Integration of Diffpack with FEM Applications

Wigand Rathmann, Frank Vogel

inuTech GmbH
Nürnberg, Germany

Summary

Sometimes problems arise, for which standard FEM-codes are not adequate. Examples for such problems are control concepts of flexible structures based on the level of non discretized partial differential equations or acoustic - structure couplings with sensors and actuators. For solving these problems it is necessary to extend existing FEA-codes. On the other side element types of standard programs are well tested and familiar in the FEM-community. This fact motivates to combine finite element formulations of a standard program, e.g. the structural formulation of ANSYS, with other simulation tools, e.g. control algorithms. This can be implemented using the developing environment Diffpack. The integration will be demonstrated by a wing structure to which a control is applied. The structural formulation is done with ANSYS and the control is computed by an application developed using Diffpack.

Keywords

Diffpack, ANSYS, control of continuous systems

0. Integration of Diffpack with FEM Applications

ANSYS is a very powerful FE code, covering a wide range of practical applications. But the hopes and wishes on simulation are proportional to the increasing computing power. The user is not longer satisfied with rough FE models, the meshes go more and more into details. This fact is not limited to spatial discretization only, there is also an attempt to model more and more details of the physics itself. The simulation of coupled problems became popular during the last years. Also the models considered in control theory became more and more detailed. A lot of research has been done over the last few decades on the control of continuous systems. One main field of interest has been the control of mechanical structures. It is time to apply these control concepts to engineering problems and to fill the gap between theoretical concepts and the “real-word” engineering application. In this paper we present an external control software implemented using Diffpack as development platform and, which employs model features (such as nodes, elements, system matrices, etc. ...) generated by ANSYS.

Diffpack is designed as component based software to develop solvers for non-standard or specialized simulation problems. It is complementary to programs like ANSYS or NASTRAN, since the user has to design the physics by himself. So one can model problems, which can not be solved by standard FE solvers. Diffpack is applicable to most simulation problems with the possibility of exact modeling of physical effects. Thus the user is able to obtain highest quality simulation results. All this requires an understanding of the physics and the mathematics of simulation problems and experiences in C++ programming.

1. Control concepts

We consider a mechanical system, which can be described by a linear partial differential equation (PDE) as given in eqn. (1). Examples for control problems are to steer a system at the final time T to a prescribed state (displacement u_D and velocity $\dot{u}_D$ are given) or to stabilize the system at a given state u_D over the time.
The control goal is formulated as a functional. The functional for the stabilization strategy is

$$J = \int_0^T \|F(\tau)\| + \|u(\tau) - u_D\| \, \mathrm{d}\tau$$

and for the final state

$$J = \frac{1}{2}\int_0^T \|F(\tau)\| \mathrm{d}\tau + \frac{k}{2}\|u(T) - u_D\| + \frac{k}{2}\|\dot{u}(T) - \dot{u}_D\| .$$

The question now is how to compute the control forces and moments denoted by F in the general model (1). In the study presented here, we only work with open loop controls. The application of feed back controls is now our ongoing work. We also do not consider controllability in the mathematical sense.
The mathematical machinery to compute the control F is well-known: The control problem to be solved is to minimize functional J with respect to the control F. We thus compute the variation of J w.r.t F to obtain a system of two coupled PDE’s (1) and (2), the necessary optimality condition. The direct or forward running system (1) describes the behavior of our structure, and from the adjoint or backward running system (2) we compute the control F.
In the following we consider the undeformed state as the state (i.e. u_D) to be stabilized.

1.1. The direct system

The system is described by a PDE in general form

$$
\begin{aligned}
\mathcal{M}\ddot{u} + \mathcal{A}u &= f, \quad x \in \Omega, \\
u\big|_{\partial\Omega} &= 0, \\
\mathcal{B}u\big|_{\partial\Omega_{con}} &= F, \\
u(0,x) &= u_{0,} \\
\dot{u}(0,x) &= u_{1.}
\end{aligned}
\tag{1}
$$

Ω denotes the domain of the structure, Ω_{con} the subdomain, where the control is applied. In our implementation we either support a boundary or distributed interior control. The model may further contain Neumann or Robin type boundary conditions. $\mathcal{M}$, $\mathcal{A}$ and $\mathcal{B}$ are the operators on continuous level. On the discretized level these operators are represented by FE matrices.

1.2 The adjoint system

The adjoint system is very similar to the direct system. For the first cost functional in 1. the adjoint system reads

$$
\begin{aligned}
\mathcal{M}\ddot{\varphi} + \mathcal{A}\varphi &= u, \quad x \in \Omega, \\
\varphi\big|_{\partial\Omega} &= 0, \\
\mathcal{B}\varphi\big|_{\Omega_{con}} &= 0, \\
\varphi(T) &= 0, \\
\dot{\varphi}(T) &= 0.
\end{aligned}
\tag{2}
$$

Note that the right hand side of the adjoint system (2) is given through the solution of the direct system (1). The boundary conditions are homogenous and the initial state at the final time T is zero. The control F in the direct system we obtain from the adjoint solution φ at Ω_{con} by

$$
\mathcal{B}u = -\varphi, \ x \in \Omega_{con} \tag{3}
$$

for boundary controls. In case of distributed controls, $-\varphi$ must be applied at the right hand side of the direct system properly.
In case of the second functional, the right hand side of the PDE is homogenous and the initial condition for $\varphi(T)$ and $\dot{\varphi}(T)$ are computed from $u(T)$ and $\dot{u}(T)$

1.3. Discretization

We see, that the FE representation of the direct and the adjoint system is the same. Only the initial state and the boundary condition changes. So we can use the FE matrices, assembled in a standard FE code like ANSYS to solve the forward and backward running system.

1.4. Algorithm

To find the optimal control F we must solve the optimality system (1), (2) simultaneously. We simplify that by formulating an iteration procedure to compute F. In one iteration we have to solve (1) and (2) as a single system. We write down the iteration procedure already for the discrete system. We start the iteration with a simulation without control:

$$
\begin{aligned}
\mathrm{M}\ddot{u}^0 + \mathrm{A}u^0 &= f, \\
u_i^0 &= 0, \\
\mathrm{B}u^0 &= 0, \\
u^0(0) &= u_0, \\
\dot{u}^0(0) &= u_1
\end{aligned}
\tag{4}
$$

i denotes the Dirichlet type bc nodes. Now we start the iteration procedure for $k = 1, \ldots$. At first we have to solve the adjoint system

$$
\begin{aligned}
M\ddot{\varphi}^k + A\varphi^k &= u^{k-1}, \\
\varphi_i^k &= 0, \\
B\varphi^k &= 0, \\
\varphi^k(T) &= 0, \\
\varphi^k(T) &= 0.
\end{aligned} \tag{5}
$$

The control is now computed by

$$F^k = F^{k-1} - \varphi \tag{6}$$

at the controlled FE nodes. We set $F^0 = 0$. For distributed controls the right side f^k has to be updated properly. Now we solve the direct system

$$
\begin{aligned}
M\ddot{u}^k + Au^k &= f^k, \\
u_i^k &= 0, \\
Bu^k &= F^k, \\
u^k(0) &= u_0, \\
\dot{u}^k(0) &= u_1.
\end{aligned} \tag{6}
$$

As stopping criteria one may choose $\left\| u^k - u^{k-1} \right\| \leq \delta$.

For time integration we employed the Newmark scheme in a predictor corrector formulation. This is well known and can be found e.g. in [3].

2. Integration concept

The solution process is done in three steps.

2.1 Modeling

In a first step the structural model (Finite Element Mesh) must be prepared within ANSYS, using standard ANSYS preprocessing functionality We then define the boundary condition and the initial state. We choose the initial state as a equilibrium of applied loads. The FE matrices are generated during the solution pass in ANSYS for the initial state. For the FE matrices we use the extended format of Rev. 6.0 for Krylov subspace methods, where the boundary conditions are already incorporated in the matrices. All this is done to use the finite element matrices generated by ANSYS in our external Diffpack application. Please note that only linear elastic materials and linear geometrical computations make sense in our problem setup. The last step is, to write out the node numbers, where the control is applied.

2.2 Apply the control concept

The control concept is implemented in Diffpack. All problem data are taken from files, generated in ANSYS. The FE matrices (mass and stiffness matrix) are read from the full file. The geometry is regenerated in Diffpack from additional files or directly from a cdb file by means of Diffpack's Datafilter Toolbox. The algorithm described in 1.4 is implemented as a member function of the simulator class in Diffpack. It is thus straight-forward to add step by step different control concepts as additional member functions. The selection of a particular control algorithm can be specified at the beginning of a Diffpack run, using Diffpack's menu system.

2.3 Postprocess

During our external control solution, the results of the last simulation is written to an ANSYS result file. We are thus able to use ANSYS functionality to postprocess our control results. The current implementation only supports nodal displacements. The next step is to output the control quantities (forces and moments) in the result file as well.

3. Vibration wing with tip controllers

As a first test case we chose the simple shell model of a wing structure as shown in figure 1. Again, please note that our external control problem solver will explicitly employ the discretized shell element matrices generated by ANSYS.

3.1 Model

The wing model is shown in figure 1, the control is applied at the tip end. The load applied in the initial state (figure 2) will excite not only the first mode of the wing. Also torsion of the wing will be excited. The applied control will at first reduce the displacements at the wing tip, but we can not expect, that the higher modes are reduced.

3.2 Initial load at wing top

In the figures 2 and 3 the displacements of the node 12 and 22 are shown. One can observe, that high amplitudes of this nodes are damped, in this way the control acts as expected. There are are higher oscillating vibration in the wing, which is also easy to see in figure 2 and 3.

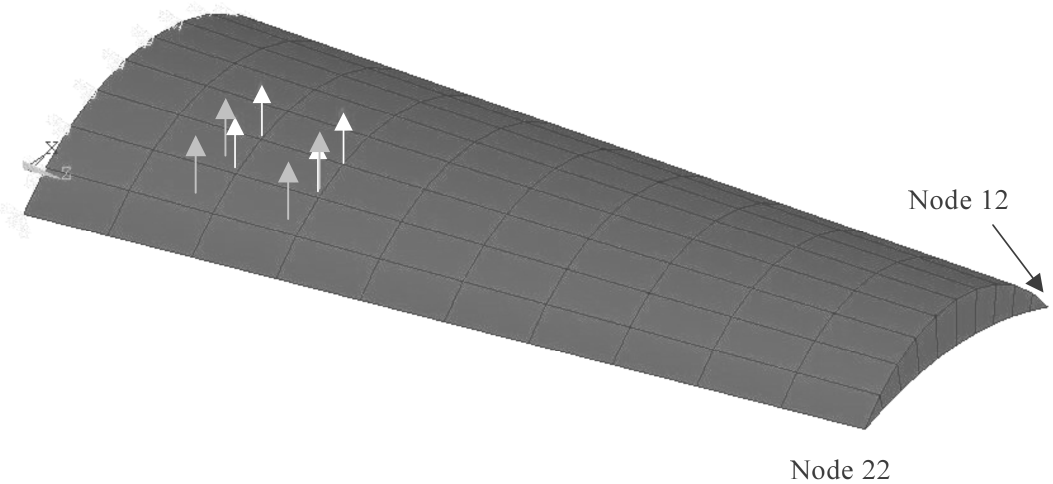

Figure 1: Wing model with load for initial condition

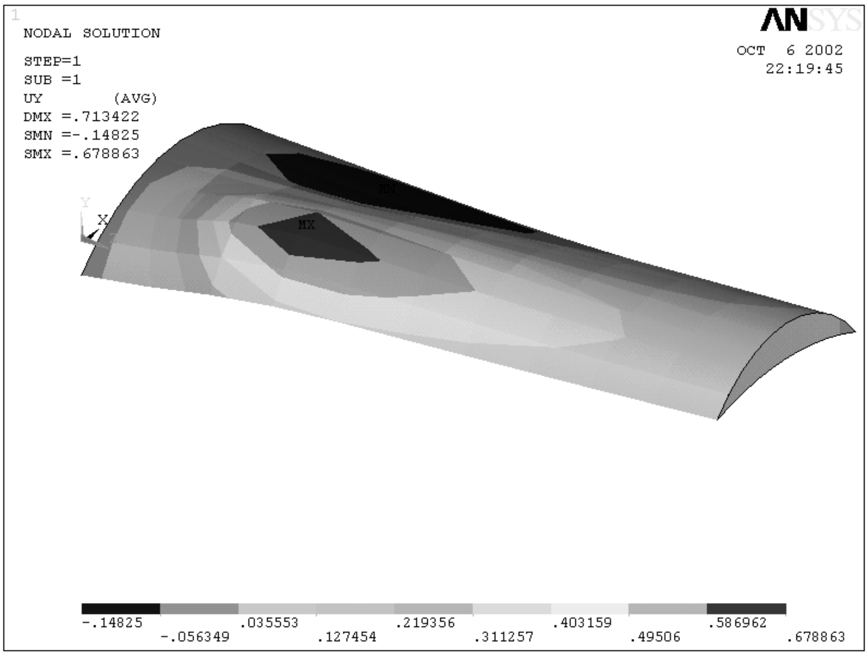

Figure 2: Initial state

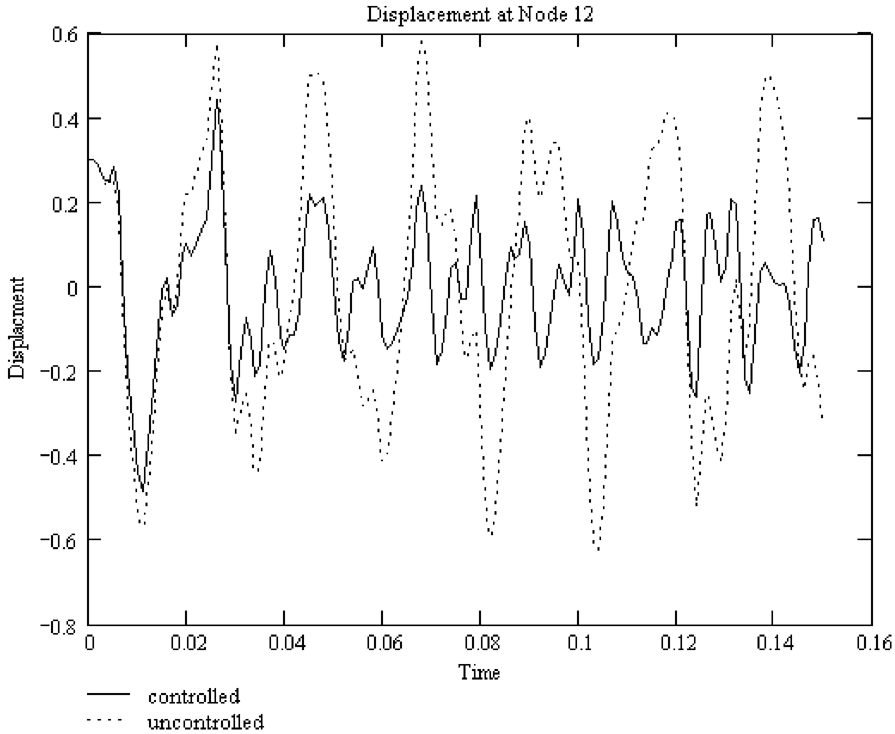

Figure 3: Comparison displacements at node 12

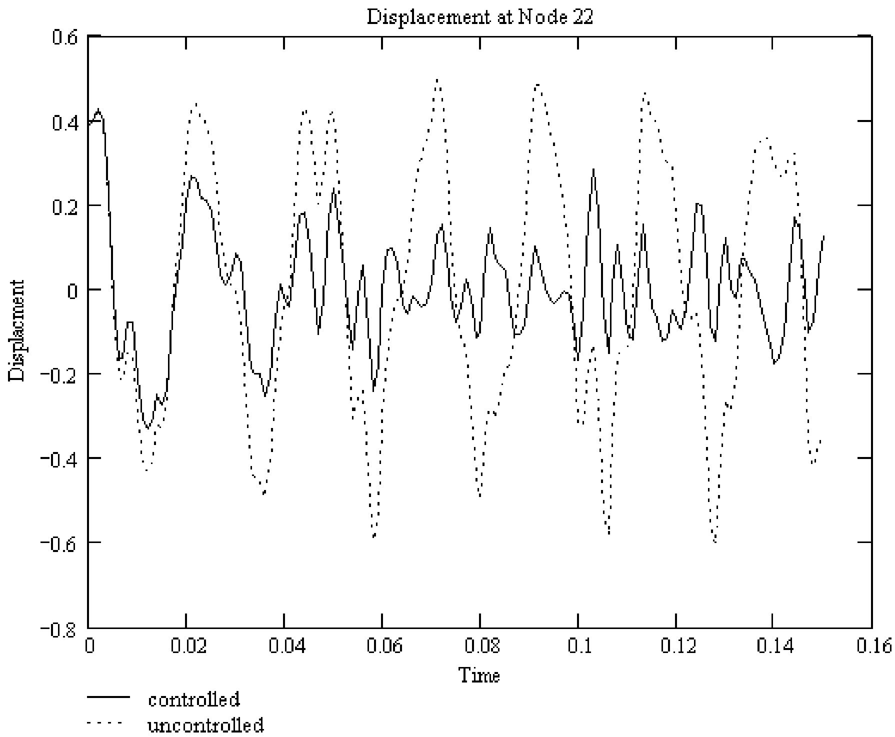

Figure 4: Comparison displacements at node 22

3.3. Initial load at wing tip end

In a second test we chose a load at the wing tip for the initial state (figure 5). This load case excites the first mode of the wing. Again, we plotted the difference between the controlled and uncontrolled system at the nodes 12 and 22. We see here, that the control works.

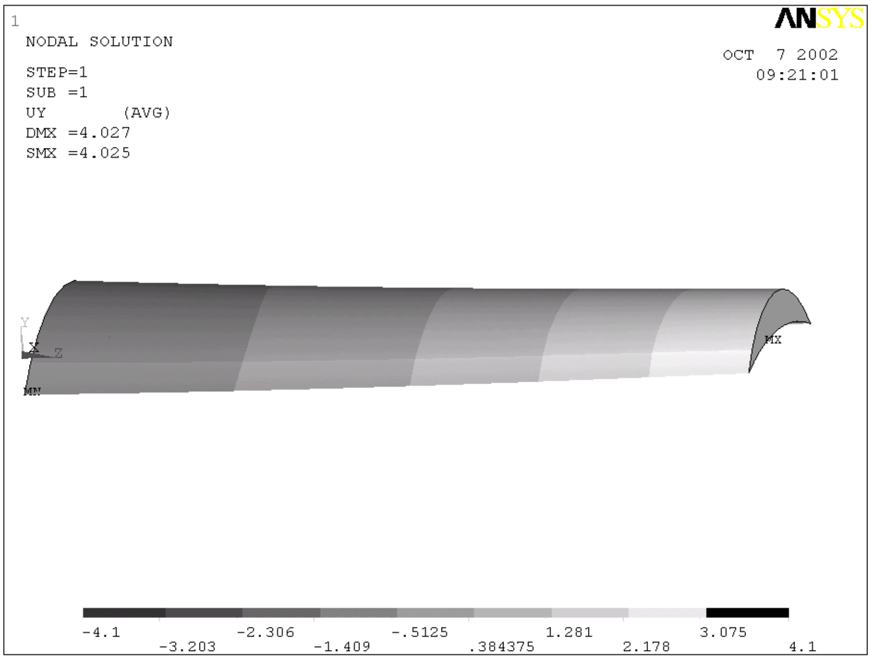

Figure 6: Initial condition with forces applied to tip end

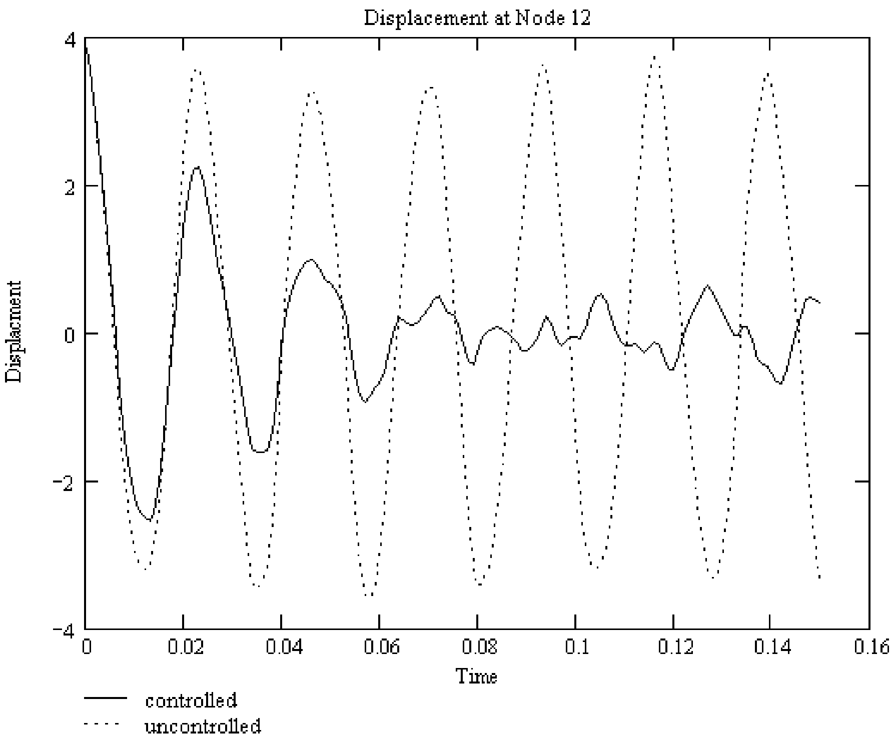

Figure 7: Comparison displacements at node 12

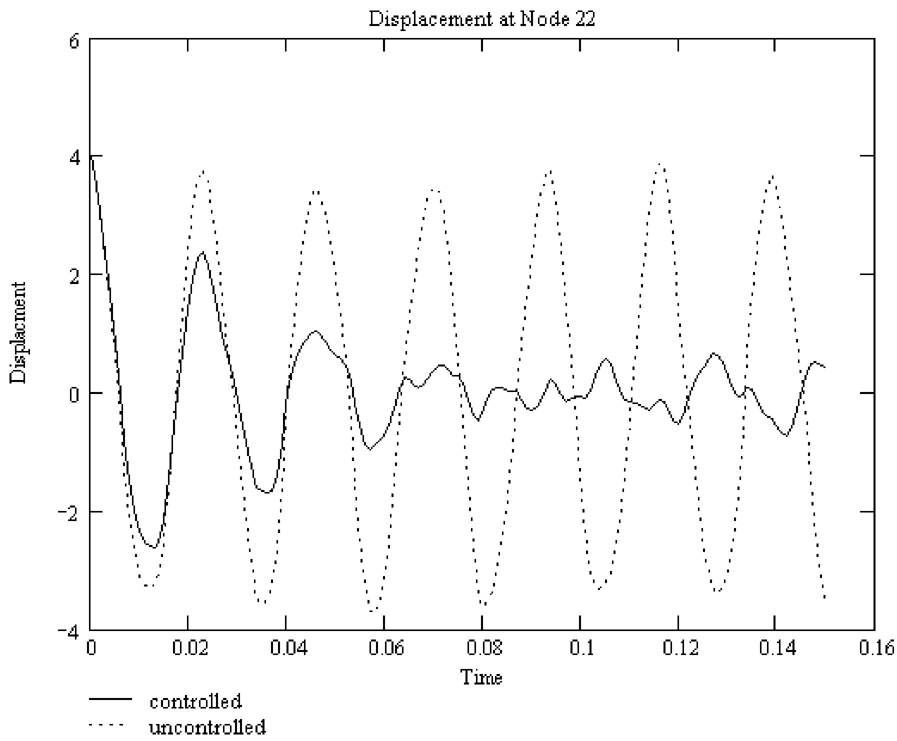

Figure 8: Comparison displacements at node 22

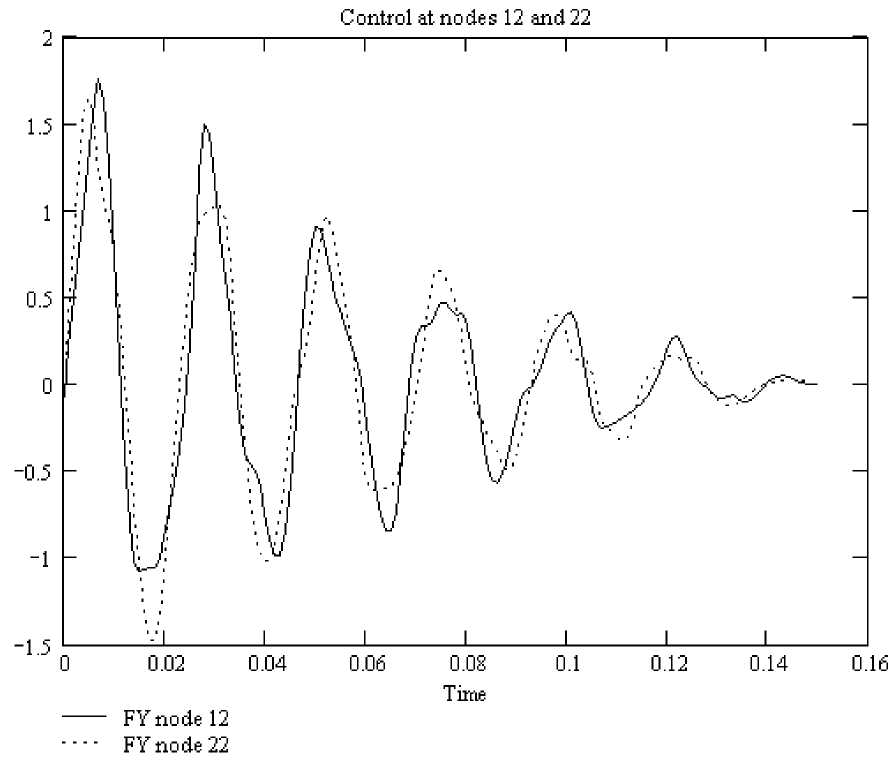

Figure 9: Control forces FY at nodes 12 and 22

Conclusions

In the paper we have presented an approach how to combine an external control solver software using the Diffpack development platform with modeling capabilities using ANSYS, to demonstrate that standard finite element packages can be easily extended for advanced and highly specialized simulation features.

A control algorithm has been presented and implemented to compute a control F by means of a simple iteration procedure based on the direct and adjoint systems. The external control algorithm explicitly imports and employs the finite element description (including element matrices) generated by ANSYS. The implementation has been applied to a control of a wing structure and promising results were presented.

In future the control approach will be extended by a so called feedback control algorithm, which is mostly requested in practice. Our focus will be the instantaneous control approach, where in a single time step of the direct system, the control applied in the next time step is computed. The advantage is, that the adjoint system is replaced by a steady state analysis. The stiffness matrix needed for the computation of the control, and generated within ANSYS, needs to be decomposed only once.

References

[1] Rathmann, W.: "Modellierung, Simulation und Steuerung von Netzwerken schwingender Balken mittels dynamischer Bereichszerlegung", Bayreuther Mathematische Schriften, vol. 60, 2000

[2] Stelzmann, U., Groth, C., Müller, G.: "FEM für Praktiker - Band 2: Strukturdynamik", Edition expertsoft Bd. 44, expert Verlag, 2001, 2. Aufl.

[3] Hughes, T.J.R. "The Finite Element Method", Prentice Hall Inc, Eglewood Cliffs, New Jersey 07632, 1987

[4] Gugat, M., Leugering G.: "Regularization of L^∞-optimal control problems for distributed parameter systems", Computational Optimization and Applications, Vol. 22, No.2, pp. 151--192, 2002

[5] Gugat, M.: "Analytic Solutions of L^∞-Optimal Control Problems for the Wave Equation", Journal of Optimization Theory and Applications 114, Vol. 114, No.2, pp. 397--421, 2002