements that make up a predictive systems model. Such a model can be derived in several different ways, depending on the type of information available. A first-principle model (also known as a mechanistic, white-box or process-based model) is composed of underlying biological, chemical or physical principles (Keesman 2011). Examples of first-principle models comprise crop and animal responses to nutrients and environment, fluid dynamics (e.g. to describe climate dynamics in greenhouses and barns or to describe fluid transport in algae reactors), actuator dynamics (e.g. in greenhouse technology, irrigation and harvesters), population dynamics (e.g. for fish and pests) and classical mechanics (e.g. to describe the motion of tractors, robots and drones). Descriptive models (also known as black-box or non-mechanistic models) are generic models that are not derived from first principles, but that describe processes based on data. Examples of descriptive models include linear regression models, neural networks, and auto-correlation models (e.g. ARX, NARX, ARMAX (Chen and Zhao 2014)).

In this overview, we make no assumptions concerning the type of models used for the model-based observation and control methods. Although the models can be either mechanistic or non-mechanistic, they are all assumed to have structures as described in this section.

2.1.1. Input-state dynamics

Farming processes are characterized by a dynamic-input response. For example, the effect of irrigation input on crop growth is not immediate, but may span several days or even weeks. This makes the scheduling of input over time a non-trivial problem. In addition to the inputs that are controlled, other inputs might not be able to be controlled but should be anticipated (e.g. precipitation, solar radiation).

In a predictive-systems model, the process dynamics and various types of input are formulated systematically. One core principle of systems theory is that each system has a specific boundary, input and state. The system boundary is determined by defining the state variables within the system and the input variables that enter the system from outside and that affect the state dynamics. A general formulation to describe the input-state dynamics in a continuous form is as follows:

$$\frac{dx(t)}{dt} = f(x(t), u(t), \varepsilon(t), \theta, t). \tag{1}$$

In this equation, $x(t)$ is the state vector that changes over time (t), with some initial condition $x(0) = x_0$. The state vector can contain variables like temperature and humidity, although it could also include spatial information to make the model specific to a given site (e.g. a temperature variable subdivided into temperature at various locations). The time derivative of the state equals f, a non-linear function describing the interaction between states and how the states respond to the input. The rates of change and the strengths of the interactions are represented by the values of the model parameters in vector θ. The control input $u(t)$ can be manipulated, and it is used to control the state dynamics. The external (i.e. uncontrollable) input $\varepsilon(t)$ can often be observed and, in some cases, its dynamics can be predicted, but it cannot be manipulated. The final argument of function f is time t, which acts as an independent variable representing changes in system response over time (e.g. changes in biological development stages) that have not already emerged from the modelled state interactions.

2.1.2. Model complexity

Organisms (e.g. animals and crops) are inherently complex. Within an organism, physiological, chemical and physical processes at the level of tissues, cells and molecules form extensive interaction networks that govern input responses that are almost always non-linear (e.g. doubling a feed ration does not generally double milk production). Whereas physical processes (e.g. mass and heat transfer) are often relatively linear, while the chemical processes underlying physiological input responses (e.g. photosynthesis, food digestion) are typically non-linear.

Model complexity is a design aspect that poses an important trade-off. Very simple models are easy to derive, and they require little computational effort for control design. At the same time, however, over-simplification may introduce model errors that have a negative influence on the accuracy of predictions. Although the development of more advanced, more complex models may improve prediction accuracy, it may also require considerable experimental and field work, knowledge acquisition and modelling expertise. Not all state estimators and control algorithms are designed to address high levels of model complexity. Furthermore, increasing complexity usually increases computational demand. Another possible drawback of high model complexity has to do with the large number of parameters, which increases the likelihood that the values of some parameters will be difficult to determine.

2.1.3. Parameter estimation

When a parameter value is unknown or uncertain, a common solution method is to estimate the maximum likelihood value based on available data. The likelihood of a parameter (vector) θ is obtained by comparing the predicted model state at discrete time instances (shorthand notation $x(\theta)_k$) with a time-series dataset of measured state $x_{data,k}$. In this calculation, k is an index representing discrete time instances ($t = k\Delta t$, with Δt representing the time-step size, and $k = 1...n$). Assuming that the measurement errors are modelled by additive Gaussian white noise with covariance matrix R_k,

$$x_{data,k} = x(\theta)_k + v_k \tag{2}$$

with $v_k \sim N(0, R_k)$ (N denotes a normal distribution). The likelihood function of parameter vector θ is then as follows:

$$L(\theta|x_{data}) \sim exp \sum_{k=1}^{n} -\frac{1}{2}\Delta x_k^T R_k^{-1} \Delta x_k, \tag{3}$$

where $\Delta x_k = x_{data,k} - x(\theta)_k$. The maximum likelihood is obtained by θ^{ML}, the value of θ that maximizes $(\theta|x_{data})$.

Many optimization algorithms are available for retrieving θ^{ML}. For relatively simple models where y is a linear function of θ (e.g. a linear regression model), the maximum likelihood can be computed directly through least-squares estimation. In general, θ^{ML} is approximated through iterative algorithms that evaluate multiple parameter values and compare their likelihoods. Two classes of algorithms can be distinguished: gradient-based algorithms and evolutionary algorithms. Gradient-based algorithms search the parameter space by following a path along which the objective function L has the strongest increase (gradient). Examples include the Newton-Raphson and the Levenberg-Marquardt (Kelley 1999) algorithms. The advantage of these methods is that the use of gradients reduces the number of iterations required, thereby increasing computational efficiency. This is a crucial advantage, especially for complex models with large integration times, given the need to integrate the model for each iteration in order to obtain L. One possible drawback is that, when multiple local optima exist, there is no guarantee that the global optimum will be found. Evolutionary algorithms use a population of candidates for θ^{ML}, the values of which evolve through some stochastic evolutionary process. This stochasticity, combined with the use of multiple candidates, increases the likelihood of finding the global optimum. One general drawback of evolutionary algorithms is that they tend to require a relatively high number of model integrations. Parameter estimation is a well-established discipline, with many good textbooks available for interested readers.

2.2. Additional equations for observation and control methods

This section describes additional equations that are useful for making predictive models compatible with the observation and control methods described in Sections 3 and 4.

2.2.1. Output dynamics

Sensing technology enables automated and dynamic measurements of system states that are of interest. It is important to note, however, that not all states can be measured directly. For example, the measurement of animal stress by sampling blood cortisol levels is an invasive method that is not very practical. In such cases, indirect measurements are performed. The sensor measurements y are related to the state vector through the following output equation:

$$y(t) = g(x(t), \theta), \tag{4}$$

where g is a function depending on the state vector x and some of the parameters contained in vector θ. For example, an output function can relate a cow's resilience against disease to measured body temperature, lying time and eating patterns (van Dixhoorn et al., 2018).

Fig. 3 shows a flow diagram of a system model, represented by Equations (1) and (4). The input-state-output structure makes it possible to close the management loop by applying a controller that connects the measured output to the control input.

2.2.2. Constraints

System states may attain critical points at which the system becomes fragile, meaning that its dynamic response can become highly sensitive to slight changes in input. At critical points, a state can shift to an undesired steady state. Organisms can attain a multitude of steady states (e.g. healthy and sick, vegetative and generative, alive and dead, fresh and spoilt). In animals with low resilience, an increase in stress or disease load (e.g. from over-stocking) can have a major impact on health (van Dixhoorn et al., 2018), growth and milk or egg production. In crops, a combination of large amounts of sun and low water supply results in wilting or even death. If the interactions underlying such tipping events are not modelled explicitly, it is important to identify the constraints of state and input under which a process should be operated in order to avoid them.

Another constraint is actuation capacity, which is almost always bounded. For example, the capacity of an air heater and the maximum rate of a harvester depend on type of equipment used, and they are always limited to some level. Although some control algorithms assume that both positive and negative actuation is possible, this is not always the case. For example, an irrigation system can apply water, but it cannot extract it. To describe the constraints on $x(t)$ and $u(t)$ that are needed in order to avoid unwanted steady states or the exceedance of input capacity, constraints are introduced in the form of a given set (b) of algebraic inequalities:

$$b(x(t), u(t)) < 0. \tag{5}$$

2.2.3. Performance

Sustainable farming is characterized by inherently conflicting objectives (typically, high production rates vs low input use). The importance of all objectives should be carefully balanced in order to arrive at suitable control advice. To this end, the model can be extended with a performance measure (usually referred to as a performance criterion), J, which can be described as some function h depending on state, input, and time:

$$J = h(x(t), u(t), t). \tag{6}$$

More specific formulations of J are presented in Sections 3 and 4. To make input selection straightforward, performance is usually represented by a scalar value. When there are multiple performance objectives, the variables corresponding to these objectives can be assigned weights according to their importance (e.g. performance equals the production rate minus 3 times the input rate). The objectives are represented mathematically by various indicators, such as energy efficiency, health deficiency (van Dixhoorn et al., 2018), the animal comfort index (Fournel et al., 2017) and the damage rate during fruit or egg transport (van Mourik et al., 2016).

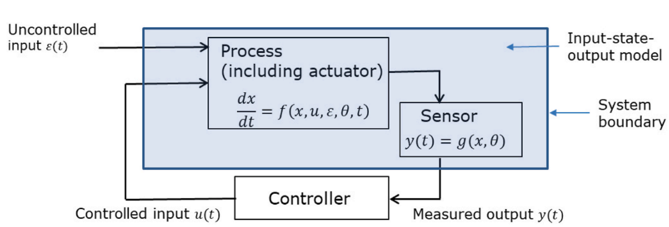

Fig. 3. Flow diagram of a controlled input-state-output system. The dynamics of state $\boldsymbol{x}$ are influenced by controlled input $\boldsymbol{u}$ and by uncontrolled input $\boldsymbol{\varepsilon}$. The blue box contains the real-life process, including actuation dynamics, which are represented by an input-state model, as well as sensor measurements, represented by an output model. The borders of the box denote the system boundary. With this input-state-output structure, the controller connects the measured output to the control input, thereby closing the management loop.

2.2.4. Disturbances and errors

Farming systems are affected by disturbances from outside the system, as well as by errors from within the system, which lead to uncertainty in observations and predictions. Consequently, this may lead to severe performance loss.

A disturbance can be seen as an unforeseen variation or fluctuation of factors coming from *outside the system* that affects the dynamics of the system and the certainty with which they can be predicted (Dorf and Bishop 2011). Examples of disturbances include unexpected fluctuations in weather, pest occurrence and disease load. Disturbances related to sensing technology include incoming sunlight and dust particles in the air (Sanderink et al., 2017). Economic disturbances (e.g. in commodity or product prices) constitute a somewhat different category, which can also affect the performance J. Due to their often-stochastic nature, the uncertainty that disturbances cause is also referred to as stochastic uncertainty (Walker et al., 2003).

An error can be regarded as a factor from *within the system* that causes its dynamics to deviate from expectations. Errors related to biophysical system properties include unexpected developmental changes, biological variations between individual plants and animals, and spatial variations in air conditions and soil properties. Errors related to sensing and actuation technology include those occurring due to poor sensor or actuator calibration, wear and tear, signal delay (Jawad et al., 2017) and spatial variations (e.g. temperature sensors located in cold areas). In control engineering, errors in sensor signals and actuators are referred to as noise (Dorf and Bishop 2011).

For purposes of observation and control, the uncertainty caused by disturbances and errors is commonly modelled as stochastic noise. The following is an example of discrete time process with stochastic noise:

$$x_{k+1} = f_k(x_k, u_k, \varepsilon_k, \theta) + w_k. \tag{7}$$

At each time instance k, the state noise w_k consists of independent draws from some distribution D: $w_k \sim D(0, Q_k)$, with Q_k the covariance matrix describing the statistical interdependency between individual noise signals. In this example, the noise is assumed to be additive (because of the + sign). The representation of stochastic, additive noise in state dynamics as well as in the output equation

$$y_k = g(x_k, \theta) + v_k, \tag{8}$$

with $v_k \sim N(0, R_k)$, is a standard way of modelling uncertainty for model-based filtering and model-predictive control. Methods like Kalman filtering and LQG feedback control are based on white noise entering the system. When the state noise is not white in terms of frequency, but exhibits auto-correlation, this can be modelled by representing white noise entering the system and adding a pre-filter within the system (see Section 3.1) that transforms the noise before it enters the state dynamics.

3. State observation

The ability to observe a system state (e.g. greenhouse climate, animal condition, crop state) is of great importance to precision management. The prediction accuracy of future state trajectories depends largely on the accuracy of the current state estimate. Current states can be estimated with models, through the use of sensing technology or through a combination of these methods. Several different methods are discussed in this section.

3.1. Data-based state estimation

System states can be estimated according to data streams obtained with sensor technology. Three common methods for this purpose are frequency-based filtering, soft sensing and machine learning.

3.1.1. Frequency-based filtering

One common way of filtering out sensing noise (see Equation (8)) involves frequency-based filtering. If it can be safely assumed that the sensing errors are in a different frequency range than the true dynamics, the errors can be filtered out based on their frequencies. For example, a low-pass filter passes the low-frequency dynamics in the sensor signal and filters out high-frequency dynamics associated with sensor noise. A high-pass filter does the opposite. A band-pass filter passes signals only within a certain bandwidth. For example, low-pass filters have been employed in order to smooth climate-sensor data in greenhouses (Rodríguez et al., 2015). A band-pass filter has been employed for automated crop-row location and tracking (Hague and Tillett 2001).

3.1.2. Soft sensing

In addition to sensing errors, limited observability can occur because the states of interest cannot be measured directly, or because direct measurements are too costly or time-consuming. In such cases, indirect measurements are performed. When an output function (Equation (4)) is available and invertible, the state can theoretically be estimated from the sensor measurements, applying the following conversion:

$$y(t) = g(x(t)), \text{into } x(t) = g^{-1}(y(t)), \tag{9}$$

This method is also known as soft sensing. A calibration curve that relates humidity (x) to measured electrical conductance (y) is an example of state estimation through an inverted output function. When y consists of multiple sensor signals, finding the relationship between the signals and the state of interest is done in a process known as sensor fusion. One example of sensor fusion involves finding the relationship between soil properties using a combination of spectroscopy, electromagnetic induction and ground-penetrating radar (Mahmood et al., 2012). One important challenge is to identify which data streams contain relevant information, and how to combine them.

3.1.3. Machine learning

In many cases, no output equation is available. Measured traits, such as the shape (Song et al., 2019), colour (Kurtulmus et al., 2011), odour (Mottram 2016) and behaviour (Berckmans, 2014) of vegetables, fruits or animals, are usually difficult to relate mechanistically to such states as ripeness, freshness, condition and health. Data-based models can be used to classify system states (e.g. a plant inside the cropping system is either a weed or not a weed; a crop is either fresh or spoilt; an animal either is or is not in good health). Various classification models for data-based machine learning have been developed, including logistic regression, support-vector machines and linear and quadratic discriminant analysis. One advantage of these models is that they do not require knowledge on the mechanics underlying specific processes. At the same time, however, this feature also prohibits the analysis of how underlying mechanics affect the system (e.g. for error analysis and in design studies). Furthermore, training sets must be carefully selected, and the systems should be tested on independent test sets or through cross-validation in order to prevent over-fitting.

For high-dimensional data (e.g. imaging data), it is even more challenging to link states to sensor measurements. One standard approach involves abstracting the data in order to obtain a lower-dimensional representation. In image processing, this is known as feature extraction, which consists of a series of image-processing steps (Gonzalez et al., 2004). The features form an abstract representation of the image data (e.g. the colour, shape and texture of tomatoes in an image), which might be indicative of the ripeness stage. One common approach involves using manually designed feature extractors with a machine-learning approach, as process-based models are often incapable of estimating states from these abstract features. The relationships are too complex to be modelled from first principles. Instead, machine-learning methods can unravel the relationships based on a set of training examples (e.g. images of tomatoes with associated ripeness values). One disadvantage of this approach is that the feature extractors are still designed manually. Recently developed deep-learning methods are able to address this by presenting an end-to-end learning approach, in which the state can be estimated directly by deep neural networks based on the raw images (Goodfellow et al., 2016). These networks optimize both feature extraction and state estimation within a single common framework based on a large training set. The deep-learning approaches have been shown to outperform classical image processing in many domains, including agriculture (Kamilaris and Prenafeta-Boldú 2018). In this paper, we focus on process-based methods. For additional background information on machine learning, see Friedman et al. (2001).

3.2. Data assimilation

Sensor information can be combined with model predictions in a process known as data assimilation. The reasoning behind this process is that both model predictions and measurements contain errors. The merging of model and sensor information results in higher estimation accuracy than is possible with measurements or predictions alone.

3.2.1. Static filter

One basic method of data assimilation involves estimating the state of interest by weighing the state prediction according to the model, x, with the measured state (assuming for now that the state can be directly measured), y, with weighting factors based on the uncertainty with which the state is predicted (σ_x^2) and the uncertainty of the measurement (σ_y^2). This results in the following estimator (Gelb 1974):

$$x = \frac{\sigma_y^2 x + \sigma_x^2 y}{\sigma_x^2 + \sigma_y^2}. \tag{10}$$

The weighting factors can be interpreted intuitively. For example, if the measurement is highly uncertain, σ_y^2 is large, and x therefore receives a higher weight. The denominator acts as a normalization term, such that all weights add up to 1. This method is suitable for estimating the state once (e.g. when x is the outcome of a dynamic process).

3.2.2. Dynamic filter

When the state is tracked over time, the estimation problem becomes more challenging, as the estimated state should be updated repeatedly according to new measurements. A dynamic filter operates in a continuous loop, using the measured input and output of the process to make a state estimation $\hat{x}(t)$. Fig. 4 shows a flow diagram of a dynamic state filter. Several dynamic filters are discussed in the following sub-sections.

3.2.3. Kalman filter

Perhaps the best-known model-based dynamic filter is the Kalman filter, the design of which is based on Equations (7) and (8). The assumptions for a standard Kalman filter are that the state function f and output function g are linear, and that state and output noise is white and normally distributed. Under these assumptions, the filter produces a

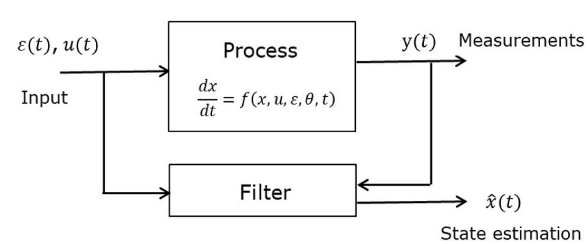

Fig. 4. Flow diagram of a model-based filter for data assimilation. The upper box represents the real-life process. The filter is based on a systems model.

maximum-likelihood state estimate, $\hat{x}_k$, together with its probability distribution, as represented by its covariance. The state estimate minimizes the expected root mean squared error between the estimated and actual state. The estimated state becomes the starting point for the model prediction at the next time instance. To estimate the state, the algorithm weighs the measurement and its uncertainty, v_k, against the model prediction based on the current state estimation and state noise, w_k, in a manner similar to Equation (10). In addition, however, it considers the uncertainty associated with the previous state estimation, and updates it accordingly. Consequently, greater uncertainty in a state estimation calls for the assignment of greater relative weight to the sensor measurement at the next time instance, and *vice versa*. This prevents unwanted bias effects. For example, if the model has a small positive bias, each new state estimate will be slightly too high and, over time, the model state will tend to drift away from the true state. Due to the increasing discrepancy between measurements and predictions, however, the state uncertainty—and thus the model-prediction uncertainty—will increase, thereby reducing the weight of model predictions. As a result, the filter tends to drive the state estimates back towards the sensor measurements.

One straightforward way to estimate the covariance matrix Q in Equation (7) is to assess the error between model predictions and measurements over a series of time. This method nevertheless requires the state to be directly measurable. The covariance matrix R in Equation (8) can be estimated based on the differences between output and model predictions, if accurate values for state and input are known, or it can be based on factory specifications about the accuracy of the sensor. The reliable estimation of Q and R imposes several requirements on state measurements. If these requirements cannot be met, autocovariance least-squares methods can be employed (Rajamani 2007).

One strong assumption on which the Kalman filter algorithm is based is that the noise is white. More specifically, the algorithm is based on the assumption that signal errors are not auto-correlated and that all frequencies are attained in a uniform manner. Although the invalidity of this assumption does not necessarily mean that the Kalman filter will not work (Julier et al., 2000), such issues should be approached with caution. If non-whiteness forms a bottleneck to filter performance, pre-filtering can be employed. In pre-filtering, the model is extended with augmented model states that transform white noise into coloured noise, which subsequently enters the process dynamics and output measurements (Salzmann et al., 1991).

Kalman filters can be used in a wide range of farm-management tasks, including the estimation of greenhouse climate and crop states (López-Cruz et al., 2017; van Mourik et al., 2019), location estimations for agricultural vehicles (Gartley and Bevly 2008), health monitoring in dairy cattle (De Mol et al., 1999) and water-level monitoring in fish farms (Ullah and Kim 2018). State estimation by Kalman filtering also forms a part of the LQG control algorithm (Section 4.1.4).

3.2.4. Extended and unscented Kalman filter

One key assumption is the linearity of the state dynamics and output function. The extended Kalman filter was designed to address non-linearity, by linearizing f and g at each state update (Grewal and Andrews 2014). Such linearization is required in order to compute the propagation of state uncertainty in a straightforward and computationally efficient manner. The extended filter also introduces linearization errors, however, as well as errors relating to the assumption of symmetric error distributions. The unscented Kalman filter was designed to circumvent these types of errors (Julier et al., 1995). This filter uses a sampling method that retains the non-linear transformation f intact (unscented) by sampling the state covariance with a few sampling points (i.e. sigma points). The unscented Kalman filter also contains three design parameters that can be adjusted to address state probabilities that are not normally distributed.

3.2.5. Particle filter

All of the aforementioned Kalman filters assume that the states have unimodal distributions. This assumption is not always realistic. Consider the following example. To estimate its location, an autonomous vehicle uses a model based on wheel-rotation speed and steering action, with GPS as a location sensor. When the vehicle is in front of a tree, it may assign high probability densities to several locations that are close to trees on the orchard map, while assigning low probabilities to locations between trees. This results in a multimodal density of location probability. This state cannot be represented using a Gaussian distribution, as is the case with the Kalman filter. The particle filter therefore represents the state estimation with a large number of weighted particles, thus allowing multi-modal probability-density functions. This comes at the cost of computation, however, as it requires the transformations f and g to be computed for each particle at each state update. The computational demand thus depends largely on the number of particles used and the computational demand for evaluating the functions f and g. For example, particle filters have been designed for localization within such repetitive environments such as orchards (Bayar et al., 2015) and barns (Vroegindeweij et al., 2016).

3.2.6. Dual estimation

Similar to online state estimation, model parameters can be adjusted online as well. The difference between this method and the parameter estimation described in Section 2.1.3 is that online state estimation is performed on a single system or subject, whereas parameter estimation often uses experimental data on a large number of subjects. The simultaneous estimation of parameters and states is known as dual estimation (Liu and Gupta 2007), and it has been used for estimating greenhouse climate (Speetjens et al., 2009) and soil moisture (Lü et al., 2011). Another online adaptation method is Bayesian forecasting (West and Harrison 2006), which estimates current states and parameters, in addition to predicting future states according to Bayesian principles. Although the basic principles are similar to those underlying Kalman filtering, Bayesian forecasting is more extensive (e.g. due to the use of discount factors that assign higher weight to current data than to older data). One potential drawback of dual estimation stems from the additional flexibility introduced by parameter adaptation. More specifically, although it decreases bias, it may increase variance in model predictions and controller performance, up to the point of instability (Rohrs et al., 1985). The bias-variance trade-off is a well-known design principle in statistical learning (Friedman et al., 2001).

3.3. Summary

The main advantages and disadvantages of the state-observation strategies discussed above are summarized in Table 1. Different methods are associated with different advantages relating to model requirements, the necessity of addressing non-linearity, multimodality and non-Gaussian error distributions. They are also associated with possible disadvantages relating to assumptions concerning error properties and computational costs. The optimal choice of filter therefore depends on the particular observation problem at hand, as well as on the key advantages and disadvantages and how they weigh against each other. For example, if a process model is non-linear and exhibits a non-Gaussian state distribution, a particle filter might be preferred. If simulations are computationally demanding, however, the need for a more time-efficient method might outweigh the choice of a particle filter.

4. Control design

Using a systems model, a controller can be designed to determine the input that would optimize performance through precise timing and dosages of inputs and actions. Two different means of control can be distinguished: feedback control (in which control actions are based on current and past system states) and model predictive control (in which

Table 1
Various filtering strategies for state estimation, together with possible advantages and disadvantages with respect to Precision Farming applications.

Method	Advantages	Disadvantages
Data-based in general	No process model required	Sensor errors not compensated
Frequency-based filter	Simple error model required	Overlapping frequency regions of errors and true dynamics cause filter errors
Soft sensing	States estimated through indirect measurements	Not designed to deal with noise
Machine learning	No output equation required	Error analysis is difficult
Data assimilation in general	Integrates model with sensor data, addresses noise	Systems model required
Static filter	Indirect measurements	Not designed for dynamic processes
Kalman filter	Dynamic process	Linearity, white noise and Gaussian distribution assumed; unimodal likelihood
Extended + unscented Kalman filter	Approximates non-linear dynamics	Extra computational costs from linearization, white noise and Gaussian distribution assumed; unimodal likelihood
Particle filters	Deals with non-linearity and multi-modality, non-Gaussian distributions	High computational costs
Dual estimation	Estimates parameters and states simultaneously	Additional variance and possible instability

control actions are based on both current states and expected future events).

4.1. Feedback control

The aim of feedback control is to steer the state in order to track a reference signal. For example, climate variables in a greenhouse should be close to a set-point chosen by the grower, or a vehicle should follow a set path. The design of signal-tracking control focuses on minimizing tracking errors (the difference between a state and a desired set-point), attenuating disturbance and maintaining stability in the process dynamics (Dorf and Bishop 2011). The basic structure of feedback control consists of a system that is connected to a controller through sensors and actuators. The controller receives sensor information about the state or states of interest, and these measurements are compared to the reference signal that the system state is intended to track. The difference between the measured signal and the reference signal is used to compute the subsequent control actions. This is usually done in a continuous fashion, forming a closed feedback-control loop (Fig. 5).

One general benefit of a feedback-control loop is the ability to react directly to tracking errors. A feedback controller adjusts the input based on the current state and, in some cases, on the past state trajectory (using integrating action; see Section 4.1.1). It nevertheless does not anticipate future events (this is done in model predictive control; see Section 4.2). The types of feedback control that are most relevant to farming operations are discussed below.

4.1.1. Proportional-integral-derivative control

Proportional-integral-derivative (PID) controllers are perhaps the most commonly used control algorithms in operational farm management. Examples include path tracking and suspension control (Yu et al., 2004) for autonomous vehicles; climate control in greenhouses (Li et al., 2018) and barns (Niu 2014); depth control in soil measurements (Mouazen et al., 2005); and water-pressure control in irrigation systems (Goodchild et al., 2018). In the abbreviation PID, the P stands for proportional action (a direct response on the tracking error), the I stands for integrating action (the adjustment of input based on the history of the tracking error) and the D stands for derivative action (the response to the trend in the tracking error). A PID controller can be expressed as follows:

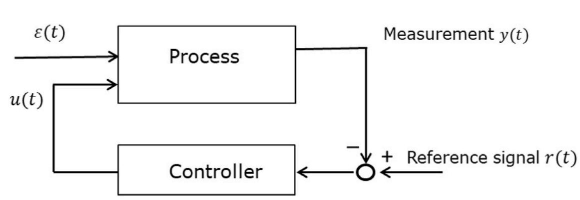

Fig. 5. Flow diagram of a feedback controller. The process state or states of interest are measured and compared to the reference signal. Based on the difference, the controller steers the state by adjusting the control variable $u(t)$.

$$u(t) = Pe(t) + I\int_0^t e(\tau)d\tau + D\frac{de(t)}{dt}, \tag{11}$$

where $e(t) = r(t) - \hat{x}(t)$ is the tracking error, with the reference value $r(t)$ representing the desired set-point trajectory and $\hat{x}(t)$ representing the estimated current state.

The proportional action (first term), provides a controller input $u(t)$, which is aimed at steering the system state towards the desired set-point. Proportional action alone, however, may result a steady-state offset for the state.

The integrating action (middle term) enables the controller to attenuate low-frequency errors and disturbances that cause an offset in the state. For example, consider a heating system that should keep the temperature inside a greenhouse at 20 °C. Suppose that, due to an actuator error, the room is not sufficiently heated, and the temperature reaches only 15 °C. This means that the state has an offset of 5 °C, which may be somewhat decreased, but not completely eliminated, by the P action. The offset will result in a steady increase in the integral term over time. Consequently, the heating input will increase and the offset will ultimately disappear. One possible disadvantage is that the integral term may respond slowly to changes in tracking error or lead to instability.

The derivative action (last term) is proportional to the changes in tracking error, and it typically speeds up the system response. Caution is needed, however, as the derivative term can be highly sensitive to violently fluctuating derivatives due to noisy measurements. These fluctuations might cause violent input dynamics that, in turn, might have a negative influence on stability, actuator wear and tear, and resource efficiency. These fluctuations can be attenuated by first filtering the tracking error with a low-pass frequency filter. The trade-off is that a low-pass filter delays the signal. In addition, it is important to be careful not to filter out true signal information.

A controller with proportional, integral or derivative action (or some combination thereof) can be tuned by loop-shaping techniques that balance sensitivity to tracking errors (with typically low-frequency dynamics) with robustness against input disturbances (with typically high-frequency dynamics) using frequency analysis tools (e.g. the Bode plot and Nyquist diagram). Bode/Nyquist-based techniques can also be employed to design controllers with additional robustness against non-linearities and modelling errors. One important advantage of PID controllers is their simplicity: they can be tuned according to measured input-output response, without the use of a process model. For example, this can be done for a system with single input and single output according to the Ziegler-Nichols tuning rules.

One possible disadvantage of PID control is reduced performance due to the simplification of assumptions concerning the actuators. In Equation (11), the controller assumes that actuation can be performed either continuously or at every discrete time instance. It is important to note, however, that this is not always the case. For example, it might be possible to irrigate a field only once every three days, as there is only one sprinkler system that is used on multiple fields.

Another assumption is that actuators can dose gradually. In practice, however, most lamps, fans and heaters can only be switched on or off. One possible solution in the case of an on-off actuator is to allow the controller to be switched on or off at a high frequency, while controlling the percentage of 'on' time with a PID controller (PWM-pulse width modulation). For example, one such controller was designed for a temperature-controlled food-storage room (Mourik et al., 2010).

As indicated in Equation (11), $u(t)$ is a symmetric function of $e(t)$. The input can thus be either positive or negative, depending on the sign of the tracking error. In reality, however, input can often only be added, and not subtracted (as is the case for irrigation water or pesticides). This can result in a problem known as wind-up. For example, if the soil-moisture level is too high, a controller can do nothing but wait until enough water has evaporated or drained. In the meantime, however, the integral term will have accumulated to a high value. For this reason, even after the water level has dropped, the high integral value will prevent the controller from taking action. Such integral wind-up behaviour has been well-documented, and various anti-wind-up approaches have been described in the literature (Azar et al., 2015).

Equation (11) applies no constraints on the input and state. A 'soft constraint' may be imposed by choosing the values for P, I and D in such a way that the input or state will seldom, if ever exceed a certain maximum or minimum value. It is possible to tune PID by loop-shaping (and not only by model-free empirical tuning). This can be done even for systems with multiple inputs and multiple outputs (Xiong et al., 2007). In addition, some PID design algorithms take input efficiency into account as control objective (Comasòlivas et al., 2012).

4.1.2. Feedforward control

Given that feedback control acts upon tracking errors caused by disturbances that affected the process dynamics at some time in the past, in principle, it will always lag behind. Feedforward control acts directly upon those disturbances by measuring them online and determining the required control input based on a model. Feedforward control is often used in combination with feedback control. One advantage of feedforward action is that it gives the controller the opportunity to react to disturbances immediately, rather than awaiting a system response and then acting upon it. Especially for systems with a slow response (e.g. in crop or animal development), feedforward control may offer a valuable solution in disturbance attenuation. One possible drawback is that the disturbances must be measured online. Such measurements can be costly or difficult to realize, and they are subject to measurement errors. In addition, the influence of the disturbances on the measurements of interest must be modelled. The added value of feedforward control therefore depends largely on the quality of the error-response model, measurement accuracy and response time.

4.1.3. LQR control

The linear quadratic regulator (LQR) is designed with the use of a linear model of the system. Under the assumptions that state dynamics are time-invariant, undisturbed, without external input and linear, model (1) is approximated as follows:

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t), \tag{12}$$

where the matrices A and B form a linearized version of function f. The control problem is commonly formulated as minimizing the performance criterion J, which indicates how well the management objectives (e.g. tracking error and low input costs) are met over a time period ranging from 0 to T,

$$J = x(T)' Sx(T) + \int_0^T x(t)' Vx(t) + u(t)' Wu(t)dt. \tag{13}$$

The criterion J makes it possible to design a controller that can weigh the costs of tracking errors against the costs of input use (the minimization of these variables usually results in a trade-off). The form of J is quite specific; it is quadratic (i.e. it consists of matrix-vector combinations, all of which yield quadratic terms) and time-invariant. The first term on the right represents the value associated with the state at end time T (e.g. produce value at harvest). The contribution of each state within the state vector $x(T)$ is weighed by matrix S. The second term is an integral containing the running costs associated with tracking errors (i.e. deviations of x from 0, weighted with matrix V) and the costs of the control input (weighted with matrix W). As with PID control, soft constraints can be imposed on the input and state by tuning the values of V and W.

Under the assumptions that the states can be observed and that the feedback is of the form $u(t) = -K(t)x(t)$, with $K(t)$ as the feedback-gain matrix, the optimal input trajectory is computed by solving the 'Ricatti equations' (Kalman 1960). These equations can be solved relatively quickly through computation, as compared to the dynamic-programming method required for non-linear, time-variant control problems (see Section 4.2.3). In addition to this computational advantage, another important benefit of LQR is that it is an optimal feedback controller, in the sense that it has an explicit performance criterion and it achieves optimal performance. One potential drawback, however, is that this optimum can be achieved only under quite strong assumptions: linear input-output response, time invariance, symmetric input, input that can be steered continuously over time and in dosage, no influence from external input, and performance that is represented by quadratic functions of tracking errors and input. Caution is advised when modelling performance with quadratic functions. Although economic costs often have a linear relationship with input (doubling the input doubles the input costs), the relationship between set-point deviations and the costs associated with spoilage or production loss may be exponential (e.g. bacteria growth in dairy products depends exponentially on temperature (Phillips and Griffiths 1987)).

The feedback mechanism in LQR offers some extent of robustness against undesired state deviations by actively steering them back towards the set-point. This robustness is limited, however, as there are no errors or disturbances in Equation (12) for which the controller is explicitly designed. Moreover, there is no integrating action (as in PID control) to compensate for offset in tracking error. Performance in practical situations is therefore heavily dependent on the validity of the assumptions. Examples of LQR applications include greenhouse climate control (Gutiérrez-Arias et al., 2015) and dust control in animal housing (Liao and Feddes 1993).

4.1.4. LQG control

Linear quadratic Gaussian (LQG) control is an extension of LQR. This extension is explicitly designed to address uncertainty in state and output measurements. For this, the process dynamics of (12) are extended with an output equation, as well as with state and output noise (see also Equations (7) and (8)),

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t) + w(t) \; y(t) = Cx(t) + \nu(t). \tag{14}$$

As with LQR, the control action consists of state feedback, $u(t) = -K(t)x(t)$, which optimizes performance with respect to the criterion

$$J = E\left[x(T)' Sx(T) + \int_0^T x(t)' V(t)x(t) + u(t)' W(t)u(t)dt\right]. \tag{15}$$

Solving for the state estimate and optimal input requires solving two matrix Riccati equations (consisting of differential equations in continuous time and difference equations in discrete time). The main differences between the LQG and LQR control problems are that, in the LQG problem, the performance criterion is minimized with respect to the *expectancy* (E) of the costs, and the state is estimated through Kalman filtering (see section 3.2.3). Furthermore, the weights are time-dependent, which may facilitate the imposition of soft constraints with regard to the timing of input (e.g. when input can be applied only at specified times). Input is more likely to occur during these intervals if W is small during designated time intervals and large elsewhere.

Despite the fact that LQG addresses stochastic state and output noise, it has no stability margin. In other words, it cannot guarantee any stable robustness against model errors, in the sense that bounded input results in bounded output (Doyle 1978). For this reason, small model error could potentially result in ever-growing oscillations in input and state. Examples of LQG control include heating and ventilation control in greenhouses (El Afou et al., 2013) and active suspension control for agricultural vehicles (Bo and Fan 2004).

4.1.5. Threshold control

Some actuators cannot provide continuous dosage, but can only be switched on or off. In on-off control (also known as bang-bang control), the timing of switching is optimized. This type of control type was developed for such purposes as the application of crop nutrients (Hooper 1988).

One very basic but commonly used type of on-off control is threshold control (also known as hysteresis control). In this type of control, a control action is performed each time a state reaches an upper or lower boundary. For example, when the temperature in a food storage room gets too high, a cooler switches on for some period of time. When the temperature gets too low, a heater turns on. This method of control may result in typical saw-tooth state dynamics within a 'hysteresis band', as illustrated in Fig. 6.

One main advantage of threshold control is its simplicity: no model is required. In addition, a threshold controller can be developed relatively easily for systems with asymmetric actuation abilities. For example, an irrigation system can apply water to the soil, but not extract it. A threshold-irrigation controller was developed, that triggered irrigation events based on crop-water status (Thompson et al., 2007).

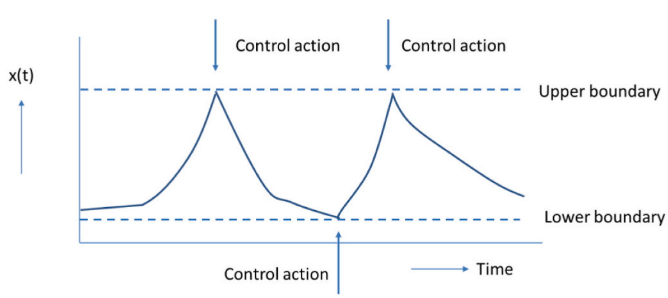

Fig. 6. Illustration of temperature dynamics with threshold control through heating and cooling actions. Each time the state reaches an upper or lower boundary, a control action takes place. This keeps the state within a desired region.

Threshold control is subject to several possible disadvantages. If a system has a slow input response, the thresholds could be violated. For example, if irrigation is stopped right at the moment at which the soil water reaches a threshold value according to an underground sensor, water will continue to trickle down through the soil, and the water content will increase a bit further. Furthermore, threshold control does not infer high precision. The state is located somewhere between two boundaries, but it does not follow a precise trajectory. Precision can be increased by bringing the upper and lower boundaries closer together to create a narrow state bandwidth. This might increase the likelihood of threshold violation, however, and it could cause the state to 'bounce' between the upper and lower bounds. For example, with the context of heating and cooling, this could cause a heater and a cooler to work against each other, thereby wasting energy.

4.1.6. Summary

Several strategies are available for feedback control, each with its own assumptions and properties. The main advantages and disadvantages of these strategies with respect to operational farm management are summarized in Table 2. As in the selection of a suitable state filter, the choice of a feedback controller depends on the particular control problem at hand, as well as on the key advantages and disadvantages, and the trade-offs between them.

4.2. Model predictive control

When future circumstances are known, or at least forecast, it may be worthwhile to anticipate them through input scheduling. For example, if the weather will be warmer tomorrow than it is today, lowering the greenhouse temperature today could save a considerable amount of heating energy, while maintaining an acceptable average temperature. Model predictive control (MPC) is the class of control methods for input scheduling while taking future changes into account. We start by describing the concept of finite horizon control.

4.2.1. Finite horizon control

In the finite-horizon form of model predictive control, the input is scheduled for a fixed period of time $[0, T]$. This is known as open-loop control, in contrast to closed-loop control (Section 4.1), as there is no feedback based on state measurements to close the management loop. The control problem consists of minimizing the performance criterion for any admissible input $u(t)$. The performance can be formulated as follows:

Table 2
Various feedback-control strategies, along with their possible advantages and disadvantages with respect to operational farm management.

Method	Advantages	Disadvantages
Feedback control in general	Robust against disturbances	Does not anticipate future events. Most methods do not explicitly consider hard state or input constraints, and most assume continuous actuation, gradual dosage and symmetric actuation
PID control (no model required)		
Proportional control (P)	Very simple tuning	Steady state offset in tracking error
P + Integrating action (I)	No steady-state offset in tracking error	Slow control response
P + I + Derivative control (D)	Accelerates control response	Sensitive to measurement noise
Model based (model required)		
Feedforward control (extension to feedback)	Anticipates slow system response	Sensitivity to model errors and sensor errors
Linear Quadratic Regulator (LQR)	Tracking quality weighed against input efficiency	Linear time invariant (LTI) system, and quadratic costs. Not designed to address errors or disturbances.
Linear Quadratic Gaussian (LQG, extension to LQR)	Designed for noisy state and output Dynamic performance weights	LTI system. No guarantee of robustness against model errors.
Threshold control	No model required. Designed for on-off actuation, hard input and state constraints	Low tracking precision, threshold violations when input response is slow, no explicit cost criterion

$$J = F(x(T)) + \int_0^T L(x(t), u(t), t)dt. \quad (16)$$

The function F represents the costs associated with the state at end time T, and function L represents the running costs associated with input and with state dynamics. In addition, J can be minimized under time-dependent boundary conditions of input and state, $b(x(t), u(t), t) < 0$. The control problem of minimizing criterion J can be seen as a generalization of the LQR problem, as both the system model and the performance criterion can be non-linear and time-varying. Given that the model includes external input $\varepsilon(t)$, minimizing J implies that the input is optimized while anticipating future external input dynamics.

Solving this control problem can be quite challenging. In the 1950s, Pontryagin and Bellman developed algorithms to design optimal input trajectories that minimize J under the constraint $b < 0$ by solving the Euler-Lagrange equations. Although the optimization problem can be solved analytically for some simple cases, a numerical solution is usually required (e.g. through dynamic programming; see Section 4.2.3).

Potential disadvantages of finite-horizon optimal control include the fact that it is not designed to attenuate any errors or disturbances. As a result, the exact value of $\varepsilon(t)$ is assumed to be known in advance. In the absence of feedback, forecast errors may cause the realized state trajectory to deviate from the predicted trajectory.

Regardless of its potential disadvantages, finite-horizon optimal control is a valuable tool for determining the potential added value of precision management in terms of timing and dosage. Comparing the theoretical optimal performance to the performance currently obtained in practice can provide a considerable amount of information about the potential gains to be realized with precision management. In greenhouse research, finite-horizon optimal control has been employed to investigate the energy-saving possibilities of precisely controlling the timing and dosing of inputs like heating and ventilating (van Straten et al., 2010). In the performance criterion, $x(T)$ represents the end state of the crop, and $L(u)$ represents the running costs for climate management. Other research applications for finite-horizon optimal control include manure spreading (Krishnan et al., 2006), robotic harvesting (Van Henten et al., 2009) and pest management (Vincent 1975).

4.2.2. Receding horizon control

Receding horizon control is an extension of finite-horizon control. In this model, both the state and the prognosis on external input are updated regularly and used to compute new input schedules. Thereafter, the time horizon is extended such that the time window across which the input is optimized shifts forward, while maintaining the same length. Receding-horizon MPC is quite popular in research on farm operations. Example applications include trajectory planning for moving agricultural machines (Coen et al., 2008), greenhouse climate control (El Ghoumari et al., 2005), irrigation (McCarthy et al., 2014), and agro-robotics (Kayacan et al., 2015). An extensive review on MPC applications in agriculture is provided in Ding et al. (2018).

The updates come at a price in terms of computation, as each update requires a recalculation of the optimal input. Given the high computational cost of solving the Euler-Lagrange equations, they may allow only a low update frequency. To avoid slow computations, the state dynamics can be linearized, thereby yielding the aforementioned Riccati equations, which can be solved much faster. This nevertheless comes at the expense of possible linearization errors. Another, increasingly popular approach is 'Explicit MPC', in which the optimization problem is solved offline for a range of operating points, and multi-parametric programming is used to express the optimal control actions as explicit functions of the states. In most cases, the end results resemble a look-up table. Although this method drastically reduces online computation requirements, it does not always guarantee that all constraints have been satisfied. For additional information on this point, see Alessio and Bemporad (2009) and Diangelakis et al. (2019).

The regular updating of open loop-control is known as open-loop feedback. Although the regular updating of states and forecasts does provide some robustness (in the sense that it can correct for errors and disturbances), it does not provide any integrating action (as is the case with PID control). For example, consider a heating system that is supposed to achieve a room temperature of 20 °C, but that has an offset of 5 °C due to actuator bias. The MPC algorithm described above will not learn from any past 'mistakes', and it will therefore continue recalculating its actions in the same way, thus preserving the offset. Possible options for addressing model offset include i) applying integrating action (Section 4.1.1) and ii) adapting the model parameters based on output response (Sections 2.1.3 and 3.2.6). Several methods have been developed in order to improve MPC performance. One example is tube MPC (Langson et al., 2004), in which a feedback controller is implemented to keep the state within a bounded area (tube) of the optimal trajectory calculated by MPC.

The basic MPC algorithm is designed to optimize performance in a deterministic manner, without taking performance uncertainty into account. By comparison, stochastic MPC is a method for mitigating performance risks associated with errors and disturbances.

4.2.3. Stochastic model predictive control

Stochastic model predictive control, or stochastic MPC, is based on the assumption that the state dynamics are disturbed by noise (Equation (7)).

The control problem consists of optimizing the input trajectory u_k, with respect to the cost criterion (Bertsekas 1995):

$$J = E\left[F(x_n) + \sum_{k=0}^{n-1} L_k(x_k, u_k, \varepsilon_k)\right]. \quad (17)$$

This criterion is similar to the one used in Equation (16), with the difference that, in this criterion, the *expectancy* of the costs is minimized. Furthermore, the formulation of J can be adapted in such a way that the variance of the costs J are included as well (Horwood 1996). Given that a decrease in variance implies a decrease in the risk of very high costs, this is also known as risk-sensitive control.

The concept of stochastic MPC has been employed within the context of irrigation optimization (Zavaleta et al., 1980), crop harvesting (Alvarez and Shepp 1998), greenhouse crop production (Mourik et al., 2016) and fish harvesting (Braumann 1999).

One important issue in stochastic MPC is computational demand. The difficulty in solving the control problem resides in the fact that there is no single optimal trajectory u_k. As time proceeds, the realizations of noise w_k will change the course of any state trajectory x_k that was planned beforehand. As a result, for each time instance k, the optimal choice of u_k will depend on the particular state at that time, x_k, such that $u_k = u(x_k, k)$. Optimization across all possible state realizations over time poses a computational challenge. Suppose that a state is discretized into m parts, and time into n parts. For a single state model, there are thus n^m possible state trajectories. A basic crop model with only one state variable, which is very coarsely discretized with $m = 10$ possible states and only $n = 10$ time instances, already has 10 billion possible state trajectories. A powerful method for overcoming this computational burden is dynamic programming (Bertsekas 1995). The method is based on the notion that the optimal control problem is solved backwards, by computing the minimal costs $J_k(x_k)$ for each time and state, starting with $k = n$. Therefore, $J_n(x_n) = F_n(x_n)$ is solved first for all possible x_n. The process is then repeated for $k = n - 1$ down to 1:

$$J_k(x_k) = \min_{u_k \varepsilon U_k(x_k)} E\left[L_k(x_k, u_k, \varepsilon_k) + J_{k+1}\left(\widehat{x}_{k+1}\right)\right], \quad (18)$$

where $U_k(x_k)$ is the admissible region of control (comparable to the previously mentioned constraint $b(x(t), u(t)) < 0$). The first term on the right of Equation (18) represents control costs between time instance k and $k + 1$, and the second term represents the costs associated with the

Table 3
Various model predictive control (MPC) strategies, along with their possible advantages and disadvantages with respect to operational farm management.

Method	Advantages	Disadvantages
MPC in general	Optimizes performance by anticipating future events. Designed for non-linear, time-varying systems, and boundaries on input and state	No integrating action, sensitive to forecast errors and model prediction errors
Finite horizon control	Exact optimal performance	Not designed to address disturbances and errors
Receding horizon (extension to finite horizon)	Addresses errors in state and forecast through state and forecast updates	Not designed explicitly for errors and disturbances Computation- intensive
Stochastic MPC (extension to receding horizon)	Explicitly addresses uncertainty in state dynamics; risk-sensitive	Highly computation- intensive

newly predicted state $\hat{x}_{k+1}$, which was already computed. This process of backward recursion reduces the computational load from n^m to only $m \times n$ cost optimizations. As the number of states increases, however, computational complexity can still increase quite rapidly. If each state is discretized into m parts, optimizing for a model with d states requires $m^d \times n$ cost optimizations. In order to achieve further reductions in computational time, approximative algorithms such as limited look-ahead policies and rollout algorithms have been developed (Bertsekas 1995). In addition to high computational demand, another potential disadvantage of this method is the assumption of white noise. Although white noise can be transformed into auto-correlated noise, it requires extending the number of states d and, consequently, the computational load.

4.2.4. *Summary*

The most important advantage of MPC over feedback control is the ability to anticipate changes in external input (not to be confused with disturbances). In addition, MPC is quite flexible, as it was designed for non-linear, time-varying systems and boundaries on input and state. It is nevertheless subject to potential disadvantages as well. The changes in external input must be predicted, thereby introducing forecast errors that may have a negative influence on performance. The control actions are based on model predictions, which makes the performance more reliant on model predictions than on feedback control, which increases the impact of modelling errors. Taken together, the choice for MPC instead of a feedback or feedforward controller requires balancing their general advantages and disadvantages. The main advantages and disadvantages of the MPC methods discussed in this section are summarized in Table 3.

5. Discussion and conclusions

Most of the observation and control methods discussed in this overview have been designed according to a systems model. A model-based approach requires some level of investment in model development. The benefit of such approaches is that they allow the *a priori* analysis of how an observer or controller will perform (i.e. before applying it in reality) and of what might be needed in order to improve the performance. For example, in the case study for Section 3 (see Supplementary Material), increased prediction accuracy is needed. In this section, we discuss the relationships between the methods discussed in this paper, as well as the types of tasks and skills that they can serve. We then present a design procedure based on these methods.

5.1. *Relationship between tasks, methods and cognitive skills*

The role of farmers has evolved through advances in technology and machine intelligence. In high-tech farms, the eyes, ears and noses of farmers are replaced by sensors and cameras, and their hands and tools are replaced by actuators. Their brains, which perceive and process information in order to take operational management decisions is assisted, or even replaced, by machine intelligence.

The methods discussed in this paper can be linked to several essential cognitive skills that make up the machine intelligence required to support farmers. The systems model represents the knowledge that is needed in order to fulfil the task of predicting how a system will respond to input. Such knowledge can be seen as a cognitive skill for model-based observation and control. The cognitive skills, and their relationships to the types of methods discussed in this paper are summarized in Table 4.

Multiple types of methods are available for each task. As demonstrated by the illustrative case studies, however, despite a multitude of methods (and types of methods), not every management task constitutes a problem with a straightforward solution. This is because the properties of the methods often do not fully match the management objectives. For example, the case study on feedback control indicates that LQG control unintentionally punishes good behaviour (in that case, by bringing the mite population below the critical level), whereas a threshold controller involves the undesired trade-off between respecting the state constraint and maximizing input efficiency. In the MPC case study, prediction uncertainty prohibited any guarantee about keeping the state constrained within acceptable bounds. In general, it is very rare for a method to satisfy all of the desired objectives and to comply with all constraints.

Another reason that method selection is not a straightforward process is the fact that it is very rare for all of the assumptions on which a method is based to be realistic. Dynamics are rarely linear, noise is seldom white and it is only in exceptional cases that perceived costs are a quadratic function of state and input. Although several methods have

Table 4
Link between cognitive skills required for prediction, observation and control, and the types of methods discussed in this paper.

Task	Cognitive skills	Method types
Prediction	Prediction of system response	Systems modelling (Section 2)
	Awareness of uncertain outcomes	Noise modelling (Section 2.2.4)
	Learning from experience	Parameter estimation (Section 2.1.3)
Observation	Observe states of interest	State observation (Section 3)
Control	Definition of goals/objectives	Performance modelling (Section 2.2.3)
	Retrieval of information on current state	See: Method types for Observation
	Prediction of system response	See: Method types for Prediction
	Awareness of state and input constraints	Constraint modelling (Section 2.2.2)
	Response to changes in state	Feedback control (Section 4.1)
	Response to changes in circumstances	Feedforward control (Section 4.1.2)
	Advance planning, anticipation of future events	Model predictive control (Section 4.2)
	Advance planning, anticipation of uncertain future events	Stochastic control (Section 4.2.3)

been designed under the assumption that no uncertainty exists, uncertainty is a prominent characteristic of almost all farming systems. In other words, there is always some gap between theory and practice. This does not mean, however, that these methods have no validity, or that they will automatically yield poor performance. Whether a violated assumption will be a true disadvantage depends on the extent to which its violation will have a negative influence on predictions, observations or control actions, as well as on how these aspects will ultimately result in the loss of performance. Assessing the relationship between the theory-practice gap and performance loss is an important frontier in the science of biosystems engineering.

5.2. Selection and design procedure

Various types of methods for each management task are displayed in Table 4. Each type comprises multiple methods, each of which has a unique set of properties and underlying assumptions. Whether these aspects translate into advantages or disadvantages depends on the type of system and the circumstances under which it is operated. Farming systems vary widely with regard to configuration, technology, production process and environmental circumstances. For engineers, therefore, selecting and designing the right methodology is likely to be a complex task. One important challenge thus involves the configuration of a selection and design procedure as a form of support for engineers.

The outline of this overview paper suggests a configuration for a basic selection and design procedure (Fig. 7). The first step involves building a systems model based on the available technology and system properties, and subsequently designing an observer and then an automatic controller. The design or choice of a model depends on relevant system characteristics (e.g. available sensing and actuation technology, external input, state dynamics, constraints and noise) and performance objectives, as well as on the requirements posed by the subsequent design of an observer and controller. For example, a given observer may require a linear model, and a given controller may require a noise model. Conversely, the choice of an observer depends on the type of systems model available (e.g. in terms of accuracy or the availability of an output equation), and the choice of controller depends on type of model available (e.g. whether performance is quadratic), as well as on the observer (e.g. the states that can be observed). In principle, therefore, the procedure for designing and selecting the right methodology is not sequential, but iterative.

The three circles depicted in Fig. 7 can be linked to the various sections of this paper. The systems-model design steps are discussed in Section 2, the observer design methods are discussed in Section 3 and the control design methods are discussed in Section 4. The associated advantages and disadvantages are summarized in Table 1, Table 2 and Table 3.

Many other relevant methods exist that could complement the overview presented here, including H_2 and H_∞ feedback control, multi-agent control, adaptive control, model selection and approximation, and reinforcement learning. An interesting avenue for follow-up research could thus be to add to the list of methods, in order to compile an elaborate yet comprehensive guide that is accessible to a broad group of engineers.

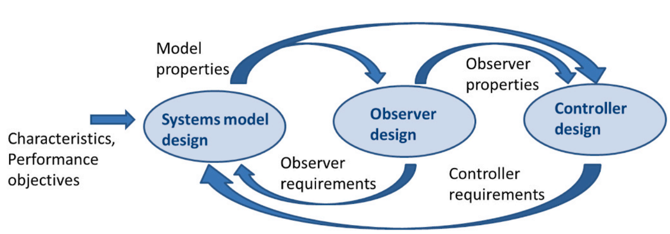

Fig. 7. The iterative procedure of selecting and designing a model, observer and controller. The arrows indicate information input for each design step. Model design is based on the system characteristics (e.g. technology, input, constraints) and performance objectives, as well as on the requirements imposed by the design of the observer and controller. Observer design is based on model properties and controller requirements. Controller design is based on the properties of the designed model, as well as on the observer.

6. Summary

This introductory overview explains the need for sustainable farming, the role of precision technology in modern farming and the demand for automation and decision support. Within this context, it clarifies the role of state observation and control methods in automation and decision support. This is followed by an overview of the basic requirements of a systems model, which provides the foundation for model-based observation and control methods. The overview subsequently shifts to a presentation of commonly used observation and control methods, along with a discussion and summary of the advantages and disadvantages of each method. The methods are then linked to specific management tasks and associated with the cognitive skills that are required in operational farm management. Finally, we present a procedural outline for method selection and design, which could serve as a basic guideline to support farming engineers. The procedures of model design and method selection (for state estimation methods, feedback control, and model predictive control) are illustrated with case studies (Supplementary Material).

The main outcomes emerging from this overview are as follows:

1. Prediction, observation and control are essential features for operational management support in farming systems (Section 1). Moreover, the required cognitive skills that make up the machine intelligence required for automation and decision support are linked to the methods discussed in this paper (Section 5.1).
2. Each method presented in this paper has a unique set of possible advantages and disadvantages (Table 1, Table 2, Table 3). This has the following implications:
   - o Farming systems are highly varied in terms of design, technology, location and performance objectives, and these variations translate into a specific set of errors, disturbances, constraints and performance criteria under which each system is operated. The selection of the best method, or combination of methods, thus requires an assessment for each particular case, based on weighing the advantages and disadvantages of each candidate method.
   - o Many of the advantages and disadvantages stem from a gap between reality and the theoretical assumptions underlying the methods. It is quite common for one or more assumptions on disturbances, dynamics, performance or other aspects to deviate from reality (as illustrated by the case studies), thus possibly resulting in a loss of performance in practice. The relationship between this theory-reality gap and possible performance loss therefore constitutes an important aspect of biosystems engineering.
3. In theory, the procedure of designing and selecting the model, observer and controller is not sequential, but iterative. This stems from the fact that model selection depends on observer requirements, while observer selection depends on model properties. Similar dependencies exist between observer and controller, as well as between controller and model (Fig. 7).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We would like to thank Dr Gjerrit Meinsma (Twente University, Netherlands); David Katzin, MSc (Wageningen University and Research, Netherlands); Dr Rachel van Ooteghem (Wageningen University and Research, Netherlands); and Prof. Michel Vellekoop (University of Amsterdam, Netherlands) for useful discussions and insightful reflections.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.envsoft.2021.105031.

References

Aerts, J.-M., Van Buggenhout, S., Vranken, E., Lippens, M., Buyse, J., Decuypere, E., Berckmans, D., 2003. Active control of the growth trajectory of broiler chickens based on online animal responses. Poultry Sci. 82 (12), 1853–1862.

Alessio, A., Bemporad, A., 2009. A Survey on Explicit Model Predictive Control. Nonlinear Model Predictive Control. Springer, pp. 345–369.

Alvarez, L.H.R., Shepp, L.A., 1998. Optimal harvesting of stochastically fluctuating populations. J. Math. Biol. 37 (2), 155–177.

Anderson, J.R., Dillon, J.L., 1992. Risk Analysis in Dryland Farming Systems. Food & Agriculture Org.

André, G., Berentsen, P.B.M., Engel, B., de Koning, C.J.A.M., Oude Lansink, A.G.J.M., 2010. Increasing the revenues from automatic milking by using individual variation in milking characteristics. J. Dairy Sci. 93 (3), 942–953.

Auernhammer, H., 2001. Precision farming — the environmental challenge. Comput. Electron. Agric. 30 (1), 31–43.

Azar, A.T., Serrano, F.E., 2015. Design and Modeling of Anti Wind up PID Controllers. Complex System Modelling and Control through Intelligent Soft Computations. Springer International Publishing, pp. 1–44. Q. Zhu and A. T. Azar. Cham.

Bayar, G., Bergerman, M., Koku, A.B., Konukseven, E.i., 2015. Localization and control of an autonomous orchard vehicle. Comput. Electron. Agric. 115, 118–128.

Berckmans, D., 2014. Precision livestock farming technologies for welfare management in intensive livestock systems. Rev. Sci. Technol. 33 (1), 189–196.

Berckmans, D., 2017. General introduction to precision livestock farming. Anim. Front. 7 (1), 6–11.

Bertsekas, D.P., 1995. Dynamic Programming and Optimal Control. Athena Scientific Belmont, MA.

Bo, L., Fan, Y., 2004. Design and simulation analysis of LQG controller of active suspension. Trans. Chin. Soc. Agric. Mach. 35 (1), 13–17.

Braumann, C.A., 1999. Variable effort fishing models in random environments. Math. Biosci. 156 (1), 1–19.

Chen, H.-F., Zhao, W., 2014. Recursive Identification and Parameter Estimation. CRC Press.

Coen, T., Saeys, W., Missotten, B., De Baerdemaeker, J., 2008. Cruise control on a combine harvester using model-based predictive control. Biosyst. Eng. 99 (1), 47–55.

Comasòlivas, R., Escobet, T., Quevedo, J., 2012. Automatic design of robust PID controllers based on QFT specifications. IFAC Proc. Vol. 45 (3), 715–720.

Day, W., 2005. Engineering precision into variable biological systems. Ann. Appl. Biol. 146 (2), 155–162.

De Mol, R., Keen, A., Kroeze, G., Achten, J., 1999. Description of a detection model for oestrus and diseases in dairy cattle based on time series analysis combined with a Kalman filter. Comput. Electron. Agric. 22 (2–3), 171–185.

Development, A.a.R., 2019. EU agricultural outlook: European agricultural labour and total income expected to decrease by 2030. Retrieved September 2020, 2020, from. https://ec.europa.eu/info/news/eu-agricultural-outlook-european-agricultural-labour-and-total-income-expected-to-decrease-by-2030_en.

Diangelakis, N.A., Oberdieck, R., Pistikopoulos, E.N., 2019. Explicit (Offline) Optimization for MPC. Handbook of Model Predictive Control. Springer, pp. 359–385.

Ding, Y., Wang, L., Li, Y., Li, D., 2018. Model predictive control and its application in agriculture: a review. Comput. Electron. Agric. 151, 104–117.

Dorf, R.C., Bishop, R.H., 2011. Modern Control Systems. Pearson.

Doyle, J., 1978. Guaranteed margins for LQG regulators. IEEE Trans. Automat. Contr. 23 (4), 756–757.

El Afou, Y., Guerbaoui, M., Ed-Dahhak, A., Outanoute, M., Lachhab, A., Belkoura, L., Bouchikhi, B., 2013. Design and implementation of lqg strategies for temperature control under greenhouse. Int. J. Eng. Adv. Technol. Stud. 1 (2).

El Ghoumari, M.Y., Tantau, H.J., Serrano, J., 2005. Non-linear constrained MPC: real-time implementation of greenhouse air temperature control. Comput. Electron. Agric. 49 (3), 345–356.

Føre, M., Frank, K., Norton, T., Svendsen, E., Alfredsen, J.A., Dempster, T., Eguiraun, H., Watson, W., Stahl, A., Sunde, L.M., 2017. Precision fish farming: a new framework to improve production in aquaculture. Biosyst. Eng. 173, 176–193.

Fournel, S., Rousseau, A.N., Laberge, B., 2017. Rethinking environment control strategy of confined animal housing systems through precision livestock farming. Biosyst. Eng. 155, 96–123.

Friedman, J., Hastie, T., Tibshirani, R., 2001. The Elements of Statistical Learning. Springer series in statistics, New York, NY, USA.

Gartley, E., Bevly, D.M., 2008. Online estimation of implement dynamics for adaptive steering control of farm tractors. IEEE ASME Trans. Mechatron. 13 (4), 429–440.

Gelb, A., 1974. Applied Optimal Estimation. MIT press.

Gonzalez, R.C., Woods, R.E., Eddins, S.L., 2004. Digital Image Processing Using MATLAB. Pearson Education India.

Goodchild, M.S., Jenkins, M.D., Whalley, W.R., Watts, C.W., 2018. A novel dielectric tensiometer enabling precision PID-based irrigation control of polytunnel-grown strawberries in coir. Biosyst. Eng. 165, 70–76.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT press.

Grewal, M.S., Andrews, A.P., 2014. Kalman Filtering: Theory and Practice with MATLAB. John Wiley & Sons.

Gutiérrez-Arias, J.M., de León-Puig, N.I.P., P érez-Osorio, G., Flores-Mena, E., Castañeda-Camacho, J., 2015. Optimal control law for the concentration of carbon dioxide in a tomato greenhouse and optimal-tuning PI control as LQR for the automatic device. Int. J. Eng. Innov. Technol. 5 (4), 1–11.

Hague, T., Tillett, N., 2001. A bandpass filter-based approach to crop row location and tracking. Mechatronics 11 (1), 1–12.

Halachmi, I., Edan, Y., Maltz, E., Peiper, U., Moallem, U., Brukental, I., 1998. A real-time control system for individual dairy cow food intake. Comput. Electron. Agric. 20 (2), 131–144.

Hemming, J., Bac, C., van Tuijl, B., Barth, R., Bontsema, J., Pekkeriet, E., van Henten, E., 2014. A Robot for Harvesting Sweet-Pepper in Greenhouses. AgEng 2014.

Hooper, A.W., 1988. Computer control of the environment in greenhouses. Comput. Electron. Agric. 3 (1), 11–27.

Horwood, J.W., 1996. Risk-sensitive optimal harvesting and control of biological populations. Math. Med. Biol. 13 (1), 35–71.

Innes, R., 2013. Economics of agricultural residuals and over fertilization: chemical fertilizer use, livestock waste, manure management, and environmental impacts. Nat. Resour. Environ. Econ. 2, 50–57.

Jawad, H., Nordin, R., Gharghan, S., Jawad, A., Ismail, M., 2017. Energy-efficient wireless sensor networks for precision agriculture: a review. Sensors 17 (8), 1781.

Julier, S., Uhlmann, J., Durrant-Whyte, H.F., 2000. A new method for the nonlinear transformation of means and covariances in filters and estimators. IEEE Trans. Automat. Contr. 45 (3), 477.

Julier, S., Uhlmann, J.K., Durrant-Whyte, H.F., 1995. A new approach for filtering nonlinear systems. American Control Conference. Proceedings of the 1995. IEEE.

Kahneman, D., 2003. Maps of bounded rationality: psychology for behavioral economics. Am. Econ. Rev. 93 (5), 1449–1475.

Kalman, R.E., 1960. A new approach to linear filtering and prediction problems. J. Basic Eng. 82 (1), 35–45.

Kamilaris, A., Prenafeta-Boldú, F.X., 2018. Deep learning in agriculture: a survey. Comput. Electron. Agric. 147, 70–90.

Kayacan, E., Kayacan, E., Ramon, H., Saeys, W., 2015. Towards agrobots: identification of the yaw dynamics and trajectory tracking of an autonomous tractor. Comput. Electron. Agric. 115, 78–87.

Keesman, K.J., 2011. System Identification: an Introduction. Springer Science & Business Media.

Kelley, C.T., 1999. Iterative Methods for Optimization. SIAM.

Krishnan, M., Foster, C.A., Strosser, R.P., Glancey, J.L., Sun, J.-Q., 2006. Adaptive modeling and control of a manure spreader for precision agriculture. Comput. Electron. Agric. 52 (1), 1–10.

Kurtulmus, F., Lee, W.S., Vardar, A., 2011. Green citrus detection using 'eigenfruit', color and circular Gabor texture features under natural outdoor conditions. Comput. Electron. Agric. 78 (2), 140–149.

Langson, W., Chryssochoos, I., Raković, S., Mayne, D.Q., 2004. Robust model predictive control using tubes. Automatica 40 (1), 125–133.

Li, Y., Ding, Y., Li, D., Miao, Z., 2018. Automatic carbon dioxide enrichment strategies in the greenhouse: a review. Biosyst. Eng. 171, 101–119.

Liao, C.-M., Feddes, J., 1993. Design of an airborne dust control system in ventilated animal housing. Appl. Math. Model. 17 (3), 133–140.

Liu, Y., Gupta, H.V.C.W., 2007. Uncertainty in hydrologic modeling: toward an integrated data assimilation framework. Water Resour. Res. 43 (7) (n/a-n/a).

López-Cruz, I.L., van Beveren, P.J.M., van Mourik, S., van Henten, E.J., 2017. Performance of Extended and Unscented Kalman Filters for State and Parameter Estimation of a Greenhouse Climate Model. International Society for Horticultural Science (ISHS), Leuven, Belgium.

Lü, H., Yu, Z., Zhu, Y., Drake, S., Hao, Z., Sudicky, E.A., 2011. Dual state-parameter estimation of root zone soil moisture by optimal parameter estimation and extended Kalman filter data assimilation. Adv. Water Resour. 34 (3), 395–406.

Mahmood, H.S., Hoogmoed, W.B., van Henten, E.J., 2012. Sensor data fusion to predict multiple soil properties. Precis. Agric. 13 (6), 628–645.

Mason, J., 2003. Sustainable Agriculture. Landlinks press.

McCarthy, A.C., Hancock, N.H., Raine, S.R., 2014. Simulation of irrigation control strategies for cotton using Model Predictive Control within the VARIwise simulation framework. Comput. Electron. Agric. 101, 135–147.

Mondaca-Duarte, F.D., van Mourik, S., Balendonck, J., Voogt, W., Heinen, M., van Henten, E.J., 2020. Irrigation, crop stress and drainage reduction under uncertainty: a scenario study. Agric. Water Manag. 230, 105990.

Mottram, T., 2016. Animal board invited review: precision livestock farming for dairy cows with a focus on oestrus detection. Animal 10 (10), 1575–1584.

Mouazen, A.M., Anthonis, J., Ramon, H., 2005. An automatic depth control system for online measurement of spatial variation in soil compaction, Part 4: improvement of compaction maps by using a proportional integrative derivative depth controller. Biosyst. Eng. 90 (4), 409–418.

Mourik, S.v., Vellekoop, M., Beveren, P.J. M.v., Ooster, A.v. t., Henten, E.J.v., 2016. Stochastic control of crop growth, a simulation study. In: Proceedings of the International Conference on Agricultural Engineering. Aarhus, Denmark.

Mourik, S.v., Zwart, H., Keesman, K.J., 2010. Switching input controller for a food storage room. Contr. Eng. Pract. 18 (5), 507–514.

Mul, M.F., van Riel, J.W., Roy, L., Zoons, J., André, G., George, D.R., Meerburg, B.G., Dicke, M., van Mourik, S., Koerkamp, P.W.G., 2017. Development of a model forecasting Dermanyssus gallinae's population dynamics for advancing Integrated Pest Management in laying hen facilities. Vet. Parasitol. 245, 128–140.
Niu, X.J., 2014. Research on the Temperature Control Algorithm of the Poultry Farm. Applied Mechanics and Materials. Trans Tech Publ.
Oliver, M., Bishop, T., Marchant, B., 2013a. Improving Food Security through Increasing the Precision of Agricultural Development. Precision Agriculture for Sustainability and Environmental Protection. Routledge.
Oliver, M., Bishop, T., Marchant, B., 2013b. An Overview of Precision Agriculture. Precision Agriculture for Sustainability and Environmental Protection. Routledge.
Phillips, J.D., Griffiths, M.W., 1987. The relation between temperature and growth of bacteria in dairy products. Food Microbiol. 4 (2), 173–185.
Pimentel, D., Giampietro, M., 1994. Implications of the limited potential of technology to increase the carrying capacity of our planet. Hum. Ecol. Rev. 1 (2), 248–251.
Prathyusha, K., Suman, M.C., 2012. Design of embedded systems for the automation of drip irrigation. IJAIEM 1 (2), 2319-4847.
Rajamani, M.R., 2007. Data-based Techniques to Improve State Estimation in Model Predictive Control. University of Wisconsin.
Rodríguez, F., Berenguel, M., Guzmán, J.L., Ramírez-Arias, A., 2015. Modeling and Control of Greenhouse Crop Growth. Springer.
Rohrs, C., Valavani, L., Athans, M., Stein, G., 1985. Robustness of continuous-time adaptive control algorithms in the presence of unmodeled dynamics. IEEE Trans. Automat. Contr. 30 (9), 881–889.
Salzmann, M., Teunissen, P., Sideris, M., 1991. Detection and Modelling of Coloured Noise for Kalman Filter Applications. Kinematic Systems in Geodesy, Surveying, and Remote Sensing. Springer, pp. 251–260.
Sanderink, F.E., Gerritsen, J.W., Koerkamp, P.W.G., van Mourik, S., 2017. Automatic detection of oestrus cows via breath sampling with an electronic nose: a pilot study. Biosyst. Eng. 156, 1–6.
Slaughter, D.C., Giles, D.K., Downey, D., 2008. Autonomous robotic weed control systems: a review. Comput. Electron. Agric. 61 (1), 63–78.
Song, X., Bokkers, E., van der Tol, P., Koerkamp, P.G., van Mourik, S., 2018. Automated body weight prediction of dairy cows using 3-dimensional vision. J. Dairy Sci. 101 (5), 4448–4459.
Song, X., Bokkers, E.A.M., van Mourik, S., Groot Koerkamp, P.W.G., van der Tol, P.P.J., 2019. Automated body condition scoring of dairy cows using 3-dimensional feature extraction from multiple body regions. J. Dairy Sci. 102 (5), 4294–4308.
Speetjens, S., Stigter, J., Van Straten, G., 2009. Towards an adaptive model for greenhouse control. Comput. Electron. Agric. 67 (1), 1–8.
Stuart, D., Schewe, R., McDermott, M., 2014. Reducing nitrogen fertilizer application as a climate change mitigation strategy: understanding farmer decision-making and potential barriers to change in the US. Land Use Pol. 36, 210–218.
Thompson, R., Gallardo, M., Valdez, L., Fernández, M., 2007. Using plant water status to define threshold values for irrigation management of vegetable crops using soil moisture sensors. Agric. Water Manag. 88 (1–3), 147–158.
Ullah, I., Kim, D., 2018. An optimization scheme for water pump control in smart fish farm with efficient energy consumption. Processes 6 (6), 65.
van Beveren, P., Bontsema, J., van Straten, G., van Henten, E., 2015a. Optimal control of greenhouse climate using minimal energy and grower defined bounds. Appl. Energy 159, 509–519.
Van Beveren, P.J.M., Bontsema, J., Van Straten, G., Van Henten, E.J., 2015b. Minimal heating and cooling in a modern rose greenhouse. Appl. Energy 137, 97–109, 0.
van Dixhoorn, I.D.E., de Mol, R.M., van der Werf, J.T.N., van Mourik, S., van Reenen, C. G., 2018. Indicators of resilience during the transition period in dairy cows: a case study. J. Dairy Sci. 101 (11), 10271–10282.
Van Henten, E.J., Van't Slot, D.A., Hol, C.W.J., Van Willigenburg, L.G., 2009. Optimal manipulator design for a cucumber harvesting robot. Comput. Electron. Agric. 65 (2), 247–257.
van Mourik, S., Alders, B., Helderman, F., van de Ven, L., Koerkamp, P.G., 2016. Predicting hairline fractures in eggs of mature hens. Poultry Sci. 96 (6), 1956–1962 pew411.
van Mourik, S., van Beveren, P.J., López-Cruz, I.L., van Henten, E.J., 2019. Improving climate monitoring in greenhouse cultivation via model based filtering. Biosyst. Eng. 181, 40–51.
van Straten, G., van Willigenburg, G., van Henten, E., van Ooteghem, R., 2010. Optimal Control of Greenhouse Cultivation. CRC press.
Vanthoor, B., De Visser, P., Stanghellini, C., Van Henten, E., 2011. A methodology for model-based greenhouse design: Part 2, description and validation of a tomato yield model. Biosyst. Eng. 110 (4), 378–395.
Verdouw, C.N., Beulens, A.J.M., van der Vorst, J.G.A.J., 2013. Virtualisation of floricultural supply chains: a review from an Internet of Things perspective. Comput. Electron. Agric. 99, 160–175.
Vincent, T.L., 1975. Pest management programs via optimal control theory. Biometrics 31 (1), 1–10.
Vroegindeweij, B.A., IJsselmuiden, J., van Henten, E.J., 2016. Probabilistic localisation in repetitive environments: estimating a robot's position in an aviary poultry house. Comput. Electron. Agric. 124, 303–317.
Walker, W.E., Harremoës, P., Rotmans, J., Van Der Sluijs, J.P., Van Asselt, M.B., Janssen, P., Krayer von Krauss, M.P., 2003. Defining uncertainty: a conceptual basis for uncertainty management in model-based decision support. Integrated Assess. 4 (1), 5–17.
Wallace, J.S., 2000. Increasing agricultural water use efficiency to meet future food production. Agric. Ecosyst. Environ. 82 (1–3), 105–119.
West, M., Harrison, J., 2006. Bayesian Forecasting and Dynamic Models. Springer Science & Business Media.
Xiong, Q., Cai, W.-J., He, M.-J., 2007. Equivalent transfer function method for PI/PID controller design of MIMO processes. J. Process Contr. 17 (8), 665–673.
Yu, S., Wuwei, C., Senren, H., 2004. Multi-body dynamic analysis of vehicle suspension and PID control. Trans. Chin. Soc. Agric. Mach. 35 (1), 4–7.
Zavaleta, L.R., Lacewell, R.D., Taylor, C.R., 1980. Open-loop stochastic control of grain sorghum irrigation levels and timing. Am. J. Agric. Econ. 62 (4), 785–792.
Zhang, N., Wang, M., Wang, N., 2002. Precision agriculture—a worldwide overview. Comput. Electron. Agric. 36 (2–3), 113–132.