A Nonlinear Model Predictive Control Tracking Application for a System of Cascaded Tanks

Guilherme N. G. dos Reis[1]*, Marcelo M. Morato[1], Julio E. Normey-Rico[1], Vladimir Stojanovic[2]
[1]Dept. de Automação e Sistemas, Univ. Fed. de Santa Catarina, Florianópolis (Brazil)
[2]Faculty of Mechanical and Civil Engineering, Univ. of Kragujevac, Kraljevo (Serbia)

Nonlinear Model Predictive Control (NMPC) formulations through quasi-Linear Parameter Varying (qLPV) embeddings have been brought to focus in recent literature. In this brief paper, we evaluate the application of this kind of control strategy to the reference tracking problem of a cascaded tank system. This benchmark application has four states and two control inputs, which represent the fluid inlets to the upper tanks. The levels of each of the four tanks dependent not only on these input flows, but also on bounded disturbance variables. The system exhibits nonlinearities due to the fluid dynamics, which are incorporated as state-dependent qLPV variables. This case study serves to illustrate how a Sequential Quadratic Program (SQP) is an elegant solution to NMPC design: the qLPV realisation of the nonlinear dynamics yields linear predictions at each sampling instant, which can be refined through sequential operations of a single QP. The resulting numerical toughness is much smaller than the Nonlinear Programs generated with “regular” NMPC design, which is very convenient. Moreover, the SQP solution provides estimates of the future scheduling parameters, with convergence properties. Using realistic simulations, we demonstrate the effectiveness of this control approach with respect to piecewise constant reference tracking and disturbance rejection, which are assessed using standard performance indexes.

Keywords: Nonlinear Model Predictive Control, Cascaded Tanks, Linear Parameter Varying Systems

1. INTRODUCTION

Model Predictive Control (MPC) represents a family of advanced control techniques, derived in the late 1980's for industrial processes subject to with constraints [1], since then, they have been intensively studied by academics and used by engineers. With thousands of industrial applications [2], some of the problems still open for research are nonlinear applications and their particularities. Nonlinear Model Predictive Control (NMPC) may, among other problems, bring real-time obstacles, given that the resulting numerical complexity increases.

The key step of MPC algorithms, solving an optimization problem, gets complicated in the majority of nonlinear systems: this procedure is not trivial and is frequently avoided through approximations. One possible solution explored more recently is the use of sophisticated tools for solving Nonlinear Programming (NP), as [3-5].

Linear Parameter Varying (LPV) framework has been, concurrently, expanded lately, as a way to circumvent these nonlinear issues. For a large class of nonlinear systems, it is possible to make use of a quasi-LPV (qLPV) embedding in order to redesign state equations by the means of linear maps parametrized by known and bounded scheduling variables, denoted ρ.

Nevertheless, qLPV embedding are not compatible with linear MPC techniques, since these realisations are based on the availability of the scheduling parameters, which depend on endogenous variables of the process. As the future process variables are unknown, qLPV model-based predictions are of uncertain computation, these are based on scheduling variables. The usual solution for this kind of problem would be Robust MPC (RMPC), but it usually implies excessive conservativeness.

Recent methods have provided alternative options to the robust design, considering qLPV embedding, [6-8]. Bearing in mind this context, a NMPC solution based on Sequential Quadratic Programming (SQP) is presented in this work. This approach contrasts with full nonlinear MPCs, making real-time applications possible, as well as with RMPCs, since the method does not imply in conservative results. The method is applied to a benchmark quadritank process [9], for which we demonstrate the effectiveness of the method in terms of reference tracking and disturbance rejection.

This paper is organized as follows: Section 1 introduces briefly the subject studied; Section 2 lays down the problem setup, passing through qLPV framework, qLPV MPC and SQP problems; Section 3 describes the process used as case study; and finally, on Section 4, results derived from a numerical simulation are shown.

2. PROBLEM SETUP

Consider the following discrete-time nonlinear system:

$$\begin{cases} x[k+1] = f(x[k], u[k]) \\ y[k] = g(x[k], u[k]) \end{cases} \tag{1}$$

*Corresponding author: Guilherme N. G. dos Reis, Dept. de Automação e Sistemas, Univ. Fed. de Santa Catarina, Florianópolis (Brazil)

where $x \in \mathbf{R}^{n_x}$, $u \in \mathbf{R}^{n_u}$ and $y \in \mathbf{R}^{n_y}$ are, respectively, state, input and output variables, all constrained to sets X, U and Y. Functions $f : \mathbf{R}^{n_x+n_u} \to \mathbf{R}^{n_x}$ and $g : \mathbf{R}^{n_x+n_u} \to \mathbf{R}^{n_y}$ are called the state and output maps, respectively.

2.1. qLPV-embedding

Consider that the Linear Differential Inclusion property is satisfied [10], which means that $\exists G(x,u) : \mathbf{R}^{n_x+n_u} \to \mathbf{R}^{(n_x+n_y)\times(n_x+n_u)}$ such that $\begin{bmatrix} f(x,u)^T & g(x,u)^T \end{bmatrix}^T = G(x,u)\begin{bmatrix} x^T & u^T \end{bmatrix}^T$.
Then, there exists a qLPV realization for the considered nonlinear system, as follows:

$$\begin{cases} x[k+1] = A(\rho[k])x[k] + B(\rho[k])u[k] \\ y[k] = C(\rho[k])x[k] + D(\rho[k])u[k] \\ \rho[k] = h(x[k],u[k]) \end{cases} \quad (2)$$

where $\rho \in \mathsf{P} \subseteq \mathbf{R}^{n_\rho}$ is called scheduling variable vector and $h : \mathbf{R}^{n_x+n_u} \to \mathbf{R}^{n_\rho}$, scheduling map. As all four matrix functions $A : \mathbf{R}^{n_\rho} \to \mathbf{R}^{n_x \times n_x}$, $B : \mathbf{R}^{n_\rho} \to \mathbf{R}^{n_x \times n_u}$, $C : \mathbf{R}^{n_\rho} \to \mathbf{R}^{n_y \times n_x}$ and $D : \mathbf{R}^{n_\rho} \to \mathbf{R}^{n_y \times n_u}$ derive from LDI matrix,

$$\begin{bmatrix} f(x[k],u[k]) \\ g(x[k],u[k]) \end{bmatrix} = \begin{bmatrix} A(\rho[k]) & B(\rho[k]) \\ C(\rho[k]) & D(\rho[k]) \end{bmatrix} \quad (3)$$

Note that Eq. (1) and Eq. (2) are equivalent, with the upper part of Eq. (2) is outsourcing the nonlinearities through the scheduling parameter interconnection $\rho = h(x,u)$, which depends on the endogenous variables x and u. Hence, we emphasize the quasi(-LPV) prefix given that ρ is not exogenous.

2.2. qLPV MPC Problem

As an intrinsically discrete-time technique, MPC solves, at each sampling instant, an optimization problem, which attempts to minimize a cost function subject to a set of constraints, with respect to a vector of decision variables. It does so by making use of a model of the system to be controlled in order to predict future values for the output vector and utilizing these predictions to evaluate performance.

Consecutively applying state equation from some initial condition x_{k_0} yields

$$\begin{cases} X_{k_0} = \mathsf{A}(\mathrm{P}_{k_0})x_{k_0} + \mathsf{B}(\mathrm{P}_{k_0})U_{k_0} \\ Y_{k_0} = \mathsf{C}(\mathrm{P}_{k_0})x_{k_0} + \mathsf{D}(\mathrm{P}_{k_0})U_{k_0} \\ \mathrm{P}_{k_0} = \mathsf{H}(X_{k_0}, U_{k_0}) \end{cases} \quad (4)$$

where

$$X_{k_0} = \begin{bmatrix} x[k_0+1|k_0] \\ x[k_0+2|k_0] \\ \vdots \\ x[k_0+N_p|k_0] \end{bmatrix} \quad U_{k_0} = \begin{bmatrix} u[k_0|k_0] \\ u[k_0+1|k_0] \\ \vdots \\ u[k_0+N_p-1|k_0] \end{bmatrix} \quad Y_{k_0} = \begin{bmatrix} y[k_0+1|k_0] \\ y[k_0+2|k_0] \\ \vdots \\ y[k_0+N_p|k_0] \end{bmatrix} \quad (5)$$

are the prediction vectors for, respectively, states, inputs and outputs. Also,

$$\mathrm{P}_{k_0} = \begin{bmatrix} \rho[k_0]^T & \rho[k_0+1|k_0]^T & \dots & \rho[k_0+N_p-1|k_0]^T \end{bmatrix}^T \quad (6)$$

represents the predictions for scheduling variables, which parametrizes all matrix functions $\mathsf{A} : \mathbf{R}^{N_p n_\rho} \to \mathbf{R}^{N_p(n_x \times n_x)}$, $\mathsf{B} : \mathbf{R}^{N_p n_\rho} \to \mathbf{R}^{N_p(n_x \times n_u)}$, $\mathsf{C} : \mathbf{R}^{N_p n_\rho} \to \mathbf{R}^{N_p(n_y \times n_x)}$ and $\mathsf{D} : \mathbf{R}^{N_p n_\rho} \to \mathbf{R}^{N_p(n_y \times n_u)}$, and map $\mathsf{H} : \mathbf{R}^{N_p(n_x+n_u)} \to \mathbf{R}^{N_p n_\rho}$ defines it. Previous vectors take values from instant k_0 to instant $k_0 + N_p$, thus comprising the full prediction horizon of N_p steps.

Accordingly, we consider the following quadratic cost function:

$$J_k = E_k^T \mathsf{Q} E_k + \Delta U_k^T \mathsf{R} \Delta U_k \quad (7)$$

as a performance index, where $E_k = R_k - Y_k$ is a reference tracking prediction vector and ΔU_k is the input variation prediction vector, weighted by matrices Q and R, implies on a nonlinear cost function for the problem $\mathsf{P}(x_{k_0}, R_k)$:

$$\min_{\Delta U_{k_0}} J_{k_0}$$
$$s.t.\begin{cases} X_{k_0} = \mathsf{A}(\mathrm{P}_{k_0})x_{k_0} + \mathsf{B}(\mathrm{P}_{k_0})U_{k_0} \\ Y_{k_0} = \mathsf{C}(\mathrm{P}_{k_0})x_{k_0} + \mathsf{D}(\mathrm{P}_{k_0})U_{k_0} \\ \mathrm{P}_{k_0} = \mathsf{H}(X_{k_0}, U_{k_0}) \\ X_{k_0} \in \mathsf{X} \\ U_{k_0} \in \mathsf{U} \\ Y_{k_0} \in \mathsf{Y} \\ P_{k_0} \in \mathsf{P} \end{cases} \quad (8)$$

Since a qLPV embedding was used to rewrite Eq. (1) as Eq. (2), all state nonlinearities were reallocated to the definition of the scheduling variable . Yet, problem P still has to include its nonlinear dynamics, which implicates a NP problem.

2.3 Sequential Quadratic Programming

In this paper, we follow the method from [6]. Accordingly, we solve a Sequential Quadratic Program, which is operated as follows: we solve the original optimization problem (6) by iterating a sequence of QPs, all parametrized by a scheduling vector P. Starting with some suitable initial guess P^0, the MPC optimization problem is solved as a QP, given that Eq. (4) based on P^0 is linear. Then, with the optimal values resulted from previous QP, P is updated to $P^1 = h(X_k, U_k)$ and this process is iterated until a convergence threshold is reached: $\left\| P^l - P^{l-1} \right\| \le \varepsilon$. This SQP is able to approximate the NP problem form, Eq. (6), very well, for any arbitrarily small positive scalar ε and can be solved more quickly.

However, equality constraint $P_k = \mathsf{H}(X_k, U_k)$ is the only nonlinear aspect of P. Thus, employing a SQP algorithm to solve it transposes a NP problem into a number of QP problems. This means any commercial QP solver would handle $\mathsf{P}_{SQP}(\mathrm{x}_{k_0}, \mathrm{R}_k, \mathrm{P}^l)$ on a reasonable time.

$$\min_{\Delta U_{k_0}} J_{k_0}$$
$$s.t.\begin{cases} X_{k_0} = \mathsf{A}(P^l)x_{k_0} + \mathsf{B}(P^l)U_{k_0} \\ Y_{k_0} = \mathsf{C}(P^l)x_{k_0} + \mathsf{D}(P^l)U_{k_0} \\ X_{k_0} \in \mathsf{X} \\ U_{k_0} \in \mathsf{U} \\ Y_{k_0} \in \mathsf{Y} \\ P^l \in \mathsf{P} \end{cases} \quad (9)$$

Algorithm 1: SQP qLPV MPC

$k \leftarrow k_0$
$x[k-1] \leftarrow x_{k_0}$
$u[k-1] \leftarrow u_{k_0}$
$\rho[k-1] \leftarrow h(x_{k_0}, u_{k_0})$
while control condition
  Measure $x[k]$
  $l \leftarrow 0$
  $P^l \leftarrow \alpha \vec{1}$
  while converge condition
    Solve $\mathsf{P}_{SQP}(x_{k_0}, \mathrm{R}_k, \mathrm{P}^l)$
    $P^{l+1} \leftarrow h(X_k^l, U_k^l)$
    $l \leftarrow l+1$
  Apply $u[k \mid k]$

3. CASCADED

In order to test the SQP method, we use the case study from [9]. Consider the scheme of four cascaded tanks as shown in Fig. 1. This system is an educational benchmark for illustration of performance limitations, with a focus on the adjustable multivariable zeros.

Consisting of four tanks, two valves, two pumps and a reservoir, it is shown in Fig. 1. Fed by the reservoir, both pumps fill crossed lower and upper tanks, regulated by the valves. Each tank has an outlet hole that either fills another tank or the reservoir, this way there is no loss of liquid.

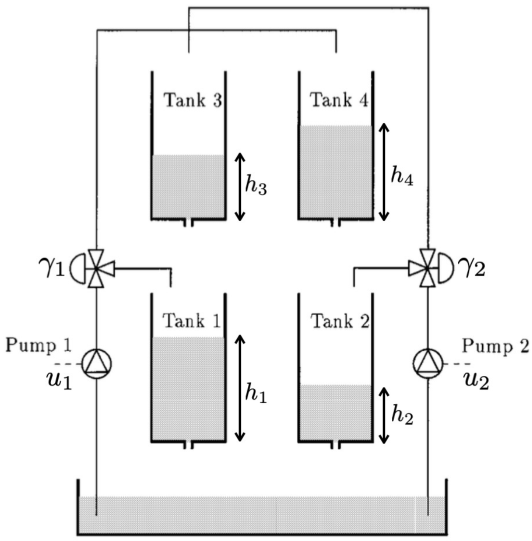

Figure 1: Cascade Tanks Process. Fig. from Johansson (2000)

From mass balance and Bernoulli's law, it is possible to model this process, yielding nonlinear ordinary differential equations, Eq. (8),

$$\frac{d}{dt}\begin{bmatrix} h_1(t) \\ h_2(t) \\ h_3(t) \\ h_4(t) \end{bmatrix} = \begin{bmatrix} -\frac{a_1}{A_1}\sqrt{2gh_1(t)} + \frac{a_3}{A_1}\sqrt{2gh_3(t)} + \frac{k_1\gamma_1(t)}{A_1}v_1(t) \\ -\frac{a_2}{A_2}\sqrt{2gh_2(t)} + \frac{a_4}{A_2}\sqrt{2gh_4(t)} + \frac{k_2\gamma_2(t)}{A_2}v_2(t) \\ -\frac{a_3}{A_3}\sqrt{2gh_3(t)} + \frac{k_2(1-\gamma_2(t))}{A_3}v_2(t) \\ -\frac{a_4}{A_4}\sqrt{2gh_4(t)} + \frac{k_1(1-\gamma_1(t))}{A_4}v_1(t) \end{bmatrix} \quad (10)$$

where h_i, $i = 1,2,3,4$, is the water level on the i-th tank, v_i, $i = 1,2$, is the voltage applied on the i-th pump and, lastly, γ_i, $i = 1,2$, is the position of the i-th valve. Parameters a_i and A_i, $i = 1,2,3,4$, are, respectively, the area of the outlet hole of the i-th tank and the bottom area of the i-th tank. Also, k_i, $i = 1,2$, is the proportionality coefficient of flow per voltage of the i-pump.

This system is clearly a nonlinear model, which can be put in state-space by taking the following state, input and output vectors, respectively:

$$\begin{aligned} x(t) &= \begin{bmatrix} h_1(t) & h_2(t) & h_3(t) & h_4(t) \end{bmatrix}^T \\ u(t) &= \begin{bmatrix} v_1(t) & v_2(t) & \gamma_1(t) & \gamma_2(t) \end{bmatrix}^T \\ y(t) &= \begin{bmatrix} h_1(t) & h_2(t) \end{bmatrix}^T \end{aligned} \quad (11)$$

Since this model satisfied the LDI property, we can obtain a qLPV realization by considering the scheduling variables vector as $\rho(t) = \begin{bmatrix} \rho_x(t)^T & \rho_u(t)^T \end{bmatrix}^T$, with $\rho_x(t) = \begin{bmatrix} 1/\sqrt{h_1(t)} & 1/\sqrt{h_2(t)} & 1/\sqrt{h_3(t)} & 1/\sqrt{h_4(t)} \end{bmatrix}^T$ and $\rho_u(t) = \begin{bmatrix} v_1(t) & v_2(t) & \gamma_1(t) & \gamma_2(t) \end{bmatrix}^T$,

$$\begin{cases} \frac{dx(t)}{dt} = A(\rho(t))x(t) + B(\rho(t))u(t) \\ y(t) = C(\rho(t))x(t) + D(\rho(t))u(t) \end{cases} \quad (12)$$

where

$$A(\rho(t)) = \begin{bmatrix} \frac{-a_1\sqrt{2g}\rho_1(t)}{A_1} & 0 & \frac{a_3\sqrt{2g}\rho_3(t)}{A_1} & 0 \\ 0 & \frac{-a_2\sqrt{2g}\rho_2(t)}{A_2} & 0 & \frac{a_4\sqrt{2g}\rho_4(t)}{A_2} \\ 0 & 0 & \frac{-a_3\sqrt{2g}\rho_3(t)}{A_3} & 0 \\ 0 & 0 & 0 & \frac{-a_4\sqrt{2g}\rho_4(t)}{A_4} \end{bmatrix}$$

$$B(\rho(t)) = \begin{bmatrix} \frac{k_1\rho_7(t)}{2A_1} & 0 & \frac{k_1\rho_5(t)}{2A_2} & 0 \\ 0 & \frac{k_2\rho_8(t)}{2A_2} & 0 & \frac{k_2\rho_6(t)}{2A_2} \\ 0 & \frac{(1-k_2/2)\rho_8(t)}{A_3} & 0 & \frac{-k_2\rho_6(t)}{2A_3} \\ \frac{(1-k_1/2)\rho_7(t)}{A_4} & 0 & \frac{-k_1\rho_5(t)}{2A_4} & 0 \end{bmatrix} \quad (13)$$

This qLPV model is discretized in order for it to be presented in the form of Eq. (2). It is noteworthy the fact that $A(\rho(t))$ depends affinely only on $\rho_x(t)$, $B(\rho(t))$ depends affinely only on $\rho_u(t)$ and C and D are constant matrices.

4. SIMULATION RESULTS

A simulation scenario is carried out to validate the SQP qLPV MPC algorithm in practice and it is described in detail in the sequence.

Starting with the model itself, it is used Eq. (1) to reproduce the process, as it is a numerical simulation, and Eq. (8) for MPC predictions, both normalized, with sampling period $t_s = 0.1s$. Table 1 displays all model parameters.

Table 1: Model Parameters

$a\ [cm^2]$	$[0.5 \quad 0.5 \quad 0.5 \quad 0.5]^T$
$A\ [cm^2]$	$[1 \quad 1 \quad 1 \quad 1]^T$
$k\ \left[\frac{cm^3}{s}\frac{1}{V}\right]$	$[1.4 \quad 1.4]^T$
$h_{k_0}\ [cm]$	$[5 \quad 5 \quad 5 \quad 5]^T$
$u_{k_0}\ [V/\%]$	$[5\sqrt{0.5} \quad 5\sqrt{0.5} \quad 0 \quad 0]^T$

Additionally, state, inputs and outputs constraint sets X, U and Y are given by

$$\mathsf{X} = \left\{ x \in \mathbb{R}^{n_x} : \begin{bmatrix}0\\0\\0\\0\end{bmatrix} \le x \le \begin{bmatrix}10\\10\\10\\10\end{bmatrix} \right\}$$

$$\mathsf{U} = \left\{ u \in \mathbb{R}^{n_u} : \begin{bmatrix}0\\0\\0\\0\end{bmatrix} \le u \le \begin{bmatrix}5\\5\\1\\1\end{bmatrix} \right\} \quad (14)$$

$$\mathsf{Y} = \left\{ y \in \mathbb{R}^{n_y} : \begin{bmatrix}0\\0\end{bmatrix} \le y \le \begin{bmatrix}10\\10\end{bmatrix} \right\}$$

as the scheduling variables constraint set can be derived from there, on an abuse of notation, $\mathsf{P} = h(\mathsf{X}, \mathsf{U}) \subseteq \mathbb{R}^{n_\rho}$.

Now, on the control system scheme, Algorithm 1 is employed. All states are considered to be available, by measurement or estimation. Initialization for P^l is taken as a constant vector, $P^0 = \rho[k] \otimes \vec{1}$, and convergence condition is a limit on the number of iterations, $l \le n_{iter}$, and a lower bound on the variation of P, $\left\| \left(P^l - P^{l-1}\right)^T \left(P^l - P^{l-1}\right) \right\|_\infty \le \varepsilon$.

Cost function J_k is the same from Eq. (5), where $\mathsf{Q} = 10I$ and $\mathsf{R} = I$, both of appropriate dimensions, meaning it is 10 times more important to track reference than to have constant input signals. Also, the prediction horizon is chosen to be $N_p = 20$ samples or $N_p = 2s$.

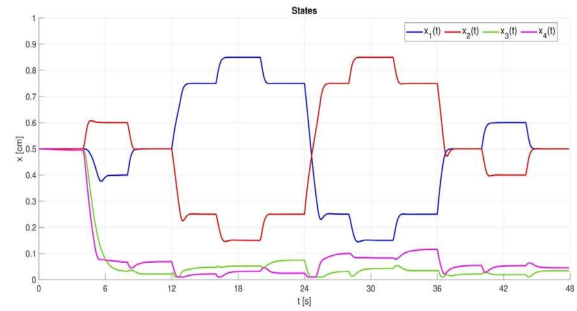

Figure 2: Cascaded Tanks: States - in blue is $x_1(t)$; in red is $x_2(t)$; in green is $x_3(t)$; in pink is $x_4(t)$;

Fig. 2 depicts, starting on h_{k_0}, all four state's trajectories: in order, on blue, red, green and pink. On Fig. 3, upper graph shows first output $y_1(t)$ and its reference signal $r_1(t)$ and lower graph, second output $y_2(t)$ and its reference signal $r_2(t)$. It is evident, by the first 12 seconds, this approach can maintain the process on stationary regime, as the pair $\left(h_{k_0}, u_{k_0}\right)$ is an equilibrium point. Likewise, as reference changes its value, both outputs can achieve a zero-error steady-state after just a few seconds. Nor first, nor second outputs do it with oscillations, although they present small overshoots when descending. Assessing disturbance rejection, every 12 seconds, starting on $t = 4s$, it is applied 4 seconds long step-like signals with 10% of max level amplitude as additive disturbances, and MPC takes just a few seconds to completely reject them, implying a zero-error steady-state.

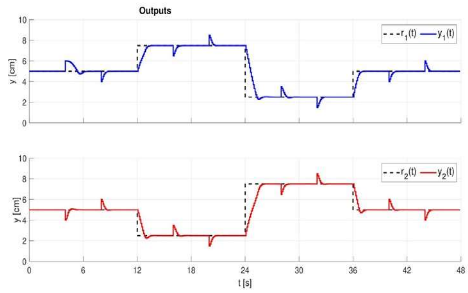

Figure 3: Cascaded Tanks: Outputs - (a) in black dashed is $r_1(t)$; in blue is $y_1(t)$; (b) in black dashed is $r_2(t)$; in red is $y_2(t)$;

Fig. 4 exhibits all four input signals, both pump voltages and valves positions. It is clear that when it is necessary, such as during reference changes, there are significant inputs variations.

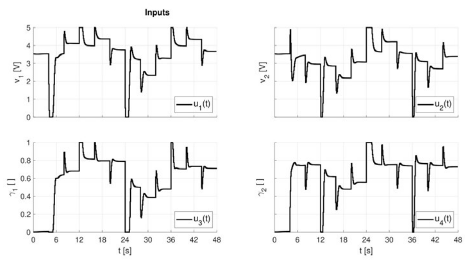

Figure 4: Cascaded Tanks: Inputs

As a way to quantify how well SQP qLPV MPC works, three indexes are proposed: integral squared error (ISE), integral average control variation (IACV) and mean value of cost function J_k. The first is an already established performance index and comprises of $ISE = \frac{1}{k_\infty - k_0}\int_{k_0}^{k_\infty}(r(\tau)-y(\tau))^2 d\tau$. The second is defined by $IACV = \frac{1}{k_\infty - k_0}\int_{k_0}^{k_\infty}|u(\tau)|d\tau$. The third is calculated by Eq. (5) divided by the number of samples. All indexes are presented in Table 2.

Table 2: Performance Indexes

$ISE(\times 10^{-3})[]$	$[3.6 \quad 3.9]^T$
$IACV(\times 10^{-3})[]$	$[14.9 \quad 17.1 \quad 14.7 \quad 18.0]^T$
$J_k(\times 10^{-3})[]$	6.2
$t_{comp}[s]$	0.3072

5. CONCLUSION

This paper offered validation through numerical simulation of an MPC algorithm for the nonlinear benchmark process Cascaded Tanks. Combining the qLPV framework and a SQP approach, the method showed satisfactory results for the tracking and rejection problems. And it does so because predictions for the scheduling variables converge to its nonlinear dynamics after a few iterations of QP optimization.

ACKNOWLEDGEMENTS

The authors thank CNPq for supporting project 304032/2019-0 and Serbian Ministry of Education, Science and Technological Development (No. 451-03-9/2021-14/200108).

REFERENCES

[1] E.F.Camacho, and C. Bordons, “Model Predictive Control”. Springer Science & Business Media, (2013)

[2] M. Alamir, “A Pragmatic Story of Model Predictive Control: Self Contained Algorithms and Case-studies”. CreateSpace Independent Publishing Platform, (2013)

[3] S. Gros, M. Zanon, R. Quirynen, A. Bemporad, and M. Diehl, “From linear to nonlinear MPC: bridging the gap via the real-time iteration”. International Journal of Control, 93(1), pp. 62–80. (2020)

[4] K.M.M. Rathai, “Synthesis and real-time implementation of parameterized NMPC schemes for automotive semi-active suspension systems”. Thèses, Université Grenoble Alpes.(2020).

[5] R. Quirynen, M. Vukov, M. Zanon, and M. Diehl, “Autogenerating microsecond solvers for nonlinear MPC: A tutorial using ACADO integrators”. Optimal Control Applications and Methods, 36(5), pp. 685–704. (2015)

[6] P.S.G. Cisneros, S. Voss, and H. Werner, “Efficient nonlinear model predictive control via quasi-LPV representation”, In 2016 IEEE 55th Conference on Decision and Control (CDC), pp. 3216–3221, (2016)

[7] M.M. Morato, J.E. Normey-Rico, and O. Sename, “Model predictive control design for linear parameter varying systems: A survey”. Annual Reviews in Control, 49, pp. 64–80. (2020)

[8] P.S. Gonzalez Cisneros, and H. Werner, “Nonlinear model predictive control for models in quasi-linear parameter varying form”. International Journal of Robust and Nonlinear Control, 30(10), pp. 3945–395, (2020)

[9] K.H. Johansson, “The quadruple-tank process: A multivariable laboratory process with an adjustable zero”. IEEE Transactions on control systems technology, 8(3), 456–465. (2000)

[10] J.S. Shamma, “An Overview of LPV Systems”, pp. 3–26. Springer US, Boston, MA. (2012)