Preprints of the 8th IFAC Symposium on Advanced Control of Chemical Processes
The International Federation of Automatic Control
Furama Riverfront, Singapore, July 10-13, 2012

Spline Dynamic Matrix: a Novel Representation of Dynamic Models

Matheus S. Escobar and Jorge O. Trierweiler

GIMSCOP – Group of Intensification, Modeling, Simulation, Control and Optimization of Process
Department of Chemical Engineering, Federal University of Rio Grande do Sul, Porto Alegre, CO 90040-040
Brazil (Tel: 5551-3308-3444; e-mail:{ mescobar, jorge} @ enq.ufrgs.br).

Abstract: In the field of dynamic simulation, Finite Impulse Response (FIR), Step Response (SR) and discrete state-space (DSS) approaches are the most widespread model representations developed so far. Such representations, however, have some disadvantages, regarding simulation speed, computational load and fixed sampling. To cope with these restrictions, this work introduces a new model representation called Spline Dynamic Matrix (SDM), which combines the dynamic matrix model with the inputs parameterized by piecewise cubic splines. Such representation results in reduced matrixes, allowing variable spacing over simulation horizon. When compared with DSS representation, SDM behaves remarkably well, with mild deviations inherent to spline interpolation, keeping model adherence. SDM also presents itself as an effective way to implement model predictive controllers (MPC), being robust when compared to traditional formulation.

Keywords: Splines, Dynamic Simulation, Process Simulation.

1. INTRODUCTION

An understanding of the dynamic behaviour of chemical processes is important from both process design and process system perspectives. Such an understanding allows a process engineer to predict behaviours, analyze trends and act preventively towards plants instabilities, assuring safety and efficiency, when it comes to plant operation, embracing also quality and reliable production. Throughout history, several methodologies were developed to, through mathematical modelling, simulate processes behaviour over time. Among several existent developments, Step Response (SR) (Cutler and Ramaker, 1980) and Discrete State Space (DSS) approaches are the most widespread ones currently (Maciejowski, 2002).

Certain concerns, however, regarding mainly computational load, simulation speed and model adherence should be taken into account. Even though these approaches simulate rather well several systems, they are fairly time consuming, where the input values at each time instant are the problem variables, being such computational load severe, depending on the model complexity.

What would happen, however, if merely a few points in space could gather all the dynamic information and use it to represent efficiently all input over time? Such proposition is presented in this paper by introducing the Spline Dynamic Matrix (SDM) concept. By representing the input along a definite horizon by a few dots in space, splines, *i.e.*, piecewise polynomial functions, can interpolate such data, in order to produce analytical expressions for inputs over an extensive horizon. By combining this methodology (Boor, 2001) with the state space representation on the continuous domain (Ogata, 2003), a dynamic matrix that is dependent on a significantly smaller amount of variables presents itself as a novel possibility.

Initially, a brief review of nowadays dynamic simulation modelling tools is presented, followed by a presentation of splines concept. The SDM approach is then presented, as a combination of both methodologies, culminating in an application for predictive controllers, the Spline Dynamic Matrix Controller (SDMC). Finally, some results are shown, in order to validate the implementation, followed by final remarks on the topic.

2. DYNAMIC SIMULATION OF LINEAR MODELS

2.1 Discrete State Space (DSS) Models

The SR model concept, based on a dynamic matrix (DM) gathering all the relevant dynamic data (Cutler and Ramaker, 1980) was developed at the early 80's, leading to other development, such as the Dynamic Matrix Controller (DMC). On a similar basis, but based on identified models or phenomenological sets of equations, DSS modelling was developed. It is derived from a set of model equations F: $\mathbf{R}^{n+s} \rightarrow \mathbf{R}^{m}$, representing m equations with n inputs and s states, as it can be seen in (1) and (2).

$$\dot{x} = Ax(t) + Bu(t) \quad (1)$$

$$y(t) = Cx(t) \quad (2)$$

Where x is a state vector, u represents the input vector and y, the desired outputs. A_c and B_c are the state matrixes. Once this equation is resolved for a continuous domain, from a certain time instant t_0 to t, the state-space representation over time can be represented by (3) (Ogata, 2003).

$$x(t) = e^{A(t-t_0)}x_0 + \int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau \quad (3)$$

This representation, however, is not equivalent to the DSS, because, evidently, it is a representation for continuous domain. From this premise, some transformations should be done, so that the discrete equivalent state-space matrixes (A, B, C and D) can be determined, according to (4) and (5) (Åström and Wittenmark, 1997).

$$A_d = e^{At_s} \tag{4}$$

$$B_d = (e^{At_s} - I)A^{-1}B \tag{5}$$

Where t_s is the sampling time and I, identity matrix. $C_d = C$ and $D_d = D$. d stands for discrete. Thus, the discrete state-space representation can be expressed by (6) and (7):

$$\Delta x_{k+1} = A_d \Delta x_k + B_d \Delta u_k \tag{6}$$

$$\Delta y_{k+1} = C_d \Delta x_{k+1} \tag{7}$$

Applying these equations over a simulation horizon, a recurrent relation appears, where earlier states variations have an impact on the system future response (Camacho and Bordons, 2004). From this relation, a dynamic matrix can be obtained, S_u, for D_d=0. Further details can be seen in (Maciejowski, 2002). The final expression for output calculation with null initial conditions is given by (8) :

$$Y_{\begin{bmatrix} k+1 \\ k+P \end{bmatrix}} = Su\Delta U_{\begin{bmatrix} k+1 \\ k+M \end{bmatrix}} \tag{8}$$

Where M is the control horizon. The idea behind SDM is slightly different from this approach. Knowing that inputs can be represented by polynomial piecewise cubic splines, what characterizes continuous functions, discrete approach loses its purpose. From (8), the whole system can be simulated, guaranteeing the development of an equivalent SR dynamic matrix that gathers the relevant dynamic data within.

2.2 Variable Reduction Strategies

Simulating systems with a smaller number of decision variables, however, is an idea already discussed in the literature. One of the simplest techniques applied to reducing the amount of variables along a simulation horizon is called blocking. It consists in keeping the same input during more than one discretized instant, creating input blocks (Maciejowski, 2002). Its usage is wide in predictive controllers, where some features are brought along its application (Gondhalekar and Imura, 2010, Cagienard et al., 2007).

Blocking introduces robustness into the system, because there are fewer variables available for decision, impacting, however, in performance loss, since responses become slower. SDM has an analogue behaviour, where input blocks behave similarly, but on each interval a piecewise polynomial cubic function is being used, which results in an input profile instead of constant values. This allows insertion of robustness minimizing controller performance loss.

3. SPLINES

Splines are tools that through knots can represent a space as polynomial functions. Among interpolating splines, the main difference is the type of algorithm used, which impacts on the piecewise functions continuity, on the tendency for oscillatory responses and on the interpolation of all knots (Foley, 2006). For this work, Hermite splines were chosen, due to continuity up to the first derivative and guarantee of smooth functions throughout the entire generation interval. They are also monotonic and can be adjusted further by certain parameters, depending on the methodology (Biswal, 2008). Hermite splines generate cubic polynomial functions between each two subsequential knots, using the derivatives in these knots as basis for spline generation. Fig. 1 illustrates this concept.

Fig. 1. Hermite spline representation for two subsequential knots.

There are several subtypes of Hermite splines. For this paper, it was chosen the Kochanek-Bartels Hermite, also known as TCB. The derivative calculation is based on an initial idea: all the derivatives arise from the tangent of several lines composed of subsequential knots, called interval deltas. Further details can be seen in Boor (2001). A schematic representation is shown in Fig. 2.

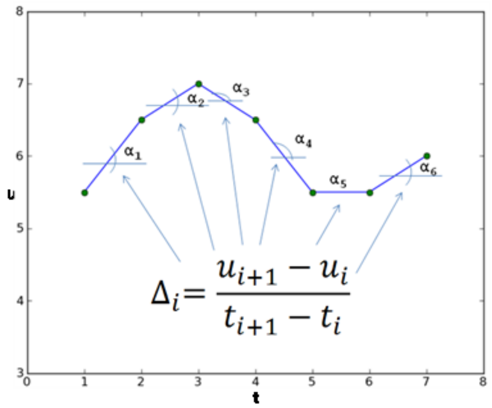

Fig. 2. Schematic representation of the interval deltas calculation.

TCB relies on the average between two deltas adjacent to a knot, combined with certain parameters that can be tuned on each knot, varying from -1 to 1. On each of these tangents, there are three parameters to be tuned. Tension (τ), which affects the tangent vector length, being responsible for the amount of oscillation permitted. Continuity (γ) affects the continuity of the functions derivative, conferring smoothness or non-smoothness to the splines generated. Bias (ω) ponders one delta over another, valuing more one trend over another.

TCB has a great flexibility when it comes to fitting an interpolated curve to the actual base function requiring representation. Furthermore, TCB does not rely on logic conditions for its conception, which allows it to be integrated in the SDM in a simpler way.

4. SPLINE DYNAMIC MATRIX – SDM

From a set of knots for each input, representing specific points in space over time, it is possible to recreate the entire input function with splines. The knots are time-fixed, which means that the only variables are, actually, the input values on each knot, not the time instants on where it will be displaced. Such arrangement must be done before the SDM generation, taking into account that knots should be placed on time instants that have significant input dynamic information.

The simulation horizon shall be discretized on P values that may, or may not be, equally spaced, in order to promptly compare its performance with other discretized strategies. From this premise, each value will have an input equivalent point given by a spline function applied on each time instant, dependent merely on the knots given.

The SDM methodology in dismembered in two parts: one regarding the input matrix representation, and another concerning output calculation and SDM composition. Further details on this methodology can be seen in (Escobar, 2012).

As for the input representation, it follows the equation presented in (9).

$$U_i(t) = T(t) \cdot Coef_i \cdot TCBMat_i \cdot H\,, \quad i = 1..N \tag{9}$$

Where N is the total interval number. T is a vector that expresses the cubic nature of the TCB spline, as can be seen in (10):

$$T(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \tag{10}$$

Coef is a square matrix 4 x 4, expressing the effect that each point in space has on each order of the piecewise cubic polynomial generated.

TCB depends on the knots position and derivatives to calculate the splines in a certain interval. Knowing that the derivatives are obtained through the knots position itself, *TCBMat* is an auxiliary matrix, which expresses the tangent assimilation in the input representation. Then, the only remaining variables are the input values on each knot, expressed by the vector H.

On every time instant U will rely only on H to be calculated, because *Coef* and *TCBMat* are constant on each interval. By applying (9) on an entire simulation horizon, and combining it with (2) and (3), a matrix with all the dynamic information and spline conception can be conceived, being dependent merely on a couple knots. Such representation can be shown by (11):

$$Y = C(X_0 + (X_U + X_{U_0} \cdot H) = C(X_0 + SDM \cdot H) \tag{11}$$

Where X_0 is a vector that expresses the inertial behaviour of a system from any initial conditions, as it can be seen in the non-integrative part of (3). X_U expresses the effect on the output arisen from input variation, in the form of an integral to be solved. X_{U0} expresses a free system response on each interval. This term would be equivalent to an accumulated inertial system response. SDM represents the sum of both of these terms. Fig. 5 illustrates the effect of each term in the final response.

From this premise, variable spacing in discretization becomes possible. Since there are analytical expressions that rule each interval, there is no need for using approximation by hold, for instance, as it happens with the DSS models, which results in equal spacing, mandatorily. It is, therefore, another clear advantage for SDM methodology. It means that for regions with small dynamic information, spacing can be wide and vice-versa, saving computational load and focusing on regions with greater dynamic variation.

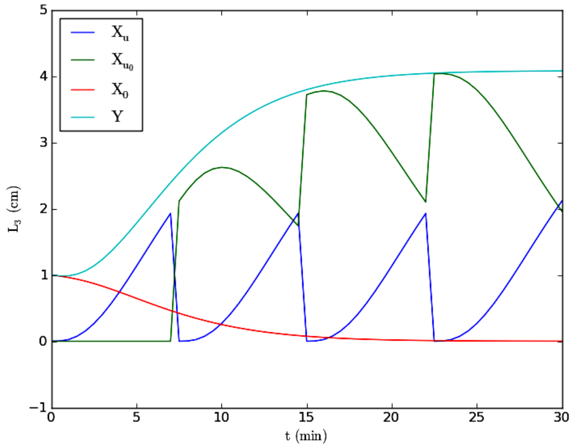

Fig. 3. X_0, X_U and X_{UO} impact when a step is applied to the input.

Alongside with this representation, it was proposed a new formulation for a predictive controller, called Spline Dynamic Matrix Controller (SDMC). The representation is structurally similar to DMC, also being expressed by a prediction horizon P and a control horizon M. For a SISO MPC, the optimization objective function is given by (12):

$$F_{OBJ} = E_y^T \Gamma E_y + E_U^T \Lambda E_U \tag{12}$$

Where E_y expresses the error between predicted and desired output values along prediction horizon, E_U expresses a penalty for input excessive variation, also called Move Supression. Finally, Γ and Λ are squared diagonal matrices P x P used for pondering. E_y is shown in (13):

$$E_y = Y_{REF} - (Y_{INER} + Y_{BIAS} + w) \tag{13}$$

Where Y_{REF} is a column vector responsible for the desired trajectory, or value, to be achieved by the MPC. Y_{INER} expresses the impact that past inputs have on the system, being obtained by the superposition of the responses generated by these inputs along P. Y_{BIAS} act as a bias for Y_{INER}, being equal to the initial stationary output. Finally, w expresses how the difference between predicted and real plant values on each sampling. E_U is conceived in a particular way, originated from SDM methodology, expressed by (14):

$$E_U = dU \cdot H \tag{14}$$

Where dU is obtained as a derivative of (9). It is important to note that, analogue to the SDM application for simulation, dU reduces the matrix size, being dependent merely on H. Therefore, both terms express quadratic sums that are dependent on H. The optimization problem is, thus, convex and quadratic programming (QP) can be used for resolution. A generic QP formulation can be seen in (15):

$$\min_H \frac{1}{2} H^T Hess H + Q^T H \tag{15}$$

S.t. to the following constraints presented in (16) and (17) :

$$EH \le d \tag{16}$$

$$GH = f \tag{17}$$

Where *Hess* is the hessian matrix and Q expresses the linear part of the optimization problem. The constraints expressed in (16) are used for input limits and those presented in (17) are applied to maintain MPC important characteristics. This problem has a dynamic trait, where optimization is performed on each sampling, changing the optimal input profile as the simulation runs.

As for parameters adjustment, there are traditional parameters, such as prediction and control horizon, sampling time and move suppression, supported by pondering of Γ and Λ, which are common of most MPCs. Other parameters, however, specific to SDM methodology, reveal themselves to have interesting features that aid the SDMC.

SDM representation itself works as a sort of blocking, strategy used for minimizing decision variables in optimization problems. Each interval has an input profile determined by external knots, diminishing the number of variables available for variation, which implies in a limited variation from one discretized point to another, increasing robustness. Other important point to be discussed refers to displacement of knots along *P*. Basically, knots should be placed in regions with inflexions or major variation on short time periods. An erroneous displacement can jeopardize the simulation consistency, what is definitely undesired.

5. RESULTS

A SISO case study is presented, where results regarding dynamic simulation and MPC performance were obtained. The computational tool applied for all implementations was Python, which is a dynamic and powerful programming language, with several applications on different fields (Pérez, Granger and Hunter 2011). The processor employed for all simulations is an Intel® Core™ i5 CPU 750 @ 2.67 GHz, with 6 GB of RAM. The system consists of three tanks arranged in sequence, as it can be seen in Fig. 6, where there is only one feed into the first tank, F_0. The system modelling is shown in (18) to (20):

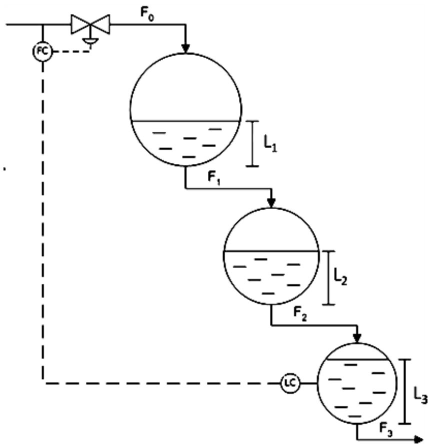

Fig. 4. Schematic representation of a three tanks laboratorial plant.

$$\frac{dL_1}{dt} = \frac{F_0 * 1000 - CD_1\sqrt{L_1}}{\pi L_1(D_1 - L_1)} \tag{18}$$

$$\frac{dL_2}{dt} = \frac{CD_1\sqrt{L_1} - CD_2\sqrt{L_2}}{\pi L_2(D_2 - L_2)} \tag{19}$$

$$\frac{dL_3}{dt} = \frac{CD_2\sqrt{L_2} - CD_3\sqrt{L_3}}{\pi L_3(D_3 - L_3)} \tag{20}$$

Where L_i is the tank level, in *cm*, and F_0 is the input flow in the first tank, in *L/min*. Parameters are CD_i, discharge coefficient on each tank, in $cm^{2.5}/min$, and D_i, tank diameter, in *cm*. These values are expressed in Table 1:

Table 1. Modelling parameters for a three tanks laboratorial plant.

	CD_i ($cm^{2.5}/min$)	D_i (*cm*)
Tank 1	1900	30
Tank 2	2000	30
Tank 3	2100	30

Validation of dynamic simulation was performed by submitting the input to an oscillatory pattern, in order to analyze model adherence to original data. In order to perform a comparison between plots, a reference curve was generated on each case, by using a Python linear simulation tool. Three curves were generated, one with 19 knots and two with 11 knots. TCB parameters were kept as zero for both 19 knots curve and one of the 11 knots curve, being altered only the bias (ω) on the other 11 knots curve. These parameters can be seen in Table 2. Fig. 7 illustrates each model adherence when compared with the reference curve. Finally, Table 3 expresses the quadratic error sum for each curve.

Table 2. Parameterized input parameters for an oscillatory response.

	Z	Knots Position	
U_{11}	11	[0;1.5;4.5;8.5;10.5;14.5;17;20.5;23.5;26.5;30]	
U_{11Opt}	11	[0;1.5;4.5;8.5;10.5;14.5;17;20.5;23.5;26.5;30]	
		ω	[0;0;0.2;-0.8;0.5;-0.5;0.4;0;0;0.2;-1]

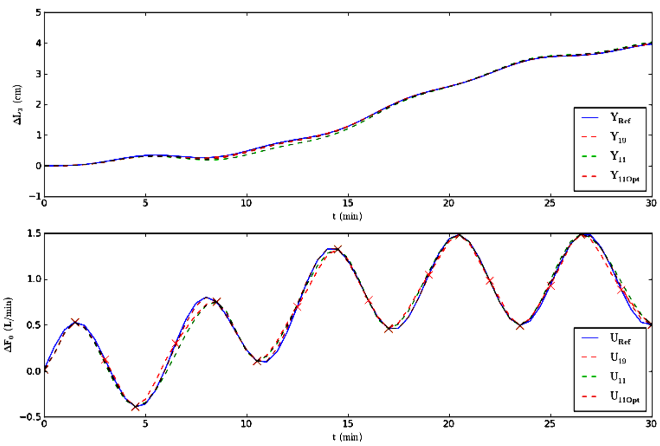

Fig. 5. Output response to an oscillatory input pattern with indication of knots position, evaluating numbers and displacement knots impact.

Table 3. Quadratic error sum for each output arisen from parameterized inputs of an oscillatory pattern.

	$\sum_{i=0}^{P}\left(Y_{\mathrm{Ref}_i}-Y_i\right)^2$
U_{19}	0.01
U_{11}	0.26
U_{11Opt}	0.03

It is clear that diminishing the number of knots has a negative impact on the model adherence, merely because fewer knots give a poor spline fit, on a general basis, as it can be seen with U_{19} and U_{11} curves. By adjusting TCB parameters, however, such error can be significantly reduced, as U_{11Opt} shows. SDM is valid, thus, as a tool for simulation, once dynamic information is coherently represented. In the end, it is all about compromise between knots and TCB parameters, related to simulation speed and robustness, and final model quality.

5.2 MPC Application

The flow F_0 serves as the input, controlling the level L_3 on the third tank. Three different operating levels were determined for the third tank, which were used for robustness and general performance evaluation of this new SDM based MPC. Fig. 8 shows a step response for the three cases and equations (21) to (23) show the equivalent transfer functions.

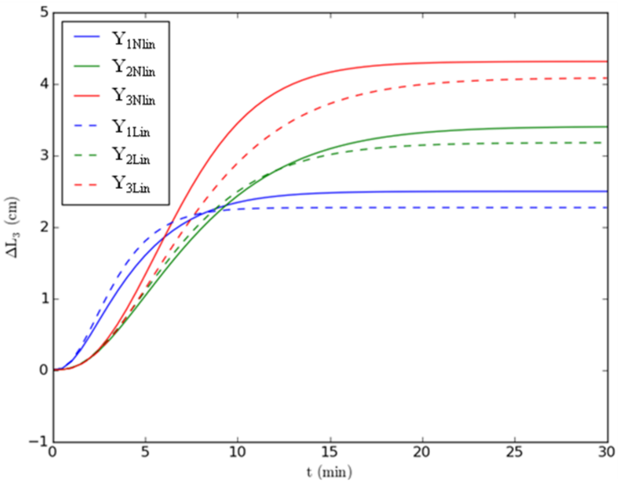

Fig. 6. Step response for three different operating points on a laboratorial three tanks plant, expressing the difference between linear and non linear simulation.

$$W_1 = \frac{0.1501}{s^3 + 1.0816s^2 + 0.3799s + 0.0429} \tag{21}$$

$$W_2 = \frac{0.9011}{s^3 + 2.2054s^2 + 1.5761s + 0.3604} \tag{22}$$

$$W_3 = \frac{0.1324}{s^3 + 0.9457s^2 + 0.2926s + 0.0294} \tag{23}$$

For results consistency, a controller that can, theoretically, control all operating points should be designed. According to the step responses, it can be inferred that the most difficult region to control is around W_3, due to being the operating point with the biggest gain and the slowest dynamic. Fig. 9 shows a simulation for an output set-point change and input disturbance with same magnitude for both controllers.

As it can be seen in Fig. 9, the MPC based on SDM, when compared with MPC-STEP, presents fewer oscillations, reaching the set point quickly. Yet, rejects rather well disturbances, being even better than MPC-STEP. Fig. 10 and Fig. 11 show the response for the same disturbances of the other two operating points, respectively, with similar results.

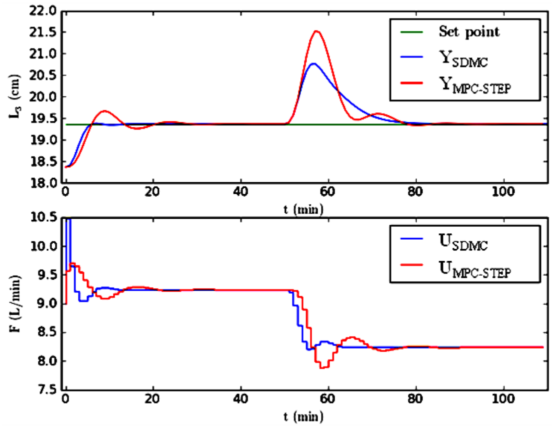

Fig. 7. Servo-regulatory behaviour of two different controllers designed for the W_3 operating point on the W_3 operating point.

In order to compare both controllers, equivalent rise-times were designed, according to the parameters expressed in Table 4.

Table 4. Base parameters used for SDMC and MPC-STEP.

	P (*min*)	M (*min*)	T_s (*min*)	Z	Knots Displacement	Γ	Λ
SDMC	30	8	1	5	[0;3;6;8;30]	1	0.5
MPC-STEP	30	10	1	-	-	1	1.2

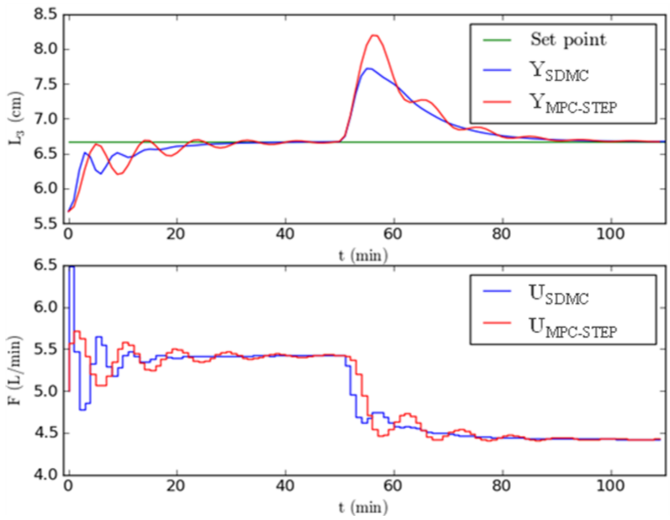

Fig. 8. Servo-reguatory behaviour of two different controllers designed for the W_3 operating point on the W_1 operating point.

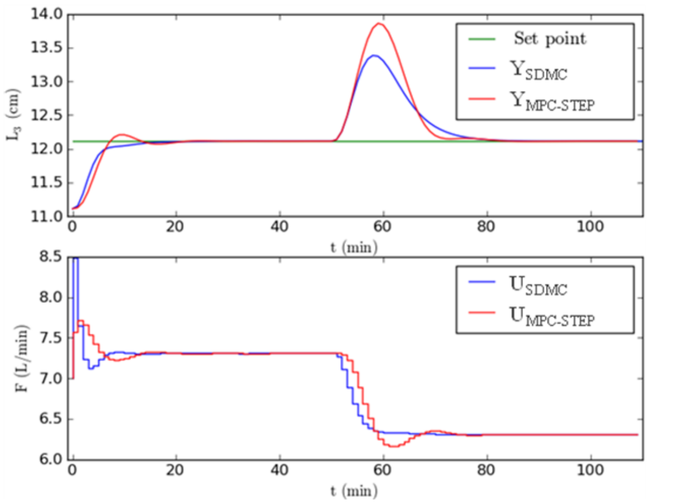

Fig. 9. Servo-regulatory behaviour of two different controllers designed for the W_3 operating point on the W_2 operating point.

Which would be the reasons for such behaviour? Referring to SDM methodology itself, it can be seen that among all points along *P* only a few ones were responsible for the input curve variation, guaranteeing more robust behaviour. Such behaviour is influenced directly by the amount of knots, as it can be seen in Fig. 12, where two different curves, one with four knots and another with seventeen, were used for input parameterization, shown in Table 5.

Table 5. Parameters used for SDMC, varying the number of knots..

	P (*min*)	M (*min*)	T_s (*min*)	Z	Knots Displacement	Γ	Λ
U_4	30	8	1	5	[0;4;7.5;30]	1	0.02
U_{17}	30	10	1	-	[0;0.5;1;1.5;2;2.5;3;3.5;4; 4.5;5;5.5;6;6.5;7;7.5;30]	1	0.02

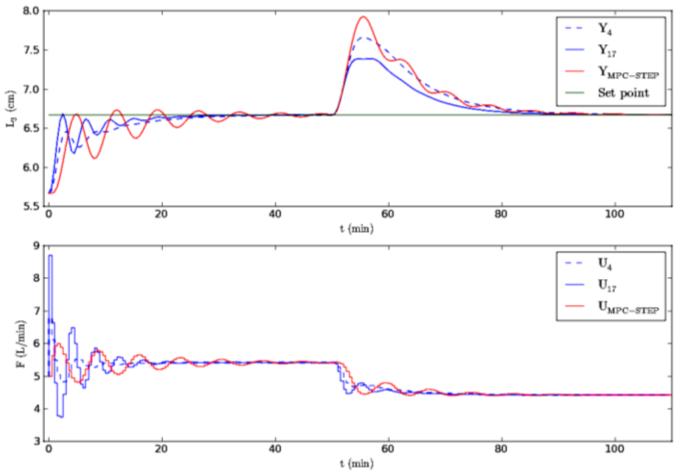

Fig. 10. Servo-regulatory behaviour of two different controllers designed for the W_3 operating point on the W_1 operating point, for different numbers of knots.

Analogue to move suppression, the amount of knots affects the response oscillation and disturbance rejection, where more knots favour rejection, but increase oscillation and vice-versa. As well as for dynamic simulation, SDMC is corroborated as an efficient and promising tool for control. It confers certain characteristics related to robustness and, on a general basis, has a faster simulation due to reduction of decision variables.

6. CONCLUSIONS

In this work a new tool for dynamic process simulation was developed combining spline theory, DM modelling and state space representation. Its application, however, is not limited to dynamic simulation only. Supported on this methodology, a new model predictive controller was proposed, SDMC. With the assimilation of SDM characteristics in output prediction and optimization problem formulation, new features were introduced into it. Robustness, for instance, was achieved without performance loss. The number of knots has a vital role on this issue, acting analogously to move suppression. This term, by the way, can even be suppressed.

Future works within this methodology include process identification and also MIMO applications, soft constraints, targets, etc. Related to dynamic information capturing, model auditing and identification present themselves as fields yet to be explored. Overall, SDM is a technique that presents a great potential, due to its intuitive and simple concept.

REFERENCES

BISWAL, P. C. 2008. Numerical Analysis, Prentice-Hall Of India Pvt. Ltd.

BOOR, C. D. 2001. A practical guide to splines, Springer.

CAGIENARD, R., GRIEDER, P., KERRIGAN, E. C. &

MORARI, M. 2007. Move blocking strategies in receding horizon control. Journal of Process Control, 17, 563-570.

CAMACHO, E. F. & BORDONS, C. 2004. Model predictive control, Springer.

CUTLER, C. R. & RAMAKER, B. L. 1980. DYNAMIC

MATRIX CONTROL - A COMPUTER CONTROL

ALGORITHM. Journal of Environmental Science and Health, Part B: Pesticides, Food Contaminants, and Agricultural Wastes, 1.

ESCOBAR, M. S. 2012. Spline Dynamic Matrix Controller: Uma Nova Formulação de Controladores Preditivos. Msc., Universidade Federal do Rio Grande do Sul.

FOLEY, J. D. 2006. Introduction to computer graphics, Addison-Wesley.

GONDHALEKAR, R. & IMURA, J.-I. 2010. Least-restrictive move-blocking model predictive control. Automatica, 46, 1234-1240.

MACIEJOWSKI, J. M. 2002. Predictive control: with constraints, Prentice Hall.

OGATA, K. 2003. Modern control engineering, Prentice Hall.

OGUNNAIKE, B. A. & RAY, W. H. 1994. Process dynamics, modeling, and control, Oxford University Press.

ÅSTRÖM, K. J. & WITTENMARK, B. 1997. Computer-controlled systems: theory and design, Prentice Hall.