The ADAMS/Controls and ADAMS/Plant Products: An Example Combining the Power of ADAMS 9.0 and Matlab 5.0

Patrick J. McNally, Mechanical Dynamics, Inc.
Scott Gray, The Mathworks, Inc.

Abstract

ADAMS is a powerful tool for mechanical system modeling, simulation, and visualization. Many mechanical systems also include sophisticated controls for defining the motions the system is to follow. For instance, a robot arm is composed of mechanical parts, constraints, and forces, but to make it move along a desired path, a control system is required. Matlab, MatrixX, and Easy5 are powerful tools for control system modeling and design. Combining the power of ADAMS with Matlab and other controls packages is highly desirable to design and test mechanical systems with controls.

This paper describes two new products that will be released in 1997 for combined design and analysis of mechanical systems with controls: ADAMS/Controls and ADAMS/Plant. Two methods of combining ADAMS with controls programs are described: control application-based model integration, and cosimulation. In control application-based model integration, the integration of the ADAMS model is done from within the control block diagram environment, along with any system states required to model the controls. In cosimulation, the ADAMS model integration is performed by ADAMS and the control integration is performed by the control applciation. Both methods are now implemented in an intuitive user interface in the ADAMS/Controls. The control application-based model integration is implemented in the ADAMS/Plant product.

An example is used to illustrate the design process of adding complex controls from Matlab/Simulink to an ADAMS model. The example illustrates the benefits of using both mechanical system simulation tools and controls design tools for tackling complex design problems.

Introduction:

Combining design and analysis tools into integrated environments has the potential to increase productivity and dramatically shorten design cycles. Currently, control design is done utilizing sophisticated control analysis packages such as Matlab (with Simulink) , MatrixX (with SystemBuild), or Easy5. These packages provide the latest in control design and analysis techniques, such as LQG design, robust control, and nonlinear control. However, these packages require the user to provide a dynamic model for the controller. This is done by building up transfer functions or providing user code. Simplifying assumptions are always made in providing a dynamic model, even though the control techniques may be very high order and quite sophisticated.

Likewise, dynamic modeling and simulation is done using ADAMS, NASTRAN, or other packages with limited or cumbersome methods for incorporating sophisticated controllers. The primary outputs of such simulation packages are the time history of states under a certain set of

predetermined forcing conditions. Many runs may be required to gain an understanding of the dynamic model response to various inputs.

This paper discusses products to be released in summer 1997 for seamlessly combining control application models with ADAMS models. The key development which allows these products is the existence of partitioning integrators in ADAMS, to segment the dynamic states of the mechanical model into independent and dependent states. An example problem is given which controls a highly nonlinear plant using a time varying nonlinear controller.

Two Products, Two Target Users

Two products have been created to provide the capability to simulate complex nonlinear mechanical systems with complex control laws. ADAMS/Plant has been created for the control system designer that does not currently use ADAMS. ADAMS/Controls has been created for the existing ADAMS users and organizations that need to add controls to their current ADAMS models. The two products are discussed in more detail in this section.

ADAMS/Plant

The ADAMS/Plant product has been created for the control system designer that needs to create nonlinear mechanical models without writing complex equations. This product was created with the non-ADAMS user in mind. The ADAMS/Plant product is bundled with ADAMS pre- and post-processing capability but without the full ADAMS integration engine. With ADAMS/Plant, the mechanical system and controls are simulated using the control application integration engine. For many mechanical systems, ADAMS/Plant provides adequate integration power along with the ability to create mechanical systems graphically.

ADAMS/Controls

The ADAMS/Controls product is an add-on product to ADAMS. This product has been created for the ADAMS user that needs to add sophisticated controls to their ADAMS models and analyze these new models in either the control application environment, or in the ADAMS environment. In the control application environment, the mechanical system and controls can be simulated using the control application integration engine, or the system can be setup so that ADAMS solves the mechanical system equations and the control application solves the control system equations. Finally, ADAMS/Controls provides the capability to read your control equations into ADAMS for full simulation using various code generation facilities such as the Real-Time Code Generation facility provided with Matlab.

Two Methods for Integrating Mechanical Systems and Controls

Two methods are available for evaluating the full system with controls. Both methods are supported in ADAMS/Controls. Only the function evaluation method is supported in ADAMS/Plant. These methods are described in this section.

1) Function Evaluation Method

We can partition the mechanical equations and setup a method to evaluate these equations numerically from the control application. This method is called the function evaluation method.

Any combined continuous plant and continuous-design control law can be reduced to the block diagram shown in figure 1. In this diagram P represents the plant equations, or for our purposes the part of the overall system that is modeled in ADAMS, and C represents the controller equations, or the part of the system that is modeled using the control package. Note that P could contain control laws as well, using the ADAMS function expression facility, and C could contain models of mechanical elements.

We will represent P by the following set of equations:

$$\begin{aligned} \dot{\mathbf{x}} &= f(\mathbf{x}, \mathbf{u}, t) \\ \mathbf{y} &= h(\mathbf{x}, \mathbf{u}, t) \end{aligned} \tag{1}$$

where $\mathbf{x}$ is the vector of all of the plant state variables. This includes velocities and positions for each of the parts, as well as constraint Lagrange multipliers and other added equations. In this equation, $\mathbf{u}$ is the vector of all of the plant external inputs and $\mathbf{y}$ is the vector of all of the plant external outputs that will be used to formulate the control law. In general, $f(\ldots)$ is a set of complex nonlinear differential and algebraic equations that include the constraint equations and $h(\ldots)$ is a set of complex nonlinear equations.

We will represent C by the following set of equations:

$$\begin{aligned} \dot{\mathbf{z}} &= g(\mathbf{z}, \mathbf{y}, t) \\ \mathbf{u} &= k(\mathbf{z}, \mathbf{y}, t) \end{aligned} \tag{2}$$

where $\mathbf{z}$ is the vector of all of the control state variables. In general, $g(\ldots)$ is a set of complex nonlinear differential and algebraic equations and $k(\ldots)$ is a set of complex nonlinear equations that describe the control law to be implemented.

One method of integrating the complete set of equations (1) and (2) is to set up the control system program to evaluate P as a set of subsystem equations without knowing the explicit form of the equations. All of the major controls programs such as Matlab/Simulink, Xmath/SystemBuild, and Easy5 allow the user to specify a subsystem block of differential and/or algebraic equations that can be called iteratively by the control program integrator and solved numerically.

ADAMS constructs the set of equations P in a greatly expanded form in order to perform a simulation. The control program needs a method for calling this subsystem equations and constructing the appropriate Jacobian matrix representation.

ADAMS models can become large and complex when compared to the size of typical control system simulations, so it makes sense to try to limit the number of states that the control program actually needs to integrate. Considering holonomic systems only (systems with constraint equations that are represented by algebraic equations - the vast majority of systems of interest here), we can partition the set of equations P into the following reduced set of differential and algebraic equations:

$$\dot{\mathbf{x}}_1 = f_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{u}, t)$$
$$0 = f_2(\mathbf{x}_1, \mathbf{x}_2, t) \tag{3}$$

Where now $\mathbf{x}_1$ is a set of independent variables (equivalent to twice the number of degrees of freedom of the system) and $\mathbf{x}_2$ is a set of dependent variables that can be found by solving the implicit algebraic equation f_2. Note that the first equation is a differential equation in the independent variables only. (see references 1 and 2 for a discussion of the partitioning algorithms in ADAMS).

So through the process of coordinate partitioning, we can reduce the set of differential equations that the control program needs to solve. ADAMS will ensure that the dependent variables are calculated properly at any iteration by first solving the implicit algebraic equation f_2. These dependent variable values are then substituted into the equation f_1 and the set of differential equations can be solved by the control programs. In function evaluation mode, the ADAMS code reads in the dynamic model, constructs the partitioned set of equations, and evaluates the function expressions of equation (3). The control program then numerically integrates for the variables $\mathbf{x}_1$ and $\mathbf{z}$ as a function of time.

2) Cosimulation Method

Suppose our plant model is numerically very stiff and cannot be integrated by the control application. We can construct a methodology for integrating the stiff dynamic states of the model using the ADAMS integrators in full simulation mode. For purposes of this paper, this method is called the cosimulation method.

Any combined sampled data plant and hybrid (discrete plus continuous) control can be reduced to the block diagram shown in figure 2. We will restrict ourselves to a single sample rate for this discussion. Now represent the plant P_s with the following set of equations:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathcal{U}_k, t)$$
$$\mathbf{y} = h(\mathbf{x}, \mathcal{U}_k, t) \tag{5}$$

In full simulation mode, ADAMS will integrate the equations (5), considering $\mathcal{U}_k$ to be constant on the interval [kT, (k+1)T). Then ADAMS will provide the outputs $\mathbf{y}$ at the end of the interval and the control application will compute $\mathcal{U}_{k+1}$ for the next interval. This method mimicks a microprocessor based sampled data control system. A rule of thumb is to sample the ADAMS system at least 5 times as fast as the required controller bandwidth.

Design Approach to ADAMS/Plant and ADAMS/Controls

The initial design of ADAMS/Plant and ADAMS/Controls used "callable ADAMS", which allows an ADAMS simulation to be driven by an external program. This design is now being upgraded to utilize the latest solver toolkit for interfacing View and Solver.

The process of setting the system up for evaluation is simple.

Mechanical System Setup

1) create input and output variables in your ADAMS model to "hook up" to your controller. Output variables are any valid ADAMS function expression that will be used as the outputs **y**
2) Input variables are set to any function expression string (such as "0.0") and will be used as the inputs **u** from the controller
3) Within your ADAMS model, designate the input forces and torques by defining their function to point to the value of the input variables
4) Save an ADAMS dataset (.adm file) of your model with the defined input and output variables

Now you are ready to create your controller using the block diagram editor of your control application. Your ADAMS model is directly connected to the controller using an interface block that has been created for each control application. The interface block allows specification of the model and variables to use for input and output. It includes code required to communicate between the control application and the ADAMS function evaluation code or the ADAMS solution engine itself.

Controller Setup

Your ADAMS model is directly connected to the controller using an interface block that has been created for each control application. The interface block allows specification of the model and variables to use for input and output. It includes code required to communicate between the control application and the ADAMS function evaluation code or the ADAMS solution engine itself. The dialog boxes for the interface block are implemented using the tools provided by each of the control applications so they are familiar to the end user.

Example: Robot Path Control using Simulink

Consider the problem of controlling the robot end effector of a two degree of freedom planar robot to move along a path by commanding joint torques. For this problem, the plant dynamics cannot be effectively linearized around an equilibrium position since the motions are large and the linear equations would not hold except over very small motions (see reference 3).

For this system, a sliding mode controller was implemented in Simulink version 2.0 running with Matlab version 5.0 (see figure 3). The system has two inputs, the joint torques, and four outputs, the joint angles and angular velocities.

A simulation was performed to move the two joints along sinusoidal paths. Joint 1 (the shoulder joint) was given a reference signal of . Joint 2 (the elbow joint) was given a reference signal of . A screen shot of the resulting scope outputs of the control signals and the joint angles is shown in figure 4. A more realistic model would include dynamics for the actuators, including torque limiting effects. The resulting animation of the robot model, with the Simulink command torques and the resulting motions, is shown in figure 5.

This simple example shows the value of using a combined ADAMS and Simulink interface for controling nonlinear mechanical systems. This same capability exists with SystemBuild (MatrixX interface) and with Easy5.

References:

1) "Code Notes: Coordinate Partitioning in ADAMS/Solver, An Explanation of the Basics," J. F. McGrath, Mechanical Dynamics, Inc. MDI-930801, October 19, 1993

2) "Why Do We Need Coordinate Partitioning Anyway?," J.F. McGrath, Mechanical Dynamics, Inc. MDI-940602, February 21, 1996

3) **Applied Nonlinear Control**, J-J. E. Slotine, W. Li, Prentice Hall, 1991

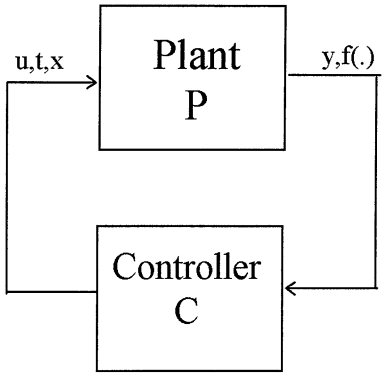

Figure 1: Reduced Form of Continuous Plant Control

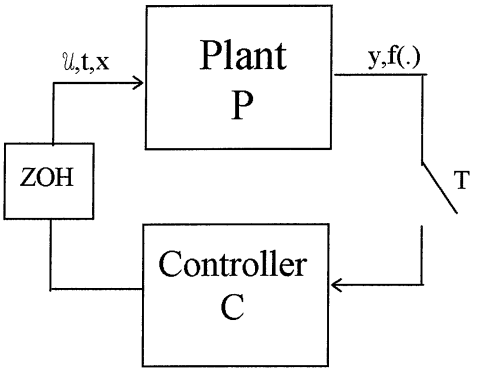

Figure 2: Reduced form of Sampled-Data plant control

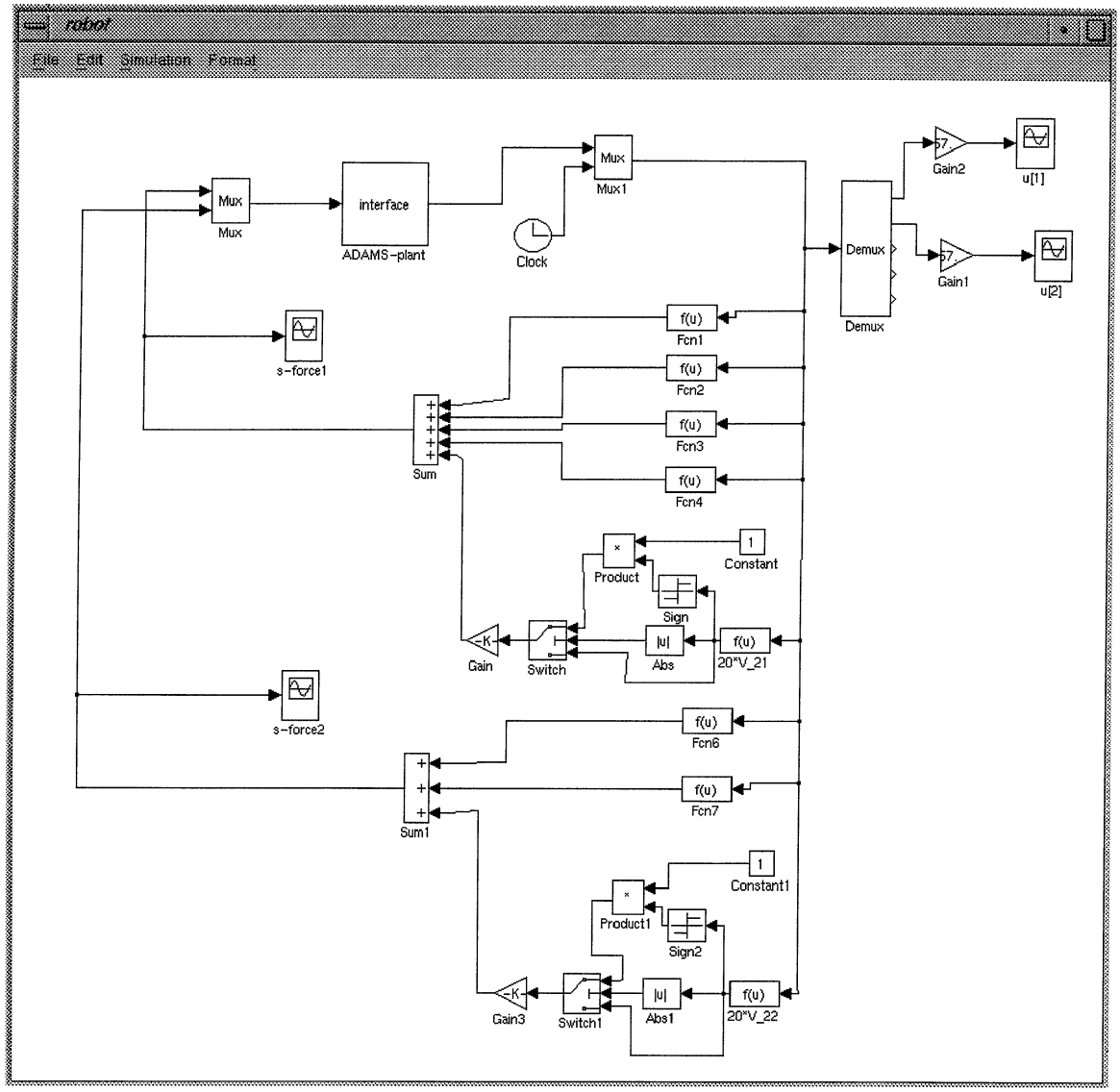

Figure 3: Sliding Mode Control Law for Robot Example in Simulink

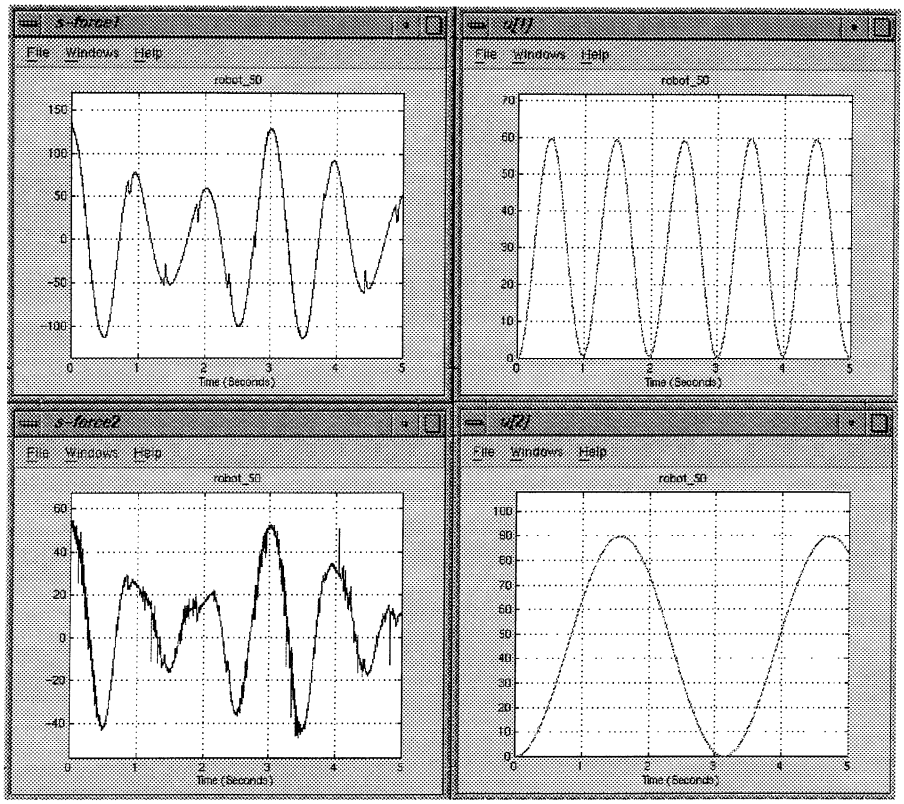

Figure 4: Joint Torques (Inputs) and Actual Joint Angles (Outputs) for Robot Example

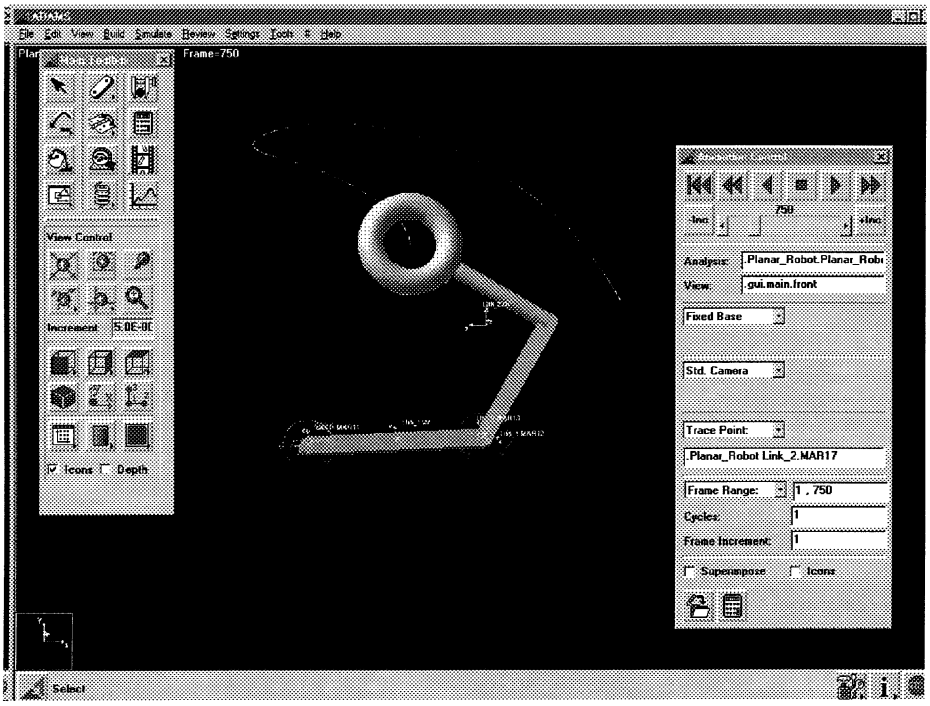

Figure 5: Final Robot Animation Showing Smooth Movement Along Controlled Path