computing@computingonline.net
www.computingonline.net

Print ISSN 1727-6209
On-line ISSN 2312-5381
International Journal of Computing

ACTOR-CRITIC REINFORCEMENT LEARNING FOR ENERGY OPTIMIZATION IN HYBRID PRODUCTION ENVIRONMENTS

Dorothea Schwung [1], Andreas Schwung [1], Steven X. Ding [2]

[1] Dept. of Automation Technology, South Westphalia University of Applied Sciences
Soest, Germany, schwung.dorothea@fh-swf.de, schwung.andreas@fh-swf.de
[2] Dept. of Automatic Control and Complex Systems, University of Duisburg-Essen
Duisburg, Germany, steven.ding@uni-due.de

Paper history:
Received 02 May 2019
Received in revised form 20 November 2019
Accepted 29 November 2019
Available online 31 December 2019

Keywords:
Machine learning;
self-learning;
actor-critic reinforcement learning;
radial-basis function neural networks;
manufacturing systems;
hybrid systems;
energy optimization.

Abstract: This paper presents a centralized approach for energy optimization in large scale industrial production systems based on an actor-critic reinforcement learning (ACRL) framework. The objective of the on-line capable self-learning algorithm is the optimization of the energy consumption of a production process while meeting certain manufacturing constraints like a demanded throughput. Our centralized ACRL algorithm works with two artificial neural networks (ANN) for function approximation using Gaussian radial-basis functions (RBF), one for the critic and another for the actor, respectively. This kind of actorcritic design enables the handling of both, a discrete and continuous state and action space, which is essential for hybrid systems where discrete and continuous actuator behavior is combined. The ACRL algorithm is exemplary validated on a dynamic simulation model of a bulk good system for the task of supplying bulk good to a subsequent dosing section while consuming as low energy as possible. The simulation results clearly show the applicability and capability of our machine learning (ML) approach for energy optimization in hybrid production environments.

1. INTRODUCTION

Energy has become a very valuable and discussed property in recent years. The main reason for this trend is the rethinking from an environmental polluting energy production to a green energy supply with the focus on renewable energy sources to reduce sustainably the emissions of detrimental greenhouse gases. This reorganization in the energy sector entails risks and costs which partly have to be borne also by the consumer, e.g. the industrial production sector. Therefore an energy efficient handling and facility operation is demanded, not least to be able to stay competitive on the market. These circumstances affect especially large-scale industrial plants with an extremely high number of energy consumers (like pumps, valves, conveyors etc.) as it is generally the case in process industry and basic material industry [1]. In this case, even minor energy savings can lead to decreasing emissions and costs [2, 3].

Generally, RL is a goal-oriented learning technique which learns the optimal policy by (long-term) rewarded trial-and-error interactions with the environment, imitating the natural learning behavior of a child or an animal. Machine learning (ML) techniques like RL became particularly popular with the success in playing the game of Go [4, 5] demonstrating super-human performance of the technical system. Considering practical real-life applications, the main benefits of RL methods are the on-line capability and additionally the capability to cope with uncertainties and changes in system dynamics [6], which makes the framework especially attractive for analytically hard describable (technical) problems.

In the literature actor-critic reinforcement learning (ACRL) methods that combine the strengths of actor-only and critic-only RL methods [7, 8], i.e. merging policy-based with value-based methods, are nowadays more present than ever before. They focus various research directions and different domains like spoken dialogue systems (SDS), i. e. task-completion dialogue policy learning with an adversarial advantage actor critic (A2C) approach [9] or ACRL with experience replay (ACER) for dialogue systems with large action spaces [10], a decentralized collaborative MARL approach based on ACRL methods especially for continuous state and action spaces [11] and ACRL for optimal control of multiple-model discrete-time systems [12], to just name a few. Particularly, ACRL with suitably chosen Gaussian radial-basis function neural networks (RBFNNs) as function approximators is efficient and notably suitable for continuous domains or hybrid systems [6, 13–15] as it is the case in process industry or manufacturing. The strength to cope with large continuous action-spaces within a hybrid system environment, is one of the most significant benefits of an RBFNN based ACRL structure.

Other important ANN types addressing the artificial intelligence research are spiking neural networks (SNNs) [16–18] and recurrent neural networks (RNNs) [19]. SNNs proceed by sequences of spikes and have their explicit strengths in applications that require very fast processing times of huge amounts of data [18], e.g. in the field of robotics. A large list of engineering applications for SNNs in combination with different learning scenarios, e.g. RL, can be found in [17]. However, SNNs are still difficult to train because by their nature they are not back-propagation capable. In [19] an approximate dynamic programming (ADP) approach for the energy management of a microgrid based on deep RNN learning is proposed, which guarantees convergence while using linear models to approximate the value function.

RL in general and recently Deep-RL using deep neural networks (DNN), has already been applied to energy management systems with special emphasis on distributed smart grids and microgrids [20, 21] and on electric vehicles [22, 23] or energy optimization in electric water heaters [24]. An example of using ANNs for the function approximation of a Q-function for RL in the context of energy optimization can be found in [25]. Especially ACRL approaches are presented in, e.g. [26] for improving variable speed wind turbine controllers to changing wind conditions dealing with continuous valued state and action spaces or [27] where a transfer actor-critic learning framework for energy efficient radio access networks is proposed. In contrast, the adequate application of ACRL techniques to energy optimization in the manufacturing and process industry domain with its inherent challenges has still open research questions. The learning set-up with an appropriate pre-elected statespace and action-set, well suited timings like episode duration and hyperparameter tuning as well as incorporating process constraints are very crucial and the basis for a successful learning behavior.

In this paper we present a centralized approach for energy optimization based on the ideas of ACRL with RBFNNs function approximators focusing the challenges of the application to hybrid manufacturing systems. We give a detailed description of the learning set-up for the actor and the critic network used in our ACRL algorithm. Furthermore we develop a bulk good process model of our physical laboratory testbed for co-simulation purposes which serves as application example for our approach. In relation to our exemplary plant we define the MDP for the energy optimization problem that can be scaled easily to larger systems. The gained results show typical learning behavior and outperform the baseline model with regard to the energy consumption and the throughput rate. A preliminary version of this paper has been presented in [1].

This paper is organized as follows. In Section 2 we state the learning problem for energy management and optimization in manufacturing systems. Section 3 describes our ACRL-based approach using Gaussian RBFNNs for function approximation. In Section 5 we present an application example for energy optimization using ACRL with RBFNNs and discuss the results obtained from a simulation model of a laboratory bulk good system. Section 6 gives the conclusions and points toward further work.

2. PROBLEM STATEMENT

The considered general structure of the production environment is illustrated in the schematic of Fig. 1. As illustrated, we consider a distributed production process with a number of possibly different subsystems interacting with each other. The interaction is assumed to take place on the physical level by exchanging energy and material flows and on the cyber level by exchanging information and control signals. To this end each subsystem has its own control system with sensoring and monitoring devices to measure its production performance and a certain number of energy consumers like electrical drives, valves and compressors to actuate the subsystem. The considered energy consumers have either discrete behavior like DOL-motors or on-off valves, continuous behavior like VSD-drives or hybrid behavior like e.g. vacuum pumps. We consider different forms of energy consumption like electrical energy or instrument air. The energy consumption of the consumers is assumed to be continuously measured by suitable energy metering devices in the local control systems and then communicated to a centralized control system for further analysis.

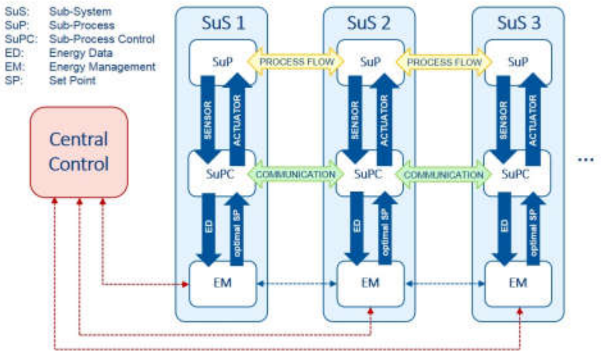

Figure 1 – General Production System Set-Up [29]

After describing the general system set-up, we will now state the problem to be solved:

We consider a distributed system $\mathcal{S}$ with $i=1\dots k$ subsystems $\mathcal{S}_i$ as illustrated in Fig. 1 for $k=3$. The system dynamics of the ith subsystem are given by

$$\dot{x}_i = f_i(x_i(t), x_{-i}(t), u_i(t), u_{-i}(t)), \qquad (1)$$

where $x_i \in \mathrm{R}^{n_i} \times \{0,1\}^{v_i}$ and $u_i \in \mathrm{R}^{m_i} \times \{0,1\}^{k_i}$ are the local states and actuators of the ith subsystem and f_i is a generally nonlinear function representing the system dynamics. Following the usual notation we denote the state and actuators of others than the ith subsystem by x_{-i} and u_{-i}. The performance of the ith subsystem is described by the performance output

$$y_i(t) = g_i(x(t), u(t)) \qquad (2)$$

with $y_i \in \mathrm{R}^{s_i}$ while the energy consumption of the ith subsystem is represented by

$$e_i(t) = h_i(u_i(t)) \qquad (3)$$

only depending on the local actuators. Furthermore, we assume necessary constraints on the state variables as

$$x_{\min} < x(t) < x_{\max} \qquad (4)$$

Then, the optimization problem is stated as follows. Given a predefined production episode $t=0,\dots T$, find the optimal energy consumption

$$\min_u \sum_{t=0}^{T} \sum_i e_i(t) \qquad (5)$$

$$s.t. \quad (1)-(4) \qquad (6)$$

$$\int_{t=0}^{T} r(y(t)) = Y, \qquad (7)$$

where Y_s is the required performance over the considered production episode previously defined. Note, that the previously scheduled production performance can depend either on the performance y_i of each module or on the performance of only a subset of modules. This relation is formally modeled by the function r. For instance, in certain processes only the last subsystems output is responsible for the overall performance while the other modules influence this output indirectly by suitable supply actions.

Some remarks to the previously defined problem are in order. The performance outputs $y_i(t)$ can be arbitrarily defined based on the given process objectives and available process measurements. Examples include product concentrations in chemical plants, mass flows in bulk good plants or processing times in manufacturing plants. The length of the considered production episode is closely related to these requirements as performance parameters might only be accessible after some processing time. Typical examples include batch operations. Hence, the episode should be at least as long as the processing times. The number of samples per episode should be determined such that the important dynamics of the energy consumption and the process parameters are represented. Particularly, the processing times and operation points of actuators strongly influence the energy consumption of the overall system and need to be carefully examined.

Note, that the problem description mainly focuses on discrete and hybrid processes where operations and system behavior are not solely continuous. Such hybrid systems containing discrete and continuous dynamics and actuation are quite common in the process and basic materials industries due to discontinuous and delaying components like buffers, reactors or conveyors as well as on-off actuators and actuators with discontinuous actions. We will introduce an example of such a process in Sec. 5.

3. ACTOR-CRITIC REINFORCEMENT LEARNING: AN INTRODUCTION

In general RL is a machine learning technique that is based upon the animal trial-and-error learning. The learner, also called agent, acts on its environment and learns from rewards gained from these interactions within a given time horizon called episode. Analytically, the environment can be formulated as a Markov decision process (MDP) with its states, possible actions to take and the resulting rewards as an evaluation of the chosen actions. Formally, the MDP is described by the tuple $(\mathcal{S}, \mathcal{A}, P, R, p_0)$ with:

- the set of states $\mathcal{S}$,
- the set of actions $\mathcal{A}$,
- the transition model $P : S \times A \times S \to [0\,1]$, such that $P(s_t, a_t, s_{t+1}) = p(s_{t+1} \,|\, s_t, a_t)$ is the transition probability from state s_t to the following state s_{t+1} by applying action a_t,
- the reward function $R : S \times A \times S \to \mathbb{R}$, which assigns a reward $r(s_t, a_t, s_{t+1})$ to each state transition $s_t \leftarrow s_{t+1}$,
- an initialization probability p_0 of the states.

The decision, which action to take next given the current state, depends on the agents policy $\pi : S \to A$. The policy can be chosen based on the agents past experiences or even randomly. The goal of the reinforcement learning problem is to find the optimal policy by interacting with the environment, i.e. the policy which results in the highest possible cumulated reward. Hence, we want to maximize the return

$$\rho^{\pi} = E[R \,|\, \pi] \tag{8}$$

with the discounted reward

$$R = \sum_t \gamma^t r_t, \tag{9}$$

where $0 \leq \gamma \leq 1$ is the discount factor.

For our optimization problem stated in Sec. 2, we choose an actor-critic (AC) framework with artificial neural network (ANN) function approximators in order to emerge a self-learning system behavior. This approach allows us to avoid a theoretically complicated solution for analyzing the condition for optimality, by using a neural network approximation within our ACRL algorithm, learning the unknown system dynamics. ACRL methods combine notions of policy iteration (PIT) with adaptive function approximation [8]. Compared to general Q-learning, the ACRL method has the advantages of reduced variance in function approximation, efficient computation in continuous domains and a high similarity to neural mechanisms in mammalian brains [28]. In contrast to the Q-learning approach in [29], where the state and action space has to be discretized, this approach allows to cope with not only a hybrid state but also a hybrid action space where continuous and discrete behavior are merged. The fundamental idea of the ACRL method is the partition into a critic part for policy evaluation (PE) and an actor part for policy improvement (PI). In our algorithm we use two normalized RBFNNs, one as policy approximator within the actor and another one as state-action value function approximator within the critic. In this context the critic evaluates the actor's policy using the SARSA(λ) method which updates the state-action value estimation and calculates a kind of temporal difference (TD) error between the state-action value at the next and the current state. Independent of the critic's PE, the actor updates the current followed policy according to its own assessment of the TD error with a second RBFNN. Fig. 2 gives a general overview of the ACRL algorithm structure.

ACRL is usually introduced with policy gradient methods augmented by a suitable evaluation of the policy. In policy gradient methods, a class of parameterized randomized policies $\{\pi_\theta(s,.), s \in \mathcal{S}, \theta \in \mathrm{R}^{d_1}\}$ is defined. Then the gradient of the average reward with respect to the policy parameters θ is estimated from the observed states, actions, and rewards. The policy is finally improved by adjusting its parameters in the direction of the estimated gradient. The average reward is usually defined as

$$J(\theta) = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \rho^{\pi}. \tag{10}$$

Hence, the optimal parameters are obtained from

$$\theta^{opt} = \text{argmax}_{\theta} J(\theta). \tag{11}$$

By means of the policy gradient theorem [30], the gradient can be calculated as

$$\nabla J(\theta) = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{a \in \mathcal{A}} \nabla \pi(s,a) A^{\pi}(s,a), \tag{12}$$

where

$$A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s) \tag{13}$$

is the advantage function, i.e. the difference between the state-action value function $Q^{\pi}(s,a)$ and the state value function $V^{\pi}(s)$ defined as

$$Q^{\pi}(s,a)=E[\sum_{t}\gamma^{t}r_{t}\,|\,s_{0},a_{0},\pi] \quad (14)$$

and

$$V^{\pi}(s)=\sum_{a\in\mathcal{A}}\pi(s,a)Q^{\pi}(s,a)\,. \quad (15)$$

The update of the policy parameters is finally obtained by

$$\theta_{t+1}=\theta_{t}+\alpha\nabla_{\theta}J(\theta) \quad (16)$$

with the learning rate α. Note, that the functions Q^{π} and V^{π} determine the expected reward to be gained when starting in state s and respectively, taking action a and then following policy π. Consequently, $Q^{\pi}\$,\$V^{\pi}$ and the advantage function A^{π} all allow to evaluate a certain policy and serve as critics during the policy learning. This ACRL variant is well known as advantage actor-critic (A2C) [31] and has even been extended to asynchronous advantage actor-critic (A3C) [32]. However, all the above mentioned functions are not available during learning but have to be estimated. Different approaches are possible including Monte-Carlo (MC) methods or temporal difference (TD) learning. In this work, we use the well known SARSA(λ)-algorithm

$$A^{\pi}(s,a)=\omega^{T}\varphi(s,a)\,, \quad (17)$$

where ω^{T} are the learning parameters and $\varphi(s,a)$ are in general, continuous differentiable nonlinear functions in the states and actions.

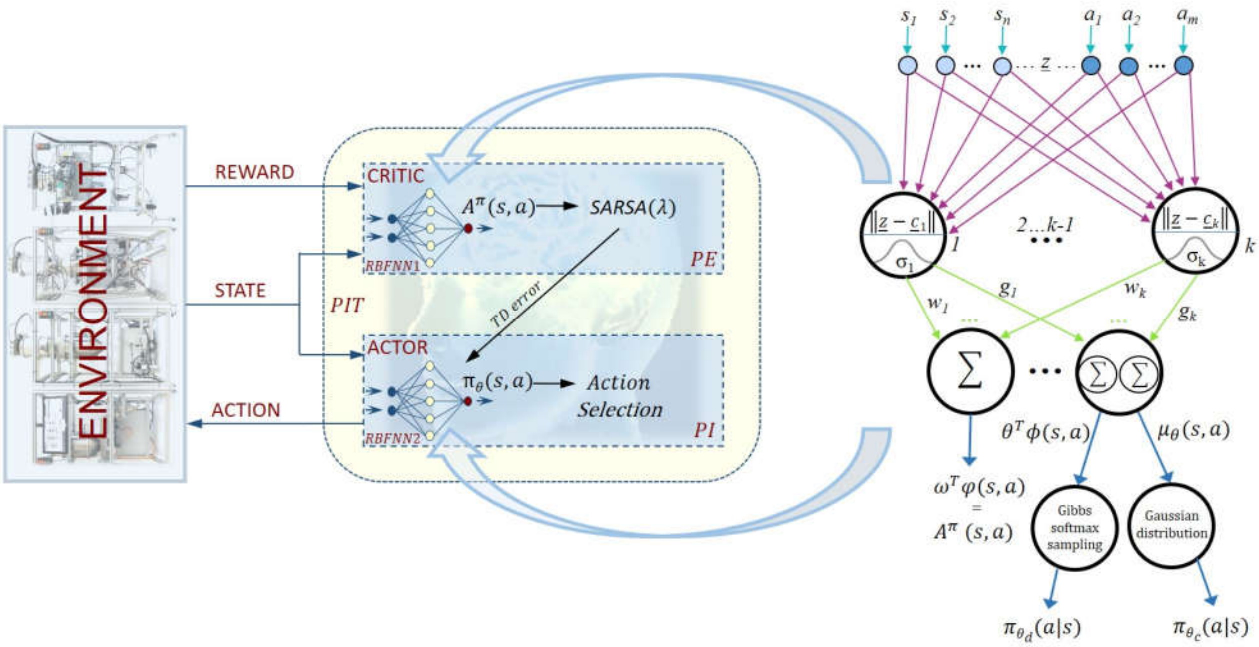

Figure 2 – Actor-Critic Reinforcement Learning Schematic with RBF Neural Networks for Function Approximation

4. RADIAL-BASIS FUNCTION NEURAL NETWORKS FOR FUNCTION APPROXIMATION IN A2C

In order to represent hybrid systems, we use radialbasis function neural networks (RBFNN) for function approximation within the critic and the actor, which should be chosen in a certain interdependency [7]. A simple RBFNN generally consists of an input layer, a hidden neural layer with RBFs and an output layer with linear neurons whose inputs are weighted. The advantage of RBFs for our application is the possibility to use a locally limited activation function (radial functions) like the Gaussian function with special approximation properties. The Gaussian function is defined as

$$y=e^{\frac{-p^{2}}{2\sigma^{2}}}=e^{-\frac{\|x-c\|^{2}}{2\sigma^{2}}}\,. \quad (18)$$

Hence, the weighted output function of the network is calculated to

$$y_{i}=\sum_{j=1}^{L}w_{i,j}\cdot e^{-\frac{\|x-c_{j}\|^{2}}{2\sigma_{j}^{2}}}$$

$$\Rightarrow A_i^{\pi}(s,a) = \sum_{j=1}^{L} \omega_{i,j} \cdot e^{-\beta_j \left\| \underline{z} - \underline{c}_j \right\|^2}, \qquad (19)$$

where $z = [s^T\ a^T]^T$, L is the number of basis functions, c_j is the mean and β_j the variance of the j-th basis function.

Hence, the normalized output, yielding accuracy improvement, can be written as

$$A_i^{\pi}(s,a) = \sum_{j=1}^{L} \omega_{i,j} \cdot \varphi(\left\| \underline{z} - \underline{c}_j \right\|), \qquad (20)$$

where

$$\varphi(\left\| \underline{z} - \underline{c}_j \right\|) = \frac{e^{-\beta_j \left\| \underline{z} - \underline{c}_j \right\|^2}}{\sum_{j=1}^{L} e^{-\beta_j \left\| \underline{z} - \underline{c}_j \right\|^2}}. \qquad (21)$$

For the sake of simplicity regarding a future PLC implementation, we reduce the learning task of the RBF network to a learning of the weights $w_{i,j}$, $\omega_{i,j}$ respectively, and define the other parameters, the centers $c_{j,k}$ and variances β_j, with special reference to the learning data pairs $\left(\tilde{\underline{x}}_i, \tilde{\underline{y}}_i\right)$, $\left(\tilde{\underline{z}}_i, \tilde{\underline{A}}_i^{\pi}(s,a)\right)$ respectively.

Finally, the resulting ACRL algorithm executes as follows:

1. Initialize learning parameters ω, θ, z to zero and choose first action a_1.
2. Execute the system using the chosen action a and observe state s_t and reward r_t.
3. Draw the next action from the distribution $\pi_{\theta}(s_t,.)$.
4. Calculate ω_t, z_t by executing SARSA($\lambda, s_{t-1}, a_{t-1}, r_t, s_t, a_t, \omega_{t-1}, z_{t-1}$)-algorithm.
5. Calculate policy parameters *θt* using Eq. (16) and (12).
6. Update states and actions.

Note, that by using the ideas of natural actor-critic (NAC) algorithms [30], the update law of the policy parameters can be further simplified to

$$\theta_{t+1} = \theta_t + \alpha \omega_t. \qquad (22)$$

In Step 3 of the above algorithm the actions have to be drawn from the policy distribution which has so far not been defined. As we will deal with hybrid action spaces, i.e. both discrete and continuous actions, the choice of the distribution has to be done differently for both classes of actions. To this end, we split the action set $\mathcal{A}$ into discrete $\mathcal{A}_d \in [0\,1]^{l_d}$ and continuous $\mathcal{A}_c \in \mathcal{R}^{l_c}$ action sets with $\mathcal{A} = \mathcal{A}_d \times \mathcal{A}_c$. Then, we draw the discrete and continuous actions independently from corresponding distributions. In the discrete case, we apply Gibb's sampling using the softmax function

$$\pi_{\theta_d}(a \mid s) = \frac{e^{\theta^T \phi(s,a)}}{\sum_{a' \in \mathcal{A}_d} e^{\theta^T \phi(s,a)}}. \qquad (23)$$

In the continuous case, we use the multivariate Gaussian distribution

$$\pi_{\theta_c}(a \mid s) = \frac{1}{\sqrt{(2\pi)^{l_c} \det(\Sigma_\theta)}} e^{-\frac{1}{2}(a - \mu_\theta(s,a))^T \Sigma_\theta^{-1} (a - \mu_\theta(s,a))} \qquad (24)$$

with positive definite matrix Σ_θ, often chosen to $\Sigma_\theta = \gamma I$ with $\gamma > 0$. The parametric mean functions $\mu_\theta(s,a)$ as well as the feature functions $\phi(s,a)$ are chosen as RBFs similar to $\varphi(s,a)$.

5. THE BULK GOOD SYSTEM: AN A2C APPLICATION

5.1 LABORATORY TESTBED

After introducing the general ACRL-approach, we will now focus on the application to a laboratory testbed as schematically illustrated in Fig. 3. As depicted, the testbed consists of four interacting modules forming a bulk good handling system. Modules 1 and 2 represent typical supply, buffer and transportation stations. Module 1 consists of a container and a continuously controlled belt conveyor from which the bulk good is carried to a mini hopper which is the interface to module 2. Module 2 consists of a vacuum pump, a buffer container and a vibration conveyor. The vacuum pump itself transports the material from module 1 into an internal container. The material is then released to the buffer container by a pneumatically actuated flap and then charged via the vibration conveyor into a mini hopper which is the interface to the dosing station module 3. It contains a further vacuum pump and a dosing unit composed of a buffer container with a weighing system and a rotary feeder. The dosed material is finally transported by a third vacuum pump to module 4 and then filled into transport boxes. Additionally, every module is equipped with its own PLC-based control system which communicate with each other via a suitable communication protocol. Each module has a set of sensors to monitor the modules state, particularly, each buffer is equipped with min/max level sensors and each mini hopper with overflow sensors. The electrical energy consumption is measured by energy metering modules. As the energy consumption of the vacuum pump and vibration conveyor is influenced by instrument air consumption, we take this ancillary into account. Note, that the testbed mimics to some extend typical large scale systems which are modularized in smaller subsystems with their own control systems and suitable communication interfaces. Besides, it is mentionable that such a system set-up is especially qualified for distributed control and decentralized optimization approaches. Furthermore, due to the system structure with different buffer containers as well as due to the inherent discontinuous behavior of the vacuum pumps, this process constitutes a typical hybrid system with a mixture of discrete and continuous behavior. For this reason a learning based energy management optimization is particularly beneficial allowing for enhancing the energy optimal operation strategy.

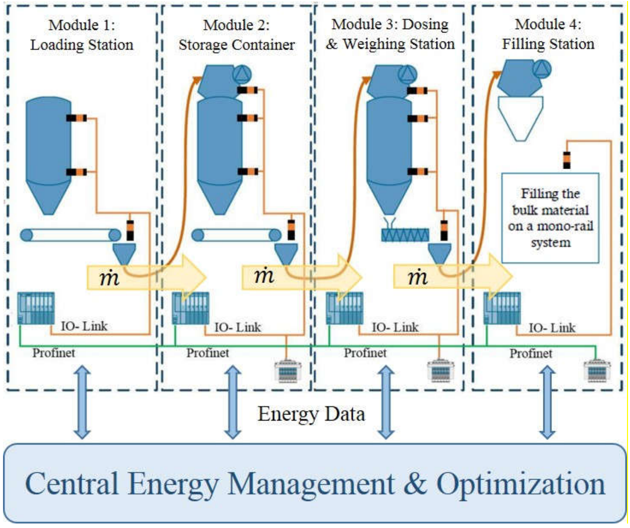
Module 1: Loading Station
Module 2: Storage Container
Module 3: Dosing & Weighing Station
Module 4: Filling Station
Filling the bulk material on a mono-rail system
ṁ
ṁ
ṁ
IO- Link
IO- Link
IO- Link
IO- Link
Profinet
Profinet
Profinet
Profinet
Energy Data
Central Energy Management & Optimization

Figure 3 – Laboratory Testbed Schematic and Modeling Set-Up

In the following experiments we will concentrate mainly on modules 1 and 2 which are the supply units for the subsequent dosing station. In particular, the target is to supply the dosing unit with the required amount of material continuously processed by the dosing station while keeping the energy consumption of all the actuators within the supply stations as low as possible. Note, that there exists no pre-programmed sequence of actuator operations in the PLC when starting with the learning process. However, to assure a safe operation of the process, some interlocks to avoid buffer overflows are implemented at the basic PLC level using the available sensor information described above [29].

5.2 BULK GOOD PROCESS MODELING

To allow for fast development times and reduce the effort to gain machine data, we additionally derive a simulation model. Hence, the ACRL can be analyzed using a co-simulation approach before testing at the real plant. To this end, we briefly state the basic system equations of the physical model based on mass-flow balance equations as well as the equations for the energy consumption used in the reward calculation. Note that the simulation model is set up as a modular model where subsystems can arbitrarily be plugged in and removed.

To define the mass-flow balances, we define a state equation for each storage element, i.e. buffer and hopper using the sum of differences between input mass flow $\dot{m}_{in,i}$ and output mass-flow $\dot{m}_{out,i}$ as

$$\rho \dot{V}_i = \rho A_i \frac{dh_i}{dt} = \sum_i \Delta \dot{m}_i = \sum_i (\dot{m}_{in,i} - \dot{m}_{out,i}) \tag{25}$$

More specific, for the first module we derive the massflow differential equations (MFDE) for the buffer (bf1) and hopper (hp1) respectively:

$$\begin{aligned} \rho A \frac{dh_{bf1}}{dt} &= \dot{m}_{in,bf1} - \dot{m}_{out,bf1} \\ &= \dot{m}_{in,bf1} - \alpha_{bc} n_{bc} \end{aligned} \tag{26}$$

$$\begin{aligned} \rho A \frac{dh_{hp1}}{dt} &= \dot{m}_{in,hp1} - \dot{m}_{out,hp1} \\ &= \alpha_{bc} n_{bc} - f_{vac1}(t_{start}, t_{stop}) \end{aligned} , \tag{27}$$

where n_{bc} denotes the speed of the belt conveyor, α_{bc} is the mass flow coefficient and $f_{vac1}(t_{start}, t_{stop})$ represents the nonlinear mass flow of the vacuum pump for the activation time $t_{stop} - t_{start}$.

Similar equations can be obtained for Module 2:

$$\begin{aligned} \rho A \frac{dh_{bf2}}{dt} &= \dot{m}_{in,bf2} - \dot{m}_{out,bf2} \\ &= f_{vac1}(t_{start}, t_{stop}) - \alpha_{vc} n_{vc} \end{aligned} \tag{28}$$

$$\begin{aligned} \rho A \frac{dh_{hp2}}{dt} &= \dot{m}_{in,hp2} - \dot{m}_{out,hp2} \\ &= \alpha_{vc} n_{vc} - f_{vac2}(t_{start}, t_{stop}) \end{aligned} \qquad (29)$$

and Module 3:

$$\begin{aligned} \rho A \frac{dh_{bf3}}{dt} &= \dot{m}_{in,bf3} - \dot{m}_{out,bf3} \quad . \\ &= f_{vac2}(t_{start}, t_{stop}) - \alpha_{rot} n_{rot} \end{aligned} \qquad (30)$$

Herein, n_{vc} and n_{rot} denote the speed and α_{vc} and α_{rot} the mass flow coefficients of vibration conveyor and rotary valve, respectively. The model output is $\dot{m}_{output} = \dot{m}_{out,bf3}$.

For the modeling of the energy consumption, we have to deal with two different energy sources, namely electrical energy E_{ei} and pneumatic energy E_{pi} in terms of instrument air. The consumptions for Module 1-3 are calculated as follows

$$E_{e1} = f_{bc}(n), \quad E_{p1} = f_{vac1}(t_{start}, t_{stop}) \qquad (31)$$

$$E_{e2} = f_{vib}(n), \quad E_{p2} = f_{vac2}(t_{start}, t_{stop}) \qquad (32)$$

$$E_{e3} = f_{rot}(n), \quad E_{p3} = f_{vac3}(t_{start}, t_{stop}) \qquad (33)$$

resulting in the overall energy consumption

$$E_{ges}(a) = \sum_{y} E_{y}(a) \, . \qquad (34)$$

Note that all above listed functions and constants used in the simulation model rely on measurements and regression analysis based on real process data.

Additionally, it is worth mentioning that the vacuum pumps exhibit a specific behavior. After switching on, first an evacuation period occurs where conveying of product is not possible. Afterward the conveyed product follows a polynomial function until the buffer in the vacuum pump is full which results in a sudden drop of mass flow. The ACRL-algorithm should be able to cope with this specific system behavior.

5.3 ENERGY MANAGEMENT SET-UP FOR THE A2C APPROACH

After the introduction to ACRL in Sec. III and a detailed description of our application example, we will now formulate our ACRL-learning framework for energy management and optimization. As application example we built a bulk good process simulation model of our laboratory testbed which has a modularized system architecture like the presented system setup illustrated in Fig. 1. To this end, we need to define the MDP for the energy optimization problem by specifying the system states, the set of possible actions and the rewards to be gained. The definition of the state and action space can be seen in Table 1.

Table 1. Definition of States and Actions

No.	Sensor	State
1	Buffer Station 1	Full Empty
2	Buffer Station 2	Full Empty
3	Mini Hopper 1	Sensor Reading
4	Mini Hopper 2	Sensor Reading

No.	Actuator	Action
1	Vacuum Pump 1	Priming Time Setting
2	Vacuum Pump 2	Priming Time Setting
3	Belt Conveyor	Speed Setting
4	Vibration Conveyor	On Off

As we are interested in energy optimization during an industrial process the state of the MDP need to mainly represent the energy flows in the system as described in Sec. 2. To this end, we assign a set of states $\mathcal{S}_{i,j}$ to each sensor measurement of the *i*th subsystem. A typical examples of such a state set is $\mathcal{S}_{i,j} = \{\text{full}, \text{empty}\}$ for the discrete states of the buffer sensors. However, also continuously operating sensors with more than two states are located in the system, like the sensors in the mini hoppers. These continuous sensor states are covered by Gaussian basis functions in the hidden layer of the RBF network. Finally, the resulting set of states of subsystem *i* than yields $\mathcal{S}_i = \mathcal{S}_{i,1} \times \ldots \times \mathcal{S}_{i,n}$.

The definition of the action space $\mathcal{A}_{i,j}$ is done by the actuators behavior which also can be either discrete or continuous. Continuous actuator behavior is captured with Gaussian basis functions likewise.

Furthermore the rewards for each state transition have to be defined. The appropriate definition of rewards is of major importance as the energy optimization problem has to fulfill different partly counteracting objectives [29]. On the one hand, the energy consumption should be minimized, but on the other hand, the plant has to supply at least a certain product amount, which costs energy. From the energy point of view a standstill would be the optimal solution. Hence, the reward function should contain both, the energy consumption during the sample period $e_{i,j}(t)$ and necessary process specifications like a certain product amount in a given time period. For this reason the reward for the ith subsystem is calculated as follows

$$r_i(s,a) = m_i(t) - \sum_j e_{i,j}(t) - p_i(t) \qquad (35)$$

where $m_i(t)$ is the mass flow of the module, $\sum_j e_{i,j}(t)$ is the energy consumption of all actuators of the module and $p_i(t)$ is a term penalizing overflow of the buffer and mini hopper.

5.4 RESULTS

In this section we present the results of our ACRL approach applied to a bulk good process simulation model.

The results are gained with the following parameter settings: discount factor $\gamma = 0.9$, learning rate $\alpha = 0.1$, vanishing for high number of episodes, and trace decay rate $\lambda = 0.9$. The setting of the centers and variances necessary for determination of the Gaussian functions rely on measurements at the laboratory bulk good system. Note that empirical investigations revealed, that variations in the location and the width of the RBFs do not have a significant influence on the results. Furthermore we fix the number of the Gaussian functions to two for discrete behavior and to three for the continuous case.

The resulting learning curves in Fig. 4 and Fig. 5 show the energy consumption and volume output over the number of episodes, respectively, where one episode comprises 15s. As indicated, a typical learning behavior with a notable exploration period at the beginning can be observed, followed by more exploitation till around learning episode 250. From episode 250 to 450 no visible changes occur anymore which is an indicator for reaching the optimal operating sequence. In the end, obviously a cyclic process sequence is the optimal result gained from the RBF-A2C learning algorithm, which is quite plausible considering the type of actuators in the bulk good handling process, in particular the vacuum pumps. The vacuum pumps exhibit a specific operation with a short evacuation period at the beginning of an operation sequence where no material can be transported, followed by a suction period where the material is transported linearly with the suction time. Thus, the operation behavior of the vacuum pumps is periodic by nature and as they are the dominant actuators in the system, it can be assumed that they have the biggest influence on the process itself. Hence, the baseline model also shows this kind of system behavior. In comparison to the baseline, the energy consumption is considerably reduced (about 22%) while generating a slightly higher volume output, which is a noticeable improvement. An obvious reason for this could be a deceleration or increased shut-off times of the conveyors in contrast to a continuous operation in the baseline. In Fig. 6 the gained reward during the learning procedure is shown. Due to the reward definition, where a multi-objective balancing between three parameters is required (high throughput demand, low energy, no overflow) in combination with the periodic process behavior, it has not to converge necessarily to the highest end value but shows also periodic behavior. In relation to the graphs of the energy consumption and the volume output, until episode 200 the exploration is clearly visible. After episode 200 the reward signal becomes more and more consistent and finally ends up in a periodic graph.

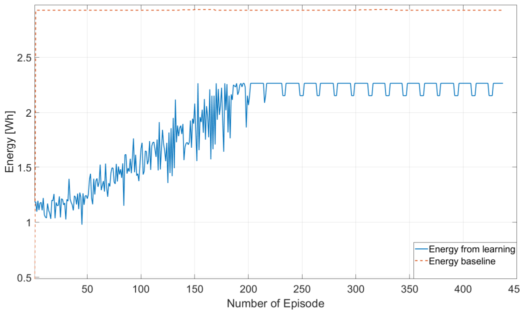

Figure 4 – Energy from ACRL vs. Energy Baseline over Episodes

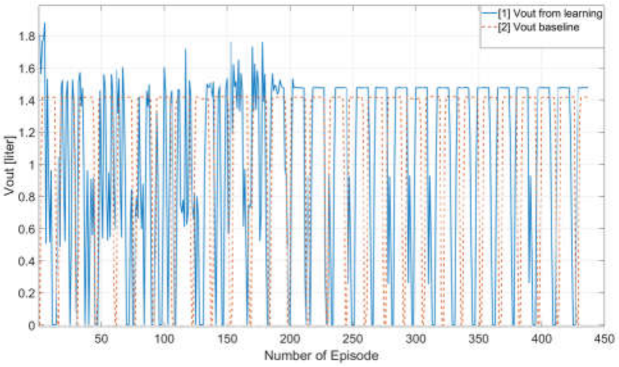

Figure 5 – Volume Output from ACRL vs. Volume Output Baseline over Episodes

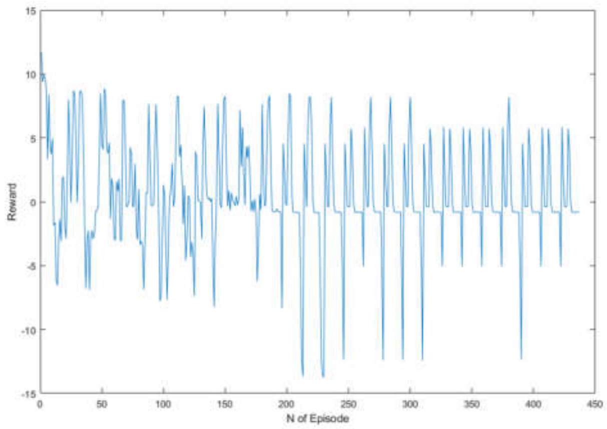

Figure 6 – Reward over Episodes

6. CONCLUSIONS

We here introduced a novel approach for energy optimization in large scale industrial process plants. The approach is based on a formulation of the optimization problem in form of an advantage actor-critic reinforcement learning (A2C) with RBFNNs as function approximators that enables to account for hybrid system behavior. The approach is applied to a bulk good process simulation model with very promising results. Hence, the energy consumption of the production process is minimized compared to the baseline model by learning from subsequent operation sequences while maintaining the production quality and performance. In future research, the developed A2C approach can be implemented on an industrial PLC for validation on the real laboratory bulk good testbed. Moreover, as the current approach requires a centralized learning process, a distributed approach for the ACRL-problem, potentially involving game theoretical ideas, leading to a game-based coordination of the multi-agent system (MAS), could be a forward-looking topic for further research activities. In this context, also the investigation of the deep deterministic policy gradient (DDPG) method [34] would be an interesting issue for a continuative research direction.

7. REFERENCES

[1] D. Schwung, A. Schwung and S. X. Ding, "On-line energy optimization of hybrid production systems using actor-critic reinforcement learning," *Proceedings of the 9th IEEE International Conference on Intelligent Systems*, 2018, pp. 147-154.

[2] A. Cannata, S. Karnouskos and M. Taisch, "Energy efficiency driven process analysis and optimization in discrete manufacturing," *Proceedings of the 35th Annual Conference of IEEE Industrial Electronics*, 2009, pp. 4449-4454.

[3] E. Oh and S.-Y. Son, "Toward dynamic energy management for green manufacturing systems," *IEEE Communications Magazine*, vol. 54, issue 10, pp. 74-79, 2016.

[4] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[5] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, Ma. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel and D. Hassabis "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[6] B. Kiumarsi, K. G. Vamvoudakis, H. Modares and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2042-2062, 2018

[7] V. R. Konda and J. N. Tsitsiklis, Actor-critic Algorithms, *Advances in Neural Information Processing Systems (NIPS)*, MIT Press, 2000.

[8] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, Cambridge, MA, USA: MIT Press, 1998.

[9] B. Peng, X. Li, J. Gao, J. Liu, Y.-N. Chen, K.-F. Wong, "Adversarial advantage actor-critic model for task-completion dialogue policy learning," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 6149-6153.

[10] G. Weisz, P. Budzianowski, P.-H. Su and M. Gasic, "Sample efficient deep reinforcement learning for dialogue systems with large action spaces," *IEEE/ACM Transactions on Audi, Speech and Language Processing*, vol. 26, no. 11, pp. 2083- 2097, 2018.

[11] K. Zhang, Z. Yang and T. Basar, "Networked multi-agent reinforcement learning in continuous spaces," *Proceedings of the IEEE Conference on Decision and Control (CDC)*, 2018, pp. 2771-2776.

[12] J. Skach, B. Kiumarsi, F. L. Lewis and O. Straka, "Actor-critic off-policy learning for optimal control of multiple-model discrete-time systems," *IEEE Transactions on Cybernatics*, vol. 48, no. 1, pp. 29-40, 2018.

[13] C. G. Li, M. Wang, Z. J. Huang and Z. F. Zhang, "An Actor-critic reinforcement learning algorithm based on adaptive RBF network," *Proceedings of the 8th International Conference on Machine Learning an Cybernetics*, 2009, pp. 984-988.

[14] H. van Hasselt and M. A. Wiering, "Reinforcement learning in continuous action spaces," *Proceedings of the IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning (ADPRL)*, 2007, pp. 272-279.

[15] Y. Wu, H. Wang, B. Zhang and K.-L. Du, "Using radial basis function networks for function approximation and classification," *International Scholarly Research Network (ISRN) Applied Mathematics*, vol. 2012, pp. 1-34, 2011.

[16] K. Voutsas and J. Adamy, "A biologically inspired spiking neural network for sound source lateralization," *IEEE Transactions on Neural Networks*, vol. 18, no. 6, pp. 1785-1799, 2007.

[17] F. Ponulak and A. Kasinski, "Introduction to spiking neural networks: Information processing, learning and applications," *Acta Neurobiologiae Experimentalis*, vol. 71, pp. 409-433, 2011.

[18] J.L. Lobo, J. Del Ser, A. Bifet and N. Kasabov, "Spiking neural networks and online learning: An overview and perspectives," *Neural Networks*, vol. 121, no. 1, pp. 88-100, 2020.

[19] P. Zeng, H. Li, H. He and S. Li, "Dynamic energy management of microgrid using approximate dynamic programming and deep recurrent neural network learning," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 4435-4445, 2019.

[20] R. Leo, R. S. Milton and S. Sibi, "Reinforcement learning for optimal energy management of a solar microgrid," *Proceedings of the IEEE Global Humanitarian Technology Conference – South Asia Satellite (GHTC-SAS)*, 2014, pp. 183-188.

[21] E. Mocanu, D. C. Mocanu, P. H. Nguyen, A. Liotta, M. E. Webber, M. Gibescu and J. G. Slootweg, "On-line building energy optimization using deep reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3698-3708, 2019.

[22] T. Liu, Y. Zou, D. Liu and F. Sun, "Reinforcement learning of adaptive energy management with transition probability for a hybrid electric tracked vehicle," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 12, pp. 7837-7846, 2015.

[23] X. Qi, Y. Luo, G. Wu, K. Boriboonsomsin and M. J. Barth, "Deep reinforcement learning-based vehicle energy efficiency autonomous learning system," *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1228-1233.

[24] F. Ruelens, B. J. Claessens, S. Quaiyum, B. De Schutter, R. Babuska and R. Belmans, "Reinforcement learning applied to an electric water heater: From theory to practice," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3792-3800, 2018.

[25] C. Liu and Y. L. Murphey, "Optimal power management based on Q-learning and neuro-dynamic programming for plugin hybrid electric vehicles," *IEEE Transactions on Neural Networks and Learning Systems*, Early Access, pp. 1-13, 2019.

[26] B. Fernandez-Gauna, U. Fernandez-Gamiz and M. Graña, "Variable speed wind turbine controller adaptation by reinforcement learning," *Integrated Computer-Aided Engineering*, vol. 24, no. 1, pp. 27-39, 2017.

[27] R. Li, Z. Zhao, X. Chen, J. Palicot and H. Zhang, "TACT: A transfer actor-critic learning framework for energy saving in cellular radio access networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 4, pp. 2000-2011, 2014.

[28] M. Patacchiola, *Research on Reinforcement Learning*, April 2018. [Online]. Available at: https://mpatacchiola.github.io/blog/2017/02/11/dissectingreinforcement-learning-4.html.

[29] D. Schwung, T. Kempe, A. Schwung and S. X. Ding, "Selfoptimization of energy consumption in complex bulk good processes using reinforcement learning," *Proceedings of the 15th IEEE International Conference on Industrial Informatics*, 2017, pp. 231-236.

[30] S. Bhatnagar and R. Sutton and M. Ghavamzadeh and M. Lee, "Natural actor-critic algorithms," *Automatica*, vol. 45, issue 11, pp. 2471-2482, 2009.

[31] P.-H. Su, P. Budzianowski, S. Ultes, M. Gasic and S. Young "Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management," *arXiv:1707.00130v2*, 2017.

[32] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Harley, T. P. Lillicrap, D. Silver and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," *Proceedings of the 33rd International Conference on Machine Learning*, vol. 48, 2016, pp. 1928-1937.

[33] G. A. Rummery and M. Niranjan, *On-line Q-learning using Connectionist Systems*, Technical Report CUED/F-INFENG/TR 166, Engineering Department, Cambridge University, 1994.

[34] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments," *arXiv:1706.02275v3*, 2018.

Dorothea Schwung *received the Dipl.-Ing. degree in electrical engineering and information technology from the University of Duisburg Essen, Duisburg, Germany, in 2010. From 2010 to 2015, she was a technical project manager for automation & control strongly involved in R&D topics with MAN Diesel & Turbo SE, Oberhausen, Germany. From 2015 to 2016, she was a software project manager in the software development department with Behr-Hella Thermocontrol, Lippstadt, Germany. Since 2016, she has been a scientist and teacher with the South Westphalia University of Applied Sciences, Soest, Germany and is currently working toward the Ph.D. degree with the University of Duisburg-Essen, Duisburg, Germany. Her research interests include machine learning with special focus on reinforcement learning for technical multi-agent systems and its relation to game theory.*

Andreas Schwung *received the Ph.D. degree in electrical engineering from the Technische University of Darmstadt, Darmstadt, Germany, in 2011. From 2011 to 2015, he was a technical project manager for automation & control with MAN Diesel & Turbo SE, Oberhausen, Germany. Since 2015, he has been a Professor of automation technology at the South Westphalia University of Applied Sciences, Soest, Germany. His research interests include model-based control, networked automation systems, and intelligent data analytics with applications in manufacturing, process industry, and electromobility.*

Steven Ding *received the Ph.D. degree in electrical engineering from the Gerhard Mercator University of Duisburg, Germany, in 1992. From 1992 to 1994, he was a R&D engineer at Rheinmetall GmbH. From 1995 to 2001, he was a professor of control engineering at the University of Applied Science Lausitz in Senftenberg, Germany, and served as vice president of this university during 1998 – 2000. Since 2001, he has been a chair professor of control engineering and the head of the Institute for Automatic Control and Complex Systems (AKS) at the University of Duisburg-Essen, Germany. His research interests are model-based and data-driven fault diagnosis, control and fault-tolerant systems as well as their applications in industry with a focus on automotive systems, chemical processes and renewable energy systems.*