# Novel Results on Output-Feedback LQR Design



Citation for the original published paper (version of record):
Ilka, A., Murgovski, N. (2022). Novel Results on Output-Feedback LQR Design. IEEE Transactions on Automatic Control, In Press. http://dx.doi.org/10.1109/TAC.2022.3218560



(article starts on next page)

# Novel Results on Output-Feedback LQR Design

Adrian Ilka, *Member, IEEE*, and Nikolce Murgovski

***Abstract*— This paper provides novel developments in output-feedback stabilization for linear time-invariant systems within the linear quadratic regulator (LQR) framework. First, we derive the necessary and sufficient conditions for output-feedback stabilizability in connection with the LQR framework. Then, we propose a novel iterative Newton's method for output-feedback LQR design and a computationally efficient modified approach that requires solving only a Lyapunov equation at each iteration step. We show that the proposed modified approach guarantees convergence from a stabilizing state-feedback to a stabilizing output-feedback solution and succeeds in solving high dimensional problems where other, state-of-the-art methods, fail. Finally, numerical examples illustrate the effectiveness of the proposed methods.**

***Index Terms*— Controller design, linear time-invariant system, linear quadratic regulator, Newton's method, output-feedback, stability.**

## I. Introduction

ONE of the most fundamental problems in control theory is the linear quadratic regulator (LQR) design problem [1]. The so-called infinite horizon linear quadratic problem of finding a control function $u(t) = -Kx(t)$ for $x_0 \in \mathbb{R}^{n_x}$ that minimizes the cost functional

$$J = \frac{1}{2}\int_0^\infty \Big(x(t)^T Q x(t) + u^T(t) R u(t) + 2x^T(t) N u(t)\Big) dt, \tag{1}$$

with $R > 0$, $Q - NR^{-1}N^T \geq 0$ subject to $x(0) = x_0$, and

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t), \end{aligned} \tag{2}$$

has been studied by many authors [1]–[4], where $x(t) \in \mathbb{R}^{n_x}$, $y(t) \in \mathbb{R}^{n_y}$, and $u(t) \in \mathbb{R}^{n_u}$ denote the state, measurable output, and the control input vectors, respectively. Furthermore, $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, and $C \in \mathbb{R}^{n_y \times n_x}$ are constant known matrices.

Often it is not possible or economically feasible to measure all the state variables. In this case, an output-feedback control law defined as

$$u(t) = -Fy(t) \tag{3}$$

would be more beneficial. However, finding an optimal output-feedback control law in the form (3) which minimizes (1), is still one of the most important open questions in control engineering, despite the availability of many approaches and numerical algorithms, as it is pointed out in survey papers [5], [6]. This is mainly due to the lack of testable necessary and sufficient conditions for output-feedback stabilizability.

Furthermore, the majority of algorithms for output-feedback LQR design are formulated in terms of linear matrix inequalities (LMIs) [7]–[14] or bilinear matrix inequalities (BMIs) [15]–[20]. These algorithms are dependent on the used BMI/LMI solvers and could work well for small/medium-sized problems, but may fail to converge to a solution or become computationally too heavy as the problem size increases [21]. In addition, available iterative numerical algorithms with guaranteed convergence such as [22]–[24], or algorithms using nonlinear programming (NLP) such as [25], [26], as well as the recently introduced ray-shooting method based approaches, e.g. [21], [27], unfortunately require a selection of an initial stabilizing output-feedback gain. However, a direct procedure for finding such a gain is unknown and could be hard to get, as highlighted in [5]. The author in [28] has proposed a state-feedback projection theory to bypass the need of a stabilizing output-feedback gain. However, the introduced iterative controller design problem results in a coupled nonlinear matrix equations, and conditions for the existence and global uniqueness are not introduced nor discussed. Furthermore, the proposed Newton approach ensures only sufficient conditions for output-feedback stabilizability. Finally, authors in [29] proposed an algorithm which iterates a Riccati equation from an initial state-feedback solution, but it applies to a restrictive problem description and its convergence has not been proven.

In general, finding stabilizing static output-feedbacks (SOFs) is suspected to be non-deterministic polynomial-time hard (NP-hard), as it is discussed in [5], [21] and [27]. The problem is known to be NP-hard if structural constraints or bounds are imposed on the entries of the controllers, see e.g. [30], [31]. Furthermore, minimal-norm SOFs with bounded entries, pole-placement and simultaneous stabilization via SOFs are also considered to be NP-hard (see [21], [30] and [32], respectively). Moreover, the authors in [5] from 1997 have reviewed results from computational complexity theory

Manuscript received Month XX, 20XX; revised Month XX, 20XX; accepted Month XX, 20XX. Date of publication Month XX, 20XX; date of current version Month XX, 20XX. Recommended by Name of the Editor.

This work has been financed by the project "International Center of Excellence for Research on Intelligent and Secure Information and Communication Technologies and Systems - II. stage" under the Operational Program Integrated Infrastructure co-financed by the European Regional Development Fund, ITMS code: 313021W404

A. Ilka is with the Water Construction Company, State Enterprise (Vodohospodárska Výstavba, Štátny Podnik), Karloveská 2, 842 04 Bratislava, Slovakia, and with the Institute of Robotics and Cybernetics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, Ilkovičova 3, 812 19 Bratislava, Slovakia (e-mail: adrian.ilka@ieee.org).

N. Murgovski is with the Department of Electrical Engineering, Chalmers University of Technology, Hörsalsvägen 9-11, SE-412 96, Gothenburg, Sweden (e-mail: nikolce.murgovski@chalmers.se).

Digital Object Identifier 10.1109/TAC.20XX.XXXXXXX

to suggest that *"such hope that someone can come up with an algorithm that can solve most of the SOFs problems in practice may not be realistic, at least for moderate and large-size problems"*. This prediction/prognosis from 20 years afar has been more or less proven since, as described above, after yeas of extensive research in this filed, there are still unsolved problems, especially if we consider large-size problems.

Even though most of the output-feedback problems are considered to be NP-hard, we have shown in our recent paper [33] that within the LQR framework, it is possible to find SOFs in a reasonable time even for large-scale systems. In this paper, we expand and complete our results from [33]. First we derive the necessary and sufficient conditions for output-feedback stabilizability in connection with the LQR framework. Then, we propose a novel iterative output-feedback LQR design approach for linear time-invariant (LTI) systems, using Newton's method. Afterwards, we show that with a simple modification a new iterative algorithm can be obtained which has a guaranteed convergence to an optimal output-feedback solution from any stabilizing state-feedback solution. In addition, the proposed modified algorithm requires solving only a Lyapunov equation at each iteration step, which is computationally much more tractable then algorithms in the literature, including approaches based on LMIs, BMIs, NLP and ray-shooting methods. Finally, we propose/review some simple and useful modifications/extensions.

The mathematical notation of the paper is as follows. The set of real and complex numbers are denoted by $\mathbb{R}$ and $\mathbb{C}$, respectively. Given a matrix $C \in \mathbb{R}^{n_y \times n_x}$, its pseudoinverse is denoted by $C^+$. For matrices $A, B \in \mathbb{R}^{n_x \times n_x}$, their Hadamard (Schur) and Kronecker products are denoted by $A \circ B$ and $A \otimes B$, respectively. Matrices, if not explicitly stated, are assumed to have compatible dimensions. The real part of a complex number $z$ is denoted by $\Re(z)$. Finally, for any positive integer $n_x$, the $n_x \times n_x$ identity and zero matrices are denoted by $I_{n_x},\ 0_{n_x} \in \mathbb{R}^{n_x \times n_x}$, respectively. For matrices, $\|.\|$ means any matrix norm, consequently $\|.\|_F$, and $\|.\|_2$ means the Frobenius and induced 2-norm, respectively.

## II. Necessary and Sufficient Conditions for Output-Feedback Stabilizability

This section formulates the necessary and sufficient conditions for output-feedback stabilizability in the LQR framework, essential for the main results.

Considering the system (2) and the output-feedback control law (3), let us first recall some related terminology.

*Definition 1:* A square matrix $A \in \mathbb{R}^{n_x \times n_x}$ is said to be *stable* if and only if for every eigenvalue $\lambda_i$ of $A$, $\Re(\lambda_i) \leq 0$.

*Definition 2:* The pair $(A, B)$ is said to be *stabilizable* if and only if there exists a real matrix $K \in \mathbb{R}^{n_u \times n_x}$ such that $A - BK$ is stable.

*Definition 3:* The pair $(A, C)$ is said to be *detectable* if and only if there exists a real matrix $L \in \mathbb{R}^{n_x \times n_y}$ such that $A - LC$ is stable.

*Definition 4:* The system (2) is said to be *static output-feedback stabilizable* if and only if there exists a real matrix $F \in \mathbb{R}^{n_u \times n_y}$ such that $A - BFC$ is stable.

Then the novel necessary and sufficient stability conditions for output-feedback stabilizability in the LQR framework can be formulated as follows.

*Theorem 1:* The system (2) is static output-feedback stabilizable if and only if the pair $(A, B)$ is stabilizable, the pair $(A, C)$ is detectable and there exist real matrices $F \in \mathbb{R}^{n_u \times n_y}$ and $G \in \mathbb{R}^{n_u \times n_x}$ such that

$$FC - R^{-1}(B^T P + N^T) = G, \tag{4}$$

where $P \in \mathbb{R}^{n_x \times n_x}$ is the real symmetric positive semi-definite solution of

$$\begin{aligned} &A^T P + PA + Q + G^T RG \\ &\quad - (PB + N)R^{-1}(B^T P + N^T) = 0, \end{aligned} \tag{5}$$

for given $Q \in \mathbb{R}^{n_x \times n_x}$, $N \in \mathbb{R}^{n_x \times n_u}$ and $R \in \mathbb{R}^{n_u \times n_u}$ matrices satisfying

$$\begin{bmatrix} Q, & N \\ N^T, & R \end{bmatrix} \geq 0,\ R > 0. \tag{6}$$

*Proof:* We will first prove the necessity of Theorem 1. Assume that $A - BFC$ is stable for some $F$, i.e. the system (2) is output-feedback stabilizable. Then the pair $(A, B)$ is stabilizable since $A - BK$ is stable for $K = FC$, and consequently the pair $(A, C)$ is detectable, since $A - LC$ is stable for $L = BF$. Furthermore, because $A - BFC$ is stable, there exists a unique symmetric positive semi-definite matrix $P$ (see Appendix I for details), such that

$$\begin{aligned} &(A - BFC)^T P + P(A - BFC) + Q \\ &\quad + C^T F^T RFC - C^T F^T N^T - NFC = 0. \end{aligned} \tag{7}$$

Rearranging (7), one can obtain

$$\begin{aligned} &A^T P + PA + Q \\ &\quad - (PB + N)R^{-1}(B^T P + N^T) \\ &\quad + \Big(FC - R^{-1}(B^T P + N^T)\Big)^T R\Big(FC \\ &\quad - R^{-1}(B^T P + N^T)\Big) = 0. \end{aligned} \tag{8}$$

Hence, setting $G = FC - R^{-1}(B^T P + N^T)$ implies the necessity of Theorem 1.

Now assume that the pair $(A, B)$ is stabilizable, the pair $(A, C)$ is detectable and there exist real matrices $F$ and $G$ satisfying (4). From equations (4) and (5) it follows that (7) is satisfied. From the second condition it follows that $A - LC$ is stable for some $L$. Noting that

$$(A - LC) = \left((A - BFC) - [L,\ -B]\begin{bmatrix} C \\ FC \end{bmatrix}\right), \tag{9}$$

it follows that the pair

$$\left(A - BFC, \begin{bmatrix} C \\ FC \end{bmatrix}\right) \tag{10}$$

is detectable as well. Since $P$ is symmetric and positive semi-definite, we conclude from (7) that $A - BFC$ is stable, and hence the sufficiency of Theorem 1 is proved as well. ■

*Remark 1:* Similar conditions for output-feedback stabilizability have been obtained in [29, Theorem 1], but for a restricted problem formulation with $Q = C^T C,\ R = I$ and $N = 0$.

From Theorem 1 follows that if the system (2) is output-feedback stabilizable, then there exists a state feedback gain $K = FC$ such that $A - BK$ is stable. If $C$ is a square and nonsingular matrix, then we can easily express the output-feedback gain as $F = KC^{-1}$. However, for most of the output-feedback problems the matrix $C$ is non-square, i.e. non-invertible. Therefore, by expressing the output-feedback gain using a pseudo-inverse as $F = KC^{+}$, a so called pseudo-inverse error appears that can be calculated as

$$G = FC - K, \tag{11}$$

which is identical to (4), since $K = R^{-1}(B^T P + N^T)$. Hence, from the above and the Theorem 1 it follows that if the system (2) is output-feedback stabilizable, then for given $Q$, $R$, and $N$ matrices satisfying (6), there exists a real positive semi-definite matrix $P$ such that (5) is fulfilled for

$$G = FC - K = FC - R^{-1}(B^T P + N^T), \tag{12}$$
$$F = KC^{+} = R^{-1}(B^T P + N^T)C^{+}. \tag{13}$$

The next Identity is straightforward and used later to obtain the main results.

*Identity 1:* Suppose that $F = R^{-1}(B^T P + N^T)C^{+}$ and $G = FC - R^{-1}(B^T P + N^T)$. Then the following statements are identical

1) $$\mathcal{R}(P) = A^T P + PA + Q + G^T RG - (PB + N)R^{-1}(B^T P + N^T), \tag{14}$$
2) $$\mathcal{R}(P) = \tilde{Q} + G^T RG + \tilde{A}^T P + P\tilde{A} - P\tilde{S}P, \tag{15}$$
3) $$\mathcal{R}(P) = (A - BFC)^T P + P(A - BFC) + Q + C^T F^T RFC - C^T F^T N^T - NFC \tag{16}$$

where $\tilde{A} = A - BR^{-1}N^T$, $\tilde{Q} = Q - NR^{-1}N^T$ and $\tilde{S} = BR^{-1}B^T$.

*Proof:* The identity can be proved by substituting back all the denotations. ∎

## III. INFINITE HORIZON OUTPUT-FEEDBACK LQR DESIGN

The equations (14), (15), and (16) are algebraic Riccati-like equations. In general, Newton's method and it's modifications are widely used to solve algebraic Riccati equations [34]–[36]. Inspired by [34] and [35], in this section we first propose a Newton's method based algorithm to design stabilizing static output-feedback controllers. Then, we show that by a simple modification, a computationally similarly tractable stabilizing output-feedback controller design approach can be obtained, while guaranteeing convergence from any initial state-feedback LQR solution. Finally, after a short sensitivity analysis, we show the relation of these approaches to the infinite horizon output-feedback LQR problem (i.e. to find a control law in the form (3), minimizing the cost function (1), subject to system dynamics (2) and initial state $x_0$).

### *A. Newton's method for stabilizing SOF controller design*

The Fréchet derivative of a matrix function $\mathcal{R} : \mathbb{F}^{n_x \times n_x} \to \mathbb{F}^{n_x \times n_x}$ at matrix $P$ is a linear function $\mathcal{L} : \mathbb{F}^{n_x \times n_x} \to \mathbb{F}^{n_x \times n_x}$, $X \to \mathcal{L}(P, X)$ such that for all $X \in \mathbb{F}^{n_x \times n_x}$

$$\mathcal{R}(P + X) - \mathcal{R}(P) - \mathcal{L}(P, X) = o(\|X\|), \tag{17}$$

where the norm is any matrix norm and $\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$ [37], [38]. The Fréchet derivative, if it exists, can be shown to be unique [39]. Consider $\mathcal{R}$ defined by the Riccati like matrix equation (15). Then its Fréchet derivative at the matrix $P$ is given by (see Appendix III)

$$\mathcal{L}(P, X) = \mathcal{H}_1^T(P)X + X\mathcal{H}_1(P) + \mathcal{H}_2^T(P)XZ + Z^T X\mathcal{H}_2(P), \tag{18}$$

where $Z = C^{+}C$, and

$$\mathcal{H}_1(P) = \tilde{A} - \tilde{S}PZ - BR^{-1}N^T Z + BR^{-1}N^T, \tag{19}$$
$$\mathcal{H}_2(P) = \tilde{S}PZ - \tilde{S}P + BR^{-1}N^T Z - BR^{-1}N^T. \tag{20}$$

Now we can formulate the Newton's method in Banach space (see [36], [40]) for the solution of (15) as follows

$$P_{j+1} = P_j + (\mathcal{L}(P_j, X_j))^{-1}\mathcal{R}(P_j), \quad j = 1, 2, \ldots. \tag{21}$$

Furthermore, we can compute $P_{j+1}$ directly from (21) as

$$P_{j+1} = P_j + X_j, \quad j = 1, 2, \ldots, \tag{22}$$

where $X_j$ is solved from

$$\mathcal{H}_1^T(P_j)X_j + X_j\mathcal{H}_1(P_j) + \mathcal{H}_2^T(P_j)X_j Z + Z^T X_j\mathcal{H}_2(P_j) = -\mathcal{R}(P_j). \tag{23}$$

The equation (23) is a generalized Sylvester equation, which can be, based on Identity 1, rewritten to the form

$$\sum_{i=1}^{4} W_{j_i} X_j U_{j_i} = -\mathcal{R}(P_j), \tag{24}$$

where

$$\begin{aligned} W_{j_1} &= A^T - C^T F_j^T B^T, & U_{j_1} &= I, \\ W_{j_2} &= G_j^T B^T, & U_{j_2} &= Z, \\ W_{j_3} &= I, & U_{j_3} &= A - BF_j C, \\ W_{j_4} &= Z^T, & U_{j_4} &= BG_j. \end{aligned}$$

*Lemma 1:* Suppose that $\bar{A} \in \mathbb{F}^{m \times n}$, $\bar{B} \in \mathbb{F}^{p \times q}$, and $\bar{X} \in \mathbb{F}^{n \times p}$. Then,

$$\mathrm{vec}(\bar{A}\bar{X}\bar{B}) = \left(\bar{B}^T \otimes \bar{A}\right)\mathrm{vec}(\bar{X}). \tag{25}$$

*Proof:* For proof see [41, Lemma 4.3.1, p. 254]. ∎

*Definition 5:* Since the Fréchet derivative $\mathcal{L}(P, X)$ is linear in $X$, applying Lemma 1 to the left hand side of (24) gives

$$\begin{aligned} \mathrm{vec}(\mathcal{L}(P_j, X_j)) &= \left(\sum_{i=1}^{4} {U_{j_i}}^T \otimes W_{j_i}\right)\mathrm{vec}(X_j) \\ &= K_{\mathcal{L}_j}\,\mathrm{vec}(X_j), \end{aligned} \tag{26}$$

where $K_{\mathcal{L}} \in \mathbb{F}^{n_x^2 \times n_x^2}$ is called the *Kronecker form of the Fréchet derivative*.

The generalized Sylvester equation (24) has a unique solution if and only if $K_{\mathcal{L}}$ is nonsingular. In this case the solution can be obtained analytically as

$$\mathrm{vec}(X_j) = K_{\mathcal{L}_j}^{-1}\mathrm{vec}(-\mathcal{R}(P_j)), \tag{27}$$

This article has been accepted for publication in IEEE Transactions on Automatic Control. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAC.2022.3218560



or can be approximated either by gradient-based iterative methods (such as [42], [43] and [44]), or by any other methods in the literature.

The proposed Newton's method for SOF controller design using (23)–(22) is summarized in Algorithm 1.

**Algorithm 1:** Newton's method for static output-feedback controller design.

```
Choose some initial guess P_1 = P_1^T such that K_L is
  nonsingular, and calculate Ã = A − B R^{-1} N^T,
  Q̃ = Q − N R^{-1} N^T, S̃ = B R^{-1} B^T, and expected
  tolerance on the numerical solution ε > 0, ε → 0.
for j=1:maxIteration do
    F_j = R^{-1}(B^T P_j + N^T) C^+,
    G_j = F_j C − R^{-1}(B^T P_j + N^T),
    R(P_j) = Q̃ + G_j^T R G_j + Ã^T P_j + P_j Ã − P_j S̃ P_j,
    if trace(R(P_j)^T R(P_j)) > ε then
        X_j ← by solving the matrix equation (24),
        P_{j+1} = P_j + X_j,
    else
        break;
    end
end
```

*Remark 2:* It follows from (24) that if $C = I$ then $Z = I$ and the generalized Sylvester equation (24) reduces to

$$\begin{aligned}(A - BK_j)^T X + X(A - BK_j) = -A^T P_j - P_j A \\ - Q + (B^T P_j + N^T)^T R^{-1}(B^T P_j + N^T).\end{aligned} \tag{28}$$

Hence, the Algorithm 1 becomes equivalent to [35, Algorithm 1.1] for state-feedback LQR design.

*Remark 3:* Algorithm 1 has a termination condition that depends on a constant $\epsilon > 0$, $\epsilon \to 0$, which describes the expected tolerance on the numerical solution. For example $\epsilon = 10^{-d}$ means $d$ digit desired accuracy in the numerical solution.

The results from this subsection are used only as an intermediate step to obtain the main results, the modified Newton's method. Therefore, global convergence and existence of a stabilizing solution remains to be proven. Although, standard local q-quadratic convergence results for Newton's method apply [41, Theorem 5.2.1], as detailed in [42, Theorem 1]. In particular, if Newton's method is started sufficiently close to a solvent for which the Fréchet derivative is non-singular, the iteration converges with a quadratic rate. Kantorovich theorem can also be applied to provide sufficient conditions for existence of a solvent and convergence of Newton's method to that solvent [41, Theorem 5.3.1].

*Remark 4:* Based on standard results for Newton's method (see [41, Theorem 5.2.1] and [42, Theorem 1]), Algorithm 1 requires an initial guess $P_1 = P_1^T$ which is close enough to a solvent for which the Fréchet derivative (i.e. $K_{\mathcal{L}}$) is non-singular. However, a direct procedure to get such initial guess is out of the topic of this paper, since the results from this subsection are only used as an intermediate step to obtain the main results, the modified Newton's method.

In the next subsection we show that with a simple modification a new iterative algorithm can be obtained which has a guaranteed convergence from any stabilizing state-feedback solution to an output-feedback solution.

## B. Modified Newton's method for stabilizing SOF controller design

In order to calculate the Newton step in the Algorithm 1, we need to solve the generalized Sylvester equation (24). In this subsection we show that with a simple modification we can approximate the Newton step and converge to a solution with similar computational effort, but with a guaranteed convergence from any state-feedback solution.

By freezing the matrix $G$ in (15), the term $G^T R G$ becomes a constant during an iteration step and the Fréchet derivative reduces to

$$\hat{\mathcal{L}}(P_j, X_j) = (\tilde{A} - \tilde{S}P_j)^T X_j + X_j(\tilde{A} - \tilde{S}P_j), \tag{29}$$

and the Newton's method to

$$(\tilde{A} - \tilde{S}P_j)^T X_j + X_j(\tilde{A} - \tilde{S}P_j) = -\mathcal{R}(P_j), \tag{30}$$

$$P_{j+1} = P_j + X_j,\ j = 1, 2, \dots. \tag{31}$$

Equation (30) is a Lyapunov equation, which can be solved efficiently and with much less computational effort than solving (24) with (27) or with other iterative methods.

The Algorithm 2 summarizes the proposed modified Newton's method for SOF controller design using (29)–(31).

**Algorithm 2:** Modified Newton's method for static output-feedback controller design.

```
Choose some ε > 0 (ε → 0), and initial guess
  P_1 = P_1^T such that Ã − S̃ P_1 is stable (such P_1 can
  be obtained via the standard state-feedback LQR
  design, see Remark 10). Then calculate
  Ã = A − B R^{-1} N^T, Q̃ = Q − N R^{-1} N^T and
  S̃ = B R^{-1} B^T.
for j=1:maxIteration do
    F_j = R^{-1}(B^T P_j + N^T) C^+,
    G_j = F_j C − R^{-1}(B^T P_j + N^T),
    R(P_j) = Q̃ + G_j^T R G_j + Ã^T P_j + P_j Ã − P_j S̃ P_j,
    if trace(R(P_j)^T R(P_j)) > ε then
        X_j ← (Ã − S̃ P_j)^T X_j + X_j (Ã − S̃ P_j) =
          −R(P_j),
        P_{j+1} = P_j + X_j,
    else
        break;
    end
end
```

*1) Convergence:* In this sub-subsection, we show that under certain assumptions, Algorithm 2 has a guaranteed convergence from a stabilizing starting guess $P_1$ (i.e. $\tilde{A} - \tilde{S}P_1$ is stable for some $\tilde{Q} \geq 0$), to a stabilizing output-feedback solution.

Let us recall some results relating to the convergence proof.

This article has been accepted for publication in IEEE Transactions on Automatic Control. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAC.2022.3218560



*Definition 6:* The *inertia* of a matrix $W \in \mathbb{R}^{n\times n}$ is the triple $\text{In}(W) = (\pi(W), \nu(W), \delta(W))$ where $\pi(W)$, $\nu(W)$, and $\delta(W)$ are the number of eigenvalues with positive, negative, and zero real part respectively [34].

*Lemma 2:* If $H = H^T \in \mathbb{R}^{n\times n}$, $A \in \mathbb{R}^{n\times n}$, and $W > 0 \in \mathbb{R}^{n\times n}$ satisfy $AH + HA^T = -W \leq 0$, and $\delta(A) = 0$, then $\text{In}(-H) \leq \text{In}(A)$.

*Proof:* For proof see [45, Proposition 1, p. 447]. ■

*Lemma 3:* Let $H = H^T \in \mathbb{R}^{n\times n}$, $A \in \mathbb{R}^{n\times n}$, $W > 0 \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{l\times n}$ satisfy $AH + HA^T = -W \leq C^TC$, where $(A, C)$ defines a detectable pair. Then $\nu(A) = n$ if and only if $\nu(H) = 0$.

*Proof:* For proof see [34, Lemma 8, p. 5]. ■

The next Proposition shows that if the conditions described in Theorem 1 hold, then with a stabilizing starting guess ($P_1$) the Algorithm 2 cannot fail due to a singular Lyapunov operator.

*Proposition 1:* Suppose that the conditions in Theorem 1 hold, and the pair $(\tilde{A}, \tilde{C}_q)$ is detectable, where $\tilde{Q} = \tilde{C}_q^T\tilde{C}_q$ is a full-rank factorisation of $\tilde{Q}$. If $P_1$ is stabilizing, and Algorithm 2 is applied to (15), then the Lyapunov operator of the Lyapunov equation in step 7 from Algorithm 2 is nonsingular for all $j$ and the sequence of approximate solutions $X_j$ is well defined.

*Proof:* Suppose that the pair $(\tilde{A}, \tilde{C}_q)$ is detectable. From step 7 from Algorithm 2 applied to (15) we can get

$$\begin{aligned}(\tilde{A} - \tilde{S}P_j)^T(P_j + X_j) + (P_j + X_j)(\tilde{A} - \tilde{S}P_j)\\ = -\tilde{Q} - G_j^TRG_j - P_j\tilde{S}P_j \leq -\tilde{Q},\end{aligned} \tag{32}$$

since $\tilde{Q}$ and $\tilde{S}$ are positive semi-definite, due to $Q - NR^{-1}N^T \geq 0$ and $R > 0$. From (32) follows that if $\tilde{A} - \tilde{S}P_j$ is stable, then $\tilde{A} - \tilde{S}(P_j + X_j)$ is also stable. Furthermore, Lemma 3 implies that $P_j + X_j$ is positive semi-definite. The Lyapunov operator corresponding to the Lyapunov equation in step 7 from Algorithm 2 is well defined, precisely as

$$\tilde{\Omega}_j(X_j) = (\tilde{A} - \tilde{S}P_j)^TX_j + X_j(\tilde{A} - \tilde{S}P_j), \tag{33}$$

for $X_j \in \mathbb{R}^{n_x\times n_x}$ and $j = 1, 2, \ldots$. ■

Let us recall the following Lemma.

*Lemma 4:* Suppose that $\{P_j\}_{j=2}^{\infty}$ is a sequence of symmetric matrices such that $\{\mathcal{R}(P_j)\}_{j=2}^{\infty}$ is bounded. If the pair $(\tilde{A}, B)$ is stabilizable and $\tilde{A} - \tilde{S}P_j$ is stable for each $j = 2, \ldots$, then $\{P_j\}_{j=2}^{\infty}$ is bounded.

*Proof:* For proof see [35, Lemma 2.3, p. 696]. ■

*Remark 5:* If $\mathcal{R}(P_j) \neq 0$, i.e., if $P_j$ is not a solution of (15), then the Newton step (of Algorithm 2) is a descent direction of $\|\mathcal{R}(P_j + X_j)\|_F$. It follows that we have $\|\mathcal{R}(P_j + X_j)\|_F \leq \|\mathcal{R}(P_j)\|_F$ and $\|\mathcal{R}(P_j + X_j)\|_F = \|\mathcal{R}(P_j)\|_F$ if and only if $\mathcal{R}(P_j) = 0$. That is, the residual decreases as long as $P_j$ is not a solution of (15).

*Remark 6:* It is important to note that $P_1 \geq P_2$, where $P_1$ is the initial guess, is not true in general. This is one of the drawbacks of Newton's methods. In [34] and [35] the authors have introduced a step-size control (for state-feedback LQR design), which can efficiently solve the problem of a potentially disastrous first Newton step.

Collecting the results so far, we have the following convergence result for the modified Newton's method.

*Theorem 2:* Suppose that the pair $(\tilde{A}, B)$ is stabilizable, the pair $(\tilde{A}, \tilde{C}_q)$ is detectable, and there exist real matrices $F$ and $G$ such that $FC - R^{-1}(B^TP + N^T) = G$. If Algorithm 2 is applied to (15) with a stabilizing starting guess $P_1$ (i.e. $\tilde{A} - BK_1$ is stable for some $\tilde{Q} \geq 0$), then $P^* = \lim_{j\to\infty} P_j$ exists and is the stabilizing solution of the generalized Riccati-like equation (15).

*Proof:* The proof follows from Theorem 1, Lemma 2, 3, 4, and Proposition 1. ■

*Remark 7:* The assumption that the pair $(\tilde{A}, \tilde{C}_q)$ is detectable, where $\tilde{Q} = \tilde{C}_q^T\tilde{C}_q$ is a full-rank factorisation of $\tilde{Q}$, is a requirement even for the standard state-feedback LQR design.

*Remark 8:* If $C = I$ then $G = 0$ and the Algorithm 2 becomes equivalent to [35, Algorithm 1.1] for state-feedback LQR design (or to [34, Algorithm 1], if we require controllability of the pair $(\tilde{A}, B)$ and observability of the pair $(\tilde{A}, \tilde{C}_q)$).

*Remark 9:* From Theorem 2 follows that the convergence rate of Algorithm 2 is at least sublinear. We have observed from the examples studied later in Section V that the convergence rate is in fact linear, if $\tilde{A} - \tilde{S}P^*$ has no eigenvalues on the imaginary axis, although further investigation is needed for a formal proof. If $\tilde{A} - \tilde{S}P^*$ has eigenvalues on the imaginary axis, the convergence behaviour remains an open problem (as it is still an open problem even for standard state-feedback LQR design, see for example [35, Remark 1.1]).

*Remark 10:* If system (2) is stabilizable and $(\tilde{A}, \tilde{C}_q)$ is detectable, then the standard state-feedback LQR solution for (2) for some $\tilde{Q} \geq 0$ always gives a $P_1$ for which $\tilde{A} - \tilde{S}P_1$ is stable.

## *C. Sensitivity analysis*

It is well known that the Newton's method based approaches, in general, are highly sensitive to ill-conditioning. Condition numbers measure the sensitivity of a problem to perturbation in the data. The unstructured absolute condition number $\text{cond}(\mathcal{R}(P))$ can be expressed in terms of the Fréchet derivative of $\mathcal{R}(P)$ in (15), evaluated at $P$

$$\text{cond}(\mathcal{R}(P)) = \max_{X\neq 0} \frac{\|\mathcal{L}(P, X)\|}{\|X\|} =: \|\mathcal{L}(X)\|. \tag{34}$$

By applying a Frobenius norm

$$\begin{aligned}\text{cond}(\mathcal{R}(P)) &= \max_{X\neq 0} \frac{\|\mathcal{L}(P, X)\|_F}{\|X\|_F}\\ &= \max_{X\neq 0} \frac{\|\text{vec}\big(\mathcal{L}(P, X)\big)\|_2}{\|\text{vec}\big(X\big)\|_2}\\ &= \|K_{\mathcal{L}}\|_2,\end{aligned} \tag{35}$$

the problem of computing $\text{cond}(\mathcal{R}(P))$ reduces to finding the 2-norm of $K_{\mathcal{L}}$. The relative condition number of $\mathcal{R}(P)$ at $P$, denoted by $\text{rcond}(\mathcal{R}(P))$, can be written in terms of the absolute condition number $\text{cond}(\mathcal{R}(P))$ (see, [46, Sec. 2, p. 776]) as

$$\text{rcond}(\mathcal{R}(P)) = \text{cond}(\mathcal{R}(P)) \frac{\|P\|}{\|\mathcal{R}(P)\|}. \tag{36}$$

This article has been accepted for publication in IEEE Transactions on Automatic Control. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAC.2022.3218560



For structured condition numbers of $\mathcal{R}(P)$ at $P$, as well as for level-2 condition numbers using higher order Fréchet derivatives, please see [46], [47] and references therein. The effect of condition numbers on convergence will be investigated later, in Section V.

### D. Connection to infinite-horizon LQR with output-feedback

This subsection describes the relation of Algorithm 1 and 2 to infinite-horizon LQR with output-feedback. First, let us recall the necessary conditions for the solution of the LQR problem with output feedback, i.e. the existence of a control law in the form (3) minimizing (1) subject to (2) with $R > 0$, $Q - NR^{-1}N^T \geq 0$ and $x(0) = x_0$.

*Lemma 5:* The necessary conditions for the solution of the LQR problem with output feedback are given by

$$\begin{aligned} 0 = A_c^T P + PA_c + Q + C^T F^T RFC - C^T F^T N^T \\ - NFC, \end{aligned} \tag{37}$$

$$0 = A_c Y + Y A_c^T + \mathcal{X}_{x_0}, \tag{38}$$

$$0 = RFCPC^T - (B^T P + N^T) Y C^T, \tag{39}$$

with $\mathcal{X}_{x_0} = x_0 x_0^T$ and $A_c = A - BFC$.

*Proof:* For proof see [48, p. 297-302]. ■

The dependence of $\mathcal{X}_{x_0}$ in (38) in the initial states $x_0$ makes the optimal gain dependent on the initial state through equation (38). In many applications $x_0$ may not be known (which is typical for output-feedback design, as it is pointed out in [48]). It is usual (see for example [49]), to sidestep this problem by replacing

$$\mathcal{X}_{x_0} \equiv E\{\mathcal{X}_{x_0}\}, \tag{40}$$

where $E\{\mathcal{X}_{x_0}\} = E\{x_0 x_0^T\}$ is the initial autocorrelation of the state. Usually, it is assumed that nothing is known of $x_0$ except that it is uniformly distributed on a surface described by $\mathcal{X}_{x_0}$. Most of the papers on output-feedback LQR design assume that the initial states are uniformly distributed on the unit sphere, i.e. $\mathcal{X}_{x_0} = I$ (e.g. [14], [18], [48], [50]).

The next theorem describes how Algorithm 1 and 2 are connected to the initial condition problem described above.

*Theorem 3:* The solution of Algorithm 1 and 2 satisfies the necessary conditions described by equations (37)–(39) in Lemma 5 if and only if $Y = I$ and $\mathcal{X}_{x_0} = -A_c - A_c^T$.

*Proof:* From (39) it follows that

$$F = R^{-1}(B^T P + N^T) Y C^T (C Y C^T)^{-1}. \tag{41}$$

By assuming that $Y = I$ the equation (41) reduces to

$$\begin{aligned} F &= R^{-1}(B^T P + N^T) C^T (CC^T)^{-1} \\ &= R^{-1}(B^T P + N^T) C^+, \end{aligned} \tag{42}$$

which is identical to the step 3 in Algorithm 1 and 2. Furthermore, from equation (38) for $Y = I$ it follows that

$$\mathcal{X}_{x_0} = -A_c - A_c^T. \tag{43}$$

Finally, setting $G = FC - R^{-1}(B^T P + N^T)$ and by rearranging (37) we can get (14) which is equivalent (see Identity 1) to the step 5 in Algorithm 1 and 2. Hence, the proof is completed. ■

*Remark 11:* The initial state $x_0$ is generally free and so is $Y$ which is a function of $\mathcal{X}_{x_0}$. Hence, instead of guessing $\mathcal{X}_{x_0} = I$, we may guess for $Y = I$, and thus the nonlinearity in $Y$ in (37)–(39) disappears. So, one can get a simple Riccati-like equation (37), which can be solved easily using Algorithm 1 or 2.

A direct comparison of setting $\mathcal{X}_{x_0} = I$ versus $Y = I$ will be investigated in Section V.

### E. Output-feedback LQR problem with known initial conditions

In the previous subsection III-D, we have shown the relation of Algorithm 1 and 2 to output-feedback LQR problem, and that the proposed algorithms involve less nonlinearities compared to other approaches in the literature when the initial conditions are not given priory, i.e. $x_0$ is unknown. In this subsection, we show (see Algorithm 3) how the Algorithm 2 can be extended if the initial conditions are known.

**Algorithm 3:** Iterative algorithm for output-feedback LQR design with known initial conditions.

1 Set $Y_1 = I$, and choose some $\epsilon > 0$ $(\epsilon \to 0)$,
2 **for** *i=1*:maxIteration **do**
3 $\quad F_i \leftarrow$ by Algorithm 2 with step 3 changed to $F_j = R^{-1}(B^T P_j + N^T) Y_i C^T (C Y_i C^T)^{-1}$,
4 $\quad A_c = A - BF_iC$,
5 $\quad \mathcal{R}(Y_i) = A_c Y_i + Y_i A_c^T + \mathcal{X}_{x_0}$,
6 $\quad$ **if** trace$(\mathcal{R}(Y_i)^T \mathcal{R}(Y_i)) > \epsilon$ **then**
7 $\quad\quad Y_{i+1} \leftarrow A_c Y_{i+1} + Y_{i+1} A_c^T + \mathcal{X}_{x_0}$,
8 $\quad$ **else**
9 $\quad\quad$ **break;**
10 $\quad$ **end**
11 **end**

The numerical examples in Section V suggest that Algorithm 3 converges to a solution if $\mathcal{R}(P)$ is well-conditioned. But at this writing, we are not aware of a proof for this conjecture. Although, if $\mathcal{X}_{x_0} = x_0 x_0^T$ is symmetric and positive definite, and if all uncontrollable state variables of the system (2) are asymptotically stable, then $A_c$ is negative definite. Hence, $Y_{i+1}$ exists and is symmetric and positive definite. It follows that $Y_{i+1}$ has a full rank and if $C Y_{i+1} C^T$ is nonsingular, the Lyapunov operator corresponding to the Lyapunov equation in step 7 from Algorithm 3 is well defined, precisely as

$$\Omega_i(Y_{i+1}) = A_c Y_{i+1} + Y_{i+1} A_c^T + \mathcal{X}_{x_0}, \tag{44}$$

for $Y_{i+1} \in \mathbb{R}^{n_x \times n_x}$ and $i = 1, 2 \ldots$. Hence, Algorithm 3 cannot fail due to a singular Lyapunov operator in step 7. Therefore, if in step 3 of Algorithm 3, the Algorithm 2 succeeds in finding $F_i$ at each step, then Algorithm 3 produces a sequence of symmetric matrices $\{Y_i\}_{i=2}^{\infty}$ and $\lim_{i\to\infty} Y_i = Y^*$ where $Y^*$ is the solution satisfying (37)–(39).

This article has been accepted for publication in IEEE Transactions on Automatic Control. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAC.2022.3218560


## IV. Useful techniques, extensions, and modifications

Control law (3) is defined in an SOF form. Many different controller structures can be transformed to this SOF form, like proportional-integral (PI), realizable proportional-integral-derivative ($\text{PID}_f$), realizable proportional-derivative ($\text{PD}_f$), realizable derivative ($\text{D}_f$), even full/reduced order dynamic output-feedback controllers (DOF), dynamic output-feedback with integral and realizable derivative part ($\text{DOFID}_f$), or dynamic output-feedback with realizable derivative part ($\text{DOFD}_f$), by augmenting the system with additional state variables. For more info, see [14].

Since the proposed algorithms (Algorithm 1, 2 and 3) belong to the LQR framework, all the well-known techniques, modifications and, extensions of the standard LQR design can be applied here as well. Therefore, one can apply

- Bryson's rules [51, Section 5.2] for selecting the weighting matrices $Q$ and $R$,
- methods/techniques in [48] for damping, decoupling, tracking, disturbance rejection, etc. controller design,
- techniques in [52] for different eigenvalue placements (pole-placement techniques in LQ) and guaranteed convergence rate,
- techniques in [53], [54] for frequency weighting (frequency shaped LQ),
- and some other methods/techniques in the LQR framework, see e.g. [48], [51] and references therein.

## V. Numerical examples

In order to show the viability of the previous proposed algorithms (Algorithm 1, 2 and 3), we have prepared two sets of examples. The first set of examples contains 1000 randomly generated SOF stabilizable state-space systems (via Matlab's rss subrutin). The second set of examples are all the SOF stabilizable examples from the COMP$l_e$ib library [55].

As algorithms to be compared, the iterative LMI (iLMI) method from [14] and the BMI formulation of the output-feedback LQR (OFLQR) problem (see Appendix II, Lemma 7) have been chosen. All examples and numerical solutions have been carried out on ASUS ZenBook UX480F (Intel(R) Core(TM) i7-8565U CPU @ 1.80 GHz, 16 GB RAM) laptop computer using Matlab 2018b [56]. Furthermore, BMI and iLMI formulations have been carried out by Penlab BMI solver [57] and by Mosek LMI solver [58] using YALMIP R20190425 [59]. Finally, the proposed algorithms (Algorithm 1, 2 and 3) have been implemented in Matlab programming language (see Listing 1, 2 and 3), where for the Algorithm 2 and 3 for the step 7 the built-in Matlab lyap subrutin has been used, and for the Algorithm 1 for the step 7 the equation (27). Furthermore, in order to simplify the code of the Algorithm 2, the Identity 1 has been used and the equation (15) has been replaced with equation (16).

Matlab implementations and examples are fully provided in Listing 1, 2 and 3. The first set of examples, can be downloaded from repository[1], while the second set of examples are all the SOF stabilizable plants from the COMP$l_e$ib library, which is freely available (see [55]).

[1] https://www.ilka.eu/FirstSetOfExamples.zip

TABLE I
Groups of examples in the first set of examples

| Group | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $n_x$ | 1 | 2 | 4 | 6 | 8 | 10 | 20 | 30 | 40 | 50 |
| $n_y$ | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| $n_u$ | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| **examples** | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |

### A. First set of examples

The first set of examples contains 1000 SOF stabilizable examples in 10 groups generated by Matlab's rss subrutin[2]. Each group represents 100 examples with different size of system order (see Table I). Hence, we can test the behaviour and effectiveness of the proposed algorithms and compare them to other output-feedback algorithms in the LQR framework for increasing number of states.

The weighting matrices have been chosen as $Q = C^T C$, $R = I$ and $N = 0$. The initial Lyapunov matrix for Algorithm 1, 2 and 3 is the optimal Lyapunov matrix from the standard state-feedback LQR design. The stopping criterion and maximal iteration number for Algorithm 1, 2 and 3 have been chosen as $\epsilon = 10^{-12}$ and $maxIteration = 9 \times 10^6$. Finally, for Algorithm 3 and for the iLMI and BMI methods, the initial state matrix has been chosen as $\mathcal{X}_{x_0} = I$, i.e. it has been assumed that the initial states are uniformly distributed on the unit sphere.

The effect of increasing the number of states on the running time of one iteration and on the number of iterations of Algorithm 1 and 2 are shown in Fig. 1 and 2. The effect of increasing the number of states on the rcond($\mathcal{R}(P)$) and on the number of iterations of Algorithm 1 and 2 is shown in Fig. 3. It can be observed that the number of iterations and therefore the overall running times of Algorithm 1 and 2 are sensitive to the relative condition number of (14) (rcond($\mathcal{R}(P)$)). That was to be expected, since it is well known that the Newton's method based approaches, in general, are sensitive to ill-conditioning. The effect of increasing the number of states on the average running times and on solved examples of Algorithm 1, 2, 3, and of the iLMI and BMI methods are shown in Fig. 4. It can be observed that the Algorithm 2 outperformed all the other algorithms and methods since it has solved all the examples in this test set, while the running time was very close or sometimes better then the running time of Algorithm 1. Furthermore, it can be observed that eventhough for the Algorithm 1 we do not have a convergence proof from a state-feedback solution (as for Algorithm 2), it has solved many more examples than the iLMI or the BMI methods. Furthermore, Algorithm 3 for $n_x \geq 30$ failed to converge to a solution for few examples. It should be noted that for all those examples the rcond($\mathcal{R}(P)$)$> 10^{10}$, and even the Algorithm 2 has struggled, since the number of iterations for those examples was higher then $10^4$, while for the rest of the examples in those groups was smaller with almost 1 or 2,

[2] The Matlab's rss subrutin generates a random stable system, therefore all the examples are SOF stabilizable as well (since one can select the output-feedback gain as zero and the closed-loop system will be stable).

This article has been accepted for publication in IEEE Transactions on Automatic Control. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAC.2022.3218560



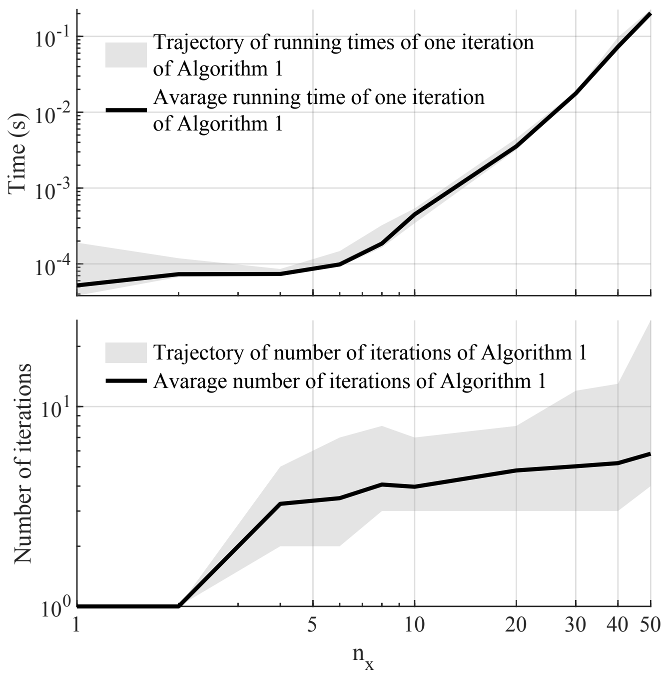


Fig. 1. The effect of increasing the number of states on the running time of one iteration and on the number of iterations of Algorithm 1.

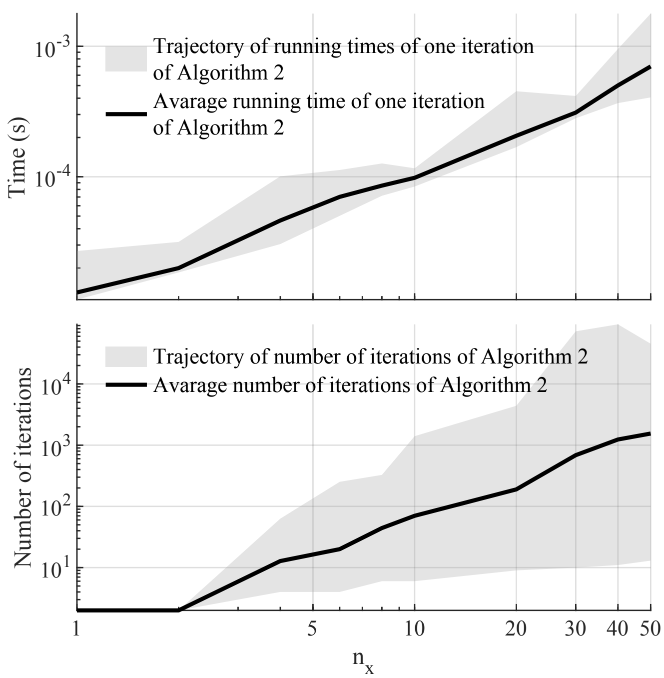


Fig. 2. The effect of increasing the number of states on the running time of one iteration and on the number of iterations of Algorithm 2.

sometimes with 3-4 orders of magnitude. This is the reason for that large trajectory of number of iterations of Algorithm 2 in Fig. 2 for $n_x > 10$.

Finally, Fig. 5 compares how far the actual linear quadratic cost is for some randomly generated initial state conditions ($x_0$ within a unit sphere) for $\mathcal{X}_{x_0} = I$ and for $Y = I$ from the optimal output-feedback cost (minimizing the linear quadratic cost for the given $x_0$). It can be observed that the

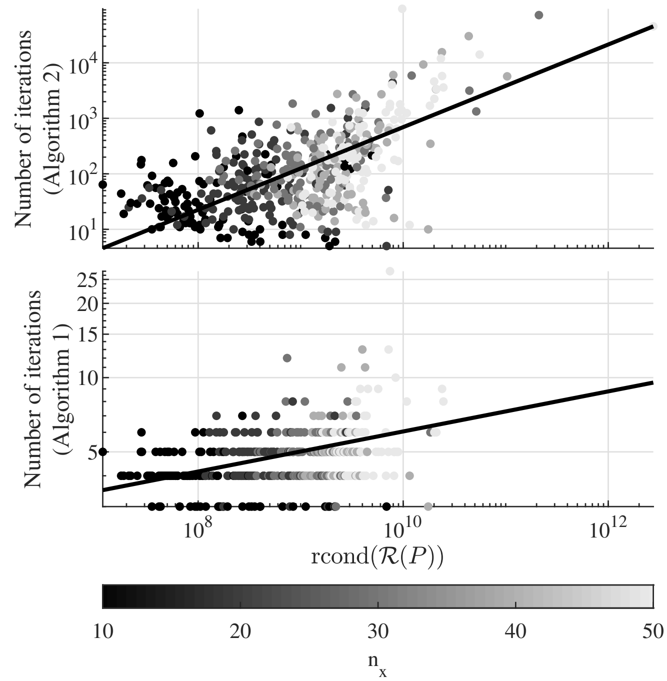


Fig. 3. The effect of increasing the number of states on the rcond($\boldsymbol{\mathcal{R}}(\boldsymbol{P})$) and on the number of iterations of Algorithm 1 and 2.

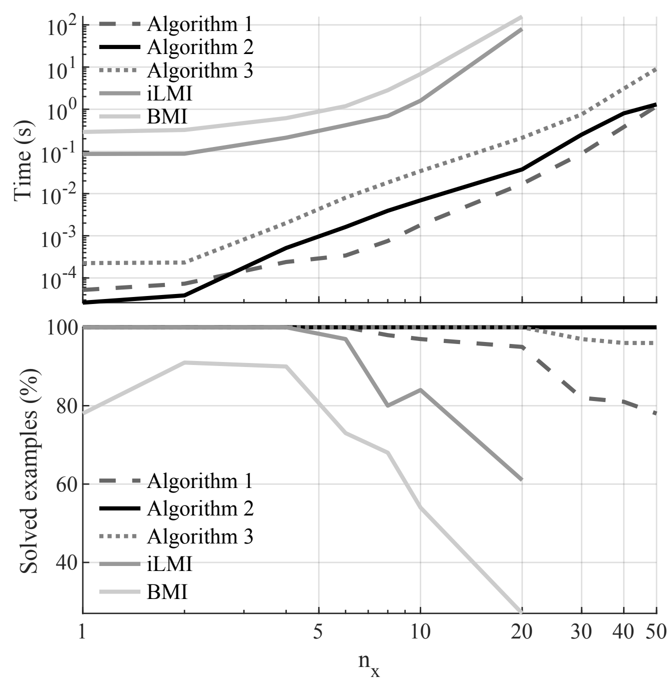


Fig. 4. The effect of increasing the number of states on the average running times and on solved examples of Algorithm 1, 2, 3, and of the iLMI and BMI methods.

distribution of distances from the optimal cost for different initial conditions for $Y = I$, i.e. for Algorithm 2, is comparable with the choice of setting $\mathcal{X}_{x_0} = I$. Hence, Algorithm 2 is a viable approach for output-feedback LQR design with unknown initial conditions.

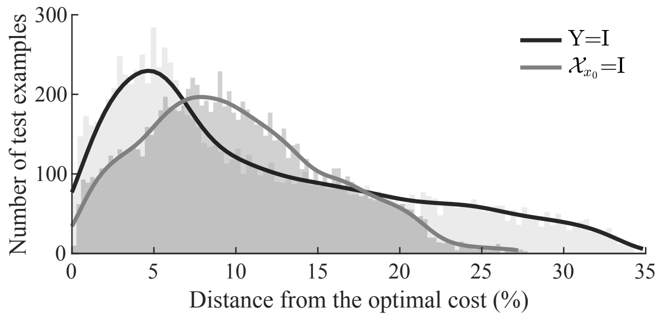


Fig. 5. Distributions of distances from optimal cost for different initial conditions.

## B. Second set of examples

The second set includes all the continuous-time SOF stabilizable examples from the COMP$l_e$ib library [55] (Table II), i.e. all the examples expect the 10 reduced order control (ROC) instances, and the 4 examples pointed out in [60], which are not continuous-time stabilizable (REA4, NN3, NN10, and NN12, indicated with the $^a$ superscript in Table II). This rich-full library contains benchmark examples from a wide spectrum of real-world applications and academic problems even with $n_x > 4000$. Therefore, we can test the behaviour and effectiveness of our proposed algorithms on large-scale stable/unstable plants as well. For better highlighting the benefits of the proposed methods, the iLMI and the BMI formulation have been evaluated on the COMP$l_e$ib library as well.

The weighting matrices for all examples have been chosen as $Q = C^T C + \alpha I$, $R = I$ and $N = 0$, where $\alpha = 10^{-9}$. The $\alpha I$ has been introduced to ensure the positive-definiteness of the $Q$ matrix, as most of the examples in the COMP$l_e$ib library are ill-conditioned causing the $C^T C$ to became negative definite due to numerical errors. The initial Lyapunov matrix for the Algorithm 2 is the optimal Lyapunov matrix from the standard state-feedback LQR design. Furthermore, the stopping criterion and maximal iteration number for Algorithm 2 have been chosen as $\epsilon = 10^{-12}$ and $maxIteration = 9 \times 10^6$. For the iLMI and BMI methods the $\mathcal{X}_{x_0}$ has been chosen as $\mathcal{X}_{x_0} = I$, and all other solver related parameters for the Mosek LMI and Penlab BMI solvers have been kept as default.

The results summarized in Table II indicate that the proposed approach is superior compared to BMI and iLMI formulations. While the proposed Algorithm 2 has solved 92% (101/110) of the examples, the iLMI formulation 63% (59/110) and the BMI formulation only 17% (19/110). That is, Algorithm 2 solved $71\%$ more examples than iLMI, and $432\%$ more than the BMI method. In addition, even with the built-in Matlab $\mathrm{lyap}$ subrutin, which is not well-suited for large-scale problems, we where able to solve examples with order higher then 4000 within minutes. The LAH example, see Table II, well demonstrates that the proposed approach is computationally much more tractable than approaches based on LMIs and/or BMIs. While the Algorithm 2 converged to a solution in $1.23\,\mathrm{ms}$, it took $31.20\,\mathrm{s}$ for the iLMI formulation, and $8.28\,\mathrm{h}$ for the BMI one.

The results with Algorithm 2 can be further divided into 4 groups.

1) Examples which can be solved without any problem with the Algorithm 2 (73%, 80/110 examples).
2) Examples where we had to use the $\mathrm{balreal}$ Matlab subrutin to balance the system matrices in order to get convergence to a solution with the Algorithm 2 (19%, 21 examples: AC9, AC14, HE5, JE1, JE2, JE3, TG1, WEC1, WEC2, WEC3, UWV, TF1, CDP, NN5, NN13, NN14, HF2D1, HF2D2, HF2D5, HF2D6, HF2D7, and HF2D8). These examples are also indicated with the $^b$ superscript and with gray color in Table II;
3) Examples where we had to allow large maximal iteration number ($> 10^9$) in order to converge to a solution with the Algorithm 2 (2%, 2 examples: AC9 and CDP).
4) Examples where the Algorithm 2 has failed to converge to a solution (8%, 9 examples: AC10, AC13, AC18, HE1, TF3, NN6, NN7, NN9, and NN17).

The COMP$l_e$ib library well demonstrates that without proper regularization or preconditioning, the proposed algorithms may fail to converge due to numerical issues. The same is true for the iLMI and BMI methods. Table II also indicates that with system balancing (in our case with the built-in Matlab $\mathrm{balreal}$ subrutin) we are able to solve $26\%$ more examples with the Algorithm 2 than without any system balancing. This number can be further increased by preconditioning the Lyapunov/Sylvester equation within the Newton's method similarly as in [36]. Furthermore, the proposed approach can be easily extended with exact line-search, similarly as it is done in [34], [35] to speed up the convergence and to reduce the overall running time even further.

In Remark 9 we have discussed that the convergence rate of the Algorithm 2 is at least sublinear. However, we have observed from the examples above that the convergence rate is in fact linear, if $\tilde{A} - \tilde{S}P^*$ has no eigenvalues on the imaginary axis. Convergence rates of the proposed Algorithm 1 and 2 on different COMP$l_e$ib plants (AC3, AC4 and DIS2), for initial Lyapunov matrices obtained from standard state-feedback LQR design, are shown in Fig. 6 and 8. Convergence rates of the Algorithm 1 and 2 on the COMP$l_e$ib plant AC3 for random initial Lyapunov matrices are shown in Fig. 7 and 9. From the figures it can be observed that the convergence rate is quadratic for Algorithm 1 and is linear for Algorithm 2, if the initial Lyapunov matrix is calculated by the standard LQR design, and that it becomes quadratic/linear in the neighbourhood of the solution when the Lyapunov matrix is randomly initialized.

In summary, the proposed algorithms, unlike other methods based on linear/bilinear matrix inequalities, NLPs, and ray-shooting methods, can solve almost all the SOF examples in the COMP$l_e$ib library (most of them within milliseconds). Until now it has been achieved only by some multivariate direct search methods applied to SOF stabilization in [60]. However, the author's attention in that publication was restricted to SOF stabilization only, i.e. no attempt was made to optimize closed-loop performance criteria relevant to control engineering. For more comparison, the readers are referred to [61], where the authors have evaluated different controller design approaches

This article has been accepted for publication in IEEE Transactions on Automatic Control. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAC.2022.3218560


on the COMP$l_e$ib library (including frequency domain approaches minimizing $H_\infty$ and/or $H_2$ norms as well).

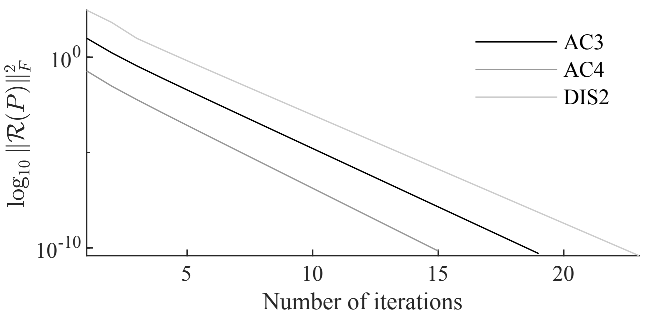


Fig. 6. Convergence rate of the Algorithm 2 on COMP$l_e$ib plants AC3, AC4 and DIS2. The initial Lyapunov matrix is obtained by the standard state-feedback LQR design. It can be observed that the convergence rate is linear.

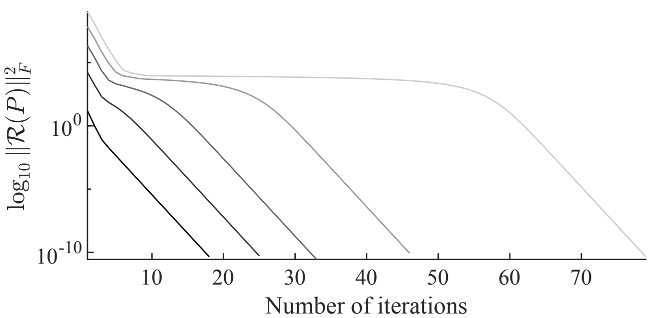


Fig. 7. Convergence rate of the Algorithm 2 on COMP$l_e$ib plant AC3 for random initial Lyapunov matrices. It can be observed that the convergence rate becomes linear in the neighbourhood of the solution.

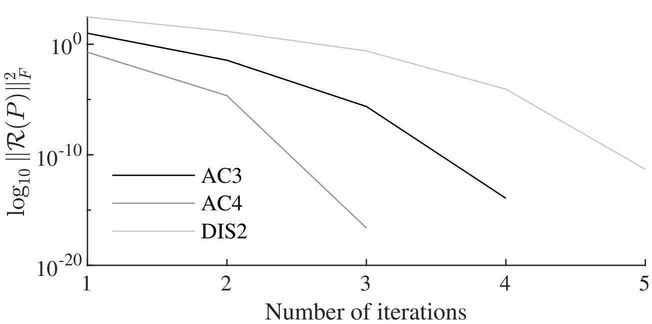


Fig. 8. Convergence rate of the Algorithm 1 on COMP$l_e$ib plants AC3, AC4 and DIS2. The initial Lyapunov matrix is obtained by the standard state-feedback LQR design. It can be observed that the convergence rate is quadratic.

## VI. Conclusions

This paper provides novel results on static output-feedback controller design for linear time-invariant systems in the LQR framework. Even though most of the output-feedback control problems are considered to be NP-hard, we show that within the LQR framework it is possible to find SOFs in sublinear (linear) time even for large-scale systems. The proposed framework, with novel necessary and sufficient conditions for

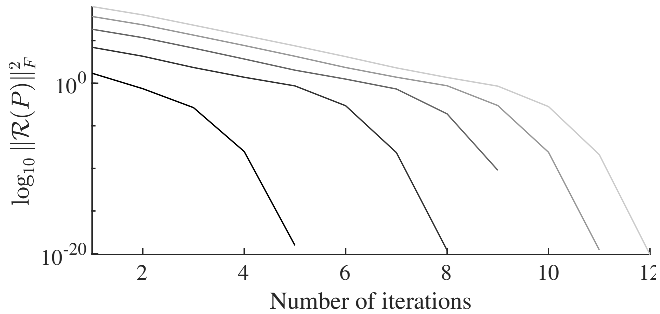


Fig. 9. Convergence rate of the Algorithm 1 on COMP$l_e$ib plant AC3 for random initial Lyapunov matrices.

output-feedback stabilizability, opens the possibility to use well known methods, such as the Newton's methods, to design SOFs with guaranteed convergence from a stabilizing state-feedback solution to a stabilizing output-feedback solution. Hence, we can get computationally efficient approaches which succeed in solving high dimensional problems where other, state-of-the-art methods fail.

The usability, tractability and effectiveness is also verified on more than 1000 numerical examples in addition to all the SOF stabilizable plants from the COMP$l_e$ib library. The proposed Algorithm 2, unlike other methods based on linear/-bilinear matrix inequalities, NLPs, can solve almost all the SOF examples in the COMP$l_e$ib library (most of them within milliseconds). Along this line, numerical results also indicate that the proposed algorithms suffer from the well known drawbacks of Newton's methods. Therefore, regularization and proper scaling is needed to improve usability of the proposed approaches for ill-conditioned problems.

In terms of future works, the Lyapunov equation can be preconditioned within the Newton's method similarly as in [36], and the proposed approaches can be easily extended with exact line-search, similarly as it is done in [34], [35].

## Appendix I
## Existence of $P \geq 0$

*Lemma 6:* Let $F \in \mathbb{R}^{n_u \times n_y}$ be given such that $A - BFC$ is stable. Substitution of $u(t) = -Fy(t) = -FCx(t)$ into the cost function (1) gives

$$J = \int_0^\infty x(t)^T \Big(Q + C^T F^T RFC - C^T F^T N^T - NFC\Big) x(t) dt. \tag{45}$$

Since $Q + C^T F^T RFC - C^T F^T N^T - NFC \geq 0$, because $R > 0$ and $Q - NR^{-1}N^T \geq 0$, and since $A - BFC$ is stable, it follows that the Lyapunov equation

$$(A - BFC)^T P + P(A - BFC) = -Q - C^T F^T RFC + C^T F^T N^T + NFC \tag{46}$$

has a unique solution $P \geq 0$.

*Proof:* For proof see [62, Lemma 12.1, p. 283]. ∎

## Appendix II
## BMI formulation of the OFLQR design problem

*Lemma 7:* The static output-feedback LQR design problem can be formulated as the following optimization problem

$$\min_{F,P}(x_0^T P x_0) \tag{47}$$

subject to BMI and LMI constraints

$$\begin{aligned}(\tilde{A}-BFC)^T P + P(\tilde{A}-BFC)& \\ +\tilde{Q}+C^T F^T RFC \leq 0&,\end{aligned} \tag{48}$$

$$P \geq 0. \tag{49}$$

*Proof:* Assume that the Lyapunov candidate

$$V(x(t)) = x(t)^T P x(t), \tag{50}$$

is positive semi-definite. Then from the Bellman-Lyapunov inequality follows

$$\dot{V}(x(t)) + J(x(t)) \leq 0 \to \dot{V}(x(t)) \leq -J(x(t)), \tag{51}$$

where

$$J = x(t)^T \tilde{Q} x(t) \geq 0, \tag{52}$$

which indicates that the closed-loop system is stable. Integrating both sides from 0 to $\infty$ we can obtain the upper bound of the cost function

$$J_\infty \leq V(x(0)) - V(x(\infty)) \leq x(0)^T P x(0), \tag{53}$$

which completes the proof. ∎

## Appendix III
## Fréchet derivative of Equation (15)

By substituting back $F = R^{-1}(B^T P + N^T)C^+$, $G = FC - R^{-1}(B^T P + N^T)$, $\tilde{S}_Q = NR^{-1}N^T$ and $\tilde{S}_A = BR^{-1}N^T$ to the equation (15), and perturbing with $X$, we can get

$$\begin{aligned}\mathcal{R}(P+X) =& \\ &+\tilde{Q}+\tilde{A}^T(P+X)+(P+X)\tilde{A}\\ &-(P+X)\tilde{S}(P+X)+Z^T(P+X)\tilde{S}(P+X)Z\\ &+Z^T(P+X)\tilde{S}_A Z - Z^T(P+X)\tilde{S}(P+X)\\ &-Z^T(P+X)\tilde{S}_A + Z^T\tilde{S}_A^T(P+X)Z\\ &+Z^T\tilde{S}_Q Z - Z^T\tilde{S}_A^T(P+X) - Z^T\tilde{S}_Q\\ &-(P+X)\tilde{S}(P+X)Z-(P+X)\tilde{S}_A Z\\ &+(P+X)\tilde{S}(P+X)+(P+X)\tilde{S}_A\\ &-\tilde{S}_A^T(P+X)Z-\tilde{S}_Q Z+\tilde{S}_A^T(P+X)+\tilde{S}_Q.\end{aligned} \tag{54}$$

By rearranging (54), we can get

$$\begin{aligned}\mathcal{R}(P+X) = \mathcal{R}(P) &+ (\tilde{A}-\tilde{S}PZ+\tilde{S}_A-\tilde{S}_A Z)^T X\\ &+X(\tilde{A}-\tilde{S}PZ+\tilde{S}_A-\tilde{S}_A Z)\\ &+(\tilde{S}PZ-\tilde{S}P+\tilde{S}_A Z-\tilde{S}_A)XZ\\ &+Z^T X(\tilde{S}PZ-\tilde{S}P+\tilde{S}_A Z-\tilde{S}_A)\\ &+Z^T X\tilde{S}XZ - Z^T X\tilde{S}X - X\tilde{S}XZ.\end{aligned} \tag{55}$$

Denoting

$$\begin{aligned}\mathcal{H}_1(P) &= (\tilde{A}-\tilde{S}PZ+\tilde{S}_A-\tilde{S}_A Z),\\ \mathcal{H}_2(P) &= (\tilde{S}PZ-\tilde{S}P+\tilde{S}_A Z-\tilde{S}_A),\\ \mathcal{L}(P,X) &= \mathcal{H}_1^T(P)X + X\mathcal{H}_1(P) + \mathcal{H}_2^T(P)XZ\\ &\quad + Z^T X\mathcal{H}_2(P),\\ \mathcal{E}_o(X) &= (Z^T X\tilde{S}XZ - Z^T X\tilde{S}X - X\tilde{S}XZ),\end{aligned}$$

we get $\mathcal{R}(P+X) = \mathcal{R}(P) + \mathcal{L}(P,X) + \mathcal{E}_o(X)$, or $\mathcal{R}(P+X) - \mathcal{R}(P) - \mathcal{L}(P,X) = \mathcal{E}_o(X)$. If $P$ is the solution of (15) then $\|\mathcal{R}(P+X)\|_F = \|\mathcal{R}(P)\|_F = 0$ and consequently $\|X\|_F = 0$. From this follows that $\lim_{\|X\|_F\to 0}\|\mathcal{E}_o(X)\|_F = 0$, and $\mathcal{L}(P,X)$ is the Fréchet derivative of (15) at $P$.

## References


[1] H. Kwakernaak and R. Sivan, *Linear optimal control systems*. Wiley-Interscience, 1972.

[2] J. Willems, "Least squares stationary optimal control and the algebraic riccati equation," *IEEE Transactions on Automatic Control*, vol. 16, no. 6, pp. 621–634, Dec 1971.

[3] B. P. Molinari, "The time-invariant linear-quadratic optimal control problem," *Automatica*, vol. 13, no. 4, pp. 347 – 357, 1977.

[4] H. L. Trentelman and J. C. Willems, *The Dissipation Inequality and the Algebraic Riccati Equation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1991, pp. 197–242.

[5] V. L. Syrmos, C. T. Abdallah, P. Dorato, and K. Grigoriadis, "Static output feedback–A survey," *Automatica*, vol. 33, no. 2, pp. 125–137, 1997.

[6] M. Sadabadi and D. Peaucelle, "From Static Output Feedback to Structured Robust Static Output Feedback: A Survey," *Annual Reviews in Control*, vol. 42, pp. 11–26, 2016.

[7] T. Iwasaki, R. Skelton, and J. Geromel, "Linear quadratic suboptimal control with static output feedback," *Systems ad Control Letters*, vol. 23, pp. 421–430, 1994.

[8] V. Veselý, "Static output feedback controller design," *Kybernetica*, vol. 37, no. 2, pp. 205–221, 2001.

[9] J. Engwerda and A. Weeren, "A result on output feedback linear quadratic control," *Automatica*, vol. 44, no. 1, pp. 265–271, 2008.

[10] V. Veselý, "Static output feedback robust controller design via LMI approach," *Journal of Electrical Engineering*, vol. 56, no. 1-2, pp. 3–8, 2005.

[11] ——, "Robust controller design for linear polytopic systems," *Kybernetika*, vol. 42, no. 1, pp. 95–110, 2006.

[12] D. Rosinová and V. Veselý, "Robust PID decentralized controller design using LMI," *International Journal of Computers, Communications & Control*, vol. 2, no. 2, pp. 195–204, 2007.

[13] V. Veselý and A. Ilka, "Design of robust gain-scheduled PI controllers," *Journal of the Franklin Institute*, vol. 352, no. 4, pp. 1476 – 1494, 2015.

[14] A. Ilka, "Matlab/Octave toolbox for structurable and robust output-feedback LQR design," *IFAC-PapersOnLine*, vol. 51, no. 4, pp. 598 – 603, 2018, 3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control PID 2018.

[15] V. Veselý and D. Rosinová, "Robust PID-PSD Controller Design: BMI Approach," *Asian Journal of Control*, vol. 15, no. 2, pp. 469–478, 2013.

[16] V. Veselý and A. Ilka, "Gain-scheduled PID controller design," *Journal of Process Control*, vol. 23, no. 8, pp. 1141–1148, Sep. 2013.

[17] A. Ilka and V. Veselý, "Gain-Scheduled Controller Design: Variable Weighting Approach," *Journal of Electrical Engineering*, vol. 65, no. 2, pp. 116–120, March-April 2014.

[18] V. Veselý and A. Ilka, "Generalized robust gain-scheduled PID controller design for affine LPV systems with polytopic uncertainty," *Systems & Control Letters*, vol. 105, pp. 6–13, 2017.

[19] A. Ilka and V. Veselý, "Robust LPV-based infinite horizon LQR design," in *2017 21st International Conference on Process Control (PC)*, June 2017, pp. 86–91.

[20] ——, "Robust guaranteed cost output-feedback gain-scheduled controller design," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11 355 – 11 360, 2017, 20th IFAC World Congress.

[21] Y. Peretz, "On application of the ray-shooting method for LQR via static-output-feedback," *Algorithms*, vol. 11, no. 1, 2018.

TABLE II
OUTPUT-FEEDBACK LQR BENCHMARKS ON COMP$l_e$IB PLANTS

| Problem description | | | | | BMI | iLMI | Alg. 2 | Problem description | | | | | BMI | iLMI | Alg. 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | $n_x$ | $n_y$ | $n_u$ | Stable | Time (s) | Time (s) | Time (s) | Name | $n_x$ | $n_y$ | $n_u$ | Stable | Time (s) | Time (s) | Time (s) |
| AC1 | 5 | 3 | 3 | Yes | - | - | 1.78E-02 | TF1[b] | 7 | 4 | 2 | Yes | - | - | 1.75E+03 |
| AC2 | 5 | 3 | 3 | Yes | - | - | 1.37E-02 | TF2 | 7 | 3 | 2 | Yes | - | - | 6.15E-02 |
| AC3 | 5 | 4 | 2 | Yes | - | 1.87E+01 | 2.30E-03 | TF3 | 7 | 3 | 2 | Yes | - | - | - |
| AC4 | 4 | 2 | 1 | No | - | 2.15E+00 | 1.59E-03 | PSM | 7 | 3 | 2 | Yes | 1.59E+00 | 3.37E+00 | 8.86E-04 |
| AC5 | 4 | 2 | 2 | No | - | - | 2.22E-03 | TL | 256 | 2 | 2 | Yes | - | - | 1.43E+00 |
| AC6 | 7 | 4 | 2 | Yes | - | 3.69E+01 | 1.49E-02 | CDP[b] | 120 | 2 | 2 | Yes | - | - | 3.57E+04 |
| AC7 | 9 | 2 | 1 | Yes | - | 9.17E+00 | 7.12E-03 | NN1 | 3 | 2 | 1 | No | - | 2.15E+01 | 1.13E-04 |
| AC8 | 9 | 5 | 1 | Yes | - | 3.18E+00 | 6.97E-03 | NN2 | 2 | 1 | 1 | Yes | 5.28E-01 | 1.10E+00 | 1.37E-04 |
| AC9[b] | 10 | 5 | 4 | Yes | - | - | 4.64E-01 | NN3[a] | 4 | 1 | 1 | No | × | × | × |
| AC10 | 55 | 2 | 2 | Yes | - | - | - | NN4 | 4 | 3 | 2 | Yes | - | - | 7.82E-03 |
| AC11 | 5 | 4 | 2 | No | - | - | 5.21E-02 | NN5[b] | 7 | 2 | 1 | Yes | - | - | 1.41E-01 |
| AC12 | 4 | 4 | 3 | Yes | - | 1.41E+00 | 1.12E-04 | NN6 | 9 | 4 | 1 | Yes | - | - | - |
| AC13 | 28 | 4 | 3 | Yes | - | - | - | NN7 | 9 | 4 | 1 | Yes | - | - | - |
| AC14[b] | 40 | 4 | 3 | Yes | - | - | 2.63E+01 | NN8 | 3 | 2 | 2 | Yes | 7.04E-01 | 5.14E+00 | 8.72E-04 |
| AC15 | 4 | 3 | 2 | Yes | - | 9.38E+00 | 1.37E-03 | NN9 | 5 | 2 | 3 | No | - | - | - |
| AC16 | 4 | 4 | 2 | Yes | - | 1.50E+00 | 1.35E-04 | NN10[a] | 8 | 3 | 3 | No | × | × | × |
| AC17 | 4 | 2 | 1 | Yes | 9.27E-01 | 6.12E+00 | 8.74E-04 | NN11 | 16 | 5 | 3 | Yes | 5.12E+01 | 3.36E+00 | 5.99E-04 |
| AC18 | 10 | 2 | 2 | Yes | - | - | - | NN12[a] | 6 | 2 | 2 | Yes | × | × | × |
| HE1 | 4 | 1 | 2 | Yes | - | - | - | NN13[b] | 6 | 2 | 2 | Yes | - | - | 1.19E+01 |
| HE2 | 4 | 2 | 2 | Yes | - | 5.81E+01 | 8.10E-03 | NN14[b] | 6 | 2 | 2 | Yes | - | - | 1.37E+01 |
| HE3 | 8 | 6 | 4 | Yes | - | 9.22E+00 | 2.19E-03 | NN15 | 3 | 2 | 2 | Yes | - | 7.89E+00 | 6.58E-04 |
| HE4 | 8 | 6 | 4 | Yes | - | 1.16E+02 | 1.03E-03 | NN16 | 8 | 4 | 4 | Yes | 1.17E+01 | 1.91E+00 | 1.51E-04 |
| HE5[b] | 8 | 2 | 4 | Yes | - | - | 1.19E+00 | NN17 | 3 | 1 | 2 | No | - | - | - |
| HE6 | 20 | 6 | 4 | Yes | - | 2.68E+02 | 4.42E-03 | NN18 | 1006 | 1 | 1 | Yes | - | - | 2.17E+01 |
| HE7 | 20 | 6 | 4 | Yes | - | 2.71E+02 | 3.88E-03 | CM1 | 20 | 2 | 1 | Yes | 3.18E+02 | 1.03E+01 | 4.18E-03 |
| JE1[b] | 30 | 5 | 3 | Yes | - | - | 1.07E+03 | CM2 | 60 | 2 | 1 | Yes | - | 1.37E+03 | 7.96E-03 |
| JE2[b] | 21 | 3 | 3 | Yes | - | - | 1.46E+00 | CM3 | 120 | 2 | 1 | Yes | - | - | 3.82E-02 |
| JE3[b] | 24 | 6 | 3 | Yes | - | - | 3.73E+02 | CM4 | 240 | 2 | 1 | Yes | - | - | 1.16E-01 |
| REA1 | 4 | 3 | 2 | No | - | 1.78E+01 | 5.42E-03 | CM5 | 480 | 2 | 1 | Yes | - | - | 8.81E-01 |
| REA2 | 4 | 2 | 2 | No | - | 1.79E+01 | 1.19E-02 | CM6 | 960 | 2 | 1 | Yes | - | - | 9.87E+00 |
| REA3 | 12 | 3 | 1 | Yes | - | 3.87E+00 | 3.13E-01 | TMD | 6 | 4 | 2 | Yes | - | 1.69E+01 | 9.92E-04 |
| REA4[a] | 8 | 1 | 1 | No | × | × | × | FS | 5 | 3 | 1 | Yes | - | - | 9.11E-02 |
| DIS1 | 8 | 4 | 4 | Yes | 4.56E+00 | 6.47E+00 | 1.35E-03 | DLR1 | 10 | 2 | 2 | Yes | 7.11E+00 | 1.18E+01 | 5.35E-04 |
| DIS2 | 3 | 2 | 2 | Yes | - | 4.13E+00 | 3.02E-03 | DLR2 | 40 | 2 | 2 | Yes | - | - | 1.12E-02 |
| DIS3 | 6 | 4 | 4 | Yes | - | 6.27E+00 | 3.58E-03 | DLR3 | 40 | 2 | 2 | Yes | - | - | 9.58E-03 |
| DIS4 | 6 | 6 | 4 | Yes | - | 9.13E-01 | 1.44E-04 | ISS1 | 270 | 3 | 3 | Yes | - | - | 4.57E-03 |
| DIS5 | 4 | 2 | 2 | No | - | - | 1.57E-03 | ISS2 | 270 | 3 | 3 | Yes | - | - | 2.55E-03 |
| TG1[b] | 10 | 2 | 2 | Yes | - | - | 3.53E+00 | CBM | 348 | 1 | 1 | Yes | - | - | 3.10E+02 |
| AGS | 12 | 2 | 2 | Yes | 7.52E+01 | 1.87E+01 | 7.21E-03 | LAH | 48 | 1 | 1 | Yes | 2.98E+04 | 3.12E+01 | 1.23E-03 |
| WEC1[b] | 10 | 4 | 3 | Yes | - | - | 4.52E-01 | HF2D1[b] | 3796 | 3 | 2 | No | - | - | 7.40E+02 |
| WEC2[b] | 10 | 4 | 3 | Yes | - | - | 3.32E-01 | HF2D2[b] | 3796 | 3 | 2 | No | - | - | 6.44E+02 |
| WEC3[2] | 10 | 4 | 3 | Yes | - | - | 3.25E-01 | HF2D3 | 4489 | 4 | 2 | Yes | - | - | 8.01E+02 |
| HF1 | 130 | 2 | 1 | Yes | - | - | 1.12E-01 | HF2D4 | 2025 | 4 | 2 | Yes | - | - | 6.21E+02 |
| BDT1 | 11 | 3 | 3 | Yes | 1.97E+01 | 9.36E+00 | 8.22E-03 | HF2D5[b] | 4489 | 4 | 2 | No | - | - | 1.36E+02 |
| BDT2 | 82 | 4 | 4 | Yes | - | 8.11E+02 | 5.41E-02 | HF2D6[b] | 2025 | 4 | 2 | No | - | - | 1.35E+02 |
| MFP | 4 | 2 | 3 | Yes | 1.75E+00 | - | 8.58E-02 | HF2D7[b] | 4489 | 4 | 2 | No | - | - | 9.01E+01 |
| UWV[b] | 8 | 2 | 2 | Yes | - | - | 1.86E+00 | HF2D8[b] | 2025 | 4 | 2 | No | - | - | 5.87E+01 |
| IH | 21 | 10 | 11 | Yes | - | - | 9.92E-03 | HF2D9 | 3481 | 2 | 2 | No | - | - | 7.48E+02 |
| CSE1 | 20 | 10 | 2 | Yes | 4.26E+01 | 2.75E+00 | 6.14E-04 | HF2D10 | 5 | 3 | 2 | No | - | 8.89E-01 | 2.54E-03 |
| CSE2 | 60 | 30 | 2 | Yes | - | 8.98E+00 | 1.21E-03 | HF2D11 | 5 | 3 | 2 | No | - | 1.89E-01 | 3.46E-04 |
| EB1 | 10 | 1 | 1 | Yes | 6.76E+00 | 6.14E+00 | 1.10E-03 | HF2D12 | 5 | 4 | 2 | Yes | 2.23E+00 | 1.58E-01 | 1.17E-04 |
| EB2 | 10 | 1 | 1 | Yes | 6.72E+00 | 6.15E+00 | 1.93E-03 | HF2D13 | 5 | 4 | 2 | Yes | - | 3.27E-01 | 4.89E-04 |
| EB3 | 10 | 1 | 1 | Yes | 1.43E+01 | 2.01E+00 | 2.15E-04 | HF2D14 | 5 | 4 | 2 | No | - | 4.86E-01 | 8.09E-04 |
| EB4 | 20 | 1 | 1 | Yes | - | 7.70E+00 | 1.37E-04 | HF2D15 | 5 | 4 | 2 | No | - | 3.36E-01 | 9.40E-04 |
| EB5 | 40 | 1 | 1 | Yes | - | - | 1.12E-03 | HF2D16 | 5 | 4 | 2 | No | - | 2.09E+00 | 1.05E-03 |
| EB6 | 160 | 1 | 1 | Yes | - | - | 1.01E-01 | HF2D17 | 5 | 4 | 2 | No | 1.47E+00 | 3.33E-01 | 9.58E-04 |
| PAS | 5 | 3 | 1 | Yes | - | - | 9.41E-04 | HF2D18 | 5 | 2 | 2 | No | - | 3.12E-01 | 5.44E-04 |

This article has been accepted for publication in IEEE Transactions on Automatic Control. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAC.2022.3218560



```
function [F,P,iteration,critFun]=algorithm1(A,B,C,Q,R,N,P,maxIteration,stopCrit)
    iteration = 0;      % initializing the iteration number
    critFun = inf;      % initializing the critterial function
    Z = pinv(C)*C;
    I = speye(size(A));
    while (iteration < maxIteration) && critFun > stopCrit
        K = R\(B'*P + N');
        F = K*pinv(C);
        Ac = (A - B*F*C);
        RE = Ac'*P + P*Ac + Q + C'*F'*R*F*C - C'*F'*N' - N*F*C;
        U = B*(F*C-K);
        KR = kron(I,Ac') + kron(Z',U') + kron(Ac',I) + kron(U',Z');
        P = P + mat(KR\vec(-RE));
        critFun = trace(RE^2);
        iteration = iteration + 1;
    end
end
```

LISTING 1. Algorithm 1 implemented in Matlab/Octave programming language.

```
function [F,P,iteration,critFun]=algorithm2(A,B,C,Q,R,N,P,maxIteration,stopCrit)
    iteration = 0;      % initializing the iteration number
    critFun = inf;      % initializing the critterial function
    while (iteration < maxIteration) && critFun > stopCrit
        K = R\(B'*P + N');
        F = K*pinv(C);
        RE = (A - B*F*C)'*P + P*(A - B*F*C) + Q + C'*F'*R*F*C - C'*F'*N' - N*F*C;
        P = P + lyap((A - B*K)',RE);
        critFun = trace(RE^2);
        iteration = iteration + 1;
    end
end
```

LISTING 2. Algorithm 2 implemented in Matlab/Octave programming language.

```
function [F,P,Y,iterationIL,critFunIL,iterationOL,critFunOL]=algorithm3(A,B,C,Q,R,N,P,Y,Xx0,
    maxIterationOL,maxIterationIL,stopCritOL,stopCritIL)
    iterationOL = 0;      % initializing the outer-loop iteration number
    critFunOL = inf;      % initializing the outer-loop critterial function
    while (iterationOL < maxIterationOL) && critFunOL > stopCritOL
        iterationIL = 0;      % initializing the inner-loop iteration number
        critFunIL = inf;      % initializing the inner-loop critterial function
        while (iterationIL < maxIterationIL) && critFunIL > stopCritIL
            K = R\(B'*P + N');
            F = K*Y*C'/(C*Y*C');
            RE = (A - B*F*C)'*P + P*(A - B*F*C) + Q + C'*F'*R*F*C - C'*F'*N' - N*F*C;
            P = P + lyap((A - B*K)',RE);
            critFunIL = trace(RE^2);
            iterationIL = iterationIL + 1;
        end
        LYE=(A-B*F*C)*Y+Y*(A-B*F*C)'+Xx0;
        Y=lyap((A-B*F*C),Xx0);
        critFunOL = trace(LYE^2);
        iterationOL = iterationOL + 1;
    end
end
```

LISTING 3. Algorithm 3 implemented in Matlab/Octave programming language.


[22] D. Moerder and A. Calise, "Convergence of a numerical algorithm for calculating optimal output feedback gains." *IEEE Transactions on Automatic Control*, vol. 30, pp. 900–903, 1985.

[23] H. Toivonen, "A globally convergent algorithm for the optimal constant output feedback problem," *International Journal of Control*, vol. 41, pp. 1589–1599, 1985.

[24] D. Moerder and A. Calise, "Convergence of a numerical algorithm for calculating optimal output feedback gains," *IEEE Transactions on Automatic Control*, vol. 30, pp. 900–903, 1985.

[25] E. J. Davison, N. S. Rau, and F. V. Palmay, "The optimal decentralized control of a power system consisting of a number of interconnected synchronous machines†," *International Journal of Control*, vol. 18, no. 6, pp. 1313–1328, 1973.

[26] D. Petersson and J. Löfberg, "LPV H2-controller synthesis using nonlinear programming," in *Proceedings of the 18th IFAC World Congress*, 2011.

[27] Y. Peretz, "A randomized approximation algorithm for the minimal-norm static-output-feedback problem," *Automatica*, vol. 63, pp. 221 – 234, 2016.

[28] J. Yu, "A Convergent Algorithm for Computing Stabilizing Static Output Feedback Gains," *IEEE Transactions on Automatic Control*, vol. 49, no. 12, pp. 2271–2275, 2004.

[29] V. Kučera and C. E. De Souza, "A necessary and sufficient conditions for output feedback stabilizability," *Automatica*, vol. 31, no. 9, pp. 1357–1359, 1995.

[30] V. Blondel and J. Tsitsiklis, "NP-Hardness of some linear control design

problems," *SIAM Journal on Control and Optimization*, vol. 35, no. 6, pp. 2118–2127, 1997.

[31] A. Nemirovskii, "Several NP-hard problems arising in robust stability analysis," *Mathematics of Control, Signals and Systems*, vol. 6, no. 2, pp. 99–105, Jun 1993.

[32] M. Fu, "Pole placement via static output feedback is NP-hard," *IEEE Transactions on Automatic Control*, vol. 49, no. 5, pp. 855–857, May 2004.

[33] A. Ilka, N. Murgovski, and J. Sjöberg, "An iterative Newton's method for output-feedback LQR design for large-scale systems with guaranteed convergence," in *18th European Control Conference (ECC)*, June 2019, pp. 4849–4854.

[34] P. Benner and R. Byers, "Newton's method with exact line search for solving the algebraic riccati equation," *IEEE Transactions on Automatic Control*, vol. 43, pp. 101–107, 1998.

[35] C. Guo and A. Laub, "On a Newton-Like Method for Solving Algebraic Riccati Equations," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 2, pp. 694–698, 2000.

[36] J.-P. Chehab and M. Raydan, "Inexact Newton's method with inner implicit preconditioning for algebraic Riccati equations," *Computational and Applied Mathematics*, vol. 36, pp. 955–969, 2017.

[37] R. Rinehart, "The derivative of a matric function," *Proceedings of the American Mathematical Society*, vol. 7, no. 1, pp. 2–5, 1956.

[38] T. Werner, "On using the complex step method for the approximation of Fréchet derivatives of matrix functions in automorphism groups," *arXiv e-prints*, pp. 1–30, Dec. 2021.

[39] N. J. Higham, *Functions of Matrices: Theory and Computation*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2008.

[40] R. G. Bartle, "Newton's method in banach spaces," *Proceedings of the American Mathematical Society*, vol. 6, no. 5, pp. 827–831, 1955. [Online]. Available: http://www.jstor.org/stable/2032941

[41] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[42] M. Dehghan and M. Hajarian, "Construction of an iterative method for solving generalized coupled Sylvester matrix equations," *Transactions of the Institute of Measurement and Control*, vol. 35, no. 8, pp. 961–970, 2013.

[43] M. Hajarian, "Extending the CGLS algorithm for least squares solutions of the generalized Sylvester-transpose matrix equations," *Journal of the Franklin Institute*, vol. 353, no. 5, pp. 1168 – 1185, 2016, special Issue on Matrix Equations with Application to Control Theory.

[44] ——, "Finite algorithms for solving the coupled Sylvester-conjugate matrix equations over reflexive and Hermitian reflexive matrices," *International Journal of Systems Science*, vol. 46, no. 3, pp. 488–502, 2015.

[45] P. Lancester and M. Tismenetsky, *The Theory of Matrices*, 2nd ed. Orlando: Academic Press, 1985.

[46] B. Arslan, V. Noferini, and F. Tisseur, "The structured condition number of a differentiable map between matrix manifolds, with applications," *SIAM Journal on Matrix Analysis and Applications*, vol. 40, no. 2, pp. 774–799, 2019.

[47] N. Higham and S. Relton, "Higher order fréchet derivatives of matrix functions and the level-2 condition number," *S I A M Journal on Matrix Analysis and Applications*, vol. 35, no. 3, pp. 1019–1037, 2014.

[48] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal control*, 3rd ed. Wiley, feb 2012.

[49] T. Johnson and M. Athans, "On the design of optimal constrained dynamic compensators for linear constant systems," *IEEE Transactions on Automatic Control*, vol. 15, no. 6, pp. 658–660, December 1970.

[50] W. Levine and M. Athans, "On the determination of the optimal constant output feedback gains for linear multivariable systems," *IEEE Transactions on Automatic Control*, vol. 15, no. 1, pp. 44–48, February 1970.

[51] A. E. Bryson and Y.-C. Ho, *Applied optimal control: optimization, estimation, and control*. New York: Wiley, 1975.

[52] P. Misra, "LQR design with prescribed damping and degree of stability," in *Proceedings of Joint Conference on Control Applications Intelligent Control and Computer Aided Control System Design*, Sep. 1996, pp. 68–70.

[53] N. K. GUPTA, "Frequency-shaped cost functionals - extension of linear-quadratic-gaussian design methods," *Journal of Guidance and Control*, vol. 3, no. 6, pp. 529–535, 1980.

[54] J. B. Moore and D. L. Mingori, "Robust frequency-shaped LQ control," *Automatica*, vol. 23, no. 5, pp. 641 – 646, 1987.

[55] F. Leibfritz, "COMPleib: COnstraint Matrix-optimization Problem library - a collection of test examples for nonlinear semidefinite programs, control system design and related problems." University of Trier, Department of Mathematics, D–54286 Trier, Germany, Tech. Rep., 2004.

[56] The Mathworks, Inc., *MATLAB R2018b*, The Mathworks, Inc., Natick, Massachusetts, 2018.

[57] J. Fiala, M. Kočvara, and M. Stingl, "Penlab: A matlab solver for nonlinear semidefinite optimization," October 2013, submitted to Mathematical Programming Computation.

[58] *MOSEK Optimization Toolbox for MATLAB*, MOSEK ApS, 2018.

[59] J. Löfberg, " YALMIP : A Toolbox for Modeling and Optimization in MATLAB," in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[60] D. Henrion, "Solving static output feedback problems by direct search optimization," in *2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control*, 2006, pp. 1534–1537.

[61] F. Leibfritz and W. Lipinski, *COMPleib 1.0 - user manual and quick reference*, University of Trier, Department of Mathematics, D 54286 Trier, Germany, 2004.

[62] W. Wonham, *Linear Multivariable Control A Geometric Approach / by W.M. Wonham.*, 3rd ed., ser. Stochastic Modelling and Applied Probability, 10. New York, NY: Springer New York : Imprint: Springer, 1985.

**Adrian Ilka** (M'21) was born in Dunajská Streda, Trnavksý Kraj, former Czechoslovakia (now Slovakia) in 1987. He received the B.Sc. degree in industrial informatics in 2010, M.Sc. degree in technical cybernetics in 2012, and Ph.D. degree in cybernetics in 2015 from the Slovak University of Technology in Bratislava, Slovakia.

From 2015 to 2017, he was a Postdoctoral Researcher with the Signal Processing group, and from 2017 to 2019, he was a Researcher with the Mechatronics group at Chalmers University of Technology, Sweden. From 2019 to 2021 he was a Specialist with the Water Construction Company, State Enterprise, Slovakia. Since 2019, he has been a Researcher with the Institute of Robotics and Cybernetics, Slovak University of Technology in Bratislava, Slovakia, and since 2022 he has been the Head of Division of Measurement, Control, and Automation of Hydroelectric Power Plants, with the Water Construction Company, State Enterprise, Slovakia. His research interests include the areas of optimal control, robust control, output-feedbacks, linear parameter-varying modelling and control, and optimization.

Dr. Ilka is a member of the IEEE Control Systems Society since 2021, and was a recipient of the IEEE students' scientific and professional activity Award of the IEEE Control Systems Society, Czechoslovakia section in 2011, the Dean's prize for excellence in M.Sc. study at the Slovak University of Technology in Bratislava in 2012, the International Scientific Conference on Control of Power Systems Best Student Paper Award in 2014, and the Rector's prize for excellence in Ph.D. study at Slovak University of Technology in Bratislava in 2015.

**Nikolce Murgovski** received the M.S. degree in software engineering from University West, Trollhättan, Sweden, in 2007, and the M.S. degree in applied physics and the Ph.D. degree in systems and control from the Chalmers University of Technology, Gothenburg, Sweden, in 2007 and 2012, respectively.

He is currently an Associate Professor with the Department of Electrical Engineering, Chalmers University of Technology. His research interest includes optimization and optimal control in the automotive area.