Controls Methods : Integrating MATLAB/MATRIXx with ADAMS

Sudhakar Medepalli, Ph.D.
Consulting Services
Mechanical Dynamics Inc.
Tel: 313 913 2551
email: smede@adams.com

ADAMS is a powerful tool for modeling and simulating large displacement multi-body dynamic systems. In addition to that, some fairly simple feedback control strategies can be implemented in ADAMS using generic system modeling elements such as DIFF, VARIABLE etc. The ADAMS/Linear module enhances these capabilities to include Eigen Value and Eigen Vector solutions as well as generation of Linear State Space Equations of the model about an operating point in a format compatible with standard control system design packages such as MATLAB and MATRIXx. Once the linear controller is designed, it can be read into ADAMS using the MATRIX, ARRAY, TFSISO and LSE elements and incorporated into the dynamic model to obtain the closed loop system. Non-Linear controllers can be implemented through the user-written subroutines such as VARSUB, DIFSUB, SFOSUB, GFOSUB etc. Note that these subroutines can be written by hand by the user or created directly by MATLAB or MATRIXx autocode generators. Minor modifications are necessary in the latter case to bring them into the format of ADAMS user-written subroutines.

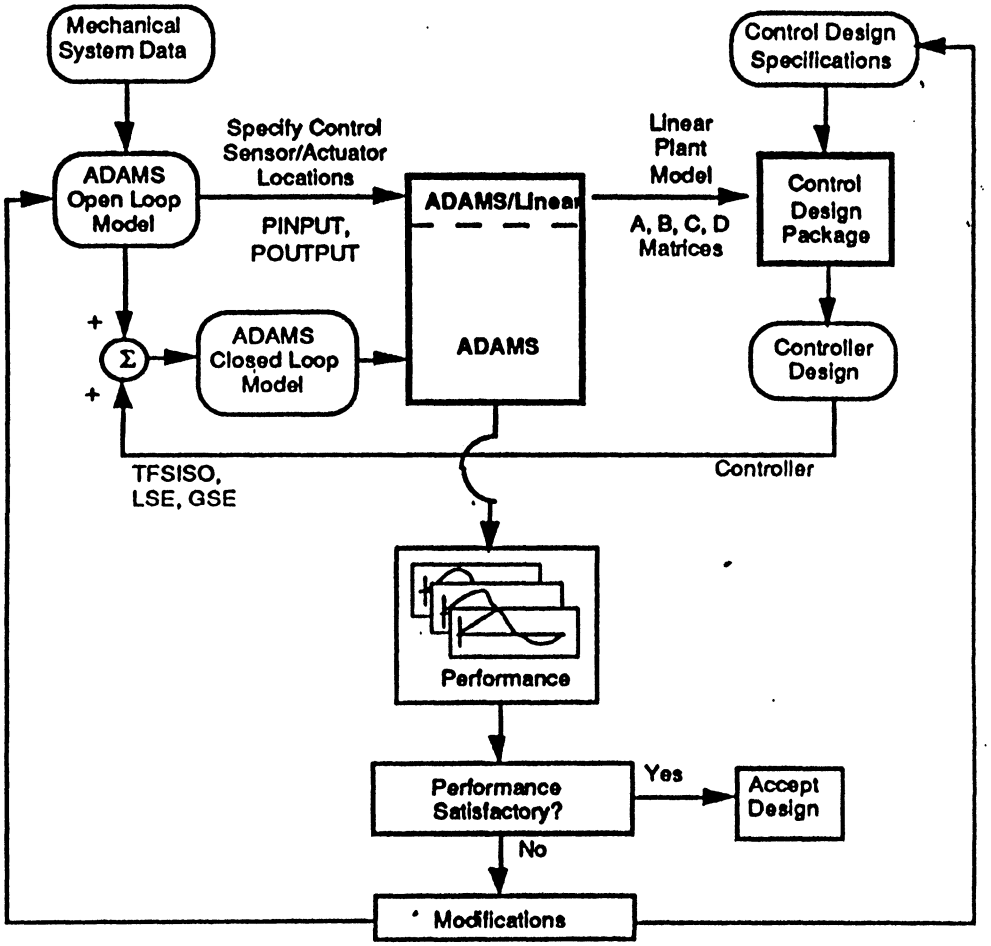
In order to implement the above mentioned controllers in ADAMS, the user needs to create several intermediate ADAMS elements such as VARIABLES, ARRAYS, MATRICES, LSE, TFSISO etc. which need to be properly related to the inputs and outputs of the open loop model under consideration using PINPUTS and POUTPUTS. This can be a little tedious at times, especially during design experimentation stage where several different control strategies need to be considered and implemented before a satisfactory design is obtained. Moreover, one needs to constantly switch back and forth between ADAMS (Solver or View) and MATRIXx or MATLAB sessions. The Concurrent Engineering Toolkit developed by Mr. Thierry Bernard of MDI France eliminates most of this inconvenience by facilitating open loop model building, controller design, analysis and synthesis through a single ADAMS/View session.

Once the toolkit is properly installed, some additional functionality is obtained in ADAMS/View. One of these is the "ANALYSIS —> SUBMIT —> SINGLE ANALYSIS" panel which is customized to include, among other things, option to perform a control system design. Plant inputs and outputs can be defined on the fly and Eigen Analysis can be peformed after a static or dynamic simulation. The state matrices of the open loop model are generated and we have the option to start an interactive MATLAB or MATRIXx session from within ADAMS/View. Once a controller is designed satisfactorily, it can be integrated back into View nad incorporated into the model automatically to obtain the closed loop model. The user also has the option to obtain Multi-Input Multi-Output frequency

response of a model and plot the results inside View. Cross-correlation between two result set components can also be obtained using the toolkit.

In order to illustrate the various features of the toolkit discussed above, we start with a simple inverted pendulum model, show the functionality of ADAMS/Linear with the help of the model and design an LQG controller for it using the conventional method, i.e., without using the toolkit. We then introduce some of the features of the toolkit and re-design the controller interactively using it and import the controller into ADAMS/View. We should mention that we only cover the control system design related functionality of the toolkit. It has many other features which are fully described in the Concurrent Engineering Toolkit User's manual. Similarly, the syntax and usage of various control system modeling elements in ADAMS are beyond the scope of this presentation and the interested users can refer to the ADAMS/View, ADAMS/Solver and ADAMS/Linear manuals for the same. The following pages show the functionality of ADAMS/Linear followed by the features in the Concurrent Engineering Toolkit that are useful for control system modeling.

Integrated control design and simulation methodology



Controller Design Elements in ADAMS

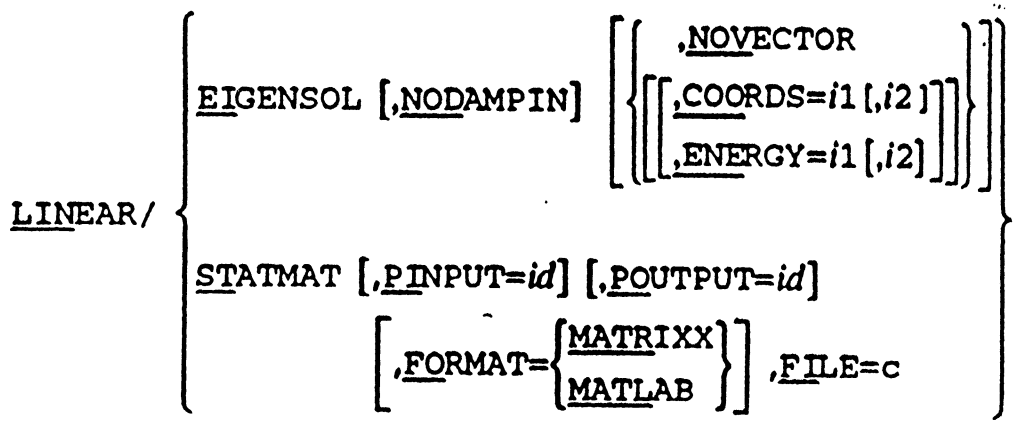
- **Algebraic Elements**
  1. VARIABLE
  2. ARRAY
  3. MATRIX

- **Inputs and Outputs**
  1. PINPUT
  2. POUTPUT

- **Dynamic Elements**
  1. DIFF
  2. TFSISO
  3. LSE
  4. GSE

- **Commands**
  1. LINEAR

Review of LINEAR/ command.



{ }	Select one item
[ ]	Optionally select the item
[[ ]]	Optionally select an item combination

EIGENSOLUTION

Generalized condensed eigenproblem

$$\mathbf{K}\,\mathbf{z} = \mu\,\mathbf{M}\,\mathbf{z}$$

μ=Eigenvalue
Z=Eigenvector

- Row and column size of **K** and **M** matrices= 2*number_of _dof + number of dynamical states

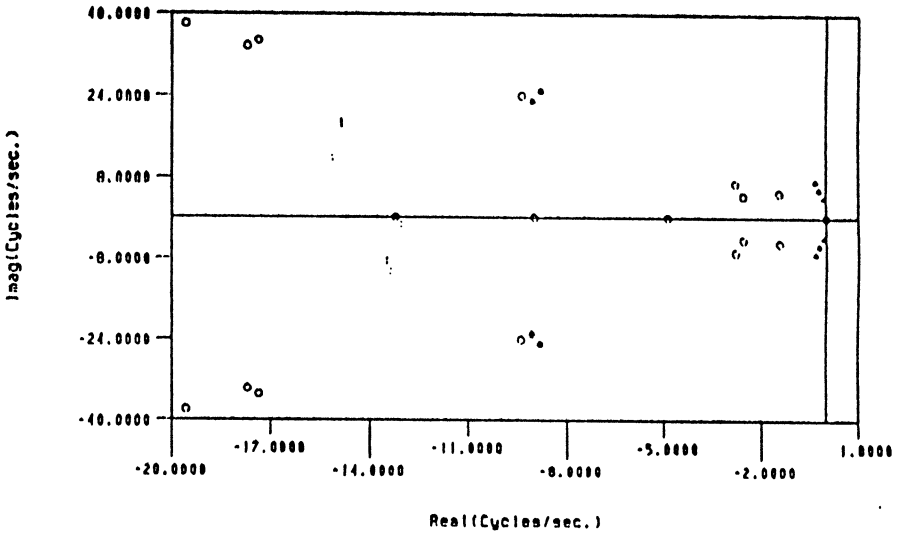
- Eigenvalue table reported to
  - Screen/Log file
  - Output (*.out) file

- Eigenvalues and Mode shape written in results (*.res) file

- Coordinate and Kinetic Energy distribution tables reported in output (*.out) file.

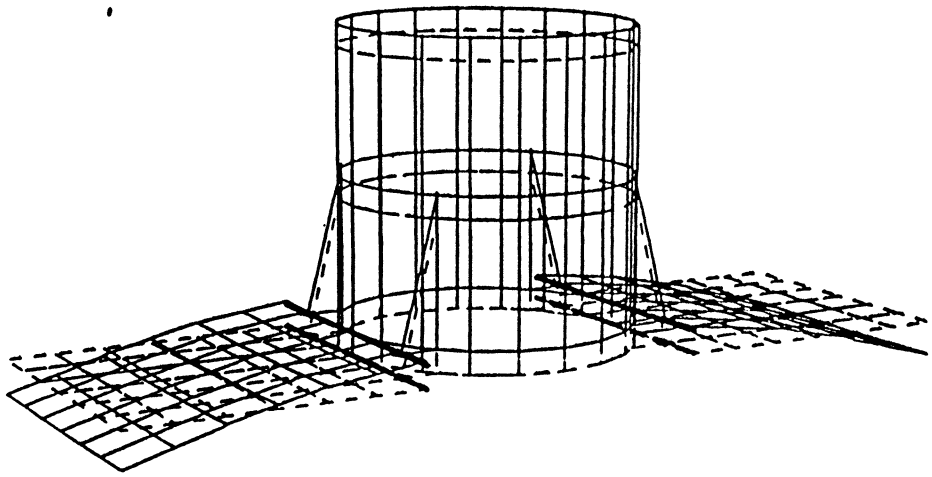
EIGENSOLUTION - VIEW post processing

- Complex scattering - pole plots



- Deformed mode shape display

- Mode shape animation



STATE MATRIX OUTPUT (Cont'd)

Information output by ADAMS

	MATLAB	MATRIXx
[A, B, C, D]	**mata, matb, matc, matd**	**mat**
States identifiers	**matst**	
Pinput identifiers	**matpi**	
Poutput identifiers	**matpo**	

3.5. Plant export & continuous LSE controller import

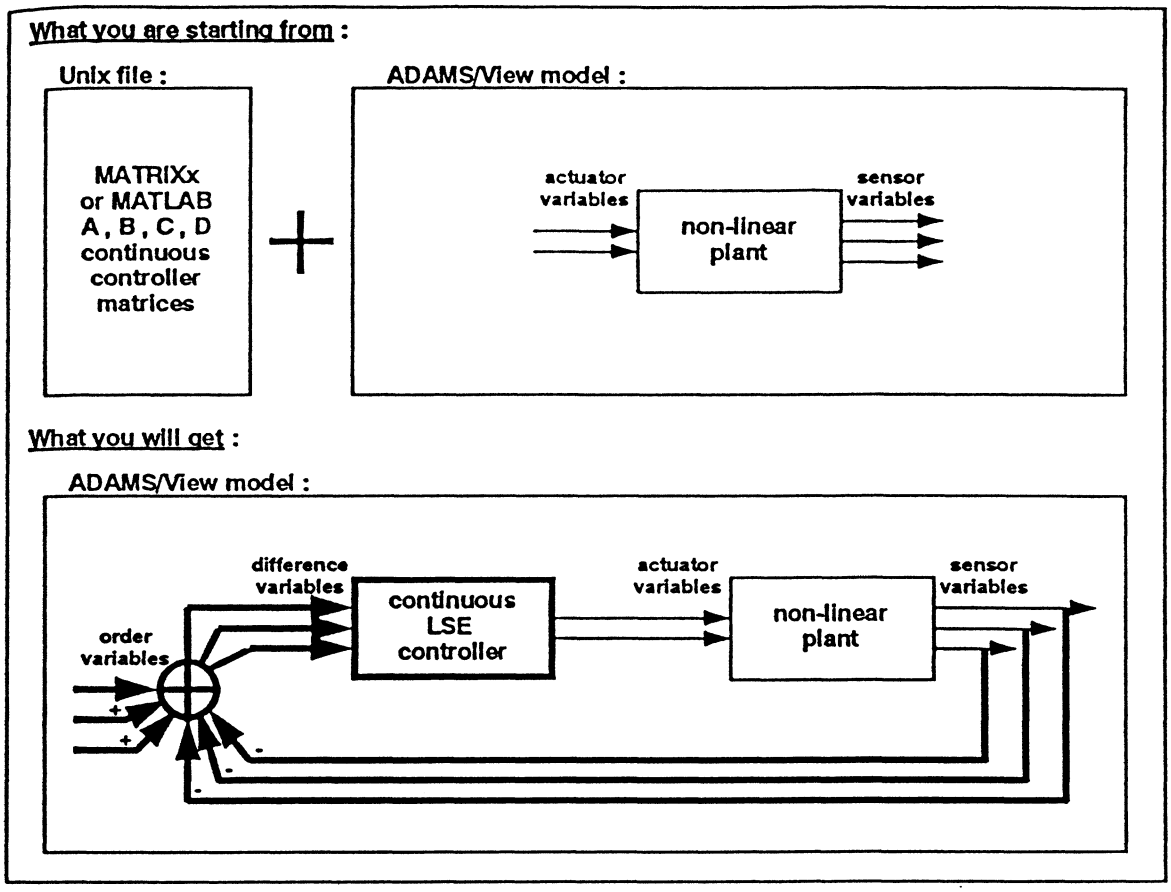
Access	MAIN - ANALYSIS - SUBMIT - SINGLE ANALYSIS - MORE - Control System Design fields MAIN - PREPROC. - PART - CREATE/DELETE - CONTROLLER - CONTINUOUS LSE
Structure	VARIABLEs : $'ctrl_name'_'sensor_name'_order/diff MATRIXs : $'ctrl_name'_A/B/C/D ARRAYs : $'ctrl_name'_U/X/Y LSE : $'ctrl_name'
Visualisation	Solver VARIABLEs and LSE icons
Limitation	

Plant export feature is enabled while using the **advanced analysis submit panel** with the following requirements :

- select **"CSD_session" option** ("xxxxx_file" will generate state matrices and put it with corresponding product startup file in some new directory ; "xxxxx_inter" will do the same and open some corresponding interactive control system design session),
- select "CSD_input_variables" to specify plant inputs,
- select "CSD_output_variables" to specify plant outputs,
- select initial or final eigenvalues analysis,
- simulate .

When using some interactive control system design session option, user can then automatically recover the plant state matrices with commented inputs, outputs and states names & values into his favorite control system design package .

Figure 9 : Continuous LSE controller impōrt .



Continuous linear controller import process requires the following previous control system design session :

- **ISI/Matrixx-Xmath** :
  - rename controller as "main.comp" dynamic system,
  <command line> adams_ex

- **TMWI/Matlab** :
  - rename controller matrices as "comp_a , comp_b , comp_c , comp_d , comp_x0 , comp_dt",
  <command line> adams_ex

, wich produces the required Adams matrices file to be imported into ADAMS/View model .

3.6. C or Fortran non-linear sampled controller import

Access	MAIN - PREPROC. - PART - CREATE/DELETE - CONTROLLER - GENERAL SAMPLER/DISCRETE
Structure	VARIABLEs : $'sampler_name'_'sensor_name'_order , $'sampler_name'_'sensor_name'_diff/chan SENSORs : $'sampler_name' , $'ctrl_name' Fortran extension : **varsub11.f**
Visualisation	Solver VARIABLEs and SENSORs icons
Limitation	one single controller output sampling period

C or Fortran non-linear sampled controller import process requires the following previous control system design session :

- **ISI/Matrixx-Systembuild** :
<systembuild window> File - Generate real-time code (use Procedure only option)

- **TMWI/Matlab-Simulink** :
<simulink window> Code - Generate code only

, wich produces the required C file to be simulated using corresponding ADAMS/Solver extensions .

ADAMS/View corresponding import process consists of the following two steps (see following pages pictures) :

- definition of controller inputs sampling (possible multiple sampling period),
- definition of controller outputs single sampling .

In **order to simulate** corresponding C or Fortran controller, **user must fill "user_subroutines" field of advanced analysis submit panel with local "varsub11.f" file,** to enable automatic executable creation before analysis if necessary .

Figure 10 : C or Fortran controller import : sampler .

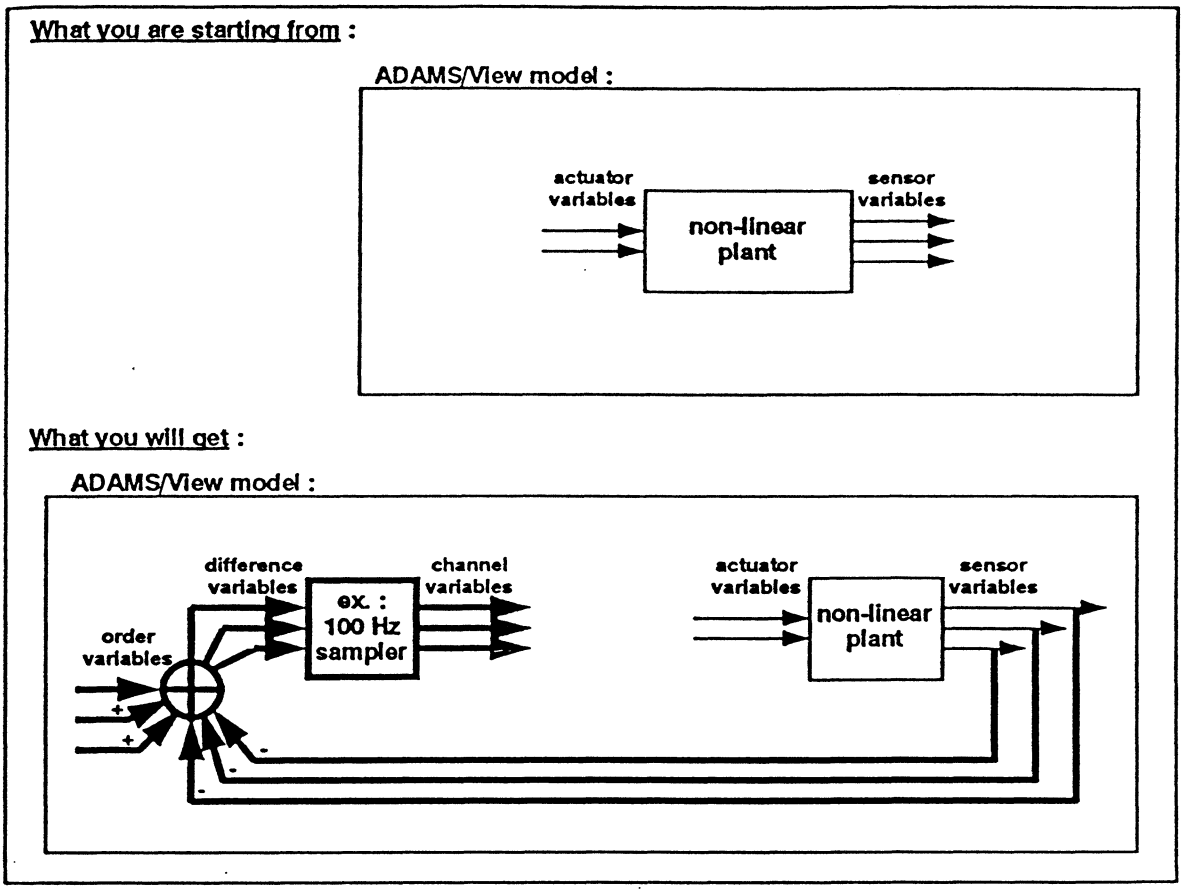
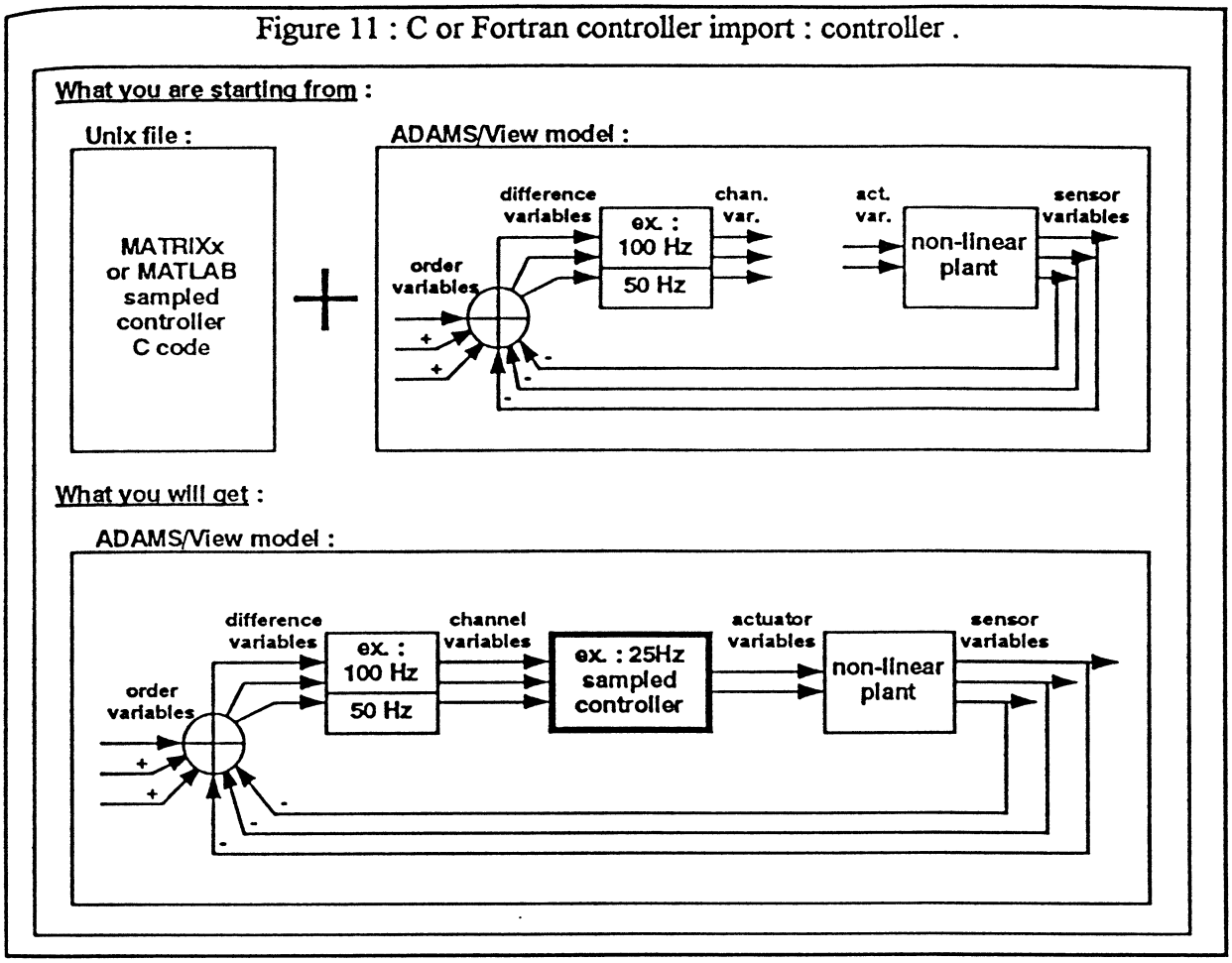


Figure 11 : C or Fortran controller import : controller .



3.7. Frequency response & cross spectrum

Access	MAIN - ANALYSIS - SUBMIT - SINGLE ANALYSIS - MORE - Control System Design fields MAIN - POSTROC. - XY_PLOTS - CROSS SPECTRUM
Structure	
Visualisation	
Limitation	

By selecting "MATRIXx_batch" (resp. "MATLAB_batch") as value for **"CSD_session" field of advanced analysis feature** together with appropriate selection of "initial" field as "static_eigen" or "final" field as "static_eigen" or "eigen", and with appropriate selection of "CSD_input_variables" and "CSD_output_variables", ADAMS/View submits some corresponding **ADAMS/Linear analysis chained by corresponding multi-input/multi-output frequency response** analysis using ISI/Matrixx (resp. TMWI/Matlab) batch mode . **Results are automatically plotted inside View** with full comprehensive comments at the end of partner product batch analysis .

Based on the same principle of using adequat specialized partner product, but using know non-linear transient result, the cross correlation spectrum feature enable users to **calculate and automatically plot inside View cross correlation spectrum of two result set components** with the following possibilities :

- user choose twice the same result set component : auto correlation spectrum is calculated and plotted,
- user choose two different result set components : cross correlation spectrum is calculated and plotted together with estimated frequency response of plant between first result set component as input signal and second as output one .