Matlab/Octave toolbox for structurable and robust output-feedback LQR design

(article starts on next page)

Matlab/Octave toolbox for structurable and robust output-feedback LQR design ⋆

Adrian Ilka

Department of Electrical Engineering, Chalmers University of Technology, Hörsalsvägen 9-11, SE-412 96, Gothenburg, Sweden. E-mail: adrian.ilka@chalmers.se www.adrianilka.eu

Abstract: In this paper, a structurable robust output-feedback infinite horizon LQR design toolbox for Matlab and Octave is introduced. The aim of the presented toolbox is to fill the gap between available toolboxes for Matlab/Octave by extending the standard infinite horizon LQR design (from Matlab/Control System Toolbox, Octave/Control package) to robust and structurable output-feedback LQR design. The toolbox allows to design a robust infinite horizon output-feedback controller in forms like proportional (P), proportional-integral (PI), realizable proportional-integral-derivative (PID), realizable proportional-derivative (PD), realizable derivative (D), dynamic output-feedback (DOF), dynamic output-feedback with integral part (DOFI), dynamic output-feedback with integral and realizable derivative part (DOFID), and dynamic output-feedback with realizable derivative part (DOFD). In addition, the controller structure for all supported controller types is fully structurable. The toolbox relies on Yalmip (A Matlab/Octave Toolbox for Modeling and Optimization) and on linear matrix inequality solvers like SeDuMi, SDPT3, etc. Notions like "simple", "highly customizable", and "user-friendly" have been used and considered as main terms during the development process.

Keywords: Linear quadratic regulator, Robust control, Output-feedback, Structured controller.

1. INTRODUCTION

One of the most fundamental problems in control theory is the linear quadratic regulator (LQR) design problem (Kwakernaak and Sivan, 1972). The so-called infinite horizon linear quadratic problem of finding a control function $u^* \in \mathbb{R}^m$ for $x_0 \in \mathbb{R}^n$ that minimizes the cost functional:

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

with $R > 0$, $Q - NR^{-1}N^T \geq 0$ subject to $\dot{x}(t) = Ax(t) + Bu(t)$, $x(0) = x_0$ has been studied by many authors (Kwakernaak and Sivan, 1972; Willems, 1971; Molinari, 1977; Trentelman and Willems, 1991). However, many times, it is not possible or economically feasible to measure all the state variables. Therefore, several new algorithms have been developed that resulted in generalization of the above state-feedback problem to output-feedback (Veselý, 2001; Rosinová et al., 2003; Engwerda and Weeren, 2008; Mukhopadhyay, 1978). Subsequently, the robust static output-feedback version of the LQR design has also been studied in many papers (Rosinová and Veselý, 2004; Veselý, 2005, 2006), as well as the LQR-based proportional-integral-derivative (PID) controller design (Rosinová and Veselý, 2007; Veselý and Rosinová, 2011, 2013). The introduction of linear parameter-varying (LPV) systems (Shamma, 2012) has opened new possibilities in LQR design. Several gain-scheduled/LPV-based LQR design techniques appeared in both static output feedback (SOF) and dynamic output-feedback (DOF), not to mention the PID controller design (Veselý and Ilka, 2013; Ilka and Veselý, 2014; Veselý and Ilka, 2015a; Ilka et al., 2016, 2015; Veselý and Ilka, 2015b, 2017; Ilka and Veselý, 2017a; Ilka and McKelvey, 2017; Ilka and Veselý, 2017b).

From this short literature survey follows that necessity for preparing a toolbox for LQR-based output-feedback approaches has come to the fore. The plan is to prepare and collect a bunch of functions for structurable LQR-based output-feedback controller design which can be used with Matlab and Octave as well. In this paper, one of the functions prepared for the toolbox (**`oflqr`** function) is presented. This function allows to design a robust infinite horizon output-feedback controller in forms like proportional (P), proportional-integral (PI), realizable proportional-integral-derivative (PID), realizable proportional-derivative (PD), realizable derivative (D), dynamic output-feedback (DOF), dynamic output-feedback with integral part (DOFI), dynamic output-feedback with integral and realizable derivative part (DOFID), and dynamic output-feedback with realizable derivative part (DOFD), for uncertain linear time-invariant (LTI) systems with polytopic uncertainty. In addition, the controller structure for all supported controller types is fully structurable. The function relies on Yalmip (Löfberg, 2004) and on linear matrix inequality solvers like SeDuMi (Sturm, 1999), SDPT3 (Toh et al., 1999) etc.

⋆ This work has been financed in part by the Swedish Energy Agency (P43322-1), and by IMPERIUM (H2020 GV-06-2015).

The rest of the paper is organized into four sections. The introduction is followed by a theoretical background of the presented function in Section 2. The **oflqr** function is described in Section 3, and numerical examples are given in Section 4. Finally, Section 5 closes the paper with some concluding remarks.

The mathematical notation of the paper is as follows. Given a symmetric matrix $P = P^T \in \mathbb{R}^{n\times n}$, the inequality $P > 0$ ($P \geq 0$) denotes the positive definiteness (semi definiteness) of the matrix. Matrices, if not explicitly stated, are assumed to have compatible dimensions. I denotes the identity matrix of corresponding dimensions.

2. THEORETICAL BACKGROUND

Consider the following uncertain linear time-invariant (LTI) system with polytopic uncertainty as follows:

$$\begin{aligned} \dot{x}(t) &= A(\xi(t))x(t) + B(\xi(t))u(t), \\ y(t) &= C(\xi(t))x(t) + D(\xi(t))u(t). \end{aligned} \tag{2}$$

where $x(t) \in \mathbb{R}^n$, $y(t) \in \mathbb{R}^l$, and $u(t) \in \mathbb{R}^m$ are the state, measurable output, and the control input vectors, respectively. Matrices $A(\xi(t)) \in \mathbb{R}^{n\times n}$, $B(\xi(t)) \in \mathbb{R}^{n\times m}$, $C(\xi(t)) \in \mathbb{R}^{l\times n}$ and $D(\xi(t)) \in \mathbb{R}^{l\times m}$ belong to the convex set, a polytope with p vertices that can be formally defined as:

$$\Xi := \left\{ S(\xi(t)) = \sum_{j=1}^{p} S_j\xi_j(t), \sum_{j=1}^{p} \xi_j(t) = 1, \xi_j(t) \geq 0 \right\} \tag{3}$$

Remark 1. In the system (2), the matrix $D(\xi(t))$ can be assumed, without loss of generality, to be zero, see (Zhou et al., 1996).

The function ***oflqr*** allows to design different controller types such as P, PI, PID, PD, D, DOF, DOFI, DOFID and DOFD. The output-feedback control law for controller type DOFID can be defined as:

$$\begin{aligned} \dot{x}_c(t) &= -A_c x_c(t) - B_{c_1} y(t) - B_{c_2} \int_0^t y_i(t)dt \\ &\quad - B_{c_3} y_{d_f}(t) \\ u(t) &= -C_c x_c(t) - K_p y(t) - K_i \int_0^t y_i(t)dt \\ &\quad - K_d y_{d_f}(t) \end{aligned} \tag{4}$$

where, $x_c \in \mathbb{R}^{n_c}$ is the state vector of the dynamic controller, $y_i \in \mathbb{R}^{l_i}$ is the measurable output vector for the integral part, $y_{d_f} \in \mathbb{R}^{l_d}$ is the vector of filtered output derivatives using a derivative filter with filter coefficient N_f:

$$G_f(s) = \frac{N_f s}{s + N_f}. \tag{5}$$

Matrices $A_c \in \mathbb{R}^{n_c\times n_c}$, $B_{c_1} \in \mathbb{R}^{n_c\times l}$, $B_{c_2} \in \mathbb{R}^{n_c\times l_i}$, $B_{c_3} \in \mathbb{R}^{n_c\times l_d}$, $C_c \in \mathbb{R}^{m\times n_c}$, are the controller's gain matrices related to the dynamic controller, furthermore $K_p \in \mathbb{R}^{m\times l}$, $K_i \in \mathbb{R}^{m\times l_i}$ and $K_d \in \mathbb{R}^{m\times l_d}$ are the proportional, integral and derivative gain matrices, respectively.

For the controller design, the system (2) is augmented (with the assumption that $D = 0$, see Remark 1) with the state vector of the dynamic controller $x_c(t)$, with integral of the outputs for the integral part $z(t) = \int_0^t y_i(t)dt$, and with the filtered outputs for the controller's derivative part $y_f(t)$:

$$\begin{aligned} \dot{\tilde{x}}(t) &= A_{aug}(\xi(t))\tilde{x}(t) + B_{aug}(\xi(t))\tilde{u}(t) \\ \tilde{y}(t) &= C_{aug}(\xi(t))\tilde{x}(t) \end{aligned} \tag{6}$$

where $\tilde{x}(t)^T = [x(t)^T, x_c(t)^T, z(t)^T, y_f(t)^T]$ is the augmented state vector, $\tilde{y}(t)^T = [y(t)^T, x_c(t)^T, z(t)^T, y_{d_f}(t)^T]$ is the augmented output vector, $\tilde{u}(t) = [u(t)^T, x_c(t)^T]$ is the augmented control input vector, and

$$A_{aug}(\xi(t)) = \begin{array}{cc} & \begin{array}{cccc} c_1 & c_2 & c_3 & c_4 \end{array} \\ \begin{array}{c} r_1 \\ r_2 \\ r_3 \\ r_4 \end{array} & \begin{bmatrix} A(\xi(t)), & 0_{n\times n_c}, & 0_{n\times l_i}, & 0_{n\times l_d} \\ 0_{n_c\times n}, & 0_{n_c\times n_c}, & 0_{n_c\times l_i}, & 0_{n_c\times l_d} \\ C_i(\xi(t)), & 0_{l_i\times n_c}, & 0_{l_i\times l_i}, & 0_{l_i\times l_d} \\ B_f C_d(\xi(t)), & 0_{l_d\times n_c}, & 0_{l_d\times l_i}, & A_f \end{bmatrix} \end{array},$$

$$B_{aug}(\xi(t)) = \begin{array}{cc} & \begin{array}{cc} c_1 & c_2 \end{array} \\ \begin{array}{c} r_1 \\ r_2 \\ r_3 \\ r_4 \end{array} & \begin{bmatrix} B(\xi(t)), & 0_{n\times n_c} \\ 0_{l_i\times m}, & I_{l_i\times n_c} \\ 0_{l_i\times m}, & 0_{l_i\times n_c} \\ 0_{l_d\times m}, & 0_{l_d\times n_c} \end{bmatrix} \end{array},$$

$$C_{aug}(\xi(t)) = \begin{array}{cc} & \begin{array}{cccc} c_1 & c_2 & c_3 & c_4 \end{array} \\ \begin{array}{c} r_1 \\ r_2 \\ r_3 \\ r_4 \end{array} & \begin{bmatrix} C(\xi(t)), & 0_{l\times n_c}, & 0_{l\times l_i}, & 0_{l\times l_d} \\ 0_{n_c\times n}, & I_{n_c\times n_c}, & 0_{n_c\times l_i}, & 0_{n_c\times l_d} \\ 0_{l_i\times n}, & 0_{l_i\times n_c}, & I_{l_i\times l_i}, & 0_{l_i\times l_d} \\ B_f C_d(\xi(t)), & 0_{l_d\times n_c}, & 0_{l_d\times l_i}, & A_f \end{bmatrix} \end{array},$$

where $C_i(\xi(t)) \in \mathbb{R}^{l_i\times n}$ is the output matrix for the integrals, and $C_d(\xi(t)) \in \mathbb{R}^{l_d\times n}$ is the output matrix for the derivatives.

Finally, for the controller design the control law (4) is transformed to a form:

$$u(t) = F\tilde{y}(t) = FC_{aug}(\xi(t))\tilde{x}(t), \tag{7}$$

where

$$F = \begin{array}{cc} & \begin{array}{cccc} c_1 & c_2 & c_3 & c_4 \end{array} \\ \begin{array}{c} r_1 \\ r_2 \end{array} & \begin{bmatrix} K_p, & C_c, & K_i, & K_d \\ B_{c_1}, & A_c, & B_{c_2}, & B_{c_3} \end{bmatrix} \end{array}.$$

Remark 2. For controller types P, PI, PID, PD, D, DOF, DOFI or DOFD one can simply neglect the unwanted parts (rows/columns of dynamic, integral or derivative parts) from (6) and (7).

Remark 3. The structure of the gain matrices can be predefined. Moreover, a fully decentralized control can be achieved (if $m = l$), by defining the gain matrices in diagonal form.

Theorem 1. For the uncertain LTI system (2) an optimal (suboptimal) stabilizing controller exists in the form (4) minimizing the cost function (1), if for the given positive definite matrix X, and weighting matrices Q, R and N, the following problem has a solution:

$$\min_{F,P} trace\,(P)\,, \tag{8}$$

subject to LMIs:

$$M_j \leq 0,\ j = 1, \ldots, p \tag{9}$$

$$P > 0 \tag{10}$$

where

$$M_j = \begin{bmatrix} A_{aug_j}^T P + PA_{aug_j} + Q + H_j, & G_j^T \\ G_j, & -R^{-1} \end{bmatrix}, \tag{11}$$

$$G_j = FC_{aug_j} - R^{-1}\left(B_{aug_j}^T P + N^T\right), \tag{12}$$

$$\begin{aligned}H_j = &-(XB_{aug_j}+N)R^{-1}(B_{aug_j}^T P + N^T)\\ &-(PB_{aug_j}+N)R^{-1}(B_{aug_j}^T X + N^T)\\ &+(XB_{aug_j}+N)R^{-1}(B_{aug_j}^T X + N^T).\end{aligned} \tag{13}$$

Proof 1. Let us choose the Lyapunov function as:

$$V(t) = \tilde{x}(t)^T P \tilde{x}(t), \tag{14}$$

The first derivative of the Lyapunov function (14) is then:

$$\begin{aligned}\dot{V}(\xi(t)) &= \dot{\tilde{x}}(t)^T P \tilde{x}(t) + \tilde{x}(t)^T P \dot{\tilde{x}}(t)\\ &= \tilde{x}(t)^T \left(A_c(\xi(t))^T P + P A_c(\xi(t))\right)\tilde{x}(t),\end{aligned} \tag{15}$$

where

$$A_c(\xi(t)) = A_{aug}(\xi(t)) + B_{aug}(\xi(t)) F C_{aug}(\xi(t)). \tag{16}$$

By substituting the control law (7) to the cost function (1) we can obtain:

$$J_\infty = \int_0^\infty \tilde{x}(t)^T J(\xi(t)) \tilde{x}(t) dt \tag{17}$$

where

$$\begin{aligned}J(\xi(t)) =& Q + C_{aug}(\xi(t))^T F^T R F C_{aug}(\xi(t))\\ &+ N F C_{aug}(\xi(t)) + C_{aug}(\xi(t))^T F^T N^T.\end{aligned} \tag{18}$$

By summarizing the equations (15) and (18) the Bellman-Lyapunov inequality can be obtained in the form:

$$M(\xi(t)) = \dot{V}(\xi(t)) + J(\xi(t)) \leq 0, \tag{19}$$

Furthermore, if P is positive definite then the Bellman-Lyapunov inequality (19) can be rewritten to this form:

$$\dot{V}(\xi(t)) + J(\xi(t)) \leq 0 \rightarrow \dot{V}(\xi(t)) \leq -J(\xi(t)) \leq 0 \tag{20}$$

Integrating both sides form 0 to ∞ one can obtain:

$$J_\infty \leq V(0) - V(\infty) = \tilde{x}(0)^T P \tilde{x}(0). \tag{21}$$

It follows that by minimizing $trace(P)$ and by satisfying $M(\xi(t)) \leq 0$ as well as $P > 0$, the closed-loop system will be quadratically stable with guaranteed cost defined by (21). In order to obtain LMI conditions, the matrix $M(\xi(t))$ can be rewritten to:

$$\begin{aligned}M(\xi(t)) =& A_c(\xi(t))^T P + P A_c(\xi(t) + Q\\ &+ C_{aug}(\xi(t))^T F^T R F C_{aug}(\xi(t))\\ &+ N F C_{aug}(\xi(t)) + C_{aug}(\xi(t))^T F^T N^T.\end{aligned} \tag{22}$$

Let us define:

$$G(\xi(t)) = F C_{aug}(\xi(t)) - R^{-1}\left(B_{aug}(\xi(t))^T P + N^T\right) \tag{23}$$

Substituting (23) to (22) and applying the Schur complement we can obtain:

$$M(\xi(t)) = \begin{bmatrix} M_{11}(\xi(t)), G(\xi(t))^T \\ G(\xi(t)), \quad -R^{-1}\end{bmatrix}, \tag{24}$$

where

$$M_{11}(\xi(t)) = A(\xi(t))^T P + P A(\xi(t)) + Q + H(\xi(t)), \tag{25}$$

$$H(\xi(t)) = -(PB(\xi(t)) + N)R^{-1}(B(\xi(t))^T P + N^T). \tag{26}$$

We can linearize the nonlinear part in (26) as:

$$\begin{aligned}&lin(H(\xi(t))) =\\ &-(XB_{aug}(\xi(t))+N)R^{-1}(B_{aug}(\xi(t))^T P + N^T)\\ &-(PB_{aug}(\xi(t))+N)R^{-1}(B_{aug}(\xi(t))^T X + N^T)\\ &+(XB_{aug}(\xi(t))+N)R^{-1}(B_{aug}(\xi(t))^T X + N^T),\end{aligned} \tag{27}$$

hence, we get an iterative procedure, where in each iteration holds $X|_i = P|_{i-1}$ (i - actual iteration step). The iteration ends if $|trace(P_i) - trace(P_{i-1})| \leq \epsilon$, where ϵ can be set by the designer.

Since $M(\xi(t))$ is convex in the uncertain parameter ξ, therefore $M(\xi(t))$ will be negative semi-definite if and only if it is negative semi-definite at the corners of ξ. Hence, semi-definiteness splits to p inequalities $\rightarrow$ (9).

Remark 4. For the first iteration $X|_1$ is a freely chosen positive definite matrix. It can be set by the designer or can be calculated/approximated by a standard LQR design using the nominal system.

Remark 5. The weighting matrices Q, R and N are also augmented since the state and control input vectors are augmented as well.

3. FUNCTION DESCRIPTION

The following command (in Matlab/Octave):
`[F,P,E]=oflqr(sys,Q,R,N,ct,Opt)` calculates the (sub)optimal robust structurable output-feedback gain matrix F such that, for a continuous-time polytopic state-space model *sys*, the output-feedback law defined with *ct* (control type: P, PI, PID, PD, D or DOF, DOFI, DOFID, DOFD) guarantees the robust closed-loop stability (quadratic stability) and minimizes the cost function (1), subject to the system dynamics:

$$\begin{aligned}\dot{x}(t) &= A_j x(t) + B_j u(t),\ j = 1,\dots,p\\ y(t) = C_j x(t);\ y_i(t) &= C_{i_j} x(t);\ y_d(t) = C_{d_j} x(t),\end{aligned} \tag{28}$$

where $x(t)$, $u(t)$ and $y(t)$ are state, control input and measurable output vectors, respectively. Furthermore, $y_i(t)$ and $y_d(t)$ are measurable output vectors for the integral and derivative parts of the controller.

INPUTS

REQUIRED:

- `SYS` - state-space LTI systems (in convex polytopic form)
  - ▷ array of ss models: `sys(:,:,1:p)`, (Matlab)
  - ▷ cels of ss objects: `sys{1:p}`, (Matlab, Octave)
  - ▷ single ss object: `sys`, (Matlab, Octave)
- `Q` - weighting matrix related to states ($Q \geq 0$ if $N = 0$)
- `R` - positive definite weighting matrix ($R > 0$)
- `N` - If $N \neq 0$ then $Q - NR^{-1}N^T \geq 0$ (use `eig` to check)
- `ct` ▷ `ct='p'`: Proportional (P) controller

$$u(t) = -K_p y(t), \tag{29}$$

$$F = [K_p]. \tag{30}$$

▷ `ct='pi'`: Proportional-Integral (PI) controller

$$u(t) = -K_p y(t) - K_i \int_0^t y_i(t) dt, \tag{31}$$

$$F = [K_p, K_i]. \tag{32}$$

▷`ct='pid'`: Proportional-Integral-Derivative (PID) controller

$$u(t) = -K_p y(t) - K_i \int_0^t y_i(t) dt - K_d y_{d_f}(t), \tag{33}$$

$$F = [K_p, K_i, K_d], \tag{34}$$

where y_{d_f} is the vector of filtered derivatives, using derivative filter (5) (default $Nf = 100$).

▷ `ct='pd'`: Proportional-Derivative (PD) controller

$$u(t) = -K_p y(t) - K_d y_{d_f}(t), \tag{35}$$
$$F = [K_p, K_d], \tag{36}$$

where y_{d_f} is the vector of filtered derivatives, using derivative filter (5) (default $Nf = 100$).

▷ `ct='d'`: Derivative (D) controller

$$u(t) = -K_d y_{d_f}(t), \tag{37}$$
$$F = [K_d], \tag{38}$$

where y_{d_f} is the vector of filtered derivatives, using derivative filter (5) (default $Nf = 100$).

▷ `ct='dof'`: Dynamic output-feedback with order n_c (default $n_c = 2$)

$$\begin{aligned} \dot{x}_c(t) &= -A_c x_c(t) - B_c y(t), \\ u(t) &= -C_c x_c(t) - K_p y(t), \end{aligned} \tag{39}$$

$$F = \begin{bmatrix} K_p, C_c \\ B_c, A_c \end{bmatrix}. \tag{40}$$

▷ `ct='dofi'`: Dynamic output-feedback with integral part and order n_c (default $n_c = 2$)

$$\begin{aligned} \dot{x}_c(t) &= -A_c x_c(t) - B_{c_1} y(t) - B_{c_2} \int_0^t y_i(t)dt \\ u(t) &= -C_c x_c(t) - K_P y(t) - K_i \int_0^t y_i(t)dt, \end{aligned} \tag{41}$$

$$F = \begin{bmatrix} K_p, C_c, K_i \\ B_{c_2}, A_c, B_{c_2} \end{bmatrix}. \tag{42}$$

▷ `ct='dofd'`: Dynamic output-feedback with filtered derivative part; order n_c (default $n_c = 2$)

$$\begin{aligned} \dot{x}_c(t) &= -A_c x_c(t) - B_{c_1} y(t) - B_{c_3} y_{d_f}(t), \\ u(t) &= -C_c x_c(t) - K_P y(t) - K_d y_{d_f}(t), \end{aligned} \tag{43}$$

$$F = \begin{bmatrix} K_p, C_c, K_d \\ B_{c_2}, A_c, B_{c_3} \end{bmatrix}. \tag{44}$$

where y_{d_f} is the vector of filtered derivatives, using derivative filter (5) (default $Nf = 100$).

▷ `ct='dofid'`: Dynamic output-feedback with integral and filtered derivative part; order n_c (default $n_c = 2$)

$$\begin{aligned} \dot{x}_c(t) &= -A_c x_c(t) - B_{c_1} y(t) \\ &\quad - B_{c_2} \int_0^t y_i(t)dt - B_{c_3} y_{d_f}(t), \\ u(t) &= -C_c x_c(t) - K_P y(t) \\ &\quad - K_i \int_0^t y_i(t)dt - K_d y_{d_f}(t), \end{aligned} \tag{45}$$

$$F = \begin{bmatrix} K_p, C_c, K_i, K_d \\ B_{c_2}, A_c, B_{c_2}, B_{c_3} \end{bmatrix}. \tag{46}$$

where y_{d_f} is the vector of filtered derivatives, using derivative filter (5) (default $Nf = 100$).

OPTIONAL:

`Opt` - options in structure:

▷ `Opt.iter`: maximal number of iterations (default: 100).

▷ `Opt.eps`: epsilon for the stopping criteria (default: $eps = 10^{-8}$).

▷ `Opt.epsP`: epsilon for the positive definiteness test $P \geq eps_P\, I$ (default: $eps_P = 2.2204 \times 10^{-16}$).

▷ `Opt.X`: Initial Lyapunov matrix for the iteration. If $X = 0$ then it is calculated by `lqr` command. (default: $X = 0$).

▷ `Opt.nc`: Order of the dynamic controller. i.e.: $n_c = 3$.(default: $n_c = 2$).

▷ `Opt.Nf`: Filter constant for the derivative filter. $G_f(s) = N_f s/(s + N_f)$. (default $N_f = 100$).

▷ `Opt.CS`: Controller structure matrix - which describes the controller structure. `CS` has the size of `F` and contains 1 or 0. i.e.: for `ct='p'`, and for $m = l$, to obtain fully decentralized control `CS=eye(m,l)`. (default `CS=ones(size(F))`).

▷ `Opt.Ci`: Output matrix for Integral part. (default $C_i = C$).

▷ `Opt.Cd`: Output matrix for Derivative part. (default $C_d = C$).

▷ `Opt.settings`: Options structure for Yalmip (sdpsettings). The structure is same as for the sdpsettings: `{'name', value,...}`. i.e.: `Opt.settings={'solver','sdpt3'}`.

OUTPUTS

`F` - static output-feedback gain matrix
`P` - Lyapunov matrix
`E` - Closed-loop system eigenvalues

OTHER INFO

Weighting matrix size (Q,R,N):

```
ct=='p':     Q(n,n), R(m,m), N(n,m)
ct=='pi':    Q(n+li,n+li), R(m,m),
             N(n+li,m)
ct=='pid':   Q(n+li+ld,n+2*li+ld),
             R(m,m), N(n+li+ld,m)
ct=='pd':    Q(n+ld,n+ld), R(m,m),
             N(n+ld,m)
ct=='d':     Q(n+ld,n+ld), R(m,m),
             N(n+ld,m)
ct=='dof':   Q(n+nc,n+nc), R(m+nc,m+nc),
             N(n+nc,m+nc)
ct=='dofi':  Q(n+nc+li,n+nc+li),
             R(m+nc,m+nc),
             N(n+nc+li,m+nc)
ct=='dofid': Q(n+nc+li+ld,n+nc+li+ld),
             R(m+nc,m+nc),
             N(n+nc+li+ld,m+nc)
```

where

`n` - number of states,
`m` - number of inputs,
`l` - number of outputs,
`li` - number of outputs for integral part, (def. `li=l`),
`ld` - number of outputs for deriv. part, (def. `ld=l`),
`nc` - order of the dynamic controller (def. `nc=2`).

REQUIREMENTS

Matlab:
- Control System Toolbox installed.
- YALMIP installed (R2015xxx or newer).
- LMI solver installed (sdpt3, sedumi, mosek, ...).

Octave:
- Control package installed and loaded.
- YALMIP installed (R2015xxx or newer).
- LMI solver installed (sdpt3, sedumi, ...).

4. EXAMPLES

In order to show the viability of the previous proposed method, the following examples have been chosen.

Example 1. The first example is the Rosenbrock system (Rosenbrock, 1970), which will be used to demonstrate and compare the proposed method with the standard LQR design. The transfer function of the system is as follows:

$$G(s) = \begin{bmatrix} \frac{1}{s+1}, & \frac{2}{s+3} \\ \frac{1}{s+1}, & \frac{1}{s+1} \end{bmatrix}, \tag{47}$$

which can be transformed to the form (2) with matrices:

$$A = \begin{bmatrix} -1, & 0, & 0, & 0 \\ 0, & -3, & 0, & 0 \\ 0, & 0, & -1, & 0 \\ 0, & 0, & 0, & -1 \end{bmatrix}, B = \begin{bmatrix} 1, 0 \\ 0, 2 \\ 1, 0 \\ 0, 1 \end{bmatrix},$$

$$C = \begin{bmatrix} 1, 1, 0, 0 \\ 0, 0, 1, 1 \end{bmatrix}, D = \begin{bmatrix} 0, 0 \\ 0, 0 \end{bmatrix}.$$

Different controller types were designed using the **oflqr** function. Beside types P, PI, PID, PD, DOF, DOFI, DOFID and DOFD, state-feedbacks like static state-feedback (SSF), dynamic state-feedback (DOF) and their variations were also designed (by changig the C matrix). Numerical solution has been carried out by SDPT3 (Toh et al., 1999) solver under OCTAVE 4.0 using YALMIP R20150918 (Löfberg, 2004). The obtained guaranteed cost (J_∞) for $x_0 = [1, 1, 1, 1]$, $Q = I_{n^*}$, $R = I_{m^*}$ and $N = 0.1\,ones(n^*, m^*)$ can be found in Table 1. (n^*, m^* denotes the augmented number of states and inputs for the given control type).

Table 1. Controller types & guaranteed costs

Controller type	J_∞
Standard infinite-horizon LQR:	
▷ SSF	0.9743
Proposed method:	
▷ SSF	0.9743
▷ SSFI	2.8167
▷ SSFID	3.5114
▷ SSFD	1.9618
▷ DSF ($n_c = 1$)	0.9649
▷ DSF ($n_c = 2$)	0.9554
▷ DSFI ($n_c = 2$)	2.8077
▷ DSFID ($n_c = 2$)	3.4493
▷ DSFD ($n_c = 2$)	1.8904
▷ Centralized P	0.9797
▷ Decentralized P	1.1566
▷ Centralized PI	2.8633
▷ Decentralized PI	3.2784
▷ Centralized PID	3.5635
▷ Decentralized PID	4.1662
▷ Centralized PD	1.9695
▷ Decentralized PD	2.7269
▷ Centralized DOF ($n_c = 1$)	0.9702
▷ Centralized DOF ($n_c = 2$)	0.9607
▷ Decentralized DOF ($n_c = 2$)	1.1412
▷ Centralized DOFI ($n_c = 2$)	2.8542
▷ Centralized DOFID ($n_c = 2$)	3.5008
▷ Decentralized DOFID ($n_c = 2$)	4.1458
▷ Centralized DOFD ($n_c = 2$)	1.8981

Example 2. The second example is the aircraft pitch control problem from the Control Tutorials for Matlab and Simulink (Messner et al., 2017). The state-space model for one of Boeing's commercial aircraft is given as:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -0.313, & 56.7, & 0 \\ -0.0139, & -0.426, & 0 \\ 0, & 56.7, & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} 0.232 \\ 0.0203 \\ 0 \end{bmatrix} \delta, \tag{48}$$
$$y = [0,\, 0,\, 1][\alpha,\, q,\, \theta]^T$$

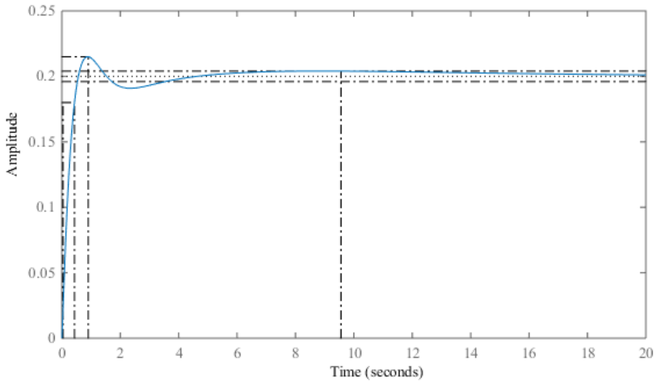

Fig. 1. Step response for 0.2 radians

where α is the angle of attack, q is the pitch rate, θ is the pitch angle and δ is the elevator deflection angle. The design requirements are the following: overshoot less than 10%, rise time less than 2 seconds, settling time less than 10 seconds, steady-state error less than 2%.

Using the LQR-based controller design approaches the controller parameter's tuning is replaced by the tuning of the weighting parameters. This can relevantly reduce the time and complexity of the tuning process, mainly for large-scale multi-input multi-output applications. For more information and tuning approaches the readers are referred to books (Athans and Falb, 1966; Dorato et al., 2000) and references therein.

After a short iterative tuning all the requirements were fulfilled by $Q = diag([0, 0, 0, 0.6, 28])$, $R = 1$ and $N = [0; 0; 0; 0; -4.5]$ (step response: Fig. 1) with rise time: 0.954 seconds, settling time: 9.5603 seconds, overshoot: 7.496%, steady-state error: 0%. The obtained realizable PID gains are: $K_p = 5.1724$, $K_i = 1.7031$, $K_d = 3.0076$, with filter coefficient $N_f = 100$.

Example 3. The third example is a simple uncertain MIMO system, which will be used to demonstrate the freedom in structurability what the **oflqr** can give. For example, different controller types can be designed for each subsystem at once. The system with parametric uncertainty is given as:

$$G(s) = \begin{bmatrix} \frac{\langle 1,2 \rangle}{10s+1}, & \frac{2}{s+1} \\ \frac{1}{s+1}, & \frac{\langle -3,-4 \rangle}{10s+1} \end{bmatrix}, \tag{49}$$

which can be transformed to the form (2) with matrices:

$$A_{1,2,3,4} = \begin{bmatrix} -0.1, & 0, & 0, & 0 \\ 0, & -1, & 0, & 0 \\ 0, & 0, & -1, & 0 \\ 0, & 0, & 0, & -0.1 \end{bmatrix}, B_1 = \begin{bmatrix} 0.1, & 0 \\ 0, & 2 \\ 1, & 0 \\ 0, & -0.3 \end{bmatrix},$$

$$B_2 = \begin{bmatrix} 0.1, & 0 \\ 0, & 2 \\ 1, & 0 \\ 0, & -0.4 \end{bmatrix}, B_3 = \begin{bmatrix} 0.2, & 0 \\ 0, & 2 \\ 1, & 0 \\ 0, & -0.3 \end{bmatrix}, B_4 = \begin{bmatrix} 0.2, & 0 \\ 0, & 2 \\ 1, & 0 \\ 0, & -0.4 \end{bmatrix}$$

$$C_{1,2,3,4} = \begin{bmatrix} 1, 1, 0, 0 \\ 0, 0, 1, 1 \end{bmatrix}, D_{1,2,3,4} = \begin{bmatrix} 0, 0 \\ 0, 0 \end{bmatrix}.$$

Assume that we want to design a fully decentralized controller, more precisely a PI controller for the first subsystem and a PID for the second subsystem. In order to do so, let's define the output matrix for the derivative part just for the second subsystem: `Opt.Cdj=Cj(2,:)`. Finally, let us construct the structure matrix `Opt.CS`:

$$CS = \begin{array}{c} \\ u_1 \\ u_2 \end{array} \begin{array}{c} \begin{array}{ccccc} y_1 & y_2 & z_1 & z_2 & y_{d_{f\,2}} \end{array} \\ \left[\begin{array}{ccccc} 1, & 0, & 1, & 0, & 0 \\ 0, & 1, & 0, & 1, & 1 \end{array}\right] \end{array}. \tag{50}$$

Numerical solution has been carried out by SDPT3 (Toh et al., 1999) solver under OCTAVE 4.0 using YALMIP R20150918 (Löfberg, 2004). The obtained PI and PID gains by weighting matrices $Q = I_{n+l_i+l_d}$, $R = I_m$ and $N = 0_{n+l_i+l_d \times m}$ are as follows:

$$\begin{aligned} &\text{PI:} \quad K_p = 1.6344,\ K_i = 1.0014, \\ &\text{PID:}\ K_p = -1.2735,\ K_i = -0.5119, \\ &\qquad\ K_d = -0.0403. \end{aligned} \tag{51}$$

5. CONCLUSION

A new toolbox for Matlab and Octave is introduced in this paper. More precisely a function from the toolbox, which can be used to design a structurable robust LQR-based output feedback controller for uncertain LTI systems. The toolbox will soon be enriched by the discrete version of the presented **oflqr** function. Moreover, the future plan is to include the author's all recent results in linear parameter-varying (LPV)/ gain-scheduled output-feedback controller design as functions in the presented toolbox. For recent updates please visit: www.adrianilka.eu/oflqrtoolbox.htm

REFERENCES

Athans, M. and Falb, P. (1966). *Optimal control*. McGraw-Hill Publishing Company, Ltd, Maidenhead, Berksh, New York. Pp. 879.

Dorato, P., Abdallah, C.T., and Cerone, V. (2000). *Linear Quadratic Control: An Introduction*. Krieger Publishing Company.

Engwerda, J. and Weeren, A. (2008). A result on output feedback linear quadratic control. *Automatica*, 44(1), 265–271.

Ilka, A. and McKelvey, T. (2017). Robust discrete-time gain-scheduled guaranteed cost PSD controller design. In *2017 21st Int. Conference on Process Control*, 54–59.

Ilka, A. and Veselý, V. (2017a). Robust LPV-based infinite horizon LQR design. In *2017 21st International Conference on Process Control (PC)*, 86–91.

Ilka, A., Ottinger, I., Ludwig, T., Tárník, M., Veselý, V., Miklovicová, E., and Murgaš, J. (2015). Robust Controller Design for T1DM Individualized Model: Gain-Scheduling Approach. *International Review of Automatic Control (IREACO)*, 8(2), 155–162.

Ilka, A. and Veselý, V. (2014). Gain-Scheduled Controller Design: Variable Weighting Approach. *Journal of Electrical Engineering*, 65(2), 116–120.

Ilka, A. and Veselý, V. (2017b). Robust guaranteed cost output-feedback gain-scheduled controller design. *IFAC-PapersOnLine*, 50(1), 11355 – 11360. 20th IFAC World Congress.

Ilka, A., Veselý, V., and McKelvey, T. (2016). Robust Gain-Scheduled PSD Controller Design from Educational Perspective. In *Preprints of the 11th IFAC Symposium on Advances in Control Education*, 354–359. Bratislava, Slovakia.

Kwakernaak, H. and Sivan, R. (1972). *Linear optimal control systems*. Wiley-Interscience.

Löfberg, J. (2004). YALMIP : A Toolbox for Modeling and Optimization in MATLAB. In *Proceedings of the CACSD Conference*. Taipei, Taiwan.

Messner, B., Tilbury, D., and JD, T. (2017). Control tutorials for matlab and simulink. Technical report, Creative Commons Attribution.

Molinari, B. (1977). The time-invariant linear-quadratic optimal control problem. *Automatica*, 13(4), 347 – 357.

Mukhopadhyay, S. (1978). P.I.D. equivalent of optimal regulatror. *Electronics Letters*, 14(25), 821–822.

Rosenbrock, H. (1970). *State-Space and Multivariable Theory*. Nelson, London, U.K.

Rosinová, D. and Veselý, V. (2004). Robust static output feedback for discrete-time systems - LMI approach. *Periodica Polytechnica Ser. El. Eng.*, 48(3-4), 151–163.

Rosinová, D. and Veselý, V. (2007). Robust PID decentralized controller design using LMI. *Int. Journal of Computers, Communications & Control*, 2(2), 195–204.

Rosinová, D., Veselý, V., and Kučera, V. (2003). A necessary and sufficient condition for static output feedback stabilizability of linear discrete-time systems. *Kybernetika*, 39(4), 447–459.

Shamma, J.S. (2012). *Control of Linear Parameter Varying Systems with Applications*, chapter An overview of LPV systems, 3–26. Springer.

Sturm, J. (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12, 625–653.

Toh, K.C., Todd, M., and Tütüncü, R.H. (1999). SDPT3 – a MATLAB software package for semidefinite programming. *Optimization methods and software*, 11, 545–581.

Trentelman, H.L. and Willems, J.C. (1991). *The Dissipation Inequality and the Algebraic Riccati Equation*, 197–242. Springer Berlin Heidelberg, Berlin, Heidelberg.

Veselý, V. (2001). Static output feedback controller design. *Kybernetica*, 37(2), 205–221.

Veselý, V. (2005). Static output feedback robust controller design via LMI approach. *Journal of Electrical Engineering*, 56(1-2), 3–8.

Veselý, V. (2006). Robust controller design for linear polytopic systems. *Kybernetika*, 42(1), 95–110.

Veselý, V. and Ilka, A. (2013). Gain-scheduled PID controller design. *Jour. of Process Cont.*, 23(8), 1141–1148.

Veselý, V. and Ilka, A. (2015a). Design of robust gain-scheduled PI controllers. *Journal of the Franklin Institute*, 352(4), 1476 – 1494.

Veselý, V. and Ilka, A. (2015b). Unified Robust Gain-Scheduled and Switched Controller Design for Linear Continuous-Time Systems. *International Review of Automatic Control (IREACO)*, 8(3), 251–259.

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

Veselý, V. and Rosinová, D. (2011). Robust PSD Controller Design. In M. Fikar and M. Kvasnica (eds.), *Proceedings of the 18th International Conference on Process Control*, 565570. Tatranská Lomnica, Slovakia.

Veselý, V. and Rosinová, D. (2013). Robust PID-PSD Controller Design: BMI Approach. *Asian Journal of Control*, 15(2), 469–478.

Willems, J. (1971). Least squares stationary optimal control and the algebraic riccati equation. *IEEE Transactions on Automatic Control*, 16(6), 621–634.

Zhou, K., Doyle, C.J., and Glover, K. (1996). *Robust and optimal control*. Prentice-Hall Inc., U.S. River, USA.