MODELING, IDENTIFICATION AND CONTROL, 1992, VOL. 13, NO. 4, 221–240
doi:10.4173/mic.1992.4.3

# Multidiscipline simulation of elastic manipulators

T. RØLVÅG†, H. P. HILDRE†, O. I. SIVERTSEN†
and Å. Ø. WALØEN†

Keywords: *Finite Element Method, kinematics, dynamics, control, multidiscipline robot simulation.*

This paper contributes to multidiscipline simulation of elastic robot manipulators in FEDEM. All developments presented in this paper are based on the formulations in FEDEM, a simulation system developed by the authors which combines finite element, mechanism and control analysis. In order to establish this general simulation system as an efficient multidiscipline robot design tool a robot control system including a high level robot programming language, interpolation algorithms, path generation algorithms, forward and inverse kinematics, control systems, gear and transmission models are implemented. These new features provide a high level of integration between traditionally separate design disciplines from the very beginning of the design and optimization process. Several simulations have shown that high fidelity mathematical models can be derived and used as a basis for dynamic analysis and controller design in FEDEM.

## 1. Introduction

A central issue in multidiscipline dynamic manipulator simulation is to integrate the separate design disciplines FEM analysis, mechanism analysis and controller design. Commercial program systems do not provide these simulation features and cannot serve as a common design tool for both mechanical and control engineers. The mechanical and controller design process have therefore traditionally been treated separately. This paper presents methodologies which are developed and implemented in FEDEM in order to obtain a high level of integration between these separate design disciplines.

This paper gives an introduction to the non-linear FEDEM formulation and a description of the new multidiscipline simulation methodologies which are developed and implemented. An application example demonstrates how FEDEM is applied in modeling and simulation of a lightweight manipulator in the space station Columbus.

FEDEM is a general simulation system for flexible mechanisms with the option of including control models (Rølvåg 1992, Hildre, Rølvåg and Waløen 1990, Rølvåg and Aamnes 1991, Iversen 1989). It is based on a non-linear finite element formulation utilizing component mode synthesis (CMS) for model reduction. Static equilibrium solution, dynamic analysis, eigenvalue analysis and options for stress calculations are available. The modeling tools consist of an extensive FEM library of 3D elements, joints, linear and non-linear springs and dampers, gravity forces, external loads, motion input and a number of other modeling options based on the finite element techniques. A

Received 24 April 1992.

† Division of Machine Design, The Norwegian Institute of Technology, N-7034 Trondheim, Norway.

library of general input functions is available for modeling time variation, for instance for loading and input motion and for modeling non-linear springs and dampers. An option for analog and digital control elements is included in the simulation model. Control elements may be chosen from a library of general elements that may easily be extended. FEDEM enables users to model the actual shape and elastic properties of each link in a mechanism, and it provides users with all typical output from a dynamic simulation such as displacements, velocities, accelerations, forces and moments, joint, spring, damper and control variables and natural frequencies. In addition, the direct coupling of FEM and mechanism simulation makes the presentation of resulting link deformations and stresses a natural part of the system.

## 2. Basic mathematical formulation

The individual links/substructures are defined on one level, and the degrees of freedom are divided into external (subscript e) and internal (subscript i) degrees of freedom. The dynamic equation of motion for a substructure may then be written as:

$$\begin{bmatrix} M_{ee} & M_{ei} \\ M_{ie} & M_{ii} \end{bmatrix}\begin{bmatrix} \ddot{v}_e \\ \ddot{v}_i \end{bmatrix}+\begin{bmatrix} C_{ee} & C_{ei} \\ C_{ie} & C_{ii} \end{bmatrix}\begin{bmatrix} \dot{v}_e \\ \dot{v}_i \end{bmatrix}+\begin{bmatrix} K_{ee} & K_{ei} \\ K_{ie} & K_{ii} \end{bmatrix}\begin{bmatrix} v_e \\ v_i \end{bmatrix}=\begin{bmatrix} Q_e \\ Q_i \end{bmatrix} \tag{1}$$

or shorter,

$$\boldsymbol{M}_s\ddot{\boldsymbol{v}}+\boldsymbol{C}_s\dot{\boldsymbol{v}}+\boldsymbol{K}_s\boldsymbol{v}=\boldsymbol{Q}_s \tag{2}$$

where $\boldsymbol{M}_s$, $\boldsymbol{C}_s$ and $\boldsymbol{K}_s$ are the substructure mass, damping and stiffness matrix respectively. The substructure matrices are assembled from conventional finite element matrices.

Using component mode synthesis (CMS) transformation the nodal displacements $\boldsymbol{v}$ may be expressed by the external degrees of freedom $\boldsymbol{v}_e$, and by the new generalized coordinates $\boldsymbol{y}$ as:

$$\boldsymbol{v}=\begin{bmatrix} \boldsymbol{v}_e \\ \boldsymbol{v}_i \end{bmatrix}=\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{B} & \boldsymbol{\Phi} \end{bmatrix}\begin{bmatrix} \boldsymbol{v}_e \\ \boldsymbol{y} \end{bmatrix}=\boldsymbol{H}\begin{bmatrix} \boldsymbol{v}_e \\ \boldsymbol{y} \end{bmatrix} \tag{3}$$

where, $\boldsymbol{B}=-\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ie}$, and $\boldsymbol{\Phi}$ is a matrix of eigenvectors for a few of the lowest modes of the substructure, selected for each substructure after eigenvalue solution ($\boldsymbol{v}_e=\boldsymbol{0}$) of the substructure model, and $\boldsymbol{I}$ is the unit matrix. This method exchanges the internal degrees of freedom $\boldsymbol{v}_i$ for a substructure by a number of eigenmodes as uncoupled generalized coordinates $\boldsymbol{y}$. If all eigenmodes are included, the transformation is exact. Usually, only a few of the lowest modes need to be included to get good results, and this will give a substantial reduction of problem size.

Combining eqn. (3) and its first and second time derivatives with eqn. (1) and premultiplying by $\boldsymbol{H}^{\mathrm{T}}$ gives

$$\boldsymbol{m}\begin{bmatrix} \ddot{\boldsymbol{v}}_e \\ \ddot{\boldsymbol{y}} \end{bmatrix}+\boldsymbol{c}\begin{bmatrix} \dot{\boldsymbol{v}}_e \\ \dot{\boldsymbol{y}} \end{bmatrix}+\boldsymbol{k}\begin{bmatrix} \boldsymbol{v}_e \\ \boldsymbol{y} \end{bmatrix}+\boldsymbol{p}=\boldsymbol{0} \tag{4}$$

where

$$\begin{aligned} \boldsymbol{m}&=\boldsymbol{H}^{\mathrm{T}}\boldsymbol{M}_S\boldsymbol{H} \\ \boldsymbol{c}&=\boldsymbol{H}^{\mathrm{T}}\boldsymbol{C}_S\boldsymbol{H} \\ \boldsymbol{k}&=\boldsymbol{H}^{\mathrm{T}}\boldsymbol{K}_S\boldsymbol{H} \\ \boldsymbol{p}&=-\boldsymbol{H}^{\mathrm{T}}\boldsymbol{Q}_S \end{aligned} \tag{5}$$

Before they are assembled into system matrices, the superelement matrices are transformed relative to a global coordinate system. The general system dynamic equation of motion at time $t$ may be written as [7]:

$$\boldsymbol{F}^I(\boldsymbol{t})+\boldsymbol{F}^D(\boldsymbol{t})+\boldsymbol{F}^S(\boldsymbol{t})=\boldsymbol{Q}(\boldsymbol{t},\boldsymbol{v},\dot{\boldsymbol{v}}) \tag{6}$$

where $\boldsymbol{F}^I$, $\boldsymbol{F}^D$ and $\boldsymbol{F}^S$ are inertia, damping and stiffness forces, respectively, while $\boldsymbol{Q}(\boldsymbol{t},\boldsymbol{v},\dot{\boldsymbol{v}})$ represents input loads. A change in force over time increment $k$ on system level may be written as:

$$(\boldsymbol{F}^I_{k+1}-\boldsymbol{F}^I_k)+(\boldsymbol{F}^D_{k+1}-\boldsymbol{F}^D_k)+(\boldsymbol{F}^S_{k+1}-\boldsymbol{F}^S_k)=(\boldsymbol{Q}_{k+1}-\boldsymbol{Q}_k) \tag{7}$$

or

$$\Delta\boldsymbol{F}^I_k+\Delta\boldsymbol{F}^D_k+\Delta\boldsymbol{F}^S_k=\Delta\boldsymbol{Q}_k \tag{8}$$

This is the equation of motion in incremental form. In the following, the different factors of (8) will be expanded.

The inertia, damping and stiffness relations are approximated by a linearization around the starting position for each increment, and the incremental system matrices, shown in the following, are generated for that position. The error introduced by this approximation may be eliminated by equilibrium iterations. The exact incremental system matrices, called the secant matrices, are a function of the unknown displacement increments sought for here, and cannot be generated in advance.

The incremental inertia forces from eqn. (8) may be written

$$\Delta\boldsymbol{F}^I_k=\boldsymbol{M}_{Ik}\cdot\Delta\ddot{\boldsymbol{r}}_k \tag{9}$$

where $\mathbf{M}_{\mathrm{Ik}}$ is the system mass matrix at the beginning of time increment k and $\Delta\ddot{\boldsymbol{r}}_k=\ddot{\boldsymbol{r}}_{k+1}-\ddot{\boldsymbol{r}}_k$ represent the change in acceleration during increment k. The system mass matrix $\mathbf{M}_{\mathrm{Ik}}$ may be constant or a function of the displacement vector $\boldsymbol{r}$, depending on what kind of element mass representation is used. The super element mass matrices **m** are constant, but undergo a geometric transformation (Rølvåg 1992) before they are added into the system matrix. If lumped mass representation is chosen, the super element mass matrix is diagonal and the geometric transformation has no effect. The system mass matrix will then also be diagonal and constant during integration. The incremental elastic forces from eqn. (8) may be written

$$\Delta\boldsymbol{F}^S_k=\boldsymbol{K}_{Ik}\cdot\Delta\boldsymbol{r}_k \tag{10}$$

where $\mathbf{K}_{\mathrm{Ik}}$ is the system stiffness matrix at the beginning of time increment k and $\Delta\boldsymbol{r}=\boldsymbol{r}_{k+1}-\boldsymbol{r}_k$ represent the increment in displacement for increment k. In mechanisms, the system stiffness matrix is in general a function of the displacement vector **r**.

The incremental damping forces from eqn. (8) may be written

$$\Delta\boldsymbol{F}^D_k=\boldsymbol{C}_{Ik}\cdot\Delta\dot{\boldsymbol{r}}_k \tag{11}$$

where $\mathbf{C}_{\mathrm{IK}}$ is the system damping matrix at the beginning of time increment k and $\Delta\dot{\boldsymbol{r}}_k=\dot{\boldsymbol{r}}_{k+1}-\dot{\boldsymbol{r}}_k$ represent the change in velocity during increment k. For this formulation the system damping matrix may be constant or a function of the displacement vector **r**.

As for the mass matrix, if the damping matrix is diagonal it will not be affected by the geometric transformation, and will therefore be constant. Two different forms of damping are included in FEDEM. One is called proportional damping, where the damping matrix is a linear combination of the stiffness and the mass matrices.

$$\boldsymbol{C}_{Ik}=\alpha_1\boldsymbol{M}_{Ik}+\alpha_2\boldsymbol{K}_{Ik} \tag{12}$$

where the constants $\alpha_1$ and $\alpha_2$ are calculated to achieve desired damping. In addition the system matrices and vectors may be modified from a number of linear, non-linear and time-dependent springs, dampers and loads including masses on individual degrees of freedom. The geometric transformation and the assembling of the system matrices are shown in Rølvåg (1992).

The general form of the incremental (linearized) dynamic equation of motion may now be written as follows.

$$\boldsymbol{M}_{Ik}\Delta\ddot{\boldsymbol{r}}_k+\boldsymbol{K}_{Ik}\Delta\boldsymbol{r}_K=\Delta\boldsymbol{Q}_k \tag{13}$$

In general $\boldsymbol{M}_{Ik}$, $\boldsymbol{C}_{Ik}$ and $\boldsymbol{K}_{Ik}$ are recalculated for each time increment. Solutions of eqn. (13) by a time integration algorithm (Newmarks method [Rølvåg 1992]) gives $\Delta\boldsymbol{r}_k$, $\Delta\boldsymbol{r}_k$ and $\Delta\ddot{\boldsymbol{r}}_k$.
The total solution at the end of the increment is then:

$$\begin{aligned}\boldsymbol{r}_{k+1}&=\boldsymbol{r}_k+\Delta\boldsymbol{r}_k\\ \dot{\boldsymbol{r}}_{k+1}&=\dot{\boldsymbol{r}}_k+\Delta\dot{\boldsymbol{r}}_k\\ \ddot{\boldsymbol{r}}_{k+1}&=\boldsymbol{r}\ \ +\Delta\ddot{\boldsymbol{r}}_k\end{aligned} \tag{14}$$

The solution at the end of the increment (14) may be used to calculate $\boldsymbol{F}^I_{k+1}$, $\boldsymbol{F}^D_{k+1}$ and $\boldsymbol{F}^S_{k+1}$, and because of the linearization there will be unbalanced forces at the end of the increment.

$$\Delta\hat{\boldsymbol{F}}_{k+1}=\boldsymbol{Q}_{k+1}-[\boldsymbol{F}^I_{k=1}+\boldsymbol{F}^D_{k+1}+\boldsymbol{F}^S_{k+1}] \tag{15}$$

These residual forces may be added to the load increment for the next step, see eqn. (13).

$$\begin{aligned}\Delta\boldsymbol{Q}_k&=\boldsymbol{Q}_{k+1}-\boldsymbol{Q}_k+\Delta\hat{\boldsymbol{F}}_k\\ &=\boldsymbol{Q}_{k+1}-[\boldsymbol{F}^I_k+\boldsymbol{F}^D_k+\boldsymbol{F}^S_k]\end{aligned} \tag{16}$$

Equation (13) may then be written:

$$\boldsymbol{M}_{Ik}\Delta\ddot{\boldsymbol{r}}_k+\boldsymbol{C}_{Ik}\Delta\dot{\boldsymbol{r}}_k+\boldsymbol{K}_{Ik}\Delta\boldsymbol{r}_k=\boldsymbol{Q}_{k+1}-[\boldsymbol{F}^I_k+\boldsymbol{F}^D_k+\boldsymbol{F}^S_k] \tag{17}$$

This is an approximation for the equation at time $t_{k+l}=(k+l)h$ where $h$ is the time step length. To achieve equilibrium at the end of the increment, iteration has to be used to minimize the error from the solution of eqn. (17). Iteration is accomplished by replacing $\Delta\boldsymbol{Q}_k$ in eqn. (13) by $\Delta\hat{\boldsymbol{F}}_{k+1}$ and by solving for the correction $\boldsymbol{\Delta}_k$ for $\Delta\boldsymbol{r}_k$ from:

$${}^i\boldsymbol{M}_{Ik}{}^i\ddot{\boldsymbol{\Delta}}_k+{}^i\boldsymbol{C}_{Ik}{}^i\dot{\boldsymbol{\Delta}}_k+{}^i\boldsymbol{K}_{Ik}{}^i\boldsymbol{\Delta}_k=\boldsymbol{Q}_{k+1}-[{}^{i-1}\boldsymbol{F}^I_{k+1}+{}^{i-1}\boldsymbol{F}^D_{k+1}+{}^{i-1}\boldsymbol{F}^S_{k+1}] \tag{18}$$

The super index to the left of the symbols indicates the iteration number. The displacement increment is then improved from:

$${}^i\Delta\boldsymbol{r}_k={}^{i-1}\Delta\boldsymbol{r}_k+{}^i\boldsymbol{\Delta}_k \tag{19}$$

and the total displacement vector is calculated from:

$${}^i\boldsymbol{r}_{k+1}={}^{i-1}\boldsymbol{r}_{k+1}+{}^i\boldsymbol{\Delta}_k \tag{20}$$

Velocities and accelerations are calculated the same way. Then ${}^{i}\boldsymbol{F}_{k+1}^{I}$, ${}^{i}\boldsymbol{F}_{k+1}^{D}$, ${}^{i}\boldsymbol{F}_{k+1}^{S}$ may be calculated and substituted into eqn. (18) to solve for the correction ${}^{i+1}\Delta_k$, and so on.

If $\boldsymbol{M}_{Ik}$, $\boldsymbol{C}_{Ik}$ and $\boldsymbol{K}_{Ik}$ are updated after each iteration, the process is called Newton–Raphson iteration. If the matrices $\boldsymbol{M}_{Ik}$, $\boldsymbol{C}_{Ik}$ and $\boldsymbol{K}_{Ik}$ are all left constant during iteration or only updated after some iterations, it is called modified Newton–Raphson iteration (Rølvåg 1992).

The following criteria of iteration convergence is established.

$$\|{}^{i}\Delta_k\| \leqslant \varepsilon \tag{21}$$

where $\varepsilon$ is a specified tolerance. The solutions of eqns. (17) and (18) are obtained by a Newmark time integration algorithm (Rølvåg 1992).

## 3. FEDEM modeling of the EMATS manipulator

### 3.1. *Introduction*

A majority of ESA's Microgravity Program is the Columbus Program. In this spacecraft all testing is controlled by two robots. In order to achieve the required working speed these robots have to be as light as possible and the robot movements have to be smooth and practically without vibration. To satisfy these demands it is necessary in arm design to compromise between lightweight construction and rigidity/minimum structural flexibility.

The microgravity environment in an orbiting spacecraft like Columbus is not obtainable on earth, and the robot operations in this environment must therefore be simulated on a computer in order to verify the manipulator performance. It was not an aim to perform a deep analysis of the EMATS manipulators in this work, rather this manipulator is used as an example to show the capabilities of FEDEM concerning modeling and dynamic simulation of flexible manipulators. The general robot control system developed during this work is used in order to ease the description of the operation to be simulated.

### 3.2. *The EMATS kinematic model*

The Columbus robot is a redundant manipulator, i.e. it has $\dim(\boldsymbol{\theta})=8$ axes for a maximum required $\dim(\mathbf{X})=6$ DOF of the tool frame $\{\boldsymbol{T}\}$. This means that the inverse kinematics problem is mathematically indeterminate, and there are two additional DOF that can be specified. One straightforward way to parameterize the solution latitude is to treat the longitudinal and vertical sliding base separately. Wohlfart, Rothenburger and Mau (1989) define the 6 joint angles $(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3, \boldsymbol{\theta}_4, \boldsymbol{\theta}_5, \boldsymbol{\theta}_6)$ and consider the linear drives $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}_z$ as known. The arm kinematics can then be derived analytically. The EMATS kinematic model is shown in Fig. 1. The transformation matrices relating the successive coordinate frames are given in Wohlfart *et al.* (1989).

The closed form solution of the inverse arm kinematics proposed in Wohlfart *et al.* (1989), was first included in the FEDEM kinematic module. This approach suffered from several singularity problems and didn't perform well. The final approach which is derived in Rølvåg (1992) is based on separating the EMATS manipulator arm into two parts, the arm with the first three joints for the major positioning, and the wrist with the last three joints for the major orientation. This approach is originally proposed by Pieper (1969) for robots with three consecutive axes intersecting at a point.

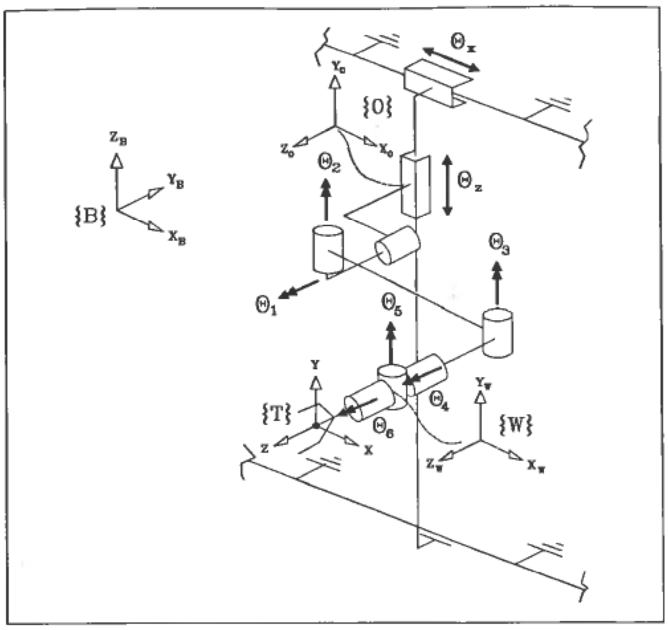

Figure 1. The EMATS kinematic structure.

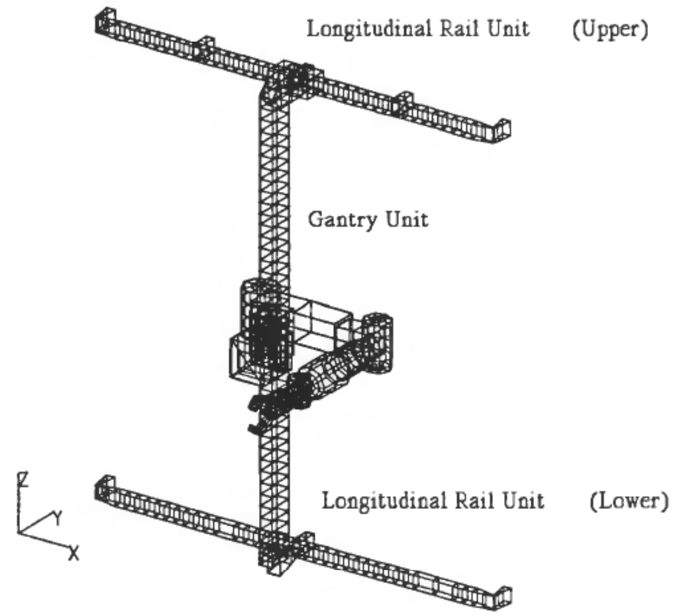
Longitudinal Rail Unit (Upper)
Gantry Unit
Longitudinal Rail Unit (Lower)
Z
Y
X

Figure 2. The EMATS FEDEM model displayed in MPGS.

### 3.3. *The EMATS Finite Element Model*

The manipulator on the right side of the spacelab is modeled in PATRAN (1991) according to the mechanical description (Wohlfart *et al.* (1989)), and an evaluation of motor and transmission for the linear vertical drive (Hildre *et al.* 1990). The Longitudinal Rail Units (LRU) and the arm are modeled by beam elements and the Gantry Unit (GU) is modeled by shell elements.

The Longitudinal Rail Unit is fixed to the satellite by springs equivalent to the rail support stiffness. The Gantry Unit is connected to the upper and lower LRU by two prismatic joints. The 6 outer links are connected by revolute joints. These joints are always kept in fixed positions while the gantry is moving.

### 3.4. *Gear modeling in FEDEM*

High gear ratios do not have a decoupling effect on FEDEM manipulator models when rotor inertias are added to the activated joints (the output side of the gear trains). Due to the finite element formulation of the mass matrix, additional masses (rotor inertias) are acting on the absolute motion and not only on the relative (joint) motion. The solution to this problem is obtained by using an implicit function (Rølvåg and Aamnes (1991)) and a gear model proposed in Hildre (1991). The main result obtained by this gear model is the possibility to add the actuator inertias on the high speed side of the gear trains. The actuator inertias will then have the desired decoupling effect.

The gear model which is shown in Fig. (3), may also include the description of the actuator dynamics and nonlinear friction effects. The input torque $\boldsymbol{M_{inp}}$ is acting on the input link $\boldsymbol{link_{inp}}$ which is modeled as a substructure with mass and inertia $\boldsymbol{J_{aj}}$ corresponding to the actuator no. $\langle \boldsymbol{j} \rangle$. This gives the input link an angular acceleration $\boldsymbol{\ddot{\theta}_{inp}}$ which is integrated for each iteration. The corresponding angle $\theta_{inp}$ is divided by the gear ratio $\eta$ and transferred to the low speed side of the gear by using an implicit function. The output angle given by the implicit function is added as a prescribed change of the stress free length of the spring acting between the two robot links $(n, n+1)$. The spring stiffness $\boldsymbol{k_{aj}}$ is representing the gear flexibility and backlash (Rølvåg and Aamnes 1991, Hildre 1991). The change in the stress free length of the spring will then generate an input torque on the output side of the gear $\boldsymbol{M_{out}}$ which will cause the arm movement.

This torque $M_{out}$ is divided by the gear ratio $\eta$ and transferred back to the high speed side of the gear train by using another implicit function. This process is repeated at each iteration until the input $M_{inp}$ and output $M_{out}$ moments are balanced and equilibrium is reached. Nonlinear friction effects (Rølvåg and Aamnes 1991, Hildre 1991) like viscous, coulomb and stiction friction may be represented by the damping coefficient $c_{aj}$. The input joint velocity $\boldsymbol{\dot{\theta}_{inp}}$ is then transferred to the low speed side and the friction force back to the high speed side by the use of similar implicit functions. This means that the gear model causes some more iterations on system level until the input/output gear forces are balanced. The reduction gears and transmissions of the

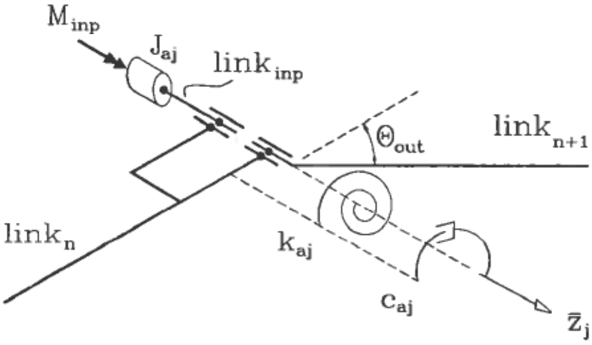

Figure 3. Gear modeling in FEDEM.

EMATS manipulator are modeled by this approach according to the evaluation results in Hildre *et al.* (1990).

### 3.5. *Computing arm inertias for controller design*

The feedback and reference tracking can be perfect only if the time-variant arm inertias are exactly estimated. If we consider a robot mechanism modeled in FEDEM, the system mass and stiffness matrices are a function of the manipulator configuration. Hence, we cannot select fixed parameters (controller gains) which will keep the poles in a desired (critical damping) location if the gear ratios are low. Therefore a control law in which the gains are time varying (as a function of the robot arm configuration) in a manner such that the system is always critically damped had to be considered.

Based on the non-linear finite element formulation in FEDEM, algorithms are implemented which provides very good estimates of the joint inertias at selected positions during simulation. These inertias are added to the motor inertias used in the controller design. The calculation of the effective inertia for each actuator is based on a numerical method for the calculations of Jacobi matrices and the principal of virtual work.

The reduced mass matrix $\boldsymbol{m}_s$ for substructure $s$ is written as

$$\boldsymbol{m}_s = \boldsymbol{H}^{\mathrm{T}}\boldsymbol{M}_s\boldsymbol{H} = \begin{bmatrix} \boldsymbol{m}_{11} & \boldsymbol{m}_{12} \\ \boldsymbol{m}_{21} & \boldsymbol{m}_{22} \end{bmatrix} \tag{22}$$

where

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{B} & \boldsymbol{\Phi} \end{bmatrix} \tag{23}$$

$$\boldsymbol{M}_s = \begin{bmatrix} \boldsymbol{M}_{ee} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}_{ii} \end{bmatrix} \tag{24}$$

$$\begin{aligned} \boldsymbol{m}_{11} &= \boldsymbol{M}_{ee} + \boldsymbol{B}^{\mathrm{T}}\boldsymbol{M}_{ii}\boldsymbol{B} \\ \boldsymbol{m}_{12} &= \boldsymbol{m}_{21} = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{M}_{ii}\boldsymbol{\Phi} \\ \boldsymbol{m}_{22} &= \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{M}_{ii}\boldsymbol{\Phi} \end{aligned} \tag{25}$$

$\boldsymbol{M}_{ee}$ and $\boldsymbol{M}_{ii}$ is a lumped or consistent mass matrix corresponding to the external and internal nodes respectively. The influence matrix $\boldsymbol{B} = -\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ie}$ gives the internal displacements as a function of the external (super node) displacements, and $\boldsymbol{\Phi}$ is a matrix of eigenvectors for a few of the lowest substructure modes. Equation (23) transforms the substructure mass matrix corresponding to the internal ($\boldsymbol{M}_{ii}$) and external ($\boldsymbol{M}_{ee}$) nodes/dofs, to a reduced mass matrix corresponding to external (subscript 1) and generalized (subscript 2) nodes/dofs (Rølvåg 1992, Rølvåg and Aamnes 1991, Newmark 1959).

Several test simulations showed that the substructure mass submatrix ($\boldsymbol{m}_{11}$) is a very good approximation of $\boldsymbol{m}_s$, unless the substructure is extremely flexible. This approximation is equivalent with not including substructure modes in the CMS transformation of the substructure mass matrix (static condensation) (Rølvåg and Aamnes 1991). For the purpose of estimating joint inertias for rigid body robot controllers this approximation is obviously good enough.

The submatrix $\boldsymbol{m}_{11}$ is transformed relative to base frame $\{\boldsymbol{B}\}$ using the corresponding corotated substructure frame at the current time incident. This geometric transformation is therefore configuration dependent and must be repeated each time the joint inertias are computed. If $\boldsymbol{m}_{11}$ is consistent, the nodal masses are lumped by adding masses on corresponding nodals dofs for each column (Rølvåg 1992) which give 6 × 6 diagonal nodal mass matrices $\boldsymbol{M}_{inode}$ referring to $\{\boldsymbol{B}\}$.

Note that lumped masses added on system level are also taken into account when computing the nodal mass matrices $\boldsymbol{M}_{inode}$. This means that additional masses representing sensors, actuators etc. will influence the calculated inertias including any payload.

Each nodal mass matrix are then transformed to equivalent joint inertias by using a Jacobian matrix. The Jacobian matrix which relates cartesian velocities of super node ⟨inode⟩ to joint rates can be computed using a numerical formulation.

$$\boldsymbol{V}_{inode}=\begin{bmatrix}\boldsymbol{v}\\ \boldsymbol{\omega}\end{bmatrix}=[\boldsymbol{J}]\cdot\dot{\boldsymbol{\theta}} \tag{26}$$

where $\mathbf{v}$ are linear velocities, $\boldsymbol{\omega}$ are angular velocities, $\dot{\boldsymbol{\theta}}$ are joint rates and

$$[\boldsymbol{J}]=\begin{bmatrix}\boldsymbol{J}_{L1} & \boldsymbol{J}_{L2} & ..\boldsymbol{J}_{Ln}\\ \boldsymbol{J}_{A1} & \boldsymbol{J}_{A2} & ..\boldsymbol{J}_{An}\end{bmatrix} \tag{27}$$

where $\boldsymbol{n}$ is the number of manipulator joints which may cause a movement of the super node ⟨inode⟩. If joint number ⟨$\boldsymbol{j}$⟩ is a prismatic joint, the corresponding column of the Jacobian is computed as

$$\begin{bmatrix}\boldsymbol{J}_{Lj}\\ \boldsymbol{J}_{Aj}\end{bmatrix}=\begin{bmatrix}\mathbf{x}_j\\ \mathbf{0}\end{bmatrix} \tag{28}$$

and if joint number ⟨$\boldsymbol{j}$⟩ is a revolute joint, the corresponding column of the Jacobian is computed as

$$\begin{bmatrix}\boldsymbol{J}_{Lj}\\ \boldsymbol{J}_{Aj}\end{bmatrix}=\begin{bmatrix}\boldsymbol{z}_j\times\boldsymbol{r}_{j,inode}\\ \boldsymbol{z}_j\end{bmatrix} \tag{29}$$

where $\mathbf{x}_j$ and $\mathbf{z}_j$ are unit vectors pointing in the direction of the prismatic and revolute axes respectively. The $\boldsymbol{r}_{j,inode}$ is a vector from the master node on the jth joint to super node number ⟨*inode*⟩.

The lumped joint inertias $\boldsymbol{M}_{\Theta}$ are most conveniently established by considering the mass contribution from each super node $\boldsymbol{M}_{inode}$. In order to make these joint inertias equivalent to the nodal masses distributed on the manipulator model, a virtual deflection field

$$[\boldsymbol{M}_{\theta,inode}\dot{\boldsymbol{\theta}}]^{\mathrm{T}}\dot{\boldsymbol{\theta}}_{virt}=[\boldsymbol{M}_{inode}\boldsymbol{V}]^{\mathrm{T}}\boldsymbol{V}_{virt} \tag{30}$$

is imposed on the model. The $\boldsymbol{M}_{inode}$ is a lumped diagonal nodal matrix and $\boldsymbol{M}_{\Theta,inode}$ is the desired equivalent diagonal joint inertia matrix (the super node's contribution to the joint inertias).

Equating the virtual work done by the distributed nodal deflection rates and the virtual work done by the equivalent virtual joint rates during the virtual displacement and substituting for $\boldsymbol{V}=[\boldsymbol{J}]\cdot\boldsymbol{\theta}$ in eqn. (30) gives

$$\begin{gathered}[\boldsymbol{M}_{\theta,inode}\boldsymbol{\theta}]^{\mathrm{T}}\boldsymbol{\theta}_{virt}=[\boldsymbol{M}_{inode}\boldsymbol{J}\boldsymbol{\theta}]^{\mathrm{T}}\boldsymbol{J}\boldsymbol{\theta}_{virt}\\ \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{M}_{\theta,inode}^{\mathrm{T}}=\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{J}^{\mathrm{T}}\boldsymbol{M}_{inode}^{\mathrm{T}}\boldsymbol{J}\\ \boldsymbol{M}_{\theta,inode}=\boldsymbol{J}^{\mathrm{T}}\boldsymbol{M}_{inode}\boldsymbol{J}\end{gathered} \tag{31}$$

The joint inertias caused by the manipulator arm (arm inertias) is then calculated as

$$M_\theta = \sum_{inode=1}^{nnode} [J_{inode}^{T} M_{inode} J_{inode}] \tag{32}$$

where index [inode⟩ runs over all external super nodes on the manipulator links. The effective joint inertias used in the controller design (Rølvåg 1992) are given as the sum of these time-variant arm inertias divided by the square of the gear ratios ($\eta > 1{\cdot}0$) and the constant actuator inertias.

## 4. FEDEM simulations of the EMATS manipulator

### 4.1. *Introduction*

Several trajectories have been generated and simulated by using the general robot control system implemented in FEDEM. These trajectories have been animated on work stations and recorded on video tapes. Both joint and cartesian space trajectories were generated and applied as inputs to PD controllers. The interpreter module allowed easy programming of robot operations, and the interpolator and kinematic modules performed satisfactorily (Rølvåg 1992).

The interpolated joint reference values are inputs to individual PD controllers, which generates the motor torque acting on the high speed side of the gear/joints. The high speed (input) side of a gear (the motor rotor and transmission) is modeled as a single link with equivalent inertia.

The data flow between the different SIMULE modules for a single joint is illustrated in fig. (4) where

$\theta_{\text{ref}}$ = the joint reference position/angle computed by the robot control system
$K_z$ = the proportional (P) gain
$K_v$ = the derivative/velocity (D) gain
$J_{\text{high}}$ = the high speed rotor inertia
$\theta_{\text{high}}$ = the joint position/angle at the high speed (input) side of the gear
$\theta_{\text{low}}$ = the joint position/angle at the low speed (output) side of the gear
$\theta_{\text{low},i}$ = the feedback joint position/angle at the low speed (output) side of the gear at iteration no. ⟨*i*⟩
$k_g$ = the gear stiffness represented by a nonlinear spring
$\eta$ = the gear ratio

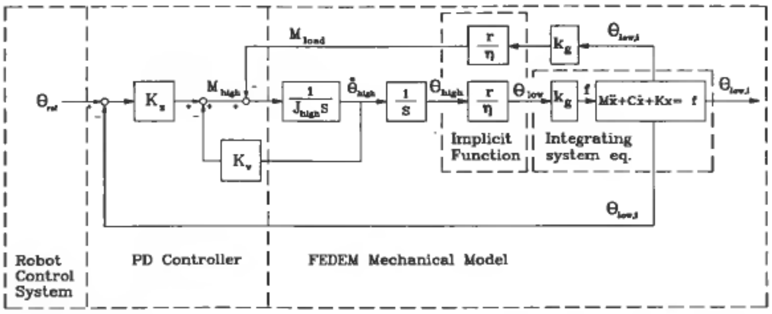

Figure 4. The data flow between the Robot Control System, the PD Controller and the FEDEM mechanical model including the implicit functions which represent a virtual gear and transmission system.

$r$ = the radius of the friction wheel (linear transmission 7 and 8)

$f$ = the force/moment caused by a change in the stress free length ($\theta_{\text{low}}$) of the nonlinear spring, acting on the low speed side of the gear.

$M_{\text{load}}$ = the load (robot arm inertia) acting on the high speed (motor) side of the gear.

$M_{\text{high}}$ = the input motor moment computed by the PD controller

The parameters used for the individual PD controllers are given in Table 1. The servo control systems are modeled/configurated in FEDEM as shown in Figure 5. Hence, 8 PD controllers are designed to handle the MIMO control problem.

Several simulations including the robot control system, gear models and PD controllers were performed in order to verify the robot simulation capabilities of FEDEM. Two examples which show the main new multidiscipline features are included in this paper.

### 4.2. *Example 1—Micro gravity simulation*

Figures 7 to 14 show how the control variables no. 1 ($\theta_{\text{ref}}$), 4 ($\tau_{\text{high}}$), 5 ($\dot{\theta}_{\text{high}}$) and 7 ($\theta_{\text{low}}$) in fig. 5 are changing for each joint during the simulation when the gravity forces are zero. The feedback values are tracking the references, and the steady state errors are zero due to no gravity forces. No overshoot in joint positions indicates that the system is critically damped or overdamped in all situations.

| Joint no. | $J_{\text{high}}$ [kgm$^2$] | $J_{\text{low}}$[1] [kgm$^2$] | $J_{\text{eff}}$ [kgm$^2$] | $\eta$ | $1/\eta, r/\eta$ | $\omega_b$ [Hz] | $\zeta$[3] | $K_v$ | $K_p$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2·70E−4 | 1·01E+1 | 4·19E−4 | 260 | 3·85E−3 | 6 | 1·0 | 5·03E−3 | 3·920 |
| 2 | 2·66E−4 | 1·86E+1 | 5·40E−4 | 260 | 3·85E−3 | 6 | 1·0 | 6·48E−3 | 5·060 |
| 3 | 1·06E−4 | 4·89E+0 | 2·28E−4 | 260 | 5·00E−3 | 6 | 1·0 | 2·74E−3 | 1·640 |
| 4 | 2·30E−5 | 5·58E−2 | 2·30E−5 | 160 | 6·25E−3 | 6 | 1·0 | 2·76E−4 | 0·132 |
| 5 | 2·30E−5 | 1·94E−1 | 3·06E−5 | 160 | 6·25E−3 | 6 | 1·0 | 3·67E−4 | 0·176 |
| 6 | 2·30E−5 | 5·40E−2 | 2·33E−5 | 160 | 6·25E−3 | 6 | 1·0 | 2·80E−4 | 0·134 |
| 7[2] | 6·50E−5 | 3·18E+1 | 6·58E−5 | 50 | 5·10E−4 | 6 | 1·0 | 7·90E−4 | 4·645 |
| 8[2] | 7·62E−4 | 3·88E+1 | 7·72E−4 | 60 | 1·59E−4 | 6 | 1·0 | 9·26E−3 | 174·35 |

[1] Computed according to the proposed algorithm.

[2] Linear transmission.

[3] Critical damping is obtained in the initial position. The change in effective inertia is expected to be negligible because of the high gear ratios, but the manipulator may be slightly under/over damped depending on the manipulator configuration.

Table 1. Parameters used in the controller design.

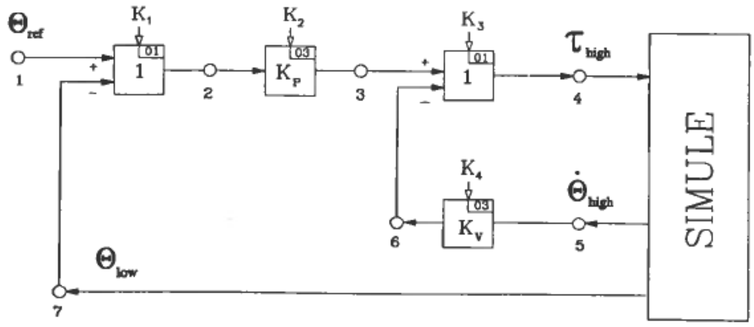

Figure 5. PD controller configuration.

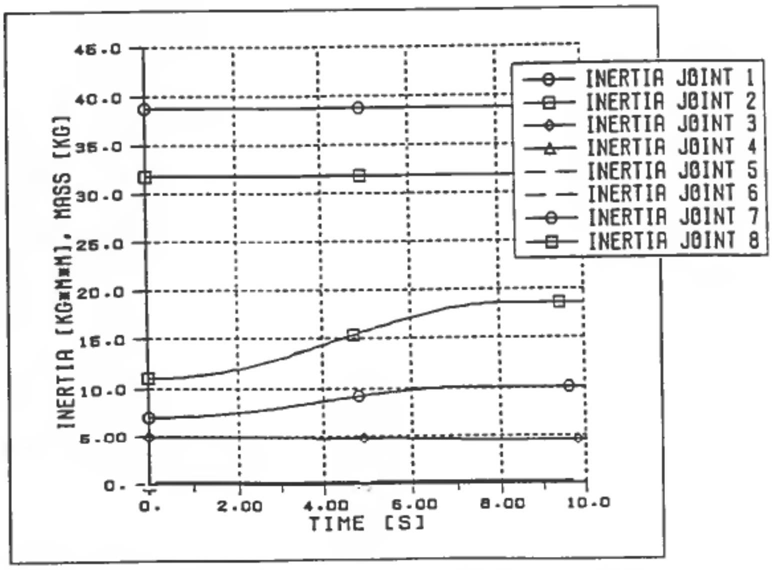

Figure 6. Variations in joint inertias during simulation.

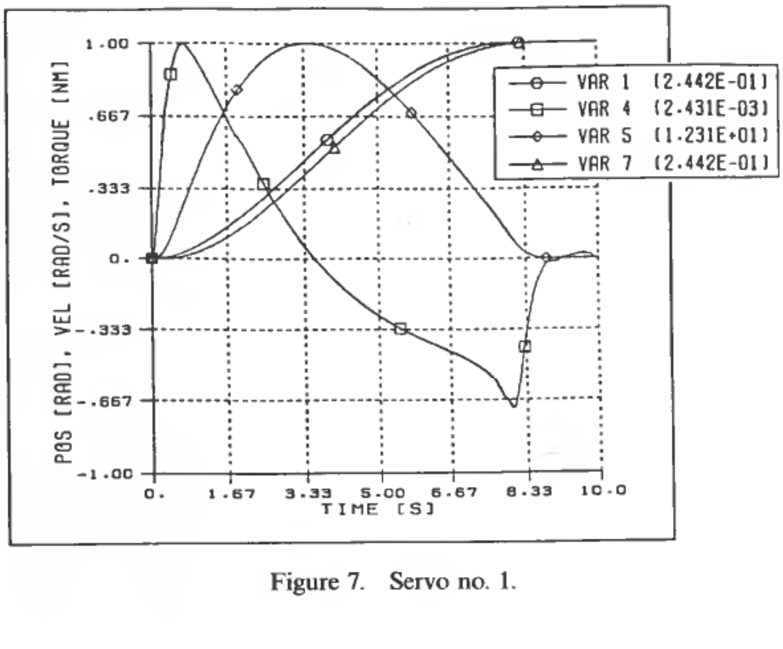

Figure 7. Servo no. 1.

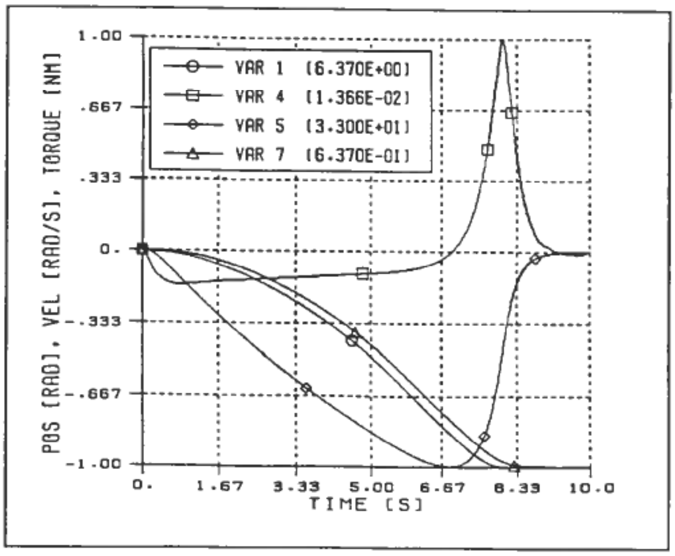

Figure 8. Servo no. 2.

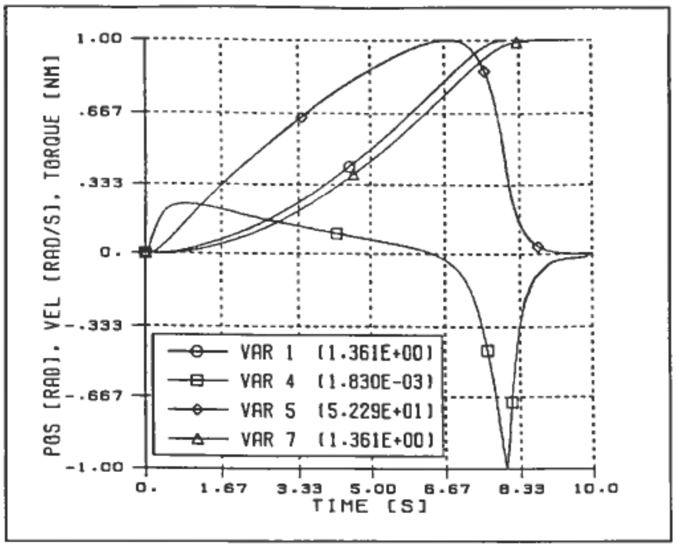

Figure 9. Servo no. 3.

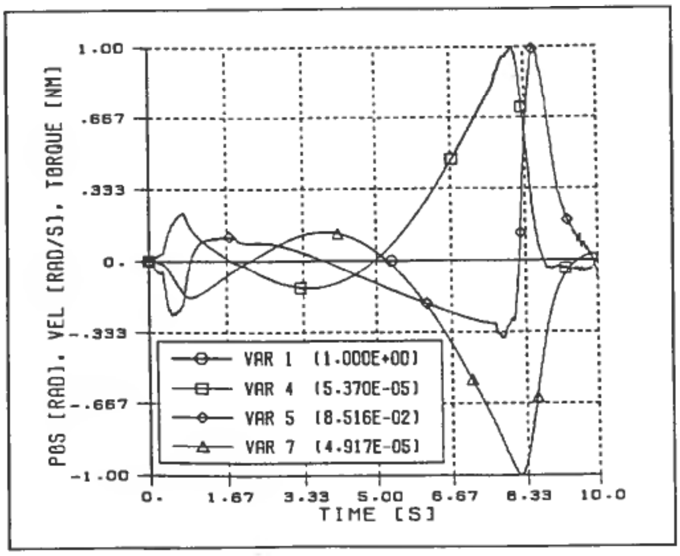

Figure 10. Servo no. 4.

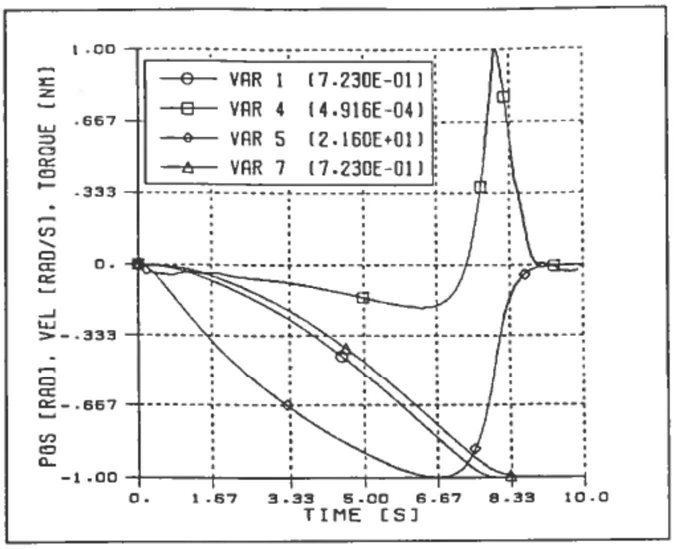

Figure 11. Servo no. 5.

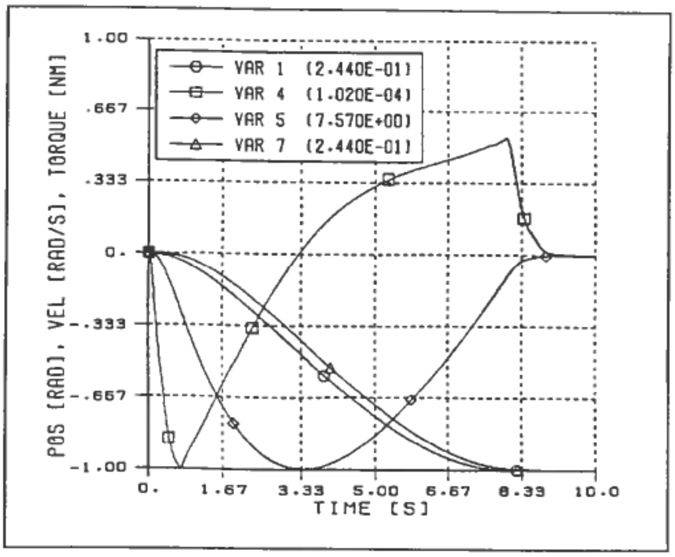

Figure 12. Servo no. 6.

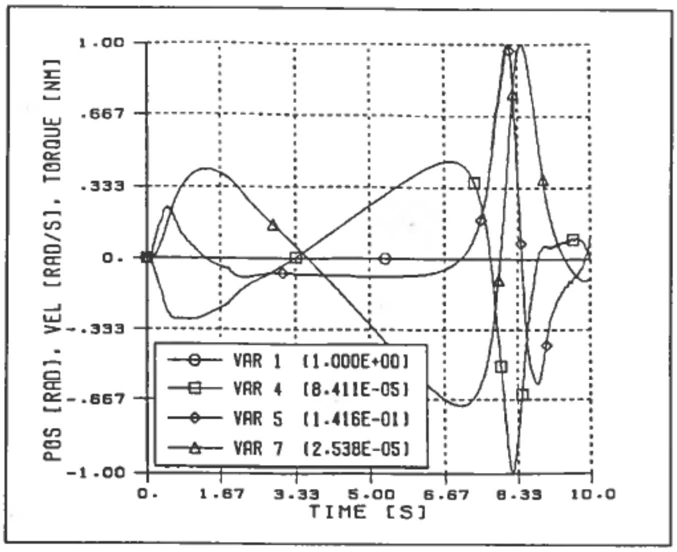

Figure 13. Servo no. 7 (XSGU).

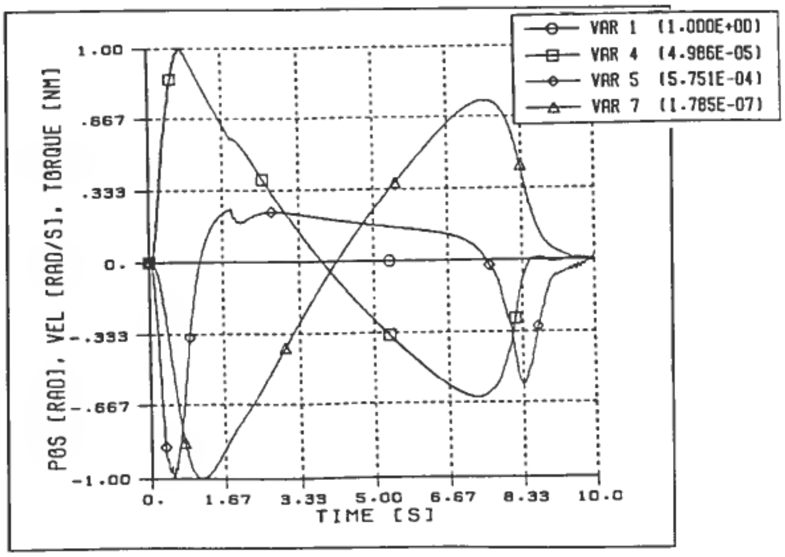

Figure 14. Servo no. 8 (ZSGU).

Table 1 shows that the time varying inertias at the low speed side ($J_{\text{low}}$) of joint no. 1, 2, 3 and 5 are a significant portion (25–50%) of the effective inertia ($J_{\text{eff}}$). To ensure that the EMATS manipulator arm never became underdamped, the value used for each joint inertia is therefore the maximum of the range of values that $J_{\text{low}}$ takes on during simulation. The maximum value of the low speed inertias are included in the effective inertia ($J_{\text{eff}}$) prior to the controller design, and since the manipulator arm was never underdamped, this indicates that the low speed inertias are well approximated by the proposed algorithms (Rølvåg 1992).

Table 1 also implies that it's not correct to make the assumption that the effective inertias for joint no. 1, 2, 3 and 5 are constant. Figure (6) shows how the arm inertia is varying 'seen' from each joint. The effective inertias ($J_{\text{eff}}$) at the high speed side of joint no. 1 and 2 are increasing by 12% and 26% respectively, while the variations in inertias for all other joints are negligible (of no importance).

To ensure that the robot arm is always critically damped, FEDEM must be able to deal with time varying controller gains. This option is planned to be included in future versions of FEDEM.

The feedback joint values $\theta_{\text{low}}$ in figs. 7 to 14 are put together in Fig. 15, and the resulting cartesian motion of the manipulator arm (TCP) is shown in Fig. 16.

Figure (16) shows that the resulting motion has been a straight line between the initial tool frame $\{\boldsymbol{T}\}$ represented by the vector $\chi_{\text{T}}^{\text{T}}=[4{\cdot}3295, -0{\cdot}161, 0{\cdot}175, 0{\cdot}0, 0{\cdot}0, 0{\cdot}0]$ and the goal frame $\{\boldsymbol{G}\}$ given by the vector $\chi_{\text{G}}^{\text{T}}=[4{\cdot}5295, -0{\cdot}161, 0{\cdot}175, 0{\cdot}0, 0{\cdot}0, 0{\cdot}0]$.

### 4.3. *Example 2—simulation with gravity*

Similar simulations including gravity forces ($g_z=9{\cdot}81$), showed that the EMATS manipulator will not perform well on future ground tests with the present choice of servo and transmission systems. The results showed that only the X-Sliding Gantry Unit ($\theta_x$) and the Z-Sliding Gantry Unit ($\theta_z$) performed as required in the specifications.

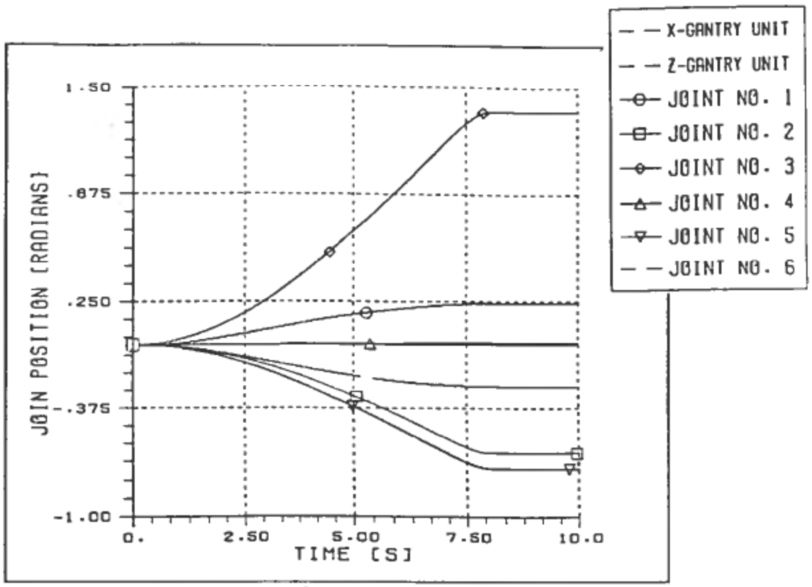

Figure 15. Joint positions ($\theta_{low}$).

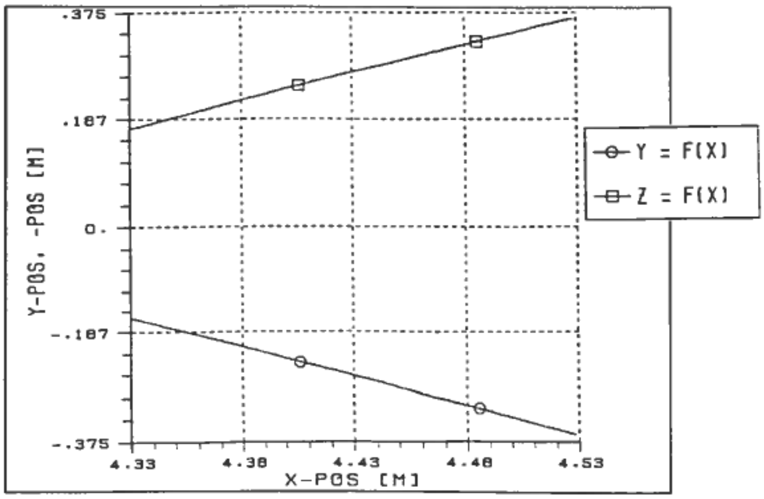

Figure 16. Y and Z positions versus X position.

The ZSGU $(\theta_z)$ which is carrying the weight of the manipulator arm caused an steady state error of the TCP of less than 0·0002 [m].

The other servo loops didn't perform well and caused a steady state error of the TCP of more than 0·12 [m] in the global *z*-direction.

The static arm moment $\tau_{\text{arm}}$, the high speed inertia $J_{\text{high}}$ and the gear ratio $\eta$ are given by the arm configuration and the choice of motor and transmission. The bandwidth of the servo loops $(\omega_b = 1/T)$ were therefore increased from 6 [Hz] to 30 [Hz]. The maximum controller bandwidth should not exceed the lowest structural resonance frequency [10 Hz], however, it was of great interest to verify if some of the 5 structural modes within the controller bandwidth (30 [Hz]) would be excited. Other ways of improving the steady state error would have been to include an integral term in the servo loops (PID control) or by increasing the high speed inertia $J_{\text{high}}$ by using an additional flywheel as used for joint no. 8 (ZSGU).

The controller gains corresponding to the desired bandwidth $(\omega_b = 30\,[\text{Hz}])$ and critical damping $(\zeta = 1)$ were calculated, and the steady state error of the TCP position in the z-direction was improved from 0·12 [m] to 0·11 [m].

Similar curves may be computed for the deflection in x and y direction, and the joint trajectories. This output capability is not a standard FEDEM feature, and requires a kinematic description of the manipulator and some postprocessing of the result files.

Figure 17 shows the deviations from the reference cartesian trajectory caused by elastic deformations and limitations in the PD controllers. The deviations caused by the elastic deformations (curve no. 1) are obtained by using the interpolated joint reference values as prescribed displacements of the joint springs rather than inputs to the servo loops. A major part of this deviation is caused by deflections in gear and transmissions. The joint positions are measured on the low speed (output) side of the transmissions so most of the elastic deflections will be compensated for by the controllers. The remaining elastic deflections in the EMATS manipulator arm (curve no. 3) are obtained as for curve no. 1, but the joint springs which are representing the stiffness in the gear and transmission systems are modeled stiff.

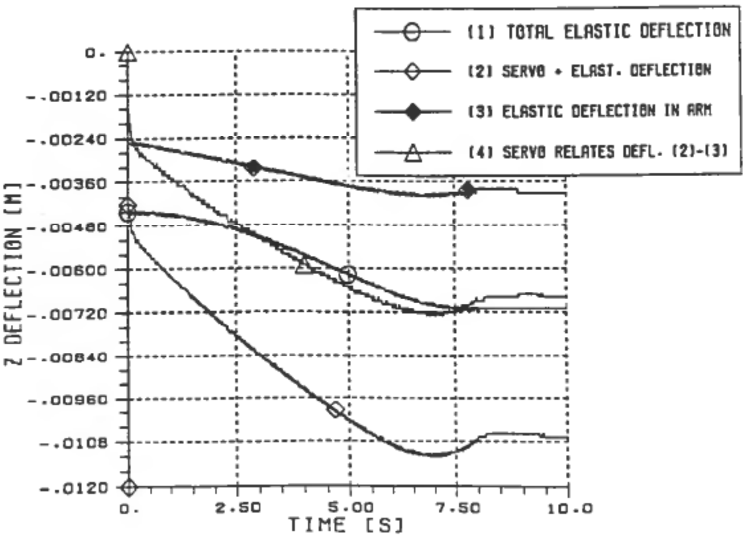

Figure 17. Z-deflection of TCP.

The total z-deflection of the TCP caused by servo limitations and compliance in arm and transmissions is shown in curve no. 2 ($\omega_b = 30$ [Hz]). The difference between curve no. 2 and 3 gives the deflection caused by servo limitations when each joint position is measured at the output side of the gear/transmission. This curve gives information about how much the steady state error can be improved by using other controllers (0·0068 [m] according to curve no. 4).

Assuming perfect control, the minimum z-deflection from the reference trajectory is given by curve no. 3. The arm deflection can only be compensated for by measuring the cartesian TCP position.

Figure (17) also indicates that none of the structural modes within the controller bandwidth of 30 [Hz] are excited. The reason may be the smooth reference trajectory which is generated, and that no disturbances are included in the servo/manipulator model.

## 5. Conclusion

For future space structures and lightweight robot manipulators multidiscipline simulation will become increasingly important. This work contributes to the integration of traditionally separate design disciplines, e.g. structure and controller design. Some of the results presented in this chapter may not be representative of the real EMATS manipulator design, but it outlines the main new multidiscipline features implemented in FEDEM.

The robot control system including the robot language interpretor module, the interpolator module and the kinematic module performed very well. The reference trajectories and information needed for the controller design of rigid body motion were easily specified by the robot programming language. The proposed algorithm provided good estimates of the time varying low speed inertias which were a significant portion of the effective inertias.

The PD controllers were designed according to the maximum value of the effective inertias, but the bandwidth of the servo loops were not limited by the lowest structural eigenvalue. The data flow between the robot control system, the controllers (servo loops) and the mechanical FEM model performed well even if further integration would improve the simulation speed (Rølvåg 1992, Hildre 1991).

The gear and transmission systems in the EMATS manipulator are modeled as described in (Wohlfart 1989, Rølvåg 1992). The proposed modeling technique, based on implicit functions, introduced the desired decoupling effect which enabled control of the rigid body motion of the FEDEM model. However, more efficient gear models should be developed, in order to improve the simulation speed.

All algorithms implemented in FEDEM during this work (Rølvåg 1992, Hildre *et al.* 1990) perform as expected, and provide a high level of integration between the different design disciplines. FEDEM offers a precision in multidiscipline dynamic simulation and modeling within robot applications which hitherto has been impossible to obtain. Commercial FEM like ANSYS, NASTRAN and ABAQUS do not include options for mechanisms and control system modeling and could not have been used.

## Acknowledgments

This work was supported by the Royal Norwegian Council for Scientific and Industrial Research.

## REFERENCES

HILDRE, H. P. (1991). Mechanical Design of High Performance Mechanisms, Division of Machine Design, Ph.D Theses, Norwegian Institute of Technology, Trondheim.

HILDRE, H. P., RØLVÅG, T. and WALØEN, Å. Ø. (1990). Robot Simulation and Optimization for Space Application, *Proceedings of the first International Conference on Dynamics of Flexible Structures in Space, UK, 15–18 May* (Springer-Verlag, New York).

IVERSEN, T. (1989). Control elements in mechanisms simulation. *Modeling, Identification and Control*, **10,** 165–172.

NEWMARK, N. M. (1959). A method of computation for structural dynamics, *Journal Engineering Mechanics Div. ASCE*, **85,** EM3.

PATRAN (1991). User Manual, PDA Engineering, 2975 Redhill Avenue, Costa Mesa, California 92626, USA.

PIEPER, D. L. (1969). The Kinematics of Manipulators under Computer Control. Ph.D Dissertation, Stanford University, USA.

RØLVÅG, T. (1992). Multidiscipline Dynamic Simulation of Elastic Manipulators and Space Structures, Ph.D. Theses, Division of Machine Design, Norwegian Institude of Technology, Trondheim.

RØLVÅG, T. and AAMNES, K., (1991). *FEDEM User's Guide, Finite Element Dynamics of Elastic Mechanisms*, SINTEF Production Engineering, Trondheim.

SIVERTSEN, O. I. and WALØEN, Å Ø. (1982). Non-linear finite element formulations for dynamic analysis of mechanisms with elastic components. *The ASME Mechanism Conference 1982* (ASME, New York).

WOHLFART, K. H., ROTHENBURGER, K. H. and MAU, K. D. (1989). EMATS Kinematic Analysis, EMATS-AND-DS-Vol-2.2, Dornier.