TWO-LEVEL PREDICTIVE BASED AXIS CONTROL FOR VIRTUAL MACHINE TOOL

M. Susanu* D. Dumur*

** Supélec, Plateau de Moulon, 3 Rue Joliot Curie, 91192 Gif-sur-Yvette cedex, France*

Abstract: Dealing with activation of trajectory constraints within machine-tools CNC is usually performed directly by the axis controllers, leading to a compromise with axis performance. This paper presents an original strategy, including classical predictive feedback axis controllers, with an upper level real time trajectory planning module. This module realized without any structural changes in the main control loop is addressed within an open architecture machine-tool framework to avoid trajectory constraints activation. This strategy is finally tested in a virtual machine-tool dedicated to advanced control strategies evaluation. *Copyright © 2005 IFAC*

Keywords: Predictive control, constrained optimization, trajectory planning, supervision, virtual reality, machine-tools.

1. INTRODUCTION

Open architecture machining platforms provide the facilities for modifying, adding or removing different embedded modules as frequently as necessary. Latest technologies available on the market or own-built modules with "plug and play" capabilities may be used without reconsidering the entire structure of the system. The increasing functionalities of such architecture imply of course the diminution of over-all costs. The portability, extendibility, interoperability and scalability criteria for estimating the openness of a controller (Pritschow, *et al*., 2001) define a new concept of machine tool, with advanced functionalities and modular structure. Still, the conventional CNC machines-tools on the market remain closed, with pre-specified controllers structures or trajectory generation modules, and in general do not enable the user to implement dedicated modules (Pritschow, *et al*., 2001).

In this context, the purpose is to implement a real-time trajectory planning module in a virtual machine-tool earlier developed while remaining within the open architecture framework. In fact, this machine-tool simulator focusing on the basic servodrives level already includes a classical trajectory generation module (Susanu and Dumur, 2004b) and advanced control modules (Susanu, *et al*., 2004a). The next step thus considers adding an independent module for handling the constraints, called in further developments "reference supervisor" although in the literature it can be generally found under the "reference governor" term, see (Bemporad, *et al*., 1997) for more details concerning the theoretical background and simulation studies on this topic.

Any type of constraints can be considered within this mechanism. The discussion will be limited here to constraints on the control signal. These constraints become active when the controller delivers values that can not be applied to the system, leading to windup effect and deterioration of the closed-loop performance. Generally, for solving this problem, either anti-windup techniques (feedback compensations) are developed, or techniques taking into account the constraints formulation even from the control design phase, which are computationally expensive. Based on the existing structure of the simulator, this paper presents a method that avoids the constraints activation, changing the setpoint with a convenient one. One of the main advantages of this structure is that adding it as a separate unit does not imply any modification of the basic designed controller modules. The planning block acts on-line while predicting the trajectory over a "trajectory prediction horizon". The tracking performances of the controlled system are entirely preserved and in the same time the constraints are not overtaken.

The question that may arise at this step is how much the initial reference trajectory will be modified? In

fact, this new trajectory planning module will act only in the limit cases where the constraints violation is caused by a certain unexpected event, otherwise, the output of this block will be the reference trajectory used as input, without any changes. Compared to the classical saturation block, the proposed one works on-line, and manages not to handle but to avoid the saturation, having a predictive structure. The performance degradation is much more reduced and the control signal is smoother. Indeed, long-time activation of this module may irreversibly degrade the performances and the global trajectory planning in time must be reconsidered.

The paper is structured as follows. In the next part the virtual machine tool is presented, under the "control level" terminology. The "trajectory supervisor level" is discussed in the third section. Section 4 presents simulation results obtained on the virtual machine tool, considering the particular case where the variant dynamic torque due to the cutting forces interferes on the control structure and causes the overtaking of the velocity limits. Some conclusions are given at the end.

2. THE CONTROL LEVEL

2.1 The virtual machine tool

The generic virtual model of a 3 axis machining centre can be synthesized under the structure of Figure 1.

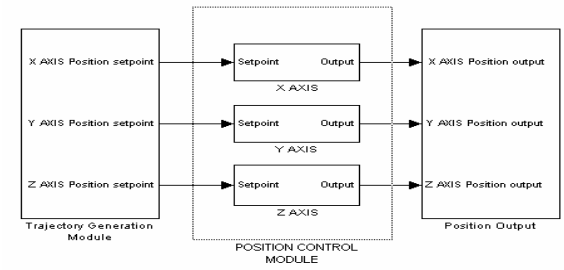

Fig. 1. Virtual machine tool simplified structure.

The first block, called here "the trajectory generation module" carries out multiple functions. In short, it represents 3D trajectories starting from user-defined points as a CAD module. Next, the CAM functionalities are simulated by breaking the generated curves into segments, similar with the G-codes generation on a machining centre. Further, it plays the interpolator role as it generates in the end the sampled reference points, decomposed on each axis of movement. It outputs them to the axis control modules using a linear interpolation algorithm following the trapezoidal velocity profile. A detailed description of the developed module can be found in (Susanu and Dumur, 2004b).

Further, the model of a 3-axis vertical milling machine is developed (Figure 1) starting from the servodrive classical model (Figure 2), replicated for all the active axes with the corresponding parameters values. It has been assumed for simplification purpose that there are no interactions between the modelled axes. Mathematical aspects of the modelling and identification phases can be found in (Susanu, *et al*., 2003). The three cascaded control loops correspond to current, velocity and position control. Classically the controllers are of proportional-integral (PI) type for the two inner loops and proportional (P) type for the outer loop. The open loop acceleration and velocity feedforward actions are added in order to minimize the tracking error. The velocity and current setpoints are saturated to some physical limits.

As the goal of this simulator was to develop a convenient structure for the open architecture demands, an advanced control module has been implemented, substituting this classical P/PI/PI control. A Generalized Predictive Control GPC (Clarke, *et al*., 1987) module in a cascaded version is designed, replacing the two external loops with appropriate predictive controllers and keeping the current loop unchanged, as it has a very fast dynamic. The feedforward open loop controllers have been deactivated as the anticipation effect is induced by the predictive structure. The developed module can be found in (Susanu, *et al*., 2004a), with very good performances in terms of tracking accuracy. Presentation will be restricted here to the main aspect of this module that will influence further formulations, that is the RST representation since the entire control structure, either the classical P/PI/PI or the advanced GPC has been formulated under this framework.

2.2 The RST numerical controller

The general framework for the design of the control laws within an open CNC architecture is the "RST" formalism, illustrated in Figure 3. The controller is implemented through a finite difference equation:

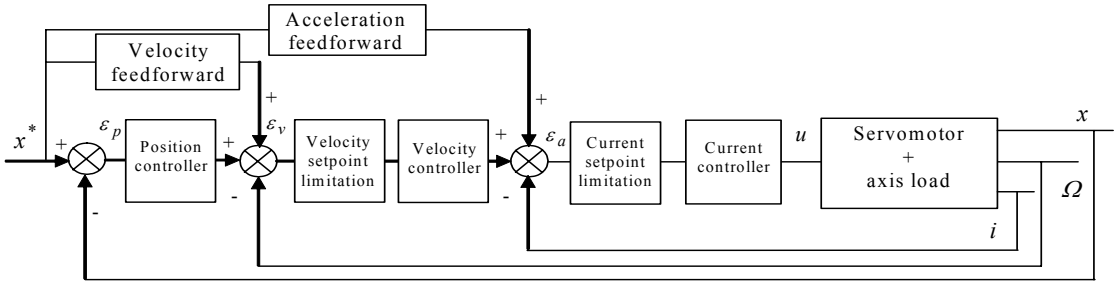

Fig. 2. Feed drive servocontrol structure.

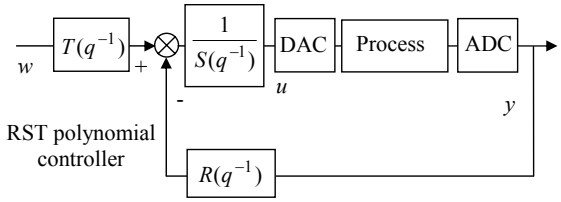

Fig. 3. The RST numerical controller.

$$S(q^{-1})\,u(t) = -R(q^{-1})\,y(t) + T(q^{-1})\,w(t) \qquad (1)$$

As main advantage, this two-degree of freedom structure allows decoupling tracking and disturbance rejection dynamics. Another interest of this generic formalism is the resulting reduced computation time as the controller polynomials are in most cases of small degrees. Implementing this strategy is then suitable even for short sampling time, as for instance in the case of high speed machining.

3. THE TRAJECTORY SUPERVISING LEVEL

3.1 The general structure

The reference supervisor can be designed for any linear time invariant controller under the RST form. However, for further implementation in CNC with advanced control structure, the GPC law will be emphasized, leading to a non causal *T* polynomial providing the anticipative behavior.

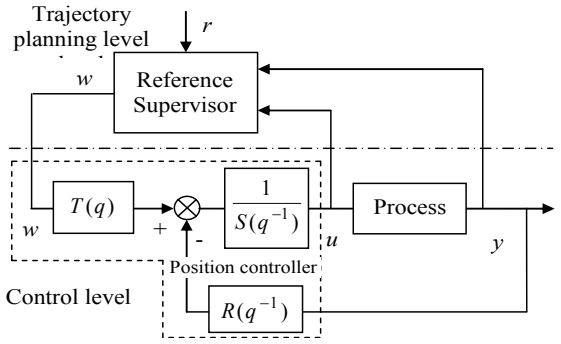

Fig. 4. The two-level predictive control structure.

The proposed two-level structure is indicated in Figure 4. The first level contains the servocontrol system described earlier (Figure 2) limited to the most external position loop, whereas a supervising nonlinear block defines the upper secondary level. The block noted "Process" represents the remaining part of the control scheme, i.e. the two loops for current and velocity with the limitation elements not removed. As mentioned before, the main point here is that adding the supervisor does not imply any changes in the initial control structure developed for best tracking performances, supposed to work in absence of any constraints. Indeed, referring to our structure, the off-line GPC law does not take into consideration any constraints.

The reference supervisor forms a new external loop on the structure. It interferes only if the constraints are to be violated by modifying the reference trajectory, acting as a trajectory planner: it finds the optimal trajectory (with respect to a cost index) for which the constraints saturation is avoided. Between the levels, a bidirectional exchange of information flow exists. The execution of the superior routine is based on the lower level actions as it will be detailed further, whereas the inferior level acts following the directions imposed by its supervisor, although there is a total independence in the construction of the base level, as seen before. The connection between the two levels from the performances point of view is that the lowest level tracking capabilities are entirely used, being applied on the specific trajectory chosen by the planner and responding with the known linear performances. Another important point underlying this hierarchic aspect of the structure is that the two levels can work with different sampling period, reducing the computational burden.

3.2 The mathematical formulation

The problem will be formulated below for constraints on the control signal, but the same procedure can be followed in order to include other constraints, for example on the output overshoot. As mentioned, the goal is to select the best trajectory points fitting the constraints demands, N_c -steps ahead, with N_c the trajectory prediction horizon. The selection is completed solving quadratic minimization criteria following the 'receding horizon' philosophy.

Generally, the limits on the control signal over the time horizon N_c can be written as:

$$\underbrace{\begin{bmatrix} \underline{u}_1 \\ \underline{u}_2 \\ \vdots \\ \underline{u}_{N_c} \end{bmatrix}}_{\underline{\mathbf{U}}} \le \begin{bmatrix} u(t) \\ u(t+1) \\ \vdots \\ u(t+N_c) \end{bmatrix} \le \underbrace{\begin{bmatrix} \overline{u}_1 \\ \overline{u}_2 \\ \vdots \\ \overline{u}_{N_c} \end{bmatrix}}_{\overline{\mathbf{U}}} \qquad (2)$$

with $\underline{\mathbf{U}} = [\underline{u}_1 \ldots \underline{u}_{N_C}]^{\mathrm{T}}$, $\overline{\mathbf{U}} = [\overline{u}_1 \ldots \overline{u}_{N_c}]^{\mathrm{T}}$ the lower and upper control signal limits. From the RST formulation of the GPC law Eq. 1, the control signal value is given in a matrix form by:

$$u(t) = [-\widetilde{S} \quad -R \quad T]\left[\mathbf{u}_p(t) \;\; \mathbf{y}_p(t) \;\; \mathbf{w}_t(t)\right]^{\mathrm{T}} \qquad (3)$$

with:

$$\mathbf{u}_p(t) = \begin{bmatrix} u(t-1) \\ \vdots \\ u(t-n_S) \end{bmatrix},\ \mathbf{y}_p(t) = \begin{bmatrix} y(t) \\ \vdots \\ y(t-n_R) \end{bmatrix}$$

$$\mathbf{w}_t(t) = \begin{bmatrix} w(t+1) \\ \vdots \\ w(t+n_T) \end{bmatrix},\ S = 1 + \widetilde{S},\ \begin{matrix} n_S = \deg(S) \\ n_T = \deg(T) \\ n_R = \deg(R) \end{matrix}$$

For further formulation taking into account the trajectory prediction horizon, $\mathbf{w}_t$ will be augmented to:

$$\mathbf{w}_t(t) = \left[w(t+1) \quad \cdots \quad w(t+n_T+N_c-1)\right]^{\mathrm{T}} \quad (4)$$

For the sake of computation with the non-causal T polynomial, the following evolution of $\mathbf{w}_t$ is chosen:

$$\mathbf{w}_t(t+k) = \left[w(t+k+1)\ldots w(t+k+n_T)\ldots \right.$$
$$\left. \ldots w(t+n_T+N_c-1)\underbrace{0\ldots 0}_{k}\right], k < N_c \quad (5)$$

Eq. 3 now becomes:

$$u(t) = \boldsymbol{\varphi}\,\boldsymbol{\theta}(t) \quad (6)$$

with:

$$\boldsymbol{\varphi} = \left[\begin{array}{cccc} -\widetilde{S} & -R & T & \underbrace{0\ldots 0}_{N_c-1} \\ & & \underbrace{\qquad}_{\hat{T}} & \end{array}\right]_{1\times(n_S+n_R+n_T+N_c)} \quad (7)$$

$$\boldsymbol{\theta}(t) = \left[\mathbf{u}_p(t)\,\mathbf{y}_p(t)\,\mathbf{w}_t(t)\right]^{\mathrm{T}}_{(n_S+n_R+n_T+N_c)\times 1} \quad (8)$$

Using Eqs. 3 and 5, and assuming the system is modeled by the discrete transfer function B/A, with polynomial degrees $n_B = n_S$ and $n_A = n_R$ (Boucher and Dumur, 1996), the evolution of $\boldsymbol{\theta}(t)$ can further be expressed as:

$$\boldsymbol{\theta}(t+k) = \boldsymbol{\Psi}\ \boldsymbol{\theta}(t+k-1) \quad (9)$$

with:

$$\boldsymbol{\Psi} = \left[\begin{array}{c|c|c} -\widetilde{S} & -R & \hat{T} \\ \mathbf{J}_{\widetilde{S}} & 0 & 0 \\ \hline B & [-\widetilde{A}\ 0] & 0 \\ 0 & \mathbf{J}_R & 0 \\ \hline 0 & 0 & \mathbf{V} \end{array}\right] \quad (10)$$

where $A = 1+\widetilde{A}$ and:

$$\mathbf{J}_x = \begin{bmatrix} 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 \end{bmatrix}_{(n_x\times n_x)}, \ x = \widetilde{S}, R \quad (11)$$

$$\mathbf{V} = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 1 \\ 0 & 0 & 0 & \ldots & 0 \end{bmatrix}_{(n_T+N_c-1)\times(n_T+N_c-1)} \quad (12)$$

Including Eq. 6 in Eq. 2 leads to:

$$\underline{\mathbf{U}} \le \underbrace{\left[\boldsymbol{\varphi} \quad \boldsymbol{\varphi}\,\boldsymbol{\Psi} \ldots \boldsymbol{\varphi}\,\boldsymbol{\Psi}^{N_c}\right]^{\mathrm{T}}}_{\boldsymbol{\Sigma}^*}\boldsymbol{\theta}(t) \le \overline{\mathbf{U}}$$

or:

$$\underline{\mathbf{U}} \le \underbrace{\left[\hat{\boldsymbol{\Sigma}}\,\middle|\,\overline{\boldsymbol{\Sigma}}\right]}_{\boldsymbol{\Sigma}^*}\left[\begin{array}{c} \mathbf{u}_p(t) \\ \mathbf{y}_p(t) \\ \hline \mathbf{w}_t(t) \end{array}\right] \le \overline{\mathbf{U}} \quad (13)$$

Finally, the multiparametric inequality:

$$\left[\begin{array}{c} -\overline{\boldsymbol{\Sigma}} \\ \hline \overline{\boldsymbol{\Sigma}} \end{array}\right]\mathbf{w}_t(t) \le \left[\begin{array}{c} \underline{\mathbf{U}} + \hat{\boldsymbol{\Sigma}}\left[\begin{array}{c} \mathbf{u}_p(t) \\ \mathbf{y}_p(t) \end{array}\right] \\ \hline \overline{\mathbf{U}} + \hat{\boldsymbol{\Sigma}}\left[\begin{array}{c} \mathbf{u}_p(t) \\ \mathbf{y}_p(t) \end{array}\right] \end{array}\right] \quad (14)$$

has to be considered for the selection of the modified trajectory N_c-steps ahead in a convex constrained quadratic optimization problem as expressed in (Casavola, *et al*., 2004). At each sampling period the reference supervisor output w_m is computed from:

$$\mathbf{w}_m(t) = \arg\min_{\mathbf{w}_t}\left\|\mathbf{w}_t(t) - \mathbf{r}(t)\right\|^2_{\boldsymbol{\Lambda}} \quad (15)$$

under the constraints in (14), with $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}^{\mathrm{T}} > 0$ a weighting matrix, $\mathbf{r}$ the initial reference sequence and $\|\mathbf{x}\|^2_{\boldsymbol{\Lambda}} := \mathbf{x}^{\mathrm{T}}\boldsymbol{\Lambda}\,\mathbf{x}$. This eventually modified reference is applied on the main controlled system input and the procedure is repeated at each sampling period, following the receding horizon principle. The prediction horizon N_c can be determined off-line (see the algorithm in (Bemporad, *et al*., 1997)).

4. SIMULATION RESULTS

The following examples are related to the case where the reference supervisor is implemented within the position control loop as an additional module of the virtual machine-tool corresponding to the Mikron UCP 710 machining centre (see (Susanu, *et al*., 2003)). Therefore, constraints will be imposed as limitation of the linear velocity setpoint on each axis of movement. For all cases, the displacements considered are on the X-axis of movement, the position control loop works at 0.006 s sampling rate as well as the trajectory supervisor. The simulations are performed in Matlab 6.5 and Simulink. The quadprog.m function was used for solving the quadratic problem. In order to observe the operating principle of the implemented module, the first example is a rough tightening of these constraints.

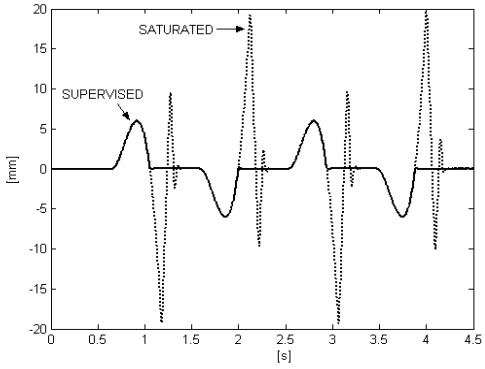

Fig. 5. $N_c = 1$. Tracking errors: saturated case – dotted, supervised case – solid.

In figures 5 to 8, the setpoint is the X axis component of a 100 mm diameter circular trajectory. The position loop control signal, i.e. the velocity setpoint, is

limited to 300mm/sec. The trajectory prediction horizon is equal to 1 in Figures 5 and 6. The tracking performances of the supervised reference case are compared with the saturated case. The windup effect due to saturation is underlined in Figure 5, where the corresponding tracking errors are plotted. The supervised reference significantly reduces the transient tracking error. Figure 6 shows the related velocity setpoints obtained under the same conditions.

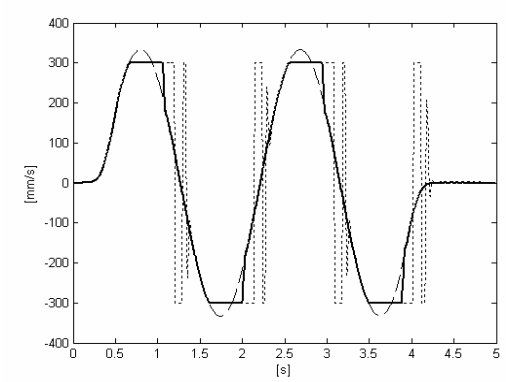

Fig. 6. $N_c = 1$. Control signal without constraints – dashed, saturated – dotted, supervised – solid.

The position performances for a family of supervisors related to prediction horizons between 1 and 25 are plotted in Figure 7. A zoom of the transient part is shown in Figure 8. The anticipative aspect of the supervisors with respect to their prediction horizons are clearly noticeable, as well as the decrease of the tracking error when N_c increases. For $N_c > 25$ however, the improvements are not significant and the computing time growth becomes unjustified.

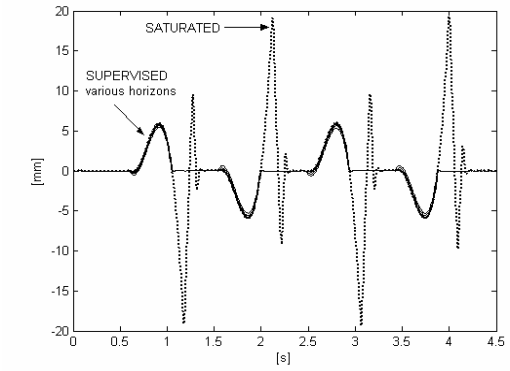

Fig. 7. Tracking errors: saturated case – dotted, supervised case for $N_c = 1 \cdots 25$ – solid.

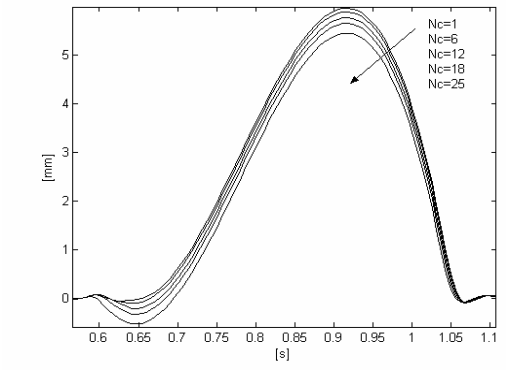

Fig. 8. Zoom on the tracking errors for $N_c = 1 \cdots 25$.

Yet such an example has not a practical meaning as long as the velocity constraints could not be that rough, as the profile is imposed from the trajectory generation phase. For our specific case of applying this strategy on the virtual machine tool drives, the next simulations consider a plant-oriented example. Nevertheless, the velocity saturation interferes rarely as the limits are managed even from the design phase, before the interpolator's action, by imposing a certain velocity over the desired profile, with the maximum admissible velocity obviously known. But this does not mean that while machining at high speeds, close to the limits, the environment will not cause different disturbances that might induce a certain 'destabilization' talking in terms of constraints.

Starting from this idea, a random noise on the torque has been considered, as disturbances. Even if the control structure is sufficiently robust and manages to settle rapid enough the perturbed system, picks on the velocity setpoint remain, with values beyond the imposed limits. This might correspond to eventual unexpected variations in the material density causing variations in the cutting forces profile. In order to simulate this, a white noise with 0.6 s sampling time has been imposed. The position setpoint is a linear displacement generated with a linear interpolation algorithm following a trapezoidal velocity profile. The linear speed is 60 mm/sec, the distance is 144.9 mm.

Figures 9 to 11 correspond to the trajectory supervisor with $N_c = 1$. Figure 9 plots the tracking errors. Two zooms on different parts of the control signals are proposed in Figures 10 and 11, to clearly assess the reduction in the transient due to the supervisor.

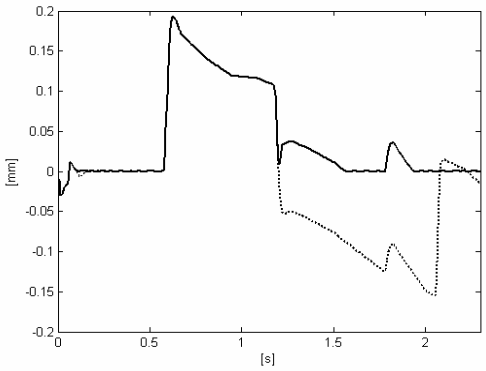

Fig. 9. $N_c = 1$. Tracking errors: saturated case – dotted, supervised case – solid.

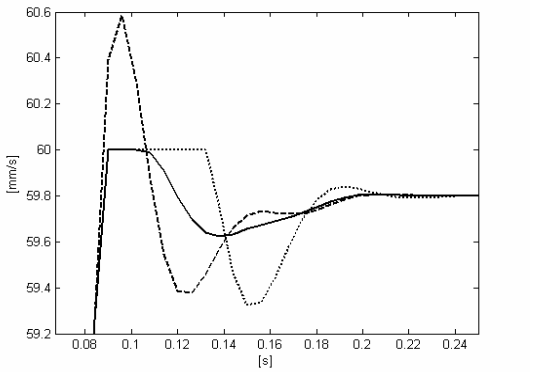

Fig. 10. $N_c = 1$. Zoom on the transient part of the control signal without constraints – dashed, saturated – dotted, supervised – solid.

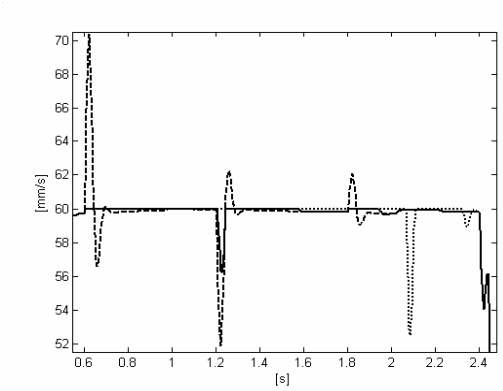

Fig. 11. $N_c = 1$. Zoom on the control signal at constant velocity: without constraints – dashed, saturated – dotted, supervised – solid.

Figures 12 and 13 show results obtained with a larger prediction horizon, $N_c = 15$.

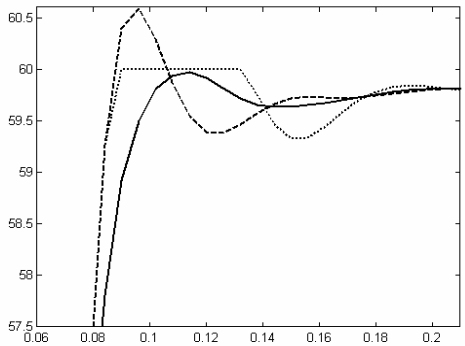

Fig. 12. $N_c = 15$. Zoom on the transient part of the control signal without constraints – dashed, saturated – dotted, supervised – solid.

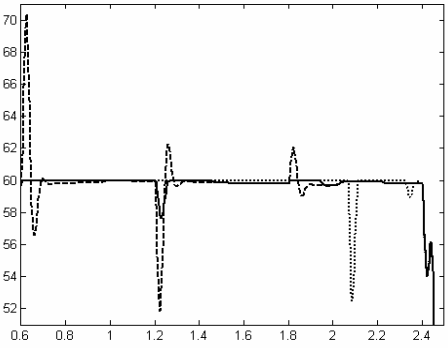

Fig. 13. $N_c = 15$. Zoom on the control signal at constant velocity: without constraints – dashed, saturated – dotted, supervised – solid.

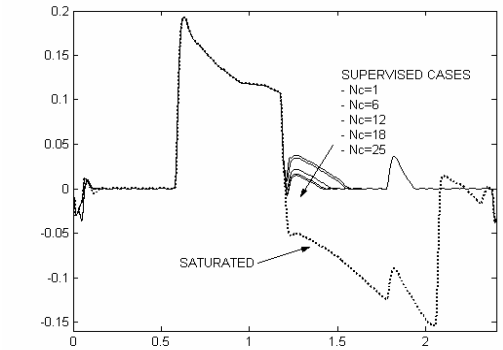

Fig. 14. Tracking errors: saturated case – dotted, supervised case for $N_c = 1 \cdots 25$ – solid.

Figure 14 shows the tracking performances for a family of trajectory supervisors with prediction horizons from 1 to 25. Again, the error decreases with the increase of the N_c prediction horizon. All these results prove that the supervisor enables a better tracking behavior. The on-line calculation of a modified reference avoids the penalizing effect of long saturation of the velocity setpoint, e.g. in Figure 11 where saturation only ends at $t \approx 2.2\,\mathrm{s}$ without supervisor, compared to $t \approx 1.2\,\mathrm{s}$ with supervisor.

5. CONCLUSIONS

The paper addressed an innovative implementation of a trajectory supervising module which comes into action by changing the setpoint once the constraints are to be overtaken. Otherwise, the reference trajectory remains unchanged. The strategy of computing the modified reference is based on predictive and sliding horizon techniques. The tracking performances of the servocontrol system equipped with this kind of module are improved compared with the classical case when the saturation elements are causing marked windup effects. Developing such an upper level within the virtual machine tool is appropriate for the open architecture framework since its integration does not imply modification of the already implemented primary axis controllers.

REFERENCES

Bemporad, A., A. Casavola and E. Mosca (1997). Nonlinear control of constrained linear systems via predictive reference management. *IEEE Trans. Automatic Control*, **Vol. 42**, pp 340-349.

Boucher P. and D. Dumur (1996). *La Commande Prédictive*. Editions Technip, Paris.

Casavola, A., E. Mosca and M. Papini (2004). Control under constraints: an application of the command governor approach to an inverted pendulum. *IEEE Transactions on control systems technology,* **Vol. 12**, pp 193-204.

Pritschow, G., *et al*. (2001). Open Controller Architecture – Past, Present and Future, *CIRP Annals*, **Vol. 50**, pp. 463-470.

Susanu, M., D. Dumur and C. Lartigue (2003). Modeling and advanced control of machine-tool in an open architecture framework. *Proceedings of the 14th Intl Conference on Control Systems and Computer Science, Bucharest*, pp. 300-305.

Susanu, M., D. Dumur, C. Lartigue and C. Tournier (2004a). Improving performance of machine tools with predictive axis controllers within an open architecture framework. *Proceedings of the 3rd Intl Conference on Advanced Manufacturing Technology, Kuala Lumpur,* pp. 322-327.

Susanu, M. and D. Dumur (2004b). Advanced axis control implementation within a virtual machine tool environment. *Proceedings of IEEE 2004 Computer Aided Control System Design Conference, Taipei,* pp. 7-12.