

# Discrete-Step, Quasi-Newton Extremum Seeking Control for Multivariable Real-Time Optimization

A. Lange\* R. King\*,<sup>1</sup>

\* Chair of Measurement and Control, Department of Process Engineering, Technische Universität Berlin, Strasse des 17. Juni 135, 10623 Berlin, Germany

---

**Abstract:** Extremum seeking control is a well known approach for multivariate real-time optimization of dynamic systems. In classical extremum seeking control schemes, the estimated gradient of the process's steady-state map is continuously integrated towards a local optimum. Gradient estimation can be done by a combination of low- and high-/bandpass filters. Advanced approaches have been developed that use extended Kalman filters, which allow for a jointed estimation of the multidimensional gradient. Within this work, a discrete-step real-time optimization scheme is investigated that is derived from the prevalent quasi-Newton method of numerical optimization. Gradient estimation is implemented by the identification of a local linear dynamic model. A significantly faster convergence to the optimum compared to classical extremum seeking is shown for an academic Hammerstein example and for the optimization of the power consumption within a multistage compressor simulation.

*Keywords:* Extremum seeking control, Real-time optimization, Multivariable control system, Nonlinear control system, Autoregressive model, Compressor

---

## 1. INTRODUCTION

As a long known real-time optimization approach, the extremum seeking control (ESC) is suitable for determining an extremum of the steady-state map within a dynamic process (Tan et al., 2010). Only a limited knowledge of the underlying system is required, such as, stability, the existence of an extremum in the static input-output map, and an approximate quantification of the relevant timescales. After the proof of stability by Krstić and Wang (2000) the research and application of this approach has been revived, see, for example, the pressure rise maximization in an axial flow compressor (Wang et al., 2000), the separation control on a high-lift configuration (Becker et al., 2007), and the active drag reduction of bluff bodies and car models (Henning et al., 2008), to name just three examples, all of which are related to aerodynamic flows. The basic component of the classical ESC is the estimation of the local gradient of the static input-output map (*static gradient*) using, e.g., a harmonic excitation and high-/low-pass filters. The gradient descent algorithm approaches a local minimum in opposite direction of the gradient. Therefore, the integral of the estimated gradient is continuously fed back. Gelbert et al. (2012) proposed an advanced algorithm for gradient estimation in a dual-input single-output (DISO) ESC using an extended Kalman-filter (EKF). Ghaffari et al. (2012) investigated the optimization of multiple-input single-output (MISO) systems with a Newton-based approach. Additionally to the  $n$ -dimensional gradient, with  $n$  representing the number of input variables, the

$n \times n$ -dimensional inverse Hessian matrix is estimated through the integration of a Riccati differential equation. It is shown that a continuous feedback of their product leads to a faster convergence compared to the classical gradient based ESC. Zhang and Ordóñez (2007) developed an ESC control scheme combining numerical optimization algorithms that provide search candidates to the extremum and a state regulator designed to regulate the state towards the search direction. The discrete optimization allows a faster convergence of the extremum seeking control and abandons the slow continuous time integration of the gradient.

In this work, a discrete real-time optimization approach is presented which is based on the well known quasi-Newton method. The estimation of the gradient takes place only at certain points in time and can be calculated from a fitted local linear dynamic model, as Krishnamoorthy et al. (2019) proposed for the single-input single-output (SISO) case. With the cyclic estimation of the gradient, the Hessian matrix is updated iteratively using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm, which is widespread in the domain of numerical optimization.

Within the context of this article, the term *continuous* refers to a continuously evolving controller-output, even though the implementation of the gradient estimation and the integration might take place in the time-discrete domain. In contrast, the developed quasi-Newton ESC conducts *discrete* optimization steps at a predefined time-grid that mainly depends on the system time constants rather than the sampling time.

The paper is structured as follows: A review of continuous multivariate extremum seeking control is given in Section

---

<sup>1</sup> Corresponding author. Tel.: +49 30 314 24720; fax: +49 30 314 21129. E-mail address: Rudibert.King@TU-Berlin.de (R. King).

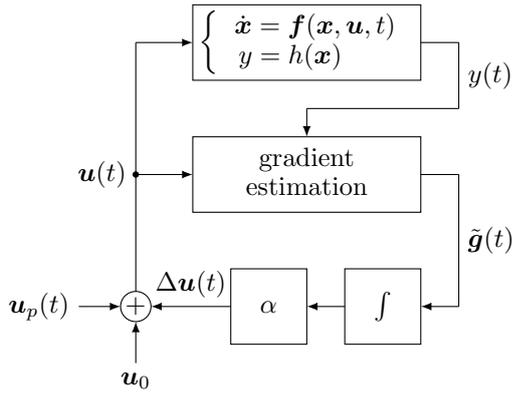


Fig. 1. Block diagram of a MISO extremum seeking feedback scheme for optimization of the output  $y$

2. The classical as well as an advanced algorithm for the multivariate gradient estimation are briefly presented. Section 3 gives an outline of the functionality of the quasi-Newton ESC. After motivating the approach, it is decomposed into its basic steps and explained in more detail. For the purpose of evaluation and comparison of the presented ESC algorithms two examples are investigated in Section 4. The first example is a simple academic system and the second one deals with a multistage compressor model.

## 2. CONTINUOUS MISO ESC

Intending to minimize the system output  $y(t)$ , the key element of the continuous gradient-based ESC is the gradient estimation of the quasi-static map  $y_s = \mathbf{F}(\mathbf{u}_s)$ , see Fig. 1. In order to obtain information about the gradient  $\mathbf{g} = (dy_s/d\mathbf{u}_s)^\top$ ,  $\mathbf{g} \in \mathbb{R}^n$ , the system input  $\mathbf{u}(t)$  is superposed with a vector of harmonic perturbations, e.g.,

$$\mathbf{u}_p = a[\sin \omega_1 t, \cos \omega_1 t, \sin \omega_2 t, \cos \omega_2 t, \dots]^\top. \quad (1)$$

Note, that there is only one output available for the estimation of the  $n$ -dimensional gradient. This requires the perturbation-input vector to be separated in phase and/or frequency. It can easily be shown that only a phase shift of  $90^\circ$  ensures an explicit separation. To assure a quasi-steady-state behavior of the system, the perturbation frequencies are limited by the largest time constant of the process  $T_{\max}$ , so that

$$1/T_{\max} \gg \omega_1 > \omega_2 > \dots \quad (2)$$

As the convergence speed of the ESC directly depends on the perturbation frequency, this results in a limit for the overall rate of convergence.

### 2.1 Classical ESC

In order to compute the estimated static gradient  $\tilde{\mathbf{g}}$  within the classical extremum-seeking control

- the mean of  $y(t)$  is removed by using a high-pass or band-pass,
- a demodulation is performed by multiplication with the perturbations  $u_{p,i}(t)/a$  and

- the remaining periodic parts are damped by a low-pass filter

for each input  $u_{p,i}$ ,  $i = 1, \dots, n$ . A more detailed presentation and a design guideline for classical ESC was shown by Xiao et al. (2014).

For the multivariate optimization, the classical ESC approach can basically be seen as  $n$  independent SISO controllers. The main disadvantage of this method is that the coupling between the inputs is not taken into account.

### 2.2 EKF based ESC

The use of an extended Kalman-filter allows the *joined* estimation of the gradient, which results in a faster overall convergence of the ESC. Here, we extend and slightly modify the approach of Gelbert et al. (2012) to the MISO case and introduce the idea briefly. As states  $\tilde{\mathbf{x}}$  of the EKF, the entries of the local gradient vector are chosen. The EKF estimates the static gradient in conjunction and therefore takes the input-coupling explicitly into account. If the integration constant  $\alpha$  is chosen to be small, the estimated static gradient  $\tilde{\mathbf{g}}$ , thus the states of the EKF  $\tilde{\mathbf{x}}$ , can be approximated as constant. This leads to the discrete-time state equation

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

Referring to the first order Taylor series expansion of the static function  $y_s = \mathbf{F}(\mathbf{u}_s)$  around the mean actual input vector  $\mathbf{u}_0 + \Delta\mathbf{u}(t)$ , results in the affine expression

$$y_s = y_0 + \mathbf{g}|_{\mathbf{u}_0 + \Delta\mathbf{u}(t)} \cdot (\mathbf{u}_s - \mathbf{u}_0 - \Delta\mathbf{u}(t)). \quad (4)$$

A proper linear measurement equation can be derived, by simply subtracting two time-points

$$\underbrace{\Delta y_s(k)}_{y_s(k) - y_s(k-k_\Delta)} = \underbrace{\Delta \mathbf{u}_s^\top(k)}_{(\mathbf{u}_s(k) - \mathbf{u}_s(k-k_\Delta))^\top} \cdot \tilde{\mathbf{x}} + v_k. \quad (5)$$

In order to respect  $\tilde{\mathbf{x}}(k)$  and  $\Delta y(k)$  being stochastic processes, Gaussian white noise  $\mathbf{w}_k \in \mathbb{R}^n$  and  $v_k \in \mathbb{R}$  are assumed.

The parameter  $k_\Delta$  is chosen, such that the time shift corresponds to  $3/4$  of the shortest perturbation period

$$k_\Delta = \frac{1}{T_s} \frac{3}{4} \frac{2\pi}{\omega_1}, \quad (6)$$

with  $T_s$  being the sampling time (Gelbert et al., 2012).

Nonlinear local observability can be shown for the system (3), (5).

## 3. DISCRETE-STEP, QUASI-NEWTON ESC

Inspired by the quasi-Newton method for unconstrained nonlinear programming the discrete-step, quasi-Newton ESC (QNEC) is proposed here. Quasi-Newton algorithms require only the gradient of the objective function to be supplied at each iterate to produce superlinear convergence. The prevalent quasi-Newton algorithm is the BFGS method. For a detailed specification of the BFGS method

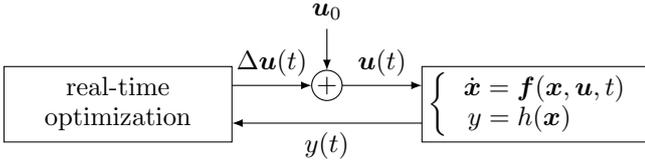


Fig. 2. Block diagram of discrete-step, quasi-Newton ESC scheme for optimization of the output  $y$

see for example Nocedal and Wright (2006). To use this algorithm within the context of real-time optimization, some adaptations have to be made. The main challenge thereby is coping with system dynamics.

The control scheme of the QNESC is shown in Fig. 2. One real-time optimization cycle  $\kappa$  can be decomposed into three basic steps:

- (1) execution of a Newton step,
- (2) evaluation of the static gradient,
- (3) iterative update of the estimated inverse Hessian.

In the following, these three steps are explained in more detail and adapted for the purpose of real-time optimization.

### 3.1 Newton step

Based on the current estimation of the inverse Hessian  $\tilde{\mathbf{H}}(\kappa)$  and the gradient  $\tilde{\mathbf{g}}(\kappa)$  of the steady-state map, the search direction is computed by

$$\mathbf{p}(\kappa) = -\tilde{\mathbf{H}}(\kappa)\tilde{\mathbf{g}}(\kappa). \quad (7)$$

The new iteration becomes

$$\Delta\mathbf{u}(\kappa + 1) = \Delta\mathbf{u}(\kappa) + \beta_\kappa\mathbf{p}(\kappa), \quad (8)$$

where the step length  $\beta_k$  is chosen to satisfy the Wolfe conditions, which ensure a sufficient decrease in the objective function and rule out unacceptably short steps, see Nocedal and Wright (2006). The simple and popular *backtracking* method has been chosen as the linesearch strategy, where the step length  $\beta_k$  is stepwisely decreased from a starting value  $\beta_0$ , until the condition

$$\mathbf{F}(\Delta\mathbf{u}(\kappa) + \beta_\kappa\mathbf{p}(\kappa)) \leq \mathbf{F}(\mathbf{u}(\kappa)) + c\beta_\kappa\tilde{\mathbf{g}}^\top(\kappa)\mathbf{p}(\kappa) \quad (9)$$

is satisfied. In some cases it might be appropriate to delimit the step size, to avoid large steps that may result from a bad estimation of the inverse Hessian, especially in the beginning of the algorithm.

As we have to cope with a dynamic system, it becomes clear that we have to wait for the system to settle after a stepwise input change, before condition (9) can be evaluated. To appraise an adequate waiting time, the following consideration might be helpful: If the system dynamic can be approximated as a first order low pass with the dominant system time constant of  $T_{\text{dom}}$ , then a waiting time of  $5 \cdot T_{\text{dom}}$  ensures the system output to reach 99.3% of its stationary value, waiting  $6 \cdot T_{\text{dom}}$  even results in 99.8%.

### 3.2 Gradient estimation

To compute the gradient of the objective function in numerical optimization, finite differences are often used. For real-time optimization of multivariate dynamic systems, this approach is highly impractical, as one has to wait for the system to converge after every step ( $n + 1$  times in case of forward difference or  $2n + 1$  times for central differences). Instead a local linear dynamic model around the current operating point is identified here, by additionally perturbing the system input with a pseudo random binary sequence (PRBS). After mean-centering the time series of the input  $\mathbf{u}(t)$  and the measured output  $y(t)$ , a MISO autoregressive model with exogenous input (ARX) model of the form

$$\mathbf{A}(q)y(t) = \mathbf{B}_1(q)u_1(t) + \dots + \mathbf{B}_n(q)u_n(t) + e(t) \quad (10)$$

can be fitted, with  $q$  being the shift operator, and

$$\mathbf{A}(q) = 1 + a_1q^{-1} + \dots + a_{n_a}q^{-n_a}, \quad (11)$$

$$\mathbf{B}_i(q) = 1 + b_{i,1}q^{-1} + \dots + b_{i,n_b}q^{-n_b+1}. \quad (12)$$

Note, that the parameters  $n_a$  and  $n_b$  have to be chosen to match the systems dynamical behavior.

The gradient of the steady-state map can be seen as the static gain of the dynamic model, that can be calculated by

$$\tilde{\mathbf{g}} = \begin{pmatrix} \mathbf{B}_1(1)/\mathbf{A}(1) \\ \vdots \\ \mathbf{B}_n(1)/\mathbf{A}(1) \end{pmatrix}. \quad (13)$$

A detailed introduction to the identification of linear models is given by Grimble et al. (2008).

### 3.3 Iterative Hessian update

The search direction (7) within the quasi-Newton algorithm depends on the inverse of the Hessian matrix. Instead of estimating the Hessian and calculating the inverse in every iteration, one can directly estimate the inverse Hessian by the update formula

$$\tilde{\mathbf{H}}(\kappa + 1) = \begin{bmatrix} \mathbf{I} - \rho(\kappa)\boldsymbol{\xi}(\kappa)\boldsymbol{\chi}^\top(\kappa) \\ \mathbf{I} - \rho(\kappa)\boldsymbol{\chi}(\kappa)\boldsymbol{\xi}^\top(\kappa) \end{bmatrix} \tilde{\mathbf{H}}(\kappa) + \rho(\kappa)\boldsymbol{\xi}(\kappa)\boldsymbol{\xi}^\top(\kappa) \quad (14)$$

with

$$\boldsymbol{\xi}(\kappa) = \Delta\mathbf{u}(\kappa + 1) - \Delta\mathbf{u}(\kappa), \quad (15)$$

$$\boldsymbol{\chi}(\kappa) = \tilde{\mathbf{g}}(\kappa + 1) - \tilde{\mathbf{g}}(\kappa) \text{ and} \quad (16)$$

$$\rho(\kappa) = \frac{1}{\boldsymbol{\chi}^\top(\kappa)\boldsymbol{\xi}(\kappa)}. \quad (17)$$

In order to avoid overshooting during the first iterations, it is suggested to choose a small positive definite matrix as the initial  $\tilde{\mathbf{H}}(0)$ . It can be shown that the estimated inverse Hessian will be positive definite whenever  $\tilde{\mathbf{H}}(0)$  is positive definite.

### 3.4 Algorithm

Even though the method cannot be explained in every single detail here, Algorithm 1 gives an idea on how the three basic steps can be implemented and how they operate together.

---

#### Algorithm 1 Discrete-step, quasi-Newton ESC

---

▷ #  $\Delta \mathbf{u}$  is applied to the system  
▷ #  $y$  is the measured system output

**Input:**  
initial inverse Hessian  $\tilde{\mathbf{H}}_0$   
initial step length  $\beta_0$   
Wolfe-condition parameter  $c_1$   
backtracking parameter  $c_2$   
wait time for stationarity  $T_w$

- 1: wait  $T_w$
- 2:  $\tilde{\mathbf{g}} \leftarrow \text{ESTIMATEGRADIENT}$
- 3:  $\tilde{\mathbf{H}} \leftarrow \tilde{\mathbf{H}}_0$   
▷ # loop over optimization cycles
- 4: **loop**
- 5:    $\beta \leftarrow \beta_0/c_2$
- 6:    $\Delta \mathbf{u}_0 \leftarrow \Delta \mathbf{u}$
- 7:    $y_0 \leftarrow y$
- 8:    $\tilde{\mathbf{g}}_0 \leftarrow \tilde{\mathbf{g}}$   
▷ # compute search direction
- 9:    $\mathbf{p} = -\tilde{\mathbf{H}}\tilde{\mathbf{g}}$   
▷ # backtracking linesearch
- 10:   **repeat**
- 11:      $\beta \leftarrow c_2\beta$
- 12:      $\Delta \mathbf{u} \leftarrow \Delta \mathbf{u}_0 + \beta\mathbf{p}$
- 13:     wait  $T_w$
- 14:     **until**  $y \leq y_0 + c_1\beta\tilde{\mathbf{g}}^T\mathbf{p}$
- 15:      $\tilde{\mathbf{g}} \leftarrow \text{ESTIMATEGRADIENT}$   
▷ # update inverse Hessian
- 16:      $\boldsymbol{\xi} \leftarrow \Delta \mathbf{u} - \Delta \mathbf{u}_0$
- 17:      $\boldsymbol{\chi} \leftarrow \tilde{\mathbf{g}} - \tilde{\mathbf{g}}_0$
- 18:      $\rho \leftarrow 1/(\boldsymbol{\chi}^T\boldsymbol{\xi})$
- 19:      $\tilde{\mathbf{H}} \leftarrow [\mathbf{I} - \rho\boldsymbol{\xi}\boldsymbol{\chi}^T]\tilde{\mathbf{H}}[\mathbf{I} - \rho\boldsymbol{\chi}\boldsymbol{\xi}^T] + \rho\boldsymbol{\xi}\boldsymbol{\xi}^T$
- 20:   **end loop**

---

- 1: **function** ESTIMATEGRADIENT
- 2:   superpose PRBS on system input  $\Delta \mathbf{u}(t)$
- 3:   measure system output  $y(t)$
- 4:   mean-center  $\Delta \mathbf{u}(t)$  and  $y(t)$
- 5:    $\text{arx} \leftarrow \text{identify ARX model}$
- 6:   **return** static gain of  $\text{arx}$
- 7: **end function**

---

## 4. EXAMPLES

### 4.1 Hammerstein model

The first example is a simple Hammerstein model that is composed of a quadratic function and a low-pass filter, see Fig. 3.

The quadratic matrix  $\mathbf{P}$  is positive definite, therefore the quadratic function  $f(\boldsymbol{\mu}) = \boldsymbol{\mu}^T\mathbf{P}\boldsymbol{\mu}$  has a global minimum at  $\boldsymbol{\mu} = \mathbf{0}$ . The paraboloid is shifted by the offsets  $\mathbf{u}_0$  and  $y_0 = \mathbf{u}_0^T\mathbf{P}\mathbf{u}_0$ , to have an optimal input  $\mathbf{u}^* \neq \mathbf{0}$  and to match the initial condition of the low-pass filter

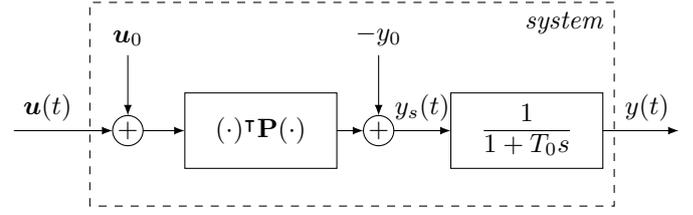


Fig. 3. Block diagram of a simple Hammerstein model

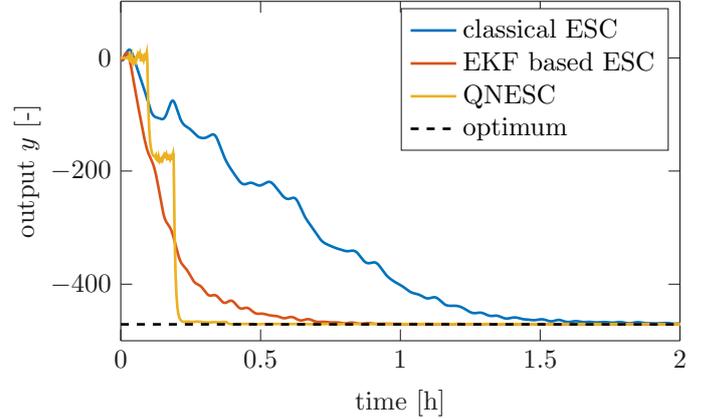


Fig. 4. Simulation results of the three presented ESC methods for the Hammerstein model

$y(0) = y_s(0) = 0$  at the same time. This makes  $\mathbf{u}^* = -\mathbf{u}_0$  the minimizer of the system with the optimal output  $y^* = -y_0$ .

A comparison of the three proposed ESC methods for the chosen Hammerstein model is presented in Fig. 4. All applied ESC approaches converge to the optimal solution. The EKF based ESC is approximately two times faster compared to the classical ESC, also the transition is much smoother. The classical algorithm is very sensitive to the integration constant  $\alpha$ . The integration constant of the EKF based approach  $\alpha_{\text{EKF}} = 5 \cdot 10^{-5}$  leads to unstable behavior for the classical approach. That is why the integration constant of the classical algorithm was chosen five times smaller ( $\alpha_{\text{classical}} = 1 \cdot 10^{-5}$ ).

It can be seen that the proposed quasi-Newton ESC approach has a significantly faster convergence compared to the continuous methods. The discrete optimization method takes less than half a time unit to converge to the optimal solution, whereas the continuous controllers need approximately one and two units respectively.

The capability of the proposed QNESC could clearly be demonstrated with this example. Also the discrete operation scheme of the QNESC, consisting of the superposition of the PRBS and the discrete optimization steps, can be seen in Fig. 4. All required system constants and ESC parameters for this simulation can be found in the Appendix A.

### 4.2 Multistage compressor

The second example deals with the regulation and optimization of the power consumption of a multistage radial compressor, and, therefore, has a technical relevance. A simulation model of a five stage main air compressor for

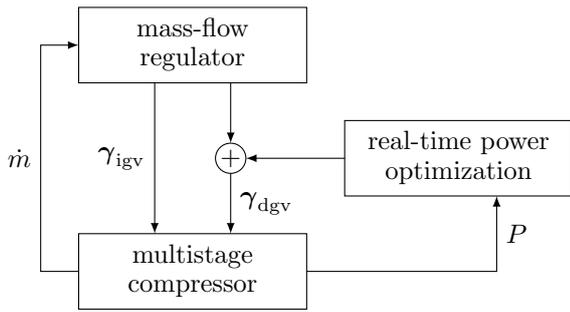


Fig. 5. Block diagram of the process-control design for the multistage compressor with independent adjustable inlet guide vanes and adjustable diffuser guide vanes for every stage

an air separation process including intercoolers was analytically derived from thermodynamical laws. A compressor stage is characterized by two static compressor-maps that specify the pressure gain and the efficiency of the compression depending on the mass-flow and the adjustable inlet guide vanes (igv) and adjustable diffuser guide vanes (dgv). The dynamic behavior of the system is mainly determined by the dynamics of the intercoolers. The dynamical mass and heat balances of the heat exchangers are solved within the simulation. The heat-transfer is modeled by a logarithmic temperature difference approach.

The control-objective is to adjust the overall mass-flow of the compressor at the operating point of maximum efficiency, which is equivalent to the point of least power consumption. The regulation of one stage influences the neighboring stages, therefore it affects the overall mass-flow and power consumption.

Fig. 5 shows the schematic process-control design, consisting of a mass-flow regulator and a real-time power optimization. The mass-flow controller uses the static compressor-maps to predetermine the optimal combinations of the manipulated variables  $\gamma_{igv}$  and  $\gamma_{dgv}$  that guarantee a certain mass-flow with maximum efficiency at every stage. Deviations of the actual mass-flow from the reference are compensated for with an integral controller. Thus, the mass-flow regulator can be seen as a combination of open-loop control with an integral closed-loop controller.

With respect to uncertainties of the model and disturbances, the compressor control with only the mass-flow regulator may lead to a non-optimal operation in terms of efficiency. In order to increase the overall efficiency and, therefore, reduce the energy costs of the air compression, a real-time power optimization is applied. The power optimization does not use any a priori information, as these are fully exploited within the mass-flow regulator already. For the purpose of power optimization, the mentioned ESC approaches have been implemented on the multistage compressor model. A dominant time constant of about 20 s has been identified for the mass-flow regulation. As mentioned before, the ESC must be significantly slower than the underlying closed-loop mass-flow control, hence, the fastest excitation frequency for the continuous ESC is set to  $\omega_1 = 2\pi/(300\text{ s})$  and the bandwidth of the PRBS signal for the discrete ESC was determined to  $\omega_{PRBS} = 2/(20\text{ s})$ . For the reason of comparison, the integration constants  $\alpha$  of the continuous approaches have the same value.

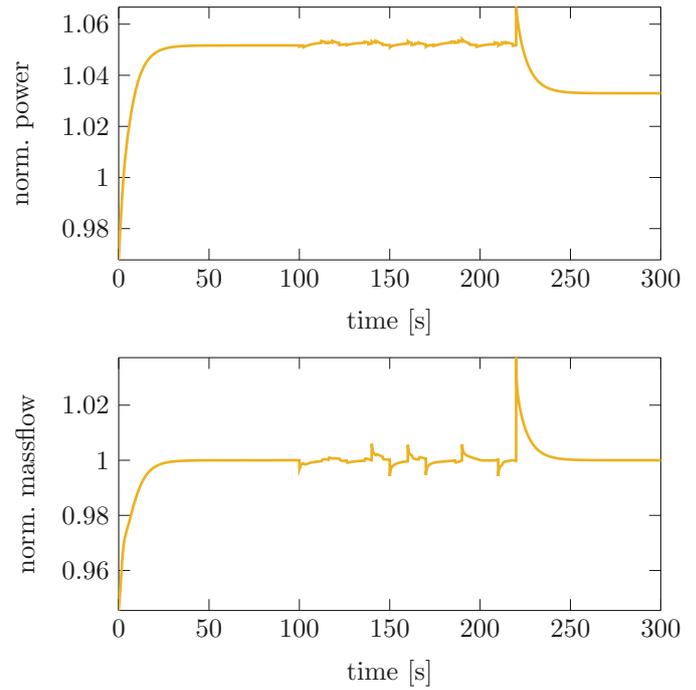


Fig. 6. Simulation results of the discrete-step QNESC for multistage compressor model

In order to compare the real-time optimization approaches, a scenario was investigated, where all five applied diffuser guide vanes have a permanent offset  $\gamma_{dgv,off} = -5^\circ$ . Fig. 6 shows the first 300 s of the simulation for the QNESC. The mass-flow is normalized to its reference value and the power is normalized to the optimal power for the reference mass-flow. Because of the constant disturbance, the mass-flow in the beginning of the simulation is less than the reference, hence, the dedicated compression power is low. With the actual mass-flow being regulated to its reference value, the compression power rises within 30 s. Due to the dgv-offset, the operation is not optimal in terms of power consumption. After the waiting time of 100 s the PRBS is superposed, followed by the first optimization step at 220 s that reduces the power consumption, once the mass-flow reaches its reference.

A comparison of the ESC methods is presented in Fig. 7. All approaches are able to reduce the dedicated compression power, thus saving energy costs for the compressor operation. The classical ESC takes about 3 h to converge, whereas the EKF based ESC is about three times faster and operates generally smoother. The proposed QNESC converges in less than 0.5 h, making it highly suitable for practical applications.

## 5. CONCLUSIONS

By adapting the well known quasi-Newton optimization method to multivariate dynamic systems a real-time optimization has been developed that converges significantly faster than both classical and advanced ESC approaches. For the purpose of gradient estimation, a local linear dynamic model is identified, which allows a determination of the accuracy of the estimated gradient by evaluating the prediction error of the identified model. The Hessian matrix is iteratively updated with the established BFGS-

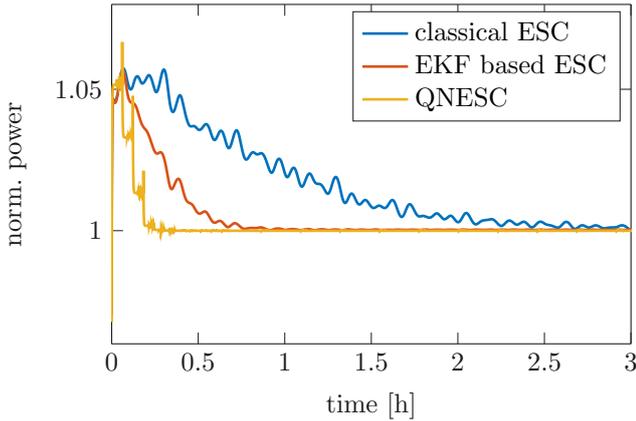


Fig. 7. Comparison of simulation results of the three presented ESC methods for multistage compressor mode

algorithm. The application examples for a simple Hammerstein model and the technically relevant compression model clearly show the benefit of the proposed discrete-step QNESC compared to the continuous methods. Additionally, the performance of the continuous approaches is very sensitive to the integration constant  $\alpha$ . A low  $\alpha$  leads to slow convergence whereas a high  $\alpha$  may cause instability. The QNESC has a small set of tuning parameters, therefore it is more robust and easy to adjust. In future works, constrained systems can be investigated by expanding the algorithm with methods of constrained nonlinear optimization, like sequential quadratic programming.

#### Appendix A. PARAMETER HAMMERSTEIN EXAMPLE

System parameters:

$$\mathbf{P} = \begin{pmatrix} 7 & 2 & -1 & -3 & -3 \\ 2 & 7 & 1 & -1 & -3 \\ -1 & 1 & 4 & 2 & 5 \\ -3 & -1 & 2 & 10 & 0 \\ -3 & -3 & 5 & 0 & 11 \end{pmatrix}$$

$$\mathbf{u}_0 = (3 \quad -1 \quad 2 \quad -5 \quad 3)^\top$$

$$T_0 = 20 \text{ s}, T_s = 0.1 \text{ s}$$

Continuous ESC parameters:

$$w_1 = 2\pi/(200 \text{ s}), w_2 = 2\pi/(267 \text{ s}), w_3 = 2\pi/(356 \text{ s}),$$

$$a = 0.1$$

Classical ESC parameters (low-pass and band-pass filter):

$$G_{\text{LP}} = 1/(1 + T_{\text{LP}}s), G_{\text{BP}} = 2\omega_{\text{BP}}s/(s + \omega_{\text{BP}})^2$$

$$T_{\text{LP},i} = 1/(0.1 \cdot \omega_i), \omega_{\text{BP},i} = \omega_i, \alpha = -1 \cdot 10^{-5}$$

EKF based ESC parameters:

$$\mathbf{Q}_{\text{EKF}} = 0.1 \cdot \mathbf{I}, R_{\text{EKF}} = 1, \mathbf{P}_{0,\text{EKF}} = 0.1 \cdot \mathbf{I}, T_w = 100 \text{ s},$$

$$\alpha = -5 \cdot 10^{-5}$$

QNESC parameters:

$$a_{\text{PRBS}} = 0.1, \Delta t_{\text{PRBS}} = 240 \text{ s}, \omega_{\text{PRBS}} = 3/(20 \text{ s}),$$

$$n_a = 1, n_b = 1, \mathbf{H}_0 = 0.01 \cdot \mathbf{I}, \beta_0 = 1, c_1 = 10^{-4}, c_2 = 0.8$$

#### ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial support of the Kopernikus project SynErgie by the Federal Ministry of Education and Research (BMBF) and the project supervision by the project management organization Projektträger Jülich.

#### REFERENCES

- Becker, R., King, R., Petz, R., and Nitsche, W. (2007). Adaptive closed-loop separation control on a high-lift configuration using extremum seeking. *AIAA Journal*, 45(6), 1382–1392. doi:10.2514/1.24941.
- Gelbert, G., Moeck, J.P., Paschereit, C.O., and King, R. (2012). Advanced algorithms for gradient estimation in one- and two-parameter extremum seeking controllers. *Journal of Process Control*, 22(4), 700–709. doi:10.1016/j.jprocont.2012.01.022.
- Ghaffari, A., Krstić, M., and Nešić, D. (2012). Multivariable newton-based extremum seeking. *Automatica*, 48(8), 1759–1767. doi:10.1016/j.automatica.2012.05.059.
- Grimble, M.J., Johnson, M.A., Garnier, H., and Wang, L. (2008). *Identification of Continuous-time Models from Sampled Data*. Springer London, London. doi:10.1007/978-1-84800-161-9.
- Henning, L., Becker, R., Feuerbach, G., Muminovic, R., King, R., Brunn, A., and Nitsche, W. (2008). Extensions of adaptive slope-seeking for active flow control. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 222(5), 309–322. doi:10.1243/09596518JSCE490.
- Krishnamoorthy, D., Ryu, J., and Skogestad, S. (2019). A dynamic extremum seeking scheme applied to gas lift optimization. *IFAC-PapersOnLine*, 52(1), 802–807. doi:10.1016/j.ifacol.2019.06.160.
- Krstić, M. and Wang, H.H. (2000). Stability of extremum seeking feedback for general nonlinear dynamic systems. *Automatica*, 36(4), 595–601. doi:10.1016/S0005-1098(99)00183-1.
- Nocedal, J. and Wright, S.J. (2006). *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer Science+Business Media LLC, New York, NY, second edition. doi:10.1007/978-0-387-40065-5.
- Tan, Y., Moase, W.H., Manzie, C., Nesic, D., and Mareels, I. (2010). Extremum seeking from 1922 to 2010. *Proceedings of the 29th Chinese Control Conference, Beijing, China*.
- Wang, H.H., Yeung, S., and Krstic, M. (2000). Experimental application of extremum seeking on an axial-flow compressor. *IEEE Transactions on Control System Technology*, 8(2), 300–309. doi:10.1109/87.826801.
- Xiao, Y., Li, Y., and Seem, J.E. (2014). Multi-variable extremum seeking control for mini-split air-conditioning system. URL <https://docs.lib.purdue.edu/iracc/1513/>.
- Zhang, C. and Ordóñez, R. (2007). Numerical optimization-based extremum seeking control with application to ABS design. *IEEE Transactions on Automatic Control*, 52(3), 454–467. doi:10.1109/TAC.2007.892389.