Proceedings of the XII International Symposium on Dynamic Problems of Mechanics (DINAME 2007),
Varoto, P. S. and Trindade, M. A. (Editors), ABCM, Ilhabela, SP, Brazil, February 26 - March 2, 2007

Simulation of systems parametrically dependent on the spatial configuration: a mechatronic approach

Maíra Martins da Silva, Wim Desmet and Hendrik Van Brussel

Katholieke Universiteit Leuven / Department of Mechanical Engineering – Celestijnenlaan 300B B-3001 Leuven Belgium

Abstract: In mechatronics, an optimal design can only be accomplished if the interaction between the control system and the structural dynamics is considered in an early design phase. Often, mechatronic systems have varying spatial configuration, such that their dynamic behavior, described by their most significant eigenfrequencies and mode-shapes, may vary. This will inevitably affect the performance and the stability of the control design. This behavior can be predicted using a flexible multibody model, which depends parametrically on the configuration. This paper presents a methodology to derive and simulate such a model, concurrently with its controllers, in a co-simulation scheme. The subsystem with configuration-dependent dynamics is modeled as a parameter-dependent linear model. This model and the controllers are implemented in Matlab/Simulink, whereas the model of the flexible multibody system is implemented in LMS Virtual.Lab Motion. For each integration step, the parameter-dependent linear model is re-evaluated, depending on the parameter. The proposed methodology has been applied to an industrial pick- and-place machine which has configuration-dependent dynamics. Experimental validation showed a confident correlation between experimental and simulated data. Based on simulations, motion and vibration controllers were designed. The control scheme adopted was a high-authority motion controller built around a low-authority vibration controller. The methodology proposed enables the evaluation of the controller not only in frequency domain but also in time domain and can be applied in an optimization procedure considering concurrently structural and control parameters.

Keywords: LPV models, model reduction, co-simulation of structure and controller

NOMENCLATURE

A,**B**,**C**,**D** = matrices of the state space model
C = damping matrix
f= analytical function
f = force
I = identity matrix
K = stiffness matrix
l = length of the beam
l = vector of varying parameters (configuration)
L = input influence matrix
M = mass matrix
p = poles of the system
q = physical degrees of freedom
u = input vector of the state space system
x = states of the system
y = output vector of the state space system
z = zeros of the system

Greek Symbols

ξ = modal damping
μ = modal mass
η = vector of modal parameters of the reduced model
θ = vector of the amplitudes of the rigid modes
η^ι = vector of the amplitudes of the internal flexible modes
Ψ = modal transformation matrix
Φ = kept normal modes
Ω = modal frequencies

Subscripts

ι associated with the internal flexible modes
c constrained modes
k kept modes
n number of the poles
s subsystem
u input of the state space model
y output of the state space model
1 2 subscript of the poles of the system in crescent order or subscript of the A matrices constants of the affine model
0,1 relative to the constant to create the affine model using poles, zeros and gains

INTRODUCTION

Active vibration control schemes are usually implemented in order to enhance the dynamic performances beyond the first resonance frequency of mechatronic systems. Moreover, some mechatronic systems have varying spatial configuration, and their dynamic behavior, described by their most significant eigenfrequencies and mode shapes, may vary depending on its configuration. An example is a pick-and-place machine with a gripper carried by a flexible beam (Fig. 1). During the motion, the length of the beam and its dynamics may vary affecting the performance and the stability of the control system (Van den Braembussche, 1998). These changes should be taken into account during the controller design (Symens, 2004). During the design phase of a mechatronic system, this behavior can be predicted using a flexible multibody model, which depends parametrically on the configuration. This model can aid the designer to evaluate and optimize the system and its controllers before a physical prototype is available.

Furthermore, mechatronic design deals with the integrated design of a mechanical system and its embedded control system (Van Amerongen & Breedveld, 2003). An optimal design can only be accomplished if the interaction between the controller and the structural dynamics is considered in an early design phase of a mechatronic system. Several works were done concerning structure-control concurrently optimization (Reyer & Papalambros, 1999; Fathy et. al., 2001). De Fonseca (2000) optimized control parameters and structural design variables of the x-axis drive chain of a milling machine for eight extreme tool positions of the working volume.

This paper presents a methodology to model and simulate a mechatronic system with configuration-dependent dynamics, using co-simulation between LMS Virtual.Lab Motion and Matlab/Simulink. Using the hereafter proposed approach, different optimization methodologies can be applied concerning the mechatronic system not only in discrete positions but also in a continuous operation. Then, both time domain and frequency domain performance of the control system can be evaluated.

In general, the controller is designed using Matlab/Simulink, while the model of the flexible multibody system is implemented in the multibody environment. However, it is not possible to simulate a flexible subsystem with dynamics depending on the configuration on a flexible multibody environment such as LMS Virtual.Lab Motion. For instance, considering the pick-and-place machine, the flexibility of the beam depends on its length. As a result, a model to simulate this behavior requires an updating finite element model for each integration step or a sliding joint applied at the flexible body. Neither of these options is available at this time in the most widespread commercial multibody softwares. In fact, some integrated environments such as Samcef can support this kind of application. However, the aim of this work is to propose an alternative way for those that are using multibody environments based on Cartesian coordinates that can not handle this kind of application, for example, MSC.Adams and LMS Virtual.Lab Motion.

During the modeling procedure, the system is divided in two parts. The subsystem that has configuration-independent dynamics is modeled in a flexible multibody environment. Whereas, the subsystem with configuration-dependent dynamics can be modeled as a parameter-dependent linear model constructed in three steps:

1. Firstly, a parametric high-order finite element model is elaborated;
2. Then, local linear models are extracted at several reference configurations using a linear reduction technique;
3. Finally, the parameter-dependent state-space model is built by affine interpolation in the configuration space between poles, zeros and gains. This parameter-dependent linear model may be simulated using Matlab/Simulink which, for each integration step, revaluates the state-space model depending on the parameter. This parameter represents the spatial configuration of the systems.

The interface between these two virtual environments allows a concurrent simulation of the flexible-multibody model, the parameter-dependent model and the control system using one integration/solver procedure.

The presented methodology has been applied to model the X-motion of an industrial pick-and-place machine, which is described in the next section. Subsequently, the methodology and experimental validations are described. Based on this virtual model, motion and vibration controllers are designed to evaluate the integrated environment capabilities. Finally, some results and conclusions are addressed.

DESCRIPTION OF THE TEST CASE SET-UP

The test-case is an industrial 3-axis pick-and-place machine (Fig. 1). The Y-motion is gantry driven by two linear motors and the X-motion (over the carriage) is also driven by a linear motor. The vertical Z-motion is actuated by a rotary brushless DC-motor which drives a vertical beam through a ball screw/nut combination. The position of the linear motors and the length of the beam are measured with optical encoders and the acceleration of the end point of the beam in the X-direction is measured with an accelerometer.

The objective is to move the beam tip as accurate and fast as possible along a prescribed trajectory in the working area. However, fast movements of the linear motors will excite the eigenfrequencies of the flexible beam, which may vary during the movement since the length of the beam is continuously changed.

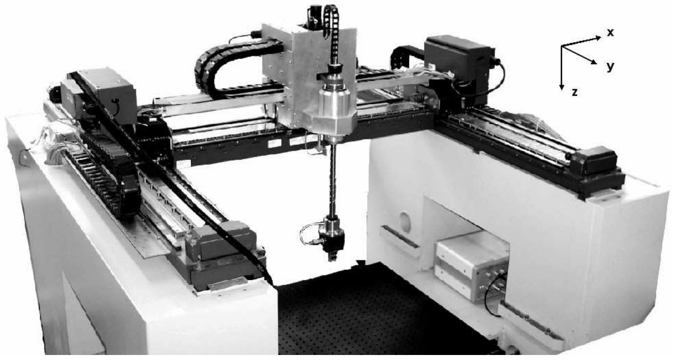


Figure 1 – Pick-and-place machine used as test-case

As mentioned before, the system can be divided in two parts: the subsystem with configuration-independent and the subsystem with configuration-dependent dynamics. The former is modeled in the multibody environment and comprises the frame, the two linear motors which drive the Y-motion, the carriage, their bushings and joints, while the latter is modeled as a parameter-dependent linear model and comprises the flexible beam and the linear motor which drives the X-motion. The modeling of the latter is addressed in the next section.

MODELING SYSTEMS PARAMETRICALLY DEPENDENT ON THE SPATIAL CONFIGURATION

Parameter-dependent linear models, also known as linear parameter varying (LPV) models, can be defined as linear systems whose describing matrices in state-space form depend on a vector of time-varying parameters, which can be measured (D'Angelo, 1970), such as described in Eq. (1).

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}(\mathbf{l})\mathbf{x} + \mathbf{B}(\mathbf{l})\mathbf{u} \\ \mathbf{y} &= \mathbf{C}(\mathbf{l})\mathbf{x} + \mathbf{D}(\mathbf{l})\mathbf{u} \end{aligned} \quad (1)$$

where **x** represents the states of the system, **u** and **y** represent, respectively, the inputs and the outputs and **l** represents a vector of varying parameters.

The LPV modeling approach follows the three-step methodology described before. Each step is described in more detail in the next sections.

Parametric finite element model

In order to have an affine LPV model, firstly, a high-order finite element model of the subsystem motor and vertical beam is created. In view of the fact that several discrete models for different lengths of the beam are necessary, a good approach to generate these finite element models is to create a parametric high-order finite element model. In this way, several finite element models can be generated automatically depending on the length of the beam. This can easily be done by changing the nodes coordinates according to the length of the beam, the mass of the clamped part and the stiffness of the clamping spring. The clamping is modeled as a linear stiffness.

Figure 2 shows a scheme of the parametric dependence of the model for two discrete positions: the maximum and the minimum lengths. *M* represents the motor mass, *x* the motor degree of freedom and *m* the equivalent mass of the part of the beam that is connected to the spring.

A model reduction technique is applied to these models in order to generate a suitable way to simulate this parameter-dependent model. The applied reduction technique is described in the next section.

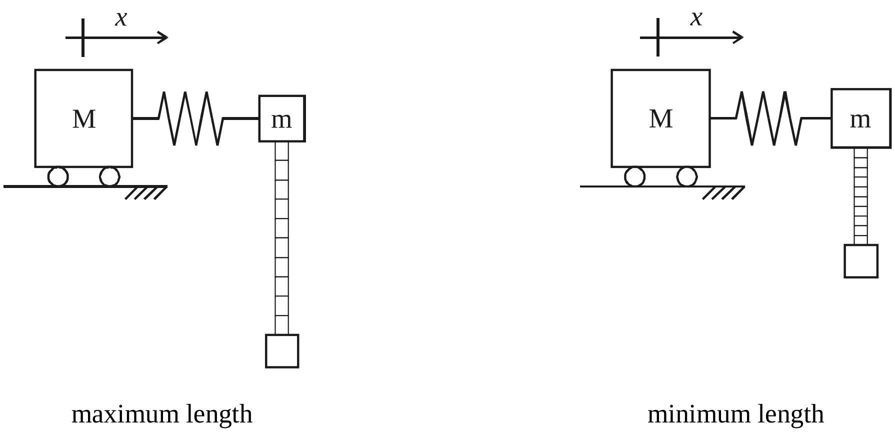


Figure 2 – Scheme of the parametric finite element model

Model reduction

Models of dynamic systems are useful primarily for two reasons: first for simulation and second for control. Any realistic model will have high complexity requiring many state variables to be adequately described. The resulting complexity is such that a simplification or model reduction will be needed in order to perform a simulation in an reasonable amount of time or for the design of a low order controller, which achieves desired objectives (Antoulas & Sorensen, 2001). Generally, for a particular kind of analysis, there is a small subset of the total number of degrees of freedom in a finite element model that are significantly active (Hintz, 1975). Consequently, there are ways to simulate complex structures using only a limited number of degrees of freedom.

Component mode synthesis (CMS) provides an appropriate solution for the reduction of a finite element model (Craig, 1981 and 1987). It is a form of substructure coupling analysis in which the dynamic behavior of each substructure is formulated as a superposition of modal contributions.

A multi-degree of freedom system can be modeled by a set of second order differential equations

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{L}\mathbf{f} \tag{2}$$

where **M** is the mass matrix, **C** is the damping matrix, **K** is the stiffness matrix, **q** are the physical degrees of freedom, **f** are the forces applied and **L** is the input force influence matrix, indicating the way the input forces act on the structure.

In any CMS technique, the displacements of the physical coordinate **q** are represented in terms of component-generalized coordinates $\boldsymbol{\eta}$ using the classical modal transformation:

$$\mathbf{q} = \boldsymbol{\Psi}\,\boldsymbol{\eta} \tag{3}$$

where the transformation $\boldsymbol{\Psi}$ consists of pre-selected component modes of the following types: normal modes, rigid body modes, constrained modes, attachment modes, inertia relief modes and inertia relief attachment modes.

There are combinations of these sets that generate a superposition of the modes, which is able to determine exactly the static response of a component submitted to external forces applied at the boundary nodes. Three sets statically complete can be defined: the Hintz's method of constrained modes superset, the Hintz's method of attachment modes superset, and the inertia-relief mode superset. The Hintz's method of constrained modes superset is defined by the rigid body modes, the constrained modes, and the inertia relief modes. The Hintz's method of attachment modes superset is defined by the rigid body modes, attachment modes, and the inertia relief modes. The inertia-relief mode superset is defined by the rigid body modes, and inertia relief attachment modes. These three static component mode supersets span the same subspace. The rigid body modes can be obtained as a linear combination of the constrained modes (Craig, 1987).

Any of these static component mode supersets may be supplemented by dynamic modes: fixed-interface, free-interface, or hybrid-interface defined by the normal modes. The well-known Craig-Bampton method only employs a static subset containing the constrained modes, but not including the inertia-relief modes; and fixed-interface normal modes to dynamically supplement it. Some other combinations can obtain more accurate results because the static component mode superset is not complete in this method. However, for the desired accuracy of the study case, an adequate reduced model can be generated using Craig-Bampton method.

So, the modal transformation $\boldsymbol{\Psi}$ used is defined by the Eq. (3), where $\boldsymbol{\Psi}_c$ are the constrained modes and $\boldsymbol{\Phi}_k$ are the kept normal modes.

$$\boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\Psi}_c & \boldsymbol{\Phi}_k \end{bmatrix} \tag{4}$$

Performing the modal transformation proposed by Eq. (3), the equation of motion can be written in modal coordinates.

$$\ddot{\boldsymbol{\eta}} + 2\boldsymbol{\xi}\boldsymbol{\Omega}\,\dot{\boldsymbol{\eta}} + \boldsymbol{\Omega}^2\boldsymbol{\eta} = \boldsymbol{\mu}^{-1}\,\boldsymbol{\Psi}_{\mathbf{u}}^T\,\mathbf{f} \tag{5}$$

where $\boldsymbol{\Omega}$ are the modal frequencies, $\boldsymbol{\xi}$ is the modal fraction of the critical damping, $\boldsymbol{\mu}$ is the modal mass, **f** is the input forces, $\boldsymbol{\Psi}_{\mathbf{u}} = \mathbf{L}^T\boldsymbol{\Psi}$ and **L** is the force influence matrix (Premount, 2002). The modal coordinates are now represented by the kept degrees of freedom, $\boldsymbol{\theta}$, representing the actuators and the modal coordinates $\boldsymbol{\eta}^{\boldsymbol{\iota}}$, Eq. (6).

$$\boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{\eta}^{\boldsymbol{\iota}} \end{bmatrix} \tag{6}$$

The output equation for a set of sensors, represented by the matrix $\mathbf{L}_{\mathbf{y}}$, can be defined by

$$\mathbf{y} = \mathbf{L}_{\mathbf{y}}^T\,\mathbf{x} = \mathbf{L}_{\mathbf{y}}^T\,\boldsymbol{\Psi}\,\boldsymbol{\eta} = \boldsymbol{\Psi}_{\mathbf{y}}\,\boldsymbol{\eta} \tag{7}$$

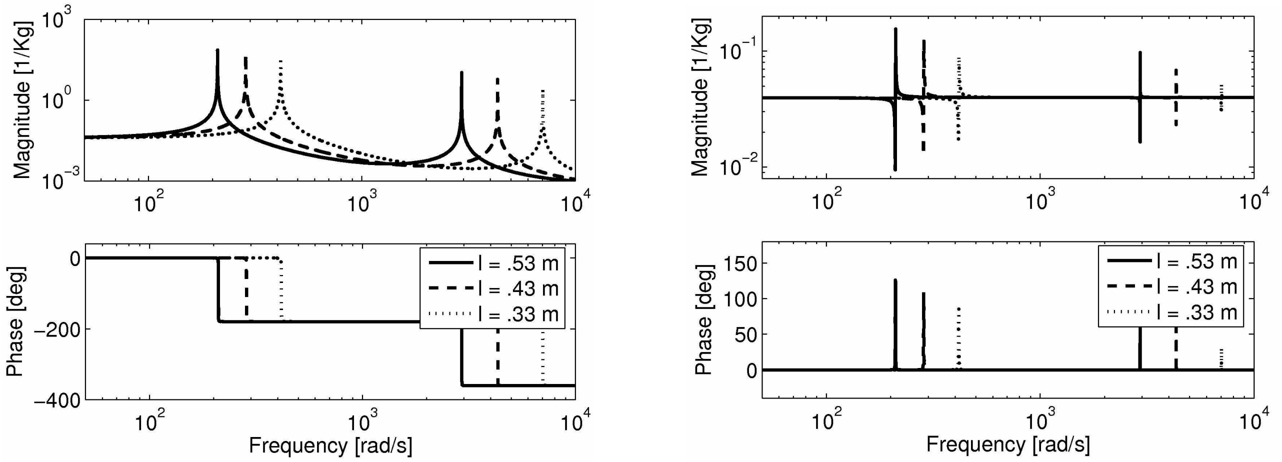
The state-space formulation can be written as Eqs. (8) and (9).

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\boldsymbol{\Omega}^2 & -2\boldsymbol{\xi}\,\boldsymbol{\Omega} \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\mu}^{-1}\,\boldsymbol{\Psi}_{\mathbf{u}}^T \end{bmatrix} \mathbf{f} \tag{8}$$

$$\mathbf{y} = \begin{bmatrix} \boldsymbol{\Psi}_{\mathbf{y}} & \mathbf{0} \end{bmatrix} \mathbf{x} \tag{9}$$

where $\mathbf{x} = \begin{bmatrix} \boldsymbol{\eta} \\ \dot{\boldsymbol{\eta}} \end{bmatrix}$ is the state variable.

This approach is performed for several models with different lengths of the beam. Each model has one degree of freedom for the actuator (linear motor), and 2 modal coordinates, representing the first and the second resonances. The model is a single input multiple outputs (SIMO) system, where the input is the force applied by the motor and the outputs are the acceleration of the beam tip and the acceleration of the motor. Figure 3 shows the frequency response functions (FRF) of the system for three different lengths (l = 0.53, 0.43, 0.33 m).



(a) **(b)**

Figure 3 – (a) FRF – Acceleration of the motor/force motor and (b) FRF – Acceleration of the beam tip/force motor for three different lengths of the beam

The local construction of the reduced-order model involves a computationally demanding numerical procedure. However, the variations of the model in the configuration space can be described by a simplified and approximated metamodel. In this paper, this metamodel is elaborated by selecting a set of discrete configurations, running a local reduction algorithm for each configuration and defining an approximated function which fits the model variation with the parameter variation. This has been implemented using affine models technique which is described in the next section.

Affine models – interpolation and simulation

The set of reduced models resulted from the procedure described in the previous section were affine using a linear interpolation technique proposed by Paijmans *et. al.* (2006) in order to create a parameter-dependent model. Experience shows that there are three criteria for a reliable interpolation technique: (1) a stability preserving interpolation is needed; (2) a performance preserving interpolation is needed and (3) the parameters of the interpolated controllers should be smooth and continuous functions of the varying parameter to avoid discontinuous controller states and output signals as a function of time.

The poles, zeros and gains are described by a linear affine interpolation dependent on the function of the scheduling parameter, $f(\mathbf{l})$. Equation (10) shows the technique for the poles vector.

$$\begin{pmatrix} p_1(\mathbf{l}) \\ p_2(\mathbf{l}) \\ \vdots \\ p_n(\mathbf{l}) \end{pmatrix} = \begin{pmatrix} p_{0,1} \\ p_{0,2} \\ \vdots \\ p_{0,n} \end{pmatrix} + \begin{pmatrix} p_{1,1} \\ p_{1,2} \\ \vdots \\ p_{1,n} \end{pmatrix} f(\mathbf{l}) \tag{10}$$

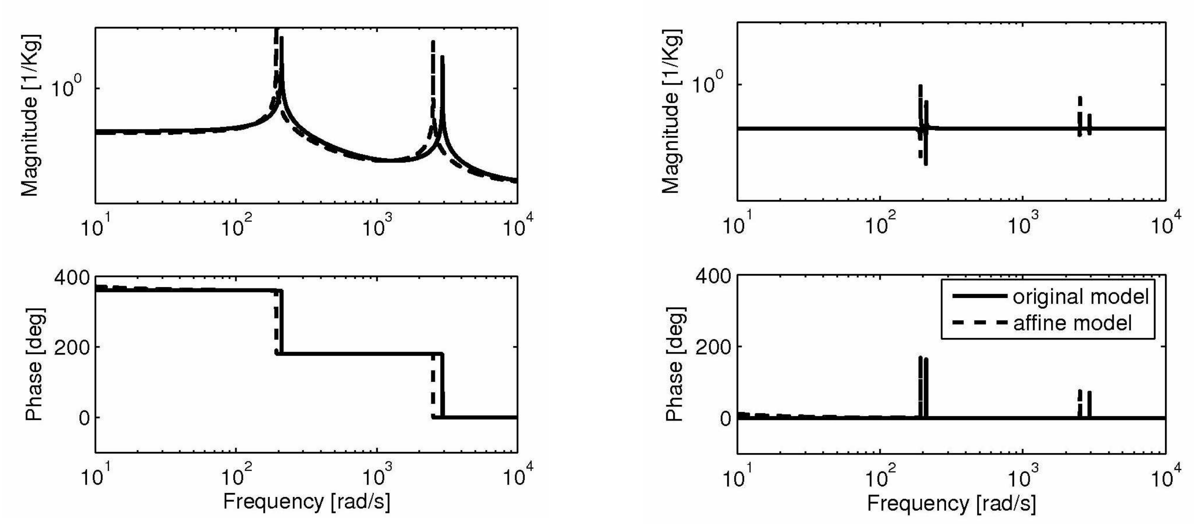
where p_1 till p_n are the poles of the system, $p_{0,1}$ till $p_{0,n}$ and $p_{1,1}$ till $p_{1,n}$ are constants and $f(\mathbf{l})$ is an analytical function of the scheduling parameter $\mathbf{l}$, e. g., the length of the beam. Similar affine functions have been derived to describe the varying zeros and gains considering the same function $f(\mathbf{l})$. The next step is to transform the affine functions of poles, zeros and gains into varying state-space matrices. Equation 11 shows the affine function for the A matrix of the state-space model.

$$\mathbf{A}(\mathbf{l}) = \mathbf{A}_0 + f(\mathbf{l}) \cdot \mathbf{A}_1 + f(\mathbf{l})^2 \cdot \mathbf{A}_2 \tag{11}$$

The main idea in this transformation is that a system defined by a gain, an amount of poles and zeros is equal to a concatenation of subsystems. These subsystems are defined by one pair of complex poles and one pair of complex zeros. The interconnection of these subsystems is performed by a concatenation of the states of the subsystems, resulting in a global affine parameter dependent state-space representation. Equation 12 illustrates how a subsystem (subscript *s*) defined by one pair of complex conjugated poles and one pair of complex conjugated zeros is transformed to state-space representation. More details on this methodology can be found in Paijmans *et. al*. (2006).

$$\begin{aligned}\mathbf{A}_s(\mathbf{l}) &= \mathrm{Re}\begin{bmatrix} p_i(\mathbf{l}) + p_{i+1}(\mathbf{l}) & -p_i(\mathbf{l})\cdot p_{i+1}(\mathbf{l}) \\ 1 & 0 \end{bmatrix} \quad B_s(\mathbf{l}) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ C_s(\mathbf{l}) &= \mathrm{Re}\begin{bmatrix} -z_i(\mathbf{l}) - z_{i+1}(\mathbf{l}) + p_i(\mathbf{l}) + p_{i+1}(\mathbf{l}) \\ z_i(\mathbf{l})\cdot z_{i+1}(\mathbf{l}) - p_i(\mathbf{l})\cdot p_{i+1}(\mathbf{l}) \end{bmatrix}^T \quad D_s(\mathbf{l}) = [1]\end{aligned} \tag{12}$$

This approach was used to build an affine model of the SIMO reduced model of the vertical beam including the linear motor inertia, described in the previous section. The system has one input: the force imposed by the linear motor in X-direction and two outputs: the accelerations of the linear motor and of the beam tip. In other to implement the methodology, this SIMO model was divided into 2 single input single output (SISO) models. The chosen analytical function was $\mathrm{f}(\mathbf{l}) = l$, the measurement of the length of the beam. Figure 4 shows the comparison between the original model and the affine model for a fixed length (0.53m).



(a) **(b)**

Figure 4 – Comparison between the original model and the affine model for l = 0.53m (a) FRF – Acceleration of the motor/force motor and (b) FRF – Acceleration of the beam tip/force motor

Several multibody environments have interface capabilities with Matlab/Simulink. For this reason, S-functions were implemented in order to run these affine SISO models. There are two inputs for the S-functions: the force of the linear motor and the length of the beam. For each integration step, the S-functions compile the ABCD-model from the affine model depending on the length. The S-functions, built in C MEX-file in order to speed up the simulation, are shown in the Fig. 5.

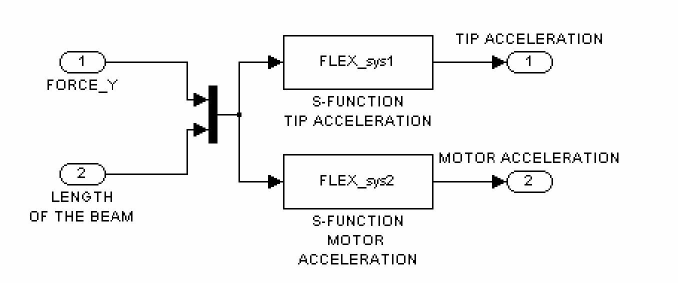


Figure 5 – Scheme of the S-function in the Simulink environment

CO-SIMULATION IN AN INTEGRATED ENVIRONMENT

LMS Virtual.Lab Motion was the multibody environment used to model the subsystem with configuration-independent dynamics (Fig. 6). The interface between Virtual.Lab Motion and Matlab/Simulink allows performing time domain simulations of arbitrary mechanisms (*plantout* in Fig. 7). The mathematical representation of the mechanism in Virtual.Lab Motion is fully nonlinear and can include rigid and/or flexible bodies connected by ideal joints. Internal or externally applied forces and time-dependent kinematics rules influence the motion of these joints and bodies. Virtual.Lab Motion provides libraries of predefined joint, driver and force elements. These modeling elements are automatically transformed into the appropriate set of nonlinear differential algebraic Newton-Euler equations of motion during the analysis. Simultaneous numerical integration of the states is performed by LMS Virtual.Lab Motion and Matlab. In this way, it is possible to simulate not only dynamic models such as described before but also control systems in Matlab environment.

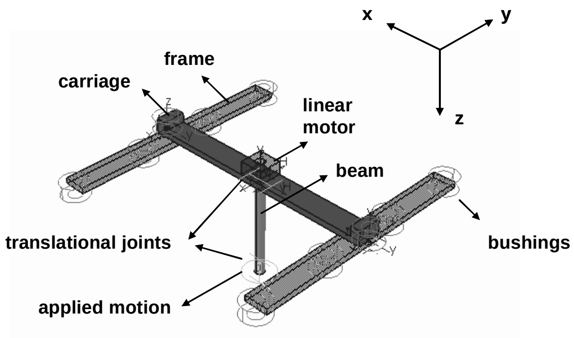


Figure 6 – Multibody model of the pick-and-place machine

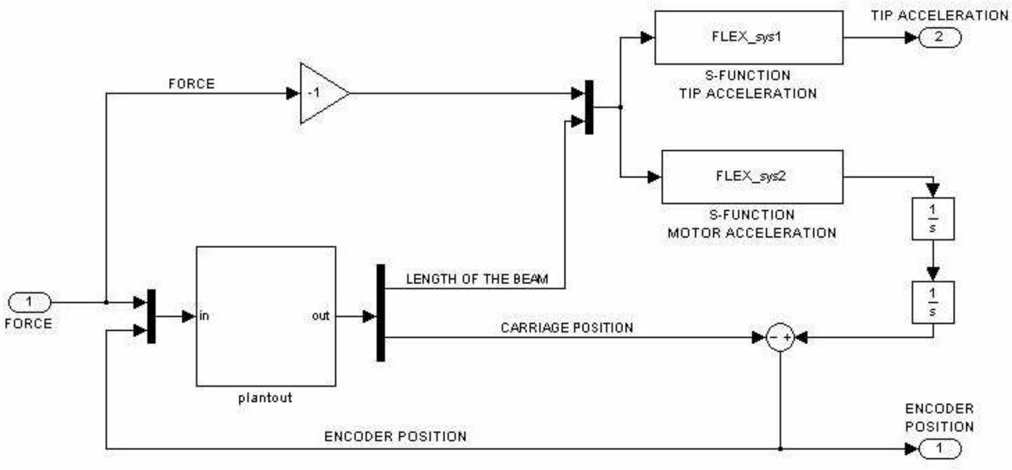


Figure 7 – Co-simulation between Matlab/Simulink and Virtual Lab Motion

The pick-and-place machine was modeled using rigid bodies, bushings, joints and forces to simulate the movement (Fig. 6). The vertical beam and the motor are fully modeled by the affine model in Matlab/Simulink environment, where the position of the motor and the acceleration of the beam tip are calculated.

In Virtual.Lab Motion environment, a sensor measures the length of the beam and this parameter is an input to the S-Function responsible for the ABCD models updating. A translational joint is applied between the carriage and the motor, and a position feedback controller is responsible for the motor position tracking in the V.Lab Motion. This keeps the subsystem motor and flexible beam at the correct place, which is quite important if the Y-motion is modeled.

A force generated by the motor is applied to the motor and to the carriage with opposite directions. The encoder of the motor measures the difference between the positions of the motor and the carriage, obtained in Matlab/Simulink and V.Lab Motion, respectively. Figure 7 shows the co-simulation between these two environments. The connection is made by the *plantout* which is a subsystem with the outputs and inputs from the Virtual Lab Motion and one S-Function that connects the environments.

RESULTS

The objective of the technique proposed here is to create an environment able to simulate mechatronic systems parametrically dependent on the spatial configuration and their controllers. Since the alternative to simulate the parameter-dependent system is to use Matlab/Simulink, the environment to simulate the controllers is ready to use. Figure 8 shows the FRFs obtained using the approach described for several fixed lengths. The simulated FRFs for a fixed length (l=0.43m) showed an acceptable correlation with the experimental FRFs until 1000 Hz (Fig. 9).

Since the approach generates a new state-space model for each time-integration step, it is possible to perform a trajectory varying the length of the beam. This enables to evaluate the performance of the system not only in the frequency domain but also in the time domain. An experimental input was applied to the model generating a trajectory response. The motor position (encoder) from this simulation is compared with the experimental data in Fig. 10. In this experiment, the length doesn't change significantly (about 20% of the total range). Therefore, the model is not validated for continuously modification. However, the varying dynamics depend quite smoothly on the length of the beam, which can assure a confident correlation between the model and the real system. A deadzone filter was used to the input data in order to simulate dry friction. The error presented by the results was about 0.5%.

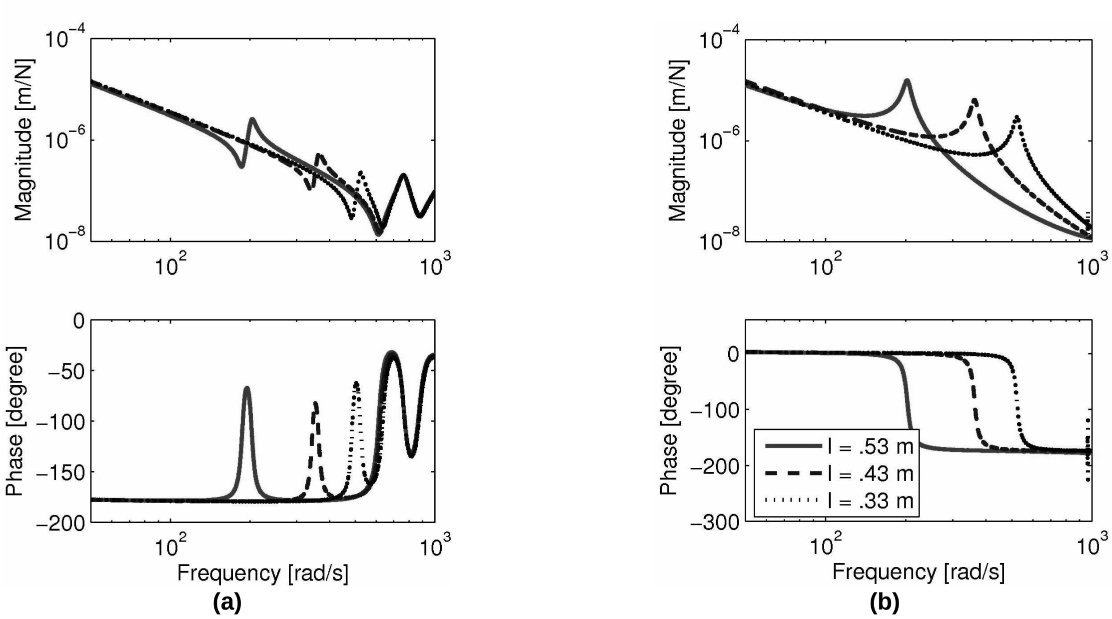


Figure 8 – (a) FRF – Encoder of the motor/Force Motor and (b) FRF – Acceleration of the beam tip/Force Motor

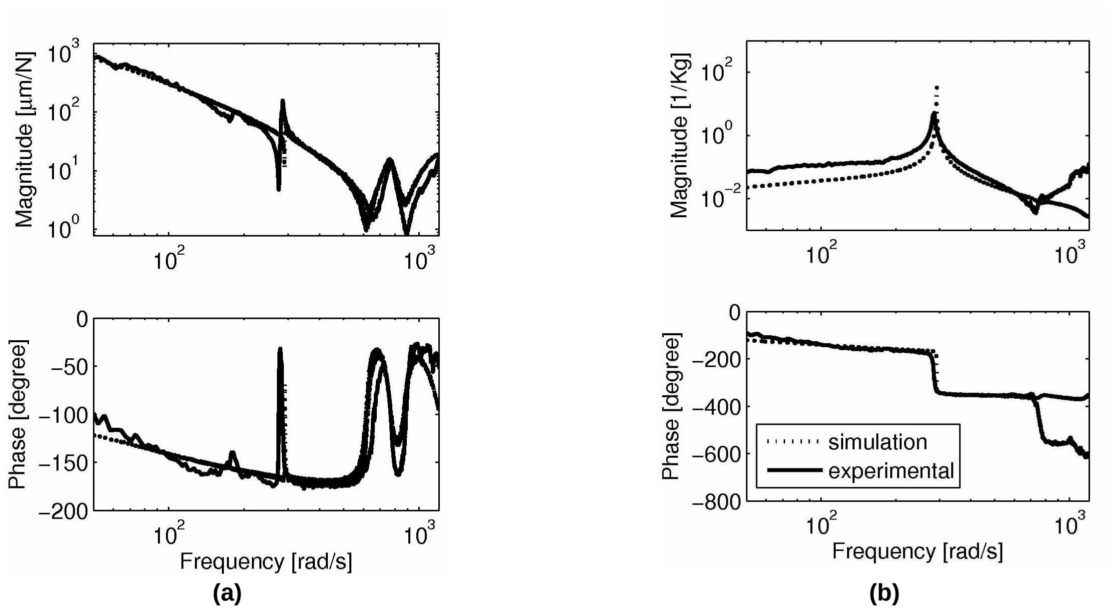


Figure 9 – Comparison between the simulation and the experimental FRF – length = -.43 m (a) Encoder: motor position (b) Accelerometer of the beam tip

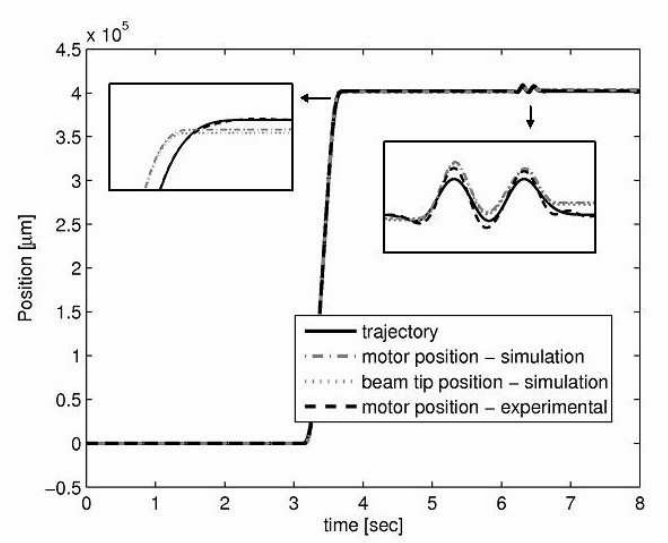


Figure 10 – Comparison between the simulated and the experimental motor position for the same input

In order to evaluate the methodology, a controller was designed and simulated. The methodology used to design the controller was proposed by Paijmans *et. al.* (2006). An appropriated way to design motion and vibration controllers is to design a high-authority motion controller (HAC) around the low authority vibration controller (LAC), Fig. 11. Details about the methodology applied to derive the controllers are described in Verscheure *et. al.* (2006). Fixed vibration controllers for different fixed configurations were derived based on H∞ mixed-sensitivity synthesis method for robust control loopshaping design. The chosen weights depend on the resonance frequency of the system. Consequently, several controllers were derived for different lengths of the beam. The vibration controllers derived from this approach were affine using the same technique described in the previous section. The result was a gain-scheduling controller, since it depends on a parameter. The motion controller was a standard PID controller fixed for all configurations. The beam tip acceleration for an input step and a fixed beam length is shown in Fig. 12. The residual vibration can be reduced tuning the PID controller, but there is a trade-off between vibration reduction and motion control. For high-bandwidth motion controller, the eigenfrequency of the beam is more excited. For low-bandwidth motion controller, the settling time is lower when used in combination with a vibration controller.

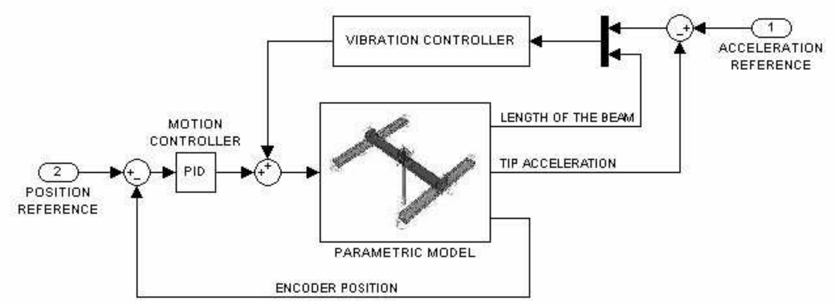


Figure 11 – Motion and vibration controllers using LAC-HAC structure

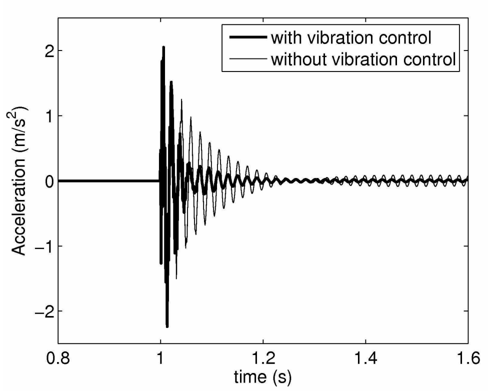


Figure 12 – Beam tip acceleration for a step input with and without vibration control

The objective of this paper is not to discuss a specific controller strategy for this pick-and-place machine, but to propose a convenient way to simulate a mechatronic system with configuration-dependent dynamics. In this way, the methodology proposed is reasonably generic and can be used in several applications.

CONCLUSIONS

During the design phase, a parametric flexible-multibody model containing structural dynamics and controllers is quite useful to aid the designer to infer about structural and control parameters. The procedure described in this work relies on the generation of affine metamodel of reduced FE models to describe the dynamics depending on the spatial configuration and the co-simulation of these affine models with a multibody system environment.

The advantage of this approach lays clearly in the use of user-friendly commercial software for multibody systems, LMS Virtual.Lab Motion, and the most widespread interface for control design, Matlab/Simulink. Moreover, using the proposed methodology, it is possible to simulate a mechatronic system which its dynamics depend on its configuration not only in discrete positions but also in continuous operations. This enables to evaluate the performance of the controller not only in the frequency domain but also in the time domain.

ACKNOWLEDGMENTS

The research work of Maíra Martins da Silva is supported by CAPES, Foundation Coordination for the Improvement of Higher Education Personnel, which is gratefully acknowledged. This work also presents research results of the Belgian Program on Inter-University Attraction Poles initiated by the Belgian Federal Science Policy Office (AMS IAP V/06).

REFERENCES

Antoulas, A.C. & Sorensen, D.C., 2001, ‘Approximation of large-scale dynamical systems: An overview”, Int. J. Appl. Math. Comp. Sci., vol. 11, pp. 1093—1121.

Craig, R.R., 1981, “Structural dynamics - An introduction to computer methods”, John Wiley & Sons, New York.

Craig, R.R., 1987, “A review of time domain and frequency domain component mode synthesis methods”, Journal of Modal Analysis, Vol. 2, No. 2, 59-72.

D’Angelo, H., 1970, “Linear time-varying systems: analysis and synthesis”, Allyn and Bacon Boston.

De Fonseca, P., 2000, “Simulation and Optimisation of the dynamic behaviour of Mechatronic Systems”. PhD Thesis, Katholieke Universiteit Leuven, Leuven (Belgium).

Fathy, H.K., Reyer, J.A., Papalambros, P.Y., Ulsoy, A.G., 2001, “On the coupling between the plant and the controller optimization problems”, Proceedings of the American Control Conference, Arlington, US, 2001, June 25-27, Arlington (2001), pp. 1864-1869.

Hintz, R.M., 1975, “Analytical methods in component modal synthesis”, AIAA J., Vol. 13, No. 8, pp.1007-1016.

Premount, A., 2002, “Vibration Control of Active Structures: An introduction”, 2nd Edition, Kluwer Academic Publishers.

Reyer, J.A., Papalambros, P.Y., 1999, “Optimal design and control of an electric DC motor”, Proceedings of the 1999 ASME Design Engineering Technical Conferences, September 12-15, 1999, Las Vegas, Nevada, DETC99/DAC-8599.

Symens, W.,2004, “Motion and vibration control of mechatronic systems with variable configuration and local non-linear friction, PhD Thesis, Katholieke Universiteit Leuven, Leuven (Belgium).

Paijmans, B., Symens, W., Van Brussel, H., Swevers, J., 2006, A Gain-Scheduling-Control Technique for Mechatronic Systems with Position-Dependent Dynamics American Control Conference, Minneapolis, Minnesota, USA, 2006 June 14-16, Paper ThB04.4.

Van Amerongen, J., Breedveld, P.,2003, “Modelling of physical systems for the design and control of mechatronic systems”, Annual Reviews in Control (2003), pp. 97-117, Pergamon, Elsevier Ltd.

Van Den Braembussche, P. (1998), “Robust motion control of machine tools with linear motors”, PhD Thesis, Katholieke Universiteit Leuven, Leuven (Belgium).

Verscheure, D, Paijmans, B., Van Brussel, H., Swevers,J., 2006, “Vibration and motion control design and trade-off for high-performance mechatronic systems”, Proceedings of the IEEE International Conference on Control Applications, 2006.

RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.