ADCONIP 2014 Hiroshima

Reference Trajectory Based Tuning Strategy for Model Predictive Controllers

Yamashita A.S., Odloak D.

***Abstract*— A time-domain based tuning technique for Model Predictive Controllers is presented. Tuning goals are specified as the sum of the square errors between closed-loop outputs and user-defined reference trajectories, and a framework is developed to guide the user through the tuning procedure, which requires process output priority assignment and input-output pairing. The tuning problem is solved in a stepwise fashion, and constraints are included in subsequent steps in order to enforce that high priority output performance will not be disrupted. The tuning strategy was applied to a DMC controller, in a 2x2 C3/C4 splitter process. Optimum tuning parameters were compared to a set of parameters obtained by trial and error, and results show that the technique is able to achieve close tracking and disturbance rejection performance, as long as the tuning problem is well defined.**

I. Introduction

Model Predictive Control (MPC) was introduced back in 1960’s. The new technology was attractive from the process and control point of view, because it allowed for direct incorporation of process constraints into the control optimization problem; moreover, multiple-input multiple-output (MIMO) processes were treated as straightforwardly as single-input single-output (SISO) processes by simple mathematical manipulation. MPC algorithms vary broadly in the model formulation and stability and robustness properties; however, the underlying framework is a constrained optimization problem. A cost function comprised of two weighted terms: (i) the sum of squared differences between the process’ predicted outputs over a prediction horizon (calculated using the current output value and process model) and a reference output value; and (ii) a term considering the sum of input variables increments over the control horizon. Problem constraints take into account physical limitations of process variables and input variable increments. The optimization problem is solved to obtain optimum input increments, at each sampling time. The first entry of the input increment vector is fed to the process, and at the next sampling time, the process is repeated. For more information on MPC, the reader is referred to [1].

Industry has widely accepted Dynamic Matrix Control (DMC), formalized by [2], as a usual control strategy because it has the advantages of MPC mentioned earlier, and it is based on step response or impulse response models, which are more intuitive than state-space models. Nonetheless, commissioning a DMC poses challenges to the control engineer, especially regarding controller tuning. The parameters that affect robustness, stability and performance in DMCs are: prediction horizon (p), control horizon (m), model horizon (N), sampling time (Ts) and cost function weighting matrices on differences between predicted outputs and reference values (Q_y) and input increments (R). The tuning technique presented in this paper will allow tuning of the weighting matrices, since sampling time is not generally available for free choice, model horizon should be chosen large enough as to ensure that the process dynamics is considered throughout all the required range, and prediction and control horizon are chosen following quantitative guidelines from literature [3], yielding good results.

Controller tuning guidelines are usually classified by tuning goal (tuning for performance, robustness or stability, however, some strategies associate multiple objectives in the objective function, or as constraints) or according to the domain in which tuning objectives are specified, i.e. time-domain or frequency domain. [4] developed a frequency-domain tuning strategy based on minimization of the *H-infinity* norm of a mixed sensitivity function in which the main tuning goal is to ensure proper disturbance rejection in DMC controllers. [5] and [6] put forward tuning techniques based on the conditioning number of the DMC system matrix, which is directly related to its ill-conditioning. The former developed qualitative guidelines regarding the choice of matrix R, so that proper conditioning of the system matrix is achieved. The latter said that a conditioning number of 500, is a good compromise between stability and performance for open-loop stable systems, and used a first order plus dead time approximation of the system dynamics to derivate analytical expression for optimum R values. [7] formulated a tuning strategy in which tuning goals are given in terms of time-domain performance specifications like overshoot, rise time, and settling time. The authors used a min-max objective function formulation to account for performance and robustness goals, and the tuning problem was solved by Particle Swarm Optimization, which is a heuristic search based on evolutionary algorithms. It simulates bird flocks behavior moving from one place to another; individual birds portrait possible

Yamashita A.S. and Odloak D. are with the Chemical Engineering department of Universidade de São Paulo, Av. Prof. Luciano Gualberto, trv.3, 380, 61548000, São Paulo, Brazil
E-mail: {andre.yamashita, odloak}@usp.br

solutions and the final direction of the flock is sought as an optimal solution.

The time-domain based tuning technique developed here considers time-domain performance objectives to find optimum values of Q_y and R for DMC controllers; however, it can be applied to virtually any parameter-based controller formulation. This paper is comprised of the following sections: first, we provide the DMC formulation used in tuning and simulations examples. Second, we explain the tuning technique for the nominal case in more detail. Later, the technique is applied to a two-input two-output model of C3/C4 industrial splitter. The paper closes with conclusions and remarks for future works.

II. Methodology

A. DMC controller

The DMC used in the paper is similar to [2]. Equations 1 to 3 show the prediction model, and how the free response vector is calculated. h_i^{kj} is the value of output j at time instant i for an unitary impulse in input k, j=1,…,ny; k=1,…,nu; i=1,…,N. It is used to calculate the dynamic matrix, D_m, in which $G_{i,j}$, i=1,…,p; j=1,…,nu are submatrices with coefficients $g_{i,k}$ of the i-th step response corresponding to the j-th input. The impulse response is obtained from a transfer function, or equivalent, process representation. The free response vector, f, is calculated according to (4), and (1) shows the prediction of process outputs over the prediction horizon. ny is the number of process outputs and nu is the number of process inputs. $\Delta u(k)$ is a vector of dimension 1 × nu, and represents an increment in process inputs, $y(k)$ is the value of process outputs at time instant k, $y^c(k+i)$, i=1,…,p is is a vector of dimension 1 × ny and represents the predicted outputs value at time instant $k+i$. $y_m(k)$ is the vector of measured outputs at time instant k.

$$\begin{bmatrix} y^c(k+1) \\ y^c(k+2) \\ \vdots \\ y^c(k+p) \end{bmatrix} = \begin{bmatrix} G_{1,1} & 0 & \cdots & 0 \\ G_{2,1} & G_{12} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ G_{p,1} & G_{p-1,2} & \cdots & G_{p-m+1,nu} \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+m-1) \end{bmatrix} + \begin{bmatrix} f(k+1) \\ f(k+2) \\ \vdots \\ f(k+p) \end{bmatrix} \tag{1}$$

$$y_j(k) = \sum_{k=1}^{nu} \sum_{i=1}^{N} h_i^{kj} u^k(k-i) \tag{2}$$

$$g_{i,k} = \sum_{j=1}^{i} h_i^{kj} \tag{3}$$

$$f(k+i) = y_m(k) + \sum_{j=1}^{N} (g_{j+1} - g_j) \Delta u(k-j) \tag{4}$$

To improve notation, (1) is be written in a compact form, seen in (5).

$$y^c = D_m \Delta u_k + f \tag{5}$$

where y^c is the vector of predicted outputs over the prediction horizon, p; Δu_k is the vector of control efforts over the control horizon, m; and f is the vector of stacked free responses $f(k+i)$, i=1,…,p.

Assuming a constant output setpoint, $y_{sp}(k)$, the set point vector is defined as $y^{sp}=[y_{sp}(k)\ldots y_{sp}(k)]$. Finally, the objective function of the DMC controller is written according to (6), and the DMC control problem, consists in minimizing (6), subject to constraints defined in (7) and (8).

$$\min_{\Delta u_k} \sum_{i=0}^{p} \left\| y^c(k+i|k) - y_{sp}(k+i) \right\|_{Q_y}^2 + \sum_{i=0}^{m-1} \left\| \Delta u(k+i|k) \right\|_R^2 \tag{6}$$

subject to

$$u_{\min} \le u(k+j) \le u_{\max},\ j = 0,\ldots,m-1 \tag{7}$$

$$-\Delta u_{\max} \le \Delta u(k+j) \le \Delta u_{\max},\ j = 0,\ldots,m-1 \tag{8}$$

u_{min} and u_{max} are the lower and upper bound of the inputs and Δu_{max} is the maximum control effort. Q_y, Q_y>0 and R, $R \geq 0$ are diagonal matrices. The problem defined by (6), (7) and (8) is efficiently solved by Quadratic Programming (QP) algorithms, however, during the tuning procedure, we will solve it analytically by disregarding constraints, as done in [8]. During simulations though, process constraints will be taken into account.

B. Nominal tuning strategy

We assume that controller and plant have the same model, i.e. model mismatch in not an issue. The tuning technique is based on minimizing the sum of squared errors (SSE) between closed-loop output trajectories and user-defined reference trajectories that take into account process characteristics like overshoot, rise time, and settling time. Similar strategies have been used in literature, for example, [9] used output envelopes that constrain process output behavior, [10] used reference trajectories for both decoupling in MIMO systems and time domain performance goals. Also, it was stated by several authors that tuning objective functions based solely on the integral of squared or absolute error of process outputs and a setpoints are faulty ([11], [12]). The method also requires that the user assigns priorities to process outputs, and that input-output pairs are established prior to solving the optimization. In this fashion, a successive tuning technique is developed, in which outputs considered more important are driven closer to their reference trajectories during the tuning procedure, and previously attained optimum values are maintained or improved in subsequent steps, while performance of low priority outputs is considered only if there is no significant sacrifice of high priority output performance.

The technique focuses on tuning weighting matrices Q_y and R. The integer variables N, p, and m will not be included because it would lead to a mixed-integer nonlinear optimization problem, which might prove difficult to solve. Moreover, the effect of p and m in the closed loop performance is indirect; high values of p and m might render the control problem computationally infeasible, and there are several references in literature that provide reliable guidelines for choosing these parameters [3].

N is chosen according to the highest settling time of the open-loop step response of the process, in order to guarantee that the step (impulse) response model has enough information on system dynamics. p is chosen as a fraction of N, 60% represents good compromise between information availability and computational load. Finally, m is usually set to a value within 3-6. Higher values of m mean more stability at the cost of higher computational load.

In open-loop unstable processes, and in the case in which the controller internal model and the plant model are different (plant-model mismatch), it is recommended to set slightly higher p and m to try to achieve more stable control, however there are no formal stability and robustness guarantees for nominal controllers. For tuning purposes, in SISO systems, it suffices to set Q_y to a fixed value and vary R to obtain an optimum control performance, because the ratio of Q_y to R unequivocally determines a control profile, even though the total control cost might not be unique. In MIMO systems, it suffices to make one entry of Q_y constant and tune the remaining coefficients the weighting matrices.

The remainder of this work uses the following notation: a system of ny outputs and nu inputs can be represented by a matrix of transfer function $G_s(s)$, the entries of the transfer function matrix are represented by $G_{i,j}(s)$, $i=1,\dots,ny$, $j=1,\dots,nu$, and s is the Laplace variable. $q_{y,i}$ and r_j, $i=1,\dots,ny$, $j=1,\dots,nu$ are diagonal entries of matrices Q_y and R.

1) *Output priority assignment*

Determining important variables among a set of different variables is a common scenario for process engineers. Important outputs are usually chosen based on economic, environmental, or safety factors. This tuning algorithm assumes a user-defined sequence of prioritized outputs.

2) *Specifying output-input pairs*

The framework behind PID control supports successive tuning of square subsystems with increasing dimensions up to the size of the original system. Most control loops in real plants are comprised of pairs of output-input variables. Since priority of outputs has already been defined, the second step is to choose a suitable input for each output according to its priority. [13] reviewed a couple of methods to select these pairs.

There are cases in which process knowledge dictates output-input pairs, however, when such information is unavailable or insufficient, pairing techniques like Relative Gain Array (RGA) or Singular Value Analysis (SVA) are employed. Observe that there may be processes in which $ny \neq nu$, and if this circumstance arises, the user is oriented to either: (i) assign more than one input to high priority outputs if $nu>ny$, or (ii) clump low priority outputs to a single input if $ny>nu$. Hereafter, subscripts i and j in y and u, as well as in q_y and r will refer to the variable order defined here.

3) *Preparing the remaining parameters*

The algorithm uses normalized transfer function, output set points, input and outputs initial points to minimize numerical problems during the tuning procedure, and to attenuate output scaling problems. It also allows for Q_y to solely correspond to the relative importance of process outputs.

4) *Specifying tuning objectives*

The tuning objectives are output reference trajectories that allow the user to set time-domain performance goals such as overshoot, rise time, settling time, and dynamic behavior. The SSE between reference trajectories and closed-loop output trajectories is the tuning objective function. Reference trajectories might be chosen according to different criteria. One example is to take the open-loop transfer function related to an output, approximate it by a first-order plus dead time transfer function and set its time constant to a fraction of the original transfer function. This strategy has its merits because control engineers are fond of stable, swift, and non-oscillatory responses. Observe that reference trajectory goals are applicable to both disturbance rejection and output tracking tuning scenarios.

Reference trajectories are given by a diagonal transfer function matrix, $G_{des}(s)$, comprised of $G_{des,i}(s)$ $i=1,\dots,ny$ terms.

5) Sequential tuning algorithm

Even though the tuning sequence follows the input-output order defined earlier, the first tuning step requires some particular procedures, and for better understanding, the tuning method will be explained in two parts. Initially, we define the tuning cost function and explain the first tuning step; later, we cover the remaining tuning steps, up to the last one.

Let us first define a function to evaluate the SSE between an output's reference trajectory and its closed-loop response:

$$J_i^\eta = \sum_{j=1}^{\theta_t}\left(y(j)_i^{ref} - y(j)_i\right)^2 \quad i=1,\dots,\eta \tag{9}$$

$$V^\eta = \sum_{j=1}^{ny} J_j^\eta \tag{10}$$

where θ_t is the tuning horizon, $y(j)_i^{ref}$ is the discretized reference trajectory of output i, $y(j)_i$ is the closed-loop trajectory of output i, $j=1,\dots,\theta_t$, and η is the current tuning step. Observe that $y(j)_i$, is evaluated using control moves calculated by an MPC in closed-loop and therefore, it is a function of the tuning parameters.

a) First step

The first step of the tuning problem is defined as follows:

$$\min_x V^1 \tag{11}$$

subject to

$$LB \le x \le UB \tag{12}$$

where x, $x \in \Re$ is the decision variable *vector.* In the first step, the vector x only takes a single element, r_1, since $q_{y,1}$ is by definition constant to avoid conditioning problems.

b) Successive steps

The method tries to improve or at least to preserve the performance of high priority outputs every time a lower priority output and its respective input pair are added to the tuned process subsystem. Usually, extra inputs provide extra degrees of freedom for better output performance; however, some systems might not allow for further improvement due to coupling properties. In order to avoid infeasibilities, performance constraints are relaxed with slack variables, heavily weighted in the tuning cost function. The tuning problem, at tuning step η, for an η x η subsystem is:

$$\min_{x,\delta} V^\eta + \delta^T S_t \delta \tag{13}$$

subject to

$$J_i^\eta - J_{old}^*(i) - \delta(i) < 0,\ i=1,\dots,\eta-1 \tag{14}$$

$$\delta \ge 0 \tag{15}$$

and (12). δ is a vector of slack variables $\delta(i)$, $i=1,\dots,(\eta-1)$ from (14); S_t is a diagonal positive definite weighting matrix, $S_t \in \Re^{(\eta-1)\times(\eta-1)}$. Considering that $q_{y,1}$ is constant, the decision variables vector at tuning step η is defined as: $x=\begin{bmatrix} r_1 & \dots & r_\eta & q_{y,2} & \dots & q_{y,\eta}\end{bmatrix}$, $x \in \Re^{(2\eta-1)}$. J^*_{old}, $J^*_{old} \in \Re^{(\eta-1)}$ is a vector containing the optimum values of (11) or (13) obtained in tuning steps, η-1, η-2,...,1.

$$J^*_{old} = \begin{bmatrix} J_1^{1*} \\ \vdots \\ J_{\eta-1}^{\eta-1*}\end{bmatrix} \tag{16}$$

III. Results and Discussion

A 2x2 C3/C4 splitter model was considered to illustrate an application of the proposed tuning method. The nominal plant model was identified from operational data, within the range of process inputs. Table 1 shows a list of process variables, ranges, and unities. For more information about the process, the reader is referred to [14].

Input-output pairs and output priority were defined from process information obtained in [14]. Non-tunable parameters were chosen as follows: N=100, p=60, m=5, the tuning method variables were chosen as θ_t = 60, the pair y_1-u_2 is considered more important than y_2-u_1. G_{des} was chosen based on first-order approximations of the open-loop step responses of transfer functions $G_{2,1}(s)$ and $G_{1,2}(s)$ of the nominal model. The time constant of the reference trajectory was set to 10% of the original value for $G_{des1,1}(s)$ and to 30% for $G_{des2,2}(s)$ as in (17). Process initial conditions and output setpoints are: y_0=[0.825 0.82]', u_0=[0.60 0.70]', y_{sp}=[0.45 0.50]', considering normalized values. The tuning problem initial guess in the first and second steps are $x_{0,1}$=[0 1 8], $x_{0,2}$=[0 1e-2 10 8 4 1]. Lower and upper bounds are LB_1=[0 1e-3 8], UB_1=[∞ 1e3 8] and LB_2=[0 1e-3 1e-3 8 0.5 0], UB_2=[∞ 1e3 1e3 8 6 1e6]. S_t=1e4.

The tuning problem is solved using *fmincon* (trust-region reflective algorithm, 1e-12 function tolerance, 1e-8 x tolerance, 4000 max function evaluations and 1e-6 constraint tolerance), and the unconstrained version of the control problem is solved analytically, in MATLAB® (version R2013a).

$$G_{des}(s) = \begin{bmatrix} \frac{1}{2s+1} & 0 \\ 0 & \frac{1}{6.6s+1} \end{bmatrix} \quad (17)$$

Figures 1 and 2 show results obtained by the tuning method after the last step, the outputs are given in engineering units. Table 2 shows values of Q_y and R obtained by the tuning technique developed here and by [14] using a trial and error approach.

The total time elapsed during steps one and two was 72 seconds, using an Intel® Core™ i5 3.20 GHz, 4Gb RAM. The low value of δ (0.001), when compared to the second step cost function value (0.1067), shows that the tuning strategy allowed for a minor decrease in y_1 performance index (illustrated in figure 1), while simultaneously adding y_2's goal to the tuning problem.

Since the upper and lower bounds for Q_y were chosen arbitrarily, there is a sufficient low value for $q_{y,2}$ so that it's deviation from the setpoint is neglected in the control cost function. In this situation, y_1 would achieve its peak performance; however, y_2 would not be controlled, which is not desirable. Figure 2 shows that performance of y_2 is already worse than y_1's, but both outputs would converge to their setpoints in longer simulations. Observe that the values of R obtained by the tuning technique were around 10 times higher than the optimum trial and error values from [14], since the desired trajectories demanded smooth responses.

A simulation study is carried on to compare results. The initial state of inputs and outputs is y_0=[1.1 50] and u_0=[3000 1700]. Upper and lower bounds of inputs and outputs are taken from table 1, and Δu_{max}=[50 25]. The simulation ran for 550 time instants, setpoint changes are listed in table 3. From time instants 200 to 205 and 400 to 405, unmeasured disturbances affected process inputs. The DMC control problem was solved using *quadprog* (default settings), in MATLAB® (version R2013a).

TABLE I. C3/C4 SPLITTER VARIABLES, RANGES, AND UNITIES.

Tag	Variable	Range	Unity
y_1	C3 % in butane stream	0.80 - 1.20	%
y_2	Top temperature	43 - 54	°C
u_1	Reflux flowrate	2000 - 4100	m^3/h
u_2	Hot fluid flowrate	1200 - 2200	m^3/h

TABLE 2: TUNING PARAMETERS OBTAINED BY TWO TUNING TECHNIQUES.

	Q_y		R	
	y_1	y_2	u_1	u_2
Final parameters	8	1	11.327	29.079
[14] parameters	1.3	1.2	1.5	1.5

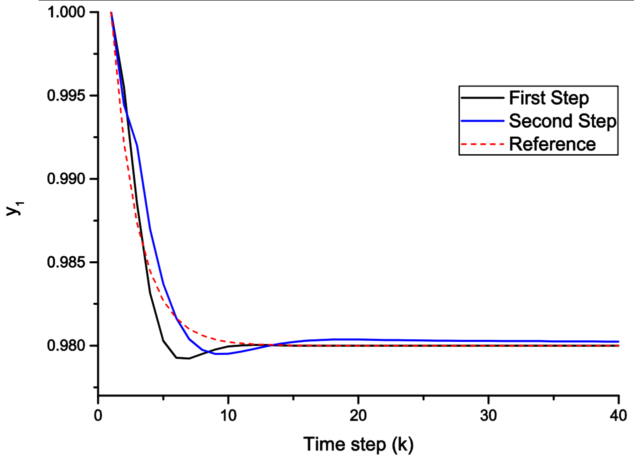

Figure 1: Comparing performance of y_1 after the first and second tuning steps.

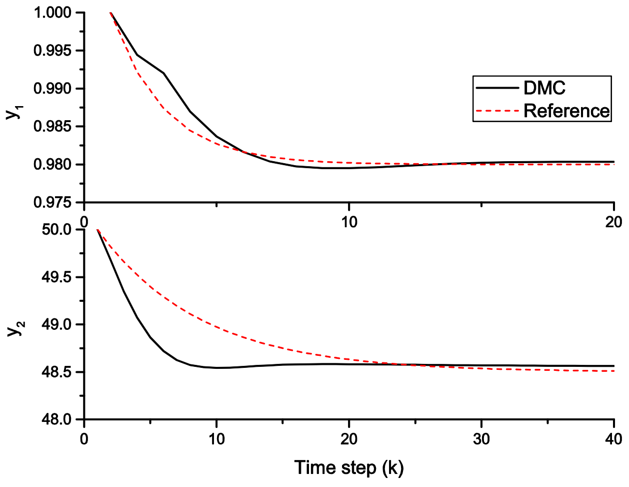

Figure 2: Tuning results, process outputs.

Figures 3 and 4 shows the process outputs and inputs, calculated using the tuning parameters obtained considering the reference trajectory approach (full lines), trial and error (dotted lines) and setpoint values (dashed lines). In figure 3, performances obtained by both tuning strategies are similar for output y_1, except in disturbance rejection scenarios, where the reference trajectory strategy outperforms the practical trial and error-tuned controller.

In output y_2, the reference trajectory-tuned controller yields more sluggish responses and larger offset. The priority difference of y_1 and y_2, given by weights Q_y, for the tuning strategy developed here (8 times) and from the trial and error tuning parameters (1.08 times) justify the observed results. Inputs behavior is very similar in both cases, as seen in the undistinguishable curves in figure 4, despite the differences in R seem in table 2, which are not great enough to lead to substantial discrepancies. In terms of Sum of Square Errors, considering the tracking error from closed-loop outputs and set-points in Figure 3, the trial-and-error tuned controller is only 2% better.

TABLE 3: SIMULATION SETPOINTS.

Time instant	$y_{1,sp}$	$y_{2,sp}$
0	0.99	48
100	0.85	49
300	0.9	51
500	0.95	50

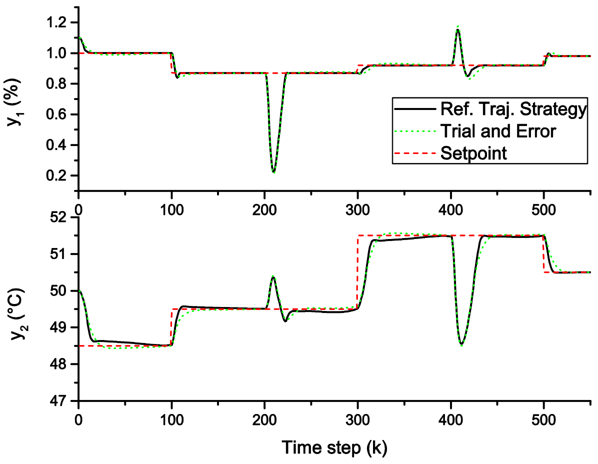

Figure 3: Process outputs of the closed loop with optimum tuning parameters obtained with the proposed tuning technique and by plant trial and error.

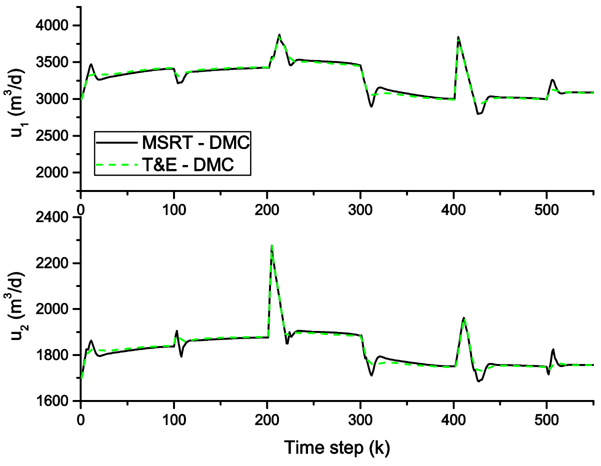

Figure 4: Process inputs of the closed loop with optimum tuning parameters obtained with the proposed tuning technique and by plant trial and error.

IV. CONCLUSION

A tuning technique for MPC, based on time-domain reference trajectory goals, was developed and tested on a 2x2 C3/C4 splitter process. User-defined parameters of the tuning procedure, such as the pairing matrix, output priorities and reference trajectories, are crucial to obtain good results. In fact, results obtained using the technique are comparable to a trial and error-tuned controller, with the advantages of saving time through the solution of a well-structured optimization problem, and higher levels of automation. The authors are working on an extension of the technique to account for polytopic model uncertainty. Our future goal is to develop a robust tuning strategy to produce a nominal controller with 'robust tuning parameters', in order to replace the need of robust controllers in industry, which are intrinsically more computationally demanding.

ACKNOWLEDGMENT

The authors would like to thank CNPq by financial support under grant 140677/2011-9.

REFERENCES

[1] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Eng. Pract.*, vol. 11, no. 7, pp. 733–764, Jul. 2003.

[2] C. R. Cutler and B. L. Ramaker, "Dynamic matrix control—a computer control algorithm," in *Joint Automatic Control Conference*, 1980.

[3] J. L. Garriga and M. Soroush, "Model Predictive Control Tuning Methods: A Review," *Ind. Eng. Chem. Res.*, vol. 49, no. 8, pp. 3505–3515, Apr. 2010.

[4] M. Francisco and P. Vega, "Automatic tuning of model predictive controllers based on multiobjective optimization" *Lat. Am. Appl. Res.*, vol. 40, pp. 255–265, 2010.

[5] J. L. Marchetti, D. A. Mellichamp, and D. E. Seborg, "Predictive control based on discrete convolution models," *Ind. Eng. Chem. Process Des. Dev.*, vol. 22, no. 3, pp. 488–495, Jul. 1983.

[6] R. Shridhar and D. J. Cooper, "A Tuning Strategy for Unconstrained SISO Model Predictive Control," *Ind. Eng. Chem. Res.*, vol. 36, pp. 729–746, 1997.

[7] K. Han, J. Zhao, and J. Qian, "A Novel Robust Tuning Strategy for Model Predictive Control," *2006 6th World Congr. Intell. Control Autom.*, pp. 6406–6410, 2006.

[8] W. Wojsznis, J. Gudaz, T. Blevins, and A. Mehta, "Practical approach to tuning MPC," *ISA Trans.*, vol. 42, no. 1, pp. 149–162, Jan. 2003.

[9] A. Al-Ghazzawi, E. Ali, A. Nouh, and E. Zafiriou, "On-line tuning strategy for model predictive controllers," *J. Process Control*, vol. 11, no. 3, pp. 265–284, Jun. 2001.

[10] V. Exadaktylos and C. J. Taylor, "Multi-objective performance optimisation for model predictive control by goal attainment," *Int. J. Control*, vol. 83, no. 7, pp. 1374–1386, Jul. 2010.

[11] K. Yamuna Rani and H. Unbehauen, "Study of Predictive Controller Tuning Methods," *Automatica*, vol. 33, no. 12, pp. 2243–2248, 1997.

[12] L. L. Giovanini and J. L. Marchetti, "Shaping Time-Domain Responses with Discrete Controllers," *Ind. Eng. Chem. Res.*, vol. 38, no. 12, pp. 4777–4789, Dec. 1999.

[13] M. van de Wal and B. de Jager, "A review of methods for input/output selection," *Automatica*, vol. 37, no. 4, pp. 487–510, Apr. 2001.

[14] C. R. Porfirio, E. Almeida Neto, and D. Odloak, "Multi-model predictive control of an industrial C3/C4 splitter," *Control Eng. Pract.*, vol. 11, no. 7, pp. 765–779, Jul. 2003.