SICOMAT : a system for SImulation and COntrol analysis of MAchine Tools

Maxime Gautier, Minh Tu Pham, Wisama Khalil, Philippe Lemoine, Philippe Poignet

HAL Id: hal-00362669

https://hal.archives-ouvertes.fr/hal-00362669

Submitted on 19 Feb 2009

SICOMAT: A system for SImulation and COntrol analysis of MAchine Tools

M. Gautier*, M.T. Pham*, W. Khalil*, Ph. Lemoine *, Ph. Poignet **

* Institut de Recherche en Communication et en Cyberntique de Nantes (I.R.C.Cy.N.)
1 rue de la Noë, BP 92101, 44321 Nantes Cedex 03, France.
** Laboratoire d'Informatique, de Robotique et de Microlectronique de Montpellier (L.I.R.M.M.)
161 rue Ada, 34392 Montpellier Cedex 05, France
Maxime.Gautier@irccyn.ec-nantes.fr
http://www.irccyn.ec-nantes.fr

Abstract— **This paper presents a software package for the simulation and the control analysis of machine tool axes. This package which is called SICOMAT (SImulation and COntrol analysis of MAchine Tools), provides a large variety of toolboxes to analyze the behavior and the control of the machine. The software takes into account several elements such as the flexibility of bodies, the interaction between several axes, the effect of numerical control and the availability to reduce models.**

I. INTRODUCTION

The need for the manufacturer to reduce the cost for designing and evaluating a prototype goes through the definition of a good simulation tool. Therefore we develop a simulation environment able to guide the designer in the choice of the components interfering in the complete mechatronic system i.e. flexible mechanical structures, actuators, sensors and controllers. The ever increasing joint speed and acceleration of machine tools leads to develop accurate models taking into account elasticity of structure and joints. A good compromise between the complexity of a distributed elasticity approach as finite element model and the simplicity of neglecting the elasticity with a rigid body model is to consider a multi-body model with lumped elasticities and rigid bodies based on a technology analysis of the system's components [1]. A block diagram and a linear state space description of one axis or several decoupled axes are implemented. A description of multi-body systems based on the robotic formalism is also implemented [2]. The originality of this approach is to provide a systematic method to automatically calculate the geometric, kinematic and dynamic models of coupled or decoupled multi-axes machine tools, whatever the number of rigid and elastic degrees of freedom (dof) of the system. Moreover the software contains several tools to completely study a given structure from the modeling (modal analysis toolbox and model reduction toolbox) to the control simulation. An other significant point of this package is the possibility to tune the controller gains considering the computer numerical controller (CNC) components (PID with feedforward controller structure, delays, quantization), the mechanical structure and the drive chain components included in the simulator. Two methods are available in **SICOMAT**. The first one is a time approach based on interactively calibrating step responses. The second one is based on a frequency automatic approach, which guarantees stability margins for a given closed loop bandwidth using optimization techniques [3][4].

The paper is organized as follows: section 2 describes the global environment of the simulator and the structure modeling, section 3 presents the different toolboxes available in the simulator: model reduction, modal analysis, control design techniques (time and frequency approaches). Finally, section 4 gives an example of a high-speed machine with a linear direct drive axis.

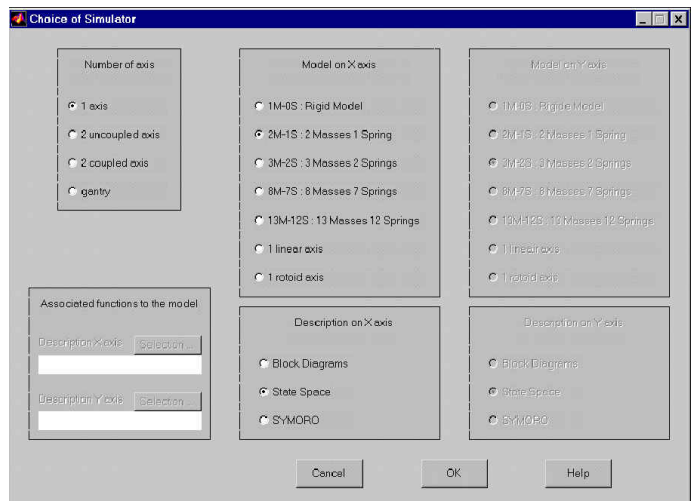## II. DESCRIPTION OF **SICOMAT**

A. Main menu of the simulator

Fig. 1. Main window of the simulator

SICOMAT is designed to easily set up the simulation environment (mechanical model, CNC, actuators and sensors model) as well as the parameter values, such as the number of masses and springs for the mechanics, the geometrical and inertial parameters, the representation of the mechanical structure with linear state-space model or with simulink blocks or with robotics formalism [2]. The user may select the desired components in a data base that contains commonly used motors, gearboxes, transmissions and so on, or define his own components. The electric actuators

dynamics and saturations, the effect of delays and quantization due to CNC are also considered [3]. The software is entirely boxes-driven and enables the user to design and modify a virtual prototype and to easily check its performances in a CAD approach without the need of the real prototype. The main window of the simulator is shown on Figure 1.

B. Trajectory generator

Several mechanical models are implemented in the library of **SICOMAT**. They are chosen through 3 boxes, Figure 1. The box 'Number of axis' allows to choose machines with one or two decoupled linear axis, or with two coupled axes. The boxes 'Model on X axis' and 'Model on Y axis' define the number of masses (M) and springs (S) of the lumped model for each axis respectively. The boxes 'Description on X axis' and 'Description on Y axis' allow to choose between a block diagram or a steady state-space or a robotic formalism description of the mechanical model. In that last case, some parameters of the description are defined in 2 files whose names are given in the box 'Associated functions to the model'. Clicking on the 'OK' button runs the automatic calculation of the Simulink blocks which are linked together and associated with Matlab routines to simulate the system while tracking different trajectories (straight line, circular interpolation,), to perform frequency and time analysis and to display results. An example of a single axis is represented on Figure 2.
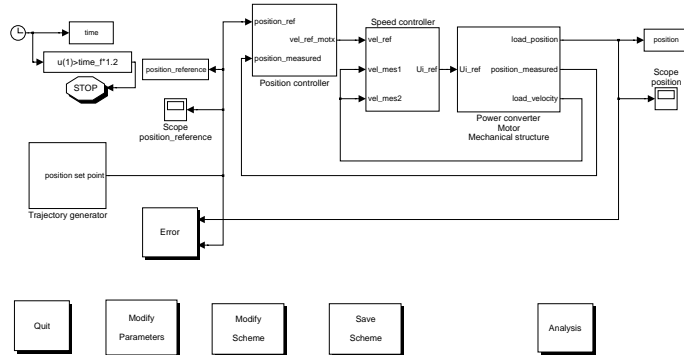

Fig. 2. Example of a single axis

C. Modeling of the mechanics

C.1 Description of the system

The example of a machine composed of a motor, a transmission gearing with two driving wheels and a belt is given to illustrate the modeling methods, Figure 3. The motion transformation from rotation to translation is achieved with a rack and a gearwheel. In this example, the model is simplified to 2 Masses and one Stiffness (2M1S), but more accurate models are available in the simulator.

The following notations are used in Figure 3: Γ_{mot} is the driving torque. J_m, θ_m are the inertia and the angular position of the rotor of the motor respectively. J_{pm}, θ_{pm}, x_{pm}, R_{pm} are the inertia, the angular and linear positions
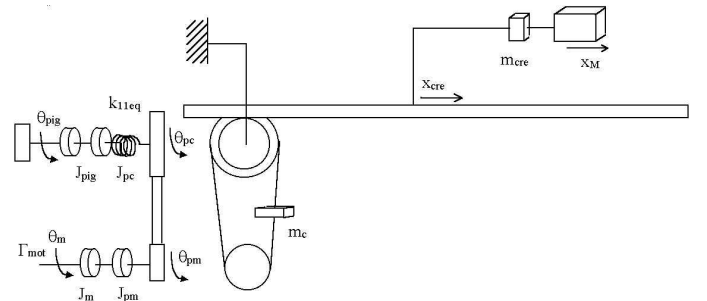

Fig. 3. An example of a machine axis

and the radius of the driving wheel supported by the motor shaft respectively, with:

$$x_{pm} = R_{pm}\theta_{pm} \tag{1}$$

J_{pc}, θ_{pc}, x_{pc}, R_{pc} are respectively the inertia, the angular and linear positions and the radius of the driving wheel supported by the rack and gearwheel, with :

$$x_{pc} = R_{pc}\theta_{pc} \tag{2}$$

$n_b = \frac{R_{pc}}{R_{pm}}$ is the ratio of the belt transmission, m_c is the mass of the belt. J_{pig}, θ_{pig}, R_{pig} are respectively the inertia, the radius and the angular position of the gearwheel. k_{11eq} is the global torsion stiffness between θ_m and $n_b\theta_{pig}$. x_{cre}, m_{cre} are respectively the linear position and the mass of the rack. x_M stands for the load position. M_{load} is the load mass.

Internal material damping, viscous and Coulomb friction forces are also considered: μ_{11eq} is the global material damping coefficient between θ_m and $n_b\theta_{pig}$, μ_{12eq} is the global material damping coefficient between x_{pc} and x_M, f_{11eq} is the global viscous friction coefficient between the load and the motor.

C.2 Linear state space model

The state space model can be obtained from the Lagrangian equation as recalled here:

$$M\ddot{X} + \mu\dot{X} + KX = F \tag{3}$$

X is the (nx1) vector of degrees of freedom of the system, n is the number of degrees of freedom F is the vector of driving forces, μ is the damping matrix, M and K are the mass and stiffness matrices respectively. In the case of the example, (3) is written with:

$$x = \begin{pmatrix} \theta_m \\ x_M \end{pmatrix}, F = \begin{pmatrix} \Gamma_{mot} \\ 0 \end{pmatrix} \tag{4}$$

$$M = \begin{pmatrix} J_{11eq} & 0 \\ 0 & J_{12eq} \end{pmatrix} \tag{5}$$

$$M_{12eq} = M_{cre} + M_{load} + \frac{J_{pig} + J_{apig} + J_{pc}}{R_{pig}^2} \tag{6}$$

$$J_{11eq} = J_m + J_{am} + J_{pm} + m_c R_{pm}^2 \tag{7}$$

J_{am} is the inertia of the motor shaft.

$$\mu = \begin{pmatrix} \mu_{11eq} + \frac{f_{11eq}}{n_b^2}R_{pig}^2 & -\frac{f_{11eq}}{n_b}R_{pig} \\ -\frac{f_{11eq}}{n_b}R_{pig} & \mu_{12eq} + f_{11eq} \end{pmatrix} \quad (8)$$

$$K = \begin{pmatrix} \frac{k_{11eq}}{n_b^2}R_{pig}^2 & -\frac{k_{11eq}}{n_b}R_{pig} \\ -\frac{k_{11eq}}{n_b}R_{pig} & k_{11eq} \end{pmatrix} \quad (9)$$

Multiplying (3) by the inverse of the mass matrix leads to:

$$\ddot{X} = M^{-1}F - M^{-1}\mu\dot{X} - M^{-1}KX \quad (10)$$

And the state equation follows:

$$\dot{z} = Az + BF \quad (11)$$

Where $z = \begin{pmatrix} \dot{X} \\ X \end{pmatrix}$ is the state vector and:

$$A = \begin{pmatrix} 0_n & I_n \\ -M^{-1}K & -M^{-1}\mu \end{pmatrix}, \ B = \begin{pmatrix} 0_n \\ M^{-1} \end{pmatrix} \quad (12)$$

I_n is the (nxn) identity matrix. 0_n is the (nxn) matrix of zeros.

C.3 Simulink block diagrams

This description uses the elementary block diagrams of the software Simulink. The user draws the model just as with pencil and paper by using click-and-drag mouse operations. This method avoids to express differential equations in a language or program. Simulink includes a comprehensive library of sinks, sources, connectors, linear and nonlinear components. This approach provides a graphical insight which can facilitate the understanding of the model, but it is limited, for sake of simplicity, to models from the rigid one to the 8 Masses and 7 Springs models. The block diagrams of the example (3), (4), (5), (6), (7), (8), (9), equivalent to the state equation (11), is given on Figure 4.
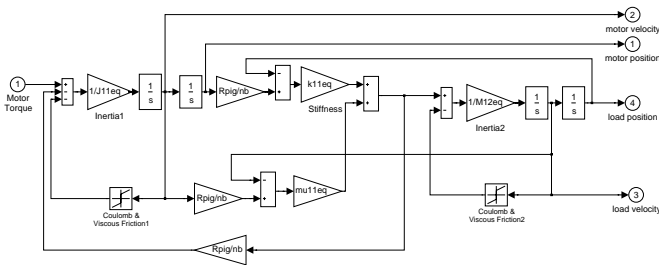

Fig. 4. Example 2m1s implemented as a Simulink block diagram

C.4 Robotics formalism

Another aspect of **SICOMAT** is to propose an original method of description based on robotics formalism. [2] proposes to adapt methods, which are derived from robotic in order to provide systematic and automatic model for system with lumped elasticities. One main application of the method concerns high-speed machine tools and robots with elastic joints. The method can provide the kinematic and dynamic models of such systems. To achieve this goal the method adapts some well-known tools and notations, which are widely used for rigid robots. The technique consists on using a description based on the modified Denavit-Hartenberg (MDH) [5] to define the kinematic of the system. Given the robot geometric parameters and its dynamic parameters, the geometric, the kinematic and the dynamic models are calculated automatically by the **SYMORO** (**SY**mbolic **MO**delling of **RO**bots) software package [6]. This software provides the different models under C or **MATLAB** code useable directly in the **MATLAB/Simulink** environment. It should be noticed that for identification purpose, the standard dynamic model can be rewritten in terms of a minimal set of parameters called the base inertial parameters which are the only parameters that can be identified [7]. These parameters can be obtained from the classical inertial parameters by eliminating those, which have no effect on the dynamic model and by regrouping some others. This procedure reduces the complexity of the dynamic model and will be used systematically hereafter in order to decrease the simulation time. Using the base parameters **SYMORO** calculates the minimal dynamic model used for simulation or control.

D. Electrical drive modeling

Most of actual machine tool actuators are synchronous motors with PWM converters and include current loop with field oriented control in the d-q frame. Thus it is assumed that the current loop can be modeled on q axis by a first order transfer function, between the current setpoint and the driving torque. The saturation of the current reference and the supplied voltage are simulated taking into account the effect of the electromotive force (back emf E) [3].

E. Computer Numerical Control (CNC)

The controllers implemented in industrial CNC are usually based on a cascade structure with a fast inner loop for torque control and outer loops for speed and position control. This structure improves the robustness of the controller with respect to the disturbances and modeling errors. The speed control loop is classically implemented with proportional and integral (PI) controllers (PI or IP structure) with anti-windup strategy. The position loop is designed with a P controller. In order to decrease the tracking errors, velocity and acceleration feedforward signals might be added to the control loops. Several aspects of the control are considered in the simulator such as continuous or discrete time control laws, poles placement method, time or frequency approaches.

III. Analysis and control tuning of a machine with SICOMAT

Four boxes are all user-selectable on Figure 2.
■ Modify Parameters

The drive chain of a machine tool usually contains a motor, a drive pulley or a gearbox and a transmission, which can be a ball-screw gear, a drive pulley or a rack and gearwheel, and a load. Manufacturer's data are given in files regrouped in a database. The names of the files needed for a model are given and can be modified with this function.

■ Modify Scheme

The main simulator, Figure 2, is automatically built from four elementary Simulink schemes given in a library: the trajectory generator, the power converter, the controller and the mechanical structure. These blocks can be changed to modify the simulator, taking new ones from the library.

■ Save Model box

Any new or modified simulator can be saved as a new simulator.

■ Analysis box

This box is dedicated to analyze the flexible modes of the mechanical structure, to reduce the model order, to compute the time and frequency responses and to optimize the controller tuning. Clicking on this box opens the following menu:
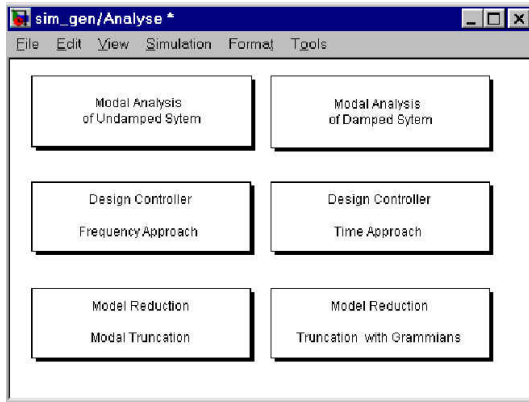

Fig. 5. Functions of the Analysis box

A. Modal analysis of the System

The "undamped system" function calculates the natural frequencies as the eigen values of $(M^{-1}K)$ which defines the undamped system ($\mu = 0$ in (3)). The "damped system" function calculates the damping coefficients and the natural frequencies of the full implemented mechanical model calculated from the 'linmod' Matlab function [8]. The components of the eigen vectors are also calculated to be interpreted as the contribution of each degree of freedom on the system. This is a help to see the dominant modes of the model and to choose the proper bandwidth for the controller.

B. Controller tuning

B.1 Initial tuning on a simplified rigid model

The user problem is then to tune the controller gains considering both the CNC components and the mechanical structure. A significant point of **SICOMAT** is to propose an automatic tuning procedure composed of two steps. Firstly gain values are calculated using a simplified rigid model. Secondly these values initialize an optimization procedure which uses the complete model implemented in the simulator to perform an accurate tuning. There are two basic approaches of designing the digital controller: the direct discrete-time design or the continuous controller redesign [9]. We choose the second one because it allows continuous time techniques to be used, specially the design in the frequency domain which ensures stability margins. Firstly, a continuous time control law is defined. Secondly it is discretized and then it is approximated by continuous transfer function, such that the gains can be calculated with a continuous controller which takes into account the effect of digital control in the bandwidth of the closed loop. Details can be found in [3], [4]. At first, the gains K_p, T_i of the speed loop,are tuned in order to get a phase margin φ_{mv} at a frequency ω_{vit} which depends on the frequency f_{meca} of the first flexible mode of the process (usually $\omega_{vit} = 2\pi f_{meca}/(4 \ to \ 10)$). Then the gain K_v of the position controller is calculated in order to get a phase margin φ_{mp}. The tuning must lead to a behavior for the speed loop similar to the Figure 6. $T_{bov}(s)$ is the open loop velocity transfer function.
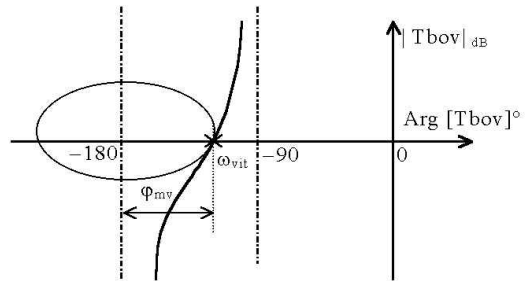

Fig. 6. Phase margin of the speed loop with the rigid model and the IP controller

B.2 Accurate tuning of the controller taking into account the complete mode

These values initialize next an optimization procedure using the full model to perform an accurate tuning. At this stage, the complete model implemented in the simulator is taken into account: lumped model of the flexible mechanics, digital control and location of measurements. The 'dlinmod' Matlab procedure automatically calculates high order discrete transfer functions which makes impossible to get analytical expressions for the gains. They are calculated with optimization techniques, using the procedure 'fmins', based on the Nelder-Mead simplex method [8] in order to satisfy the phase margins and bandwidths. In case of the complete model, the delays and the oscillatory poles and zero introduced by the numerical control and the flexible mechanics make the gains tuning very delicate. Too small gain margin may be obtained in spite of a good phase margin(typically 6dB gain margin is needed). The tuning procedure consists then in computing again the gains by optimization with gain margin criterion, starting from the values obtained with the phase margin criterion.

B.3 A time approach design

As an alternative to the automatic and optimized gain tuning, a time approach proposed by NC manufacturers is implemented in **SICOMAT**. The method consists on simulating step responses tuning the parameters of the controller successively. At each iteration, the simulator performs a simulation with the new values of the parameters of the controller. The first step is the tuning of the speed loop. The integral gain is fixed to a large value (0.5s) in order to move the integral effect to low frequencies, outside off the velocity bandwidth. The proportional gain is adjusted running simulations of the velocity closed loop in order to get a step response overshoot close to 10%. Then the integral gain is adjusted to get a step response overshoot between 20% and 40%.The second step is to adjust the proportional gain of the position controller in order to get a desired overshoot. At each iteration **SICOMAT** provides numerical information on the time rising and the overshoot.

C. Model reduction

The mathematical models obtained with **SICOMAT** are often highly order because they take account the dynamics of rigid bodies, the elasticity of bodies and mechanical transmission, the dynamics of electric actuators, the effect of numerical control as delays. One difficulties in designing controller for complex systems lies in the problem to represent a complex model by a equivalent model which has a smaller number of degrees of freedom (dof). Three methods are implemented in **SICOMAT**:

1. The first one is based on the calculation of the eigenvalues and eigenvectors of A matrix 12, which are sorted in order to cancel the m fastest modes of A with matching static gain. The state of the reduced model has lost its physical meaning with respect to the dof defined by the initial state vector.

2. The second one uses the partial Gramian operators in order to take into account some controllability and observability aspects of the system [10][11][12][13]. This method keeps the drawback of the interpretation of the dof of the reduced model.

3. The aim of the third one is to keep the most significant dof of the reduced model among those of the initial state vector X of (3), as a result of a technological analysis of the mechanical system. The reduced model is then calculated by regrouping serial or parallel masses and stiffnesses of the full model with very simple rules. This method is quite less accurate than the first two ones from a system analysis point of view, but keeps an interesting physical meaning.

IV. ANALYSIS AND SIMULATION OF A 5M4S MODEL

A. Description of the system

SICOMAT has been devoted to a large number of industrial cases. As an example, a model composed of 5 masses and 4 springs (noted 5M4S) will be considered in this paper (Figure 7). This model represents a single axis composed of a bed, a double linear synchronous motor, a linear scale and a load, which may be moved on a perpendicular axis. The two motor's primary are mounted like a gantry type, but very close side to side on a very rigid structure in order for the gantry to be considered as only one equivalent motor. The primary part is a short
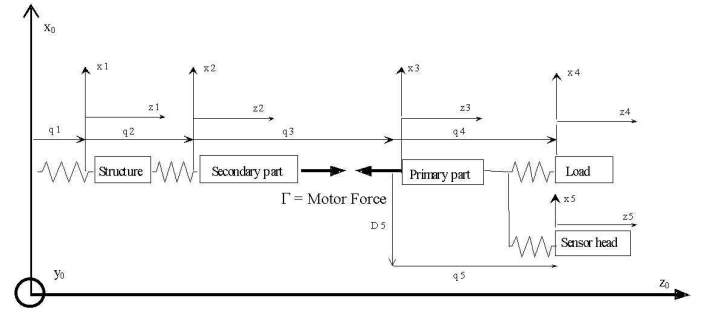

Fig. 7. 5M4S Model for a high speed machine tool axis

moving coil assembly and the secondary part is a long permanent magnet assembly. The notations used on Figure 7 are: Γ is the driving force, $R_o(O_o, x_o, y_o, z_o)$ is the reference frame, $M_{structure}$ and $k_{structure}$ are respectively the mass of the structure and its stiffness w.r.t. the bed, $m_{secondary}$ and $k_{secondary}$ are respectively the mass of the secondary part of the linear motor and its stiffness w.r.t. the structure, $m_{primary}$ is the mass of the primary part of the linear motor, m_{sensor} and k_{sensor} are respectively the mass of the linear scale sensor head and its stiffness w.r.t. the primary part, M_{load} and k_{load} are respectively the mass of the load and its stiffness w.r.t. the primary part. Shortly, the lumped model is composed of: 5 masses $M_{structure}$, $m_{secondary}$, $m_{primary}$, M_{sensor}, M_{load} and 4 stiffnesses $k_{structure}$, $k_{secondary}$, k_{sensor}, k_{load}.

We will also consider internal material damping, viscous friction coefficients and Coulomb friction forces: $\mu_{structure}$ is the material damping coefficient of the structure, $\mu_{secondary}$ is the material damping coefficient of the secondary part of the linear drive, μ_{sensor} is the material damping coefficient of the load, μ_{load} is the material damping coefficient of the sensor head support. F_{vmotor} is the viscous friction coefficient between primary and secondary part of the linear drive, $F_{Coulombmotor}$ is the Coulomb friction force between primary and secondary part of the linear drive.

These simulations are carried out with the following parameters for the 5M4S model: $M_{structure} = 3667kg$, $m_{secondary} = 232kg$, $m_{primary} = 92.305kg$, $M_{sensor} = 270.2kg$, $M_{load} = 0.008kg$, $k_{structure} = 1e9N/m$, $k_{secondary} = 1e12N/m$, $k_{sensor} = 3.8e8N/m$, $k_{load} = 1e7N/m$, $\mu_{structure} = 19.15N/(m/s)$, $\mu_{secondary} = 107.70N/(m/s)$, $\mu_{sensor} = 4.6548N/(m/s)$, $\mu_{load} = 0.0089N/(m/s)$, $F_{vmotor} = 88.1N/(m/s)$, $F_{Coulombmotor} = 142N$

F_{vmotor} and $F_{Coulombmotor}$ are issued from identification results. $\mu_{structure}$, $\mu_{secondary}$, μ_{sensor} and μ_{load} are given by technological datasheets.

B. Mechanical flexible mode analysis

The frequency analysis toolbox gives the mechanical frequencies, the pulsation and the damping coefficients of the system. We observe in this example (Figure 8) a first mode near from $f_{meca} = 80Hz$. The bandwidths of velocity and position loop can be then respectably choose equal to $\omega_{vit} = 125.6 rad/s$ ($2\pi f_{meca}/4$) and $\omega_{pos} = 31.4 rad/s$ ($2\pi f_{meca}/16$).

Fig. 8. Frequency analysis of the system

C. Controller tuning

The velocity phase margin φ_{mv} of each axis is chosen equal to 45° with a PI structure. The frequency and time responses of the velocity loop are given in Figure 9.
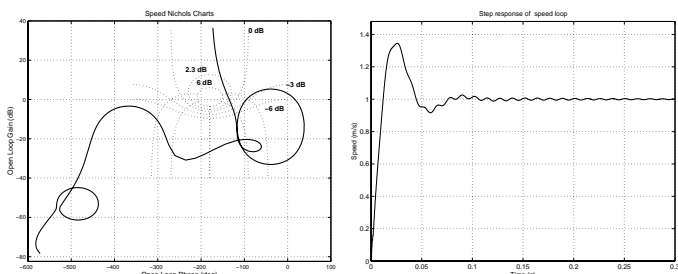

Fig. 9. Frequency and time responses of the speed loop

The different loops on Nichols diagrams of the Figure 9 are due to the poorly damping modes of the process. The phase margin φ_{mp} of the position loop is equal to 80°. The frequency and time responses of the position loop are given in Figure 10.
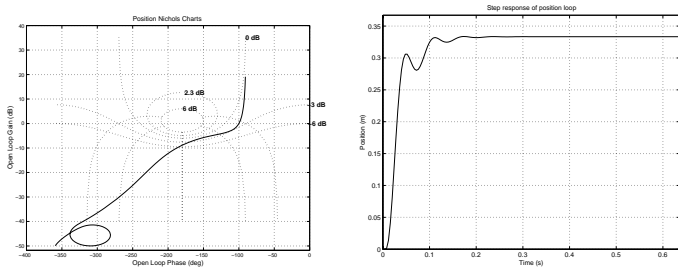

Fig. 10. Frequency and time responses of the position loop

The controller tuning is performed in order to achieve the desired phase margins. By selecting in the design controller toolbox, the frequency optimization procedure described in the Section 3.2.2, the following parameters are computed in Table I. The performances obtained with this set of parameters, in term of bandwidth are a phase margin for

the speed loop of $\varphi_{mv} = 45°$ with a bandwidth of $\omega_{vit} = 126\ rad/s$. For the position loop, we obtained $\varphi_{mp} = 80°$ with a bandwidth of $\omega_{pos} = 33.5\ rad/s$.

Parameters	$K_p\ (s^{-1})$	$T_i\ (s)$	$K_v\ (s^{-1})$
Initial tuning	94.26	0.009	24.8
Optimized tuning	97.1	0.009	30.26

TABLE I
DESIGN OF THE CONTROLLER

V. CONCLUSION

This paper presents the software package **SICOMAT**, which is devoted for the simulation and the prediction of the closed loop performances of high-speed machine tool axis. The simulations are carried out in the **Matlab/Simulink** environment. Several toolboxes give the user the opportunity to completely model and analyze its process. The most important contribution for the modeling is the mechanical modeling, which is based on a multibody structure model with lumped masses and elasticities. The analysis is performed through the calculation of the mechanical flexible modes, the controller tuning and the computation of the dynamic behavior. Classical machine tool axes are implemented in a database or can be easily created by the user. Future works will concern the extension of this work for any n axes.

REFERENCES

[1] P. J. Barre, *Control Strategies For A Numerical Machine Tool Axis In Case Of Very High Speed Machining*, Ph D. Thesis, E.N.S.A.M. Paris, 1995.

[2] W. Khalil, M. Gautier, "Modeling of mechanical systems with lumped elasticity," *Proc. IEEE Conf. On Robotics and Automation*, San Francisco, USA, 2000.

[3] P. Poignet, M. Gautier ,W. Khalil, "Modeling, Control and Simulation of High Speed Machine Tools Axes," *IEEE /ASME International Conference on Advanced Intelligent Mechatronics*, Atlanta, USA, pp. 617–622, 1999.

[4] M.T. Pham, P. Poignet, M. Gautier, "Automatic Tuning of Cascade Structure CNC Controllers," *6th International Workshop on Advanced Motion Control*, Nagoya, Japan, pp. 390-395, 2000.

[5] W. Khalil, J.-F. Kleinfinger, "A New Geometric Notation for Open and Closed-Loop Robots," *Proc. IEEE Conf. On Robotics and Automation*, San Francisco, USA, pp. 1174-1180, 1986.

[6] W. Khalil, D. Creusot, "A System for the Symbolic Modeling of Robots," *Robotica*, vol. 15, pp. 153-161, 1997.

[7] M. Gautier, "Numerical calculation of the base inertial parameters," *Journal of Robotics Systems*, vol. 8, No 4, pp. 485-506, 1991.

[8] MATHWORKS, *Matlab and Simulink Reference Guides*, The Mathworks, Inc. Natick, Mass., USA, 1998.

[9] F.L.L. Lewis, *Applied Optimal Control and Estimation*, Prentice Hall, 1992.

[10] T. Cambois, *Simplification de modèles de systèmes mécatroniques en vue de la commande. Application aux suspensions actives pour l'automobile* , Ph. D. Thesis, Université de NANTES-Ecole Centrale de NANTES, 1998.

[11] M. G. Safonov, R. Y. Chiang "A Schur method for balanced-truncation model reduction," *IEEE Trans. Automat. Contr.*, vol. 34, No 7, p. 729-733, 1989.

[12] K. Glover, "All optimal Hankel norm approximations of linear multivariable systems and their L^{∞} error bound," *Int. J. Contr.*, vol. 39, No. 6, pp. 1115-1193, 1984.

[13] B. C. Moore, "Principal component analysis in linear systems: Controllability, observability, and model reduction," *IEEE Trans. Automat. Contr.*, vol. 26, pp. 17-31, 1981.