Double Offset CV Joint Analysis Program Based on ADAMS

By

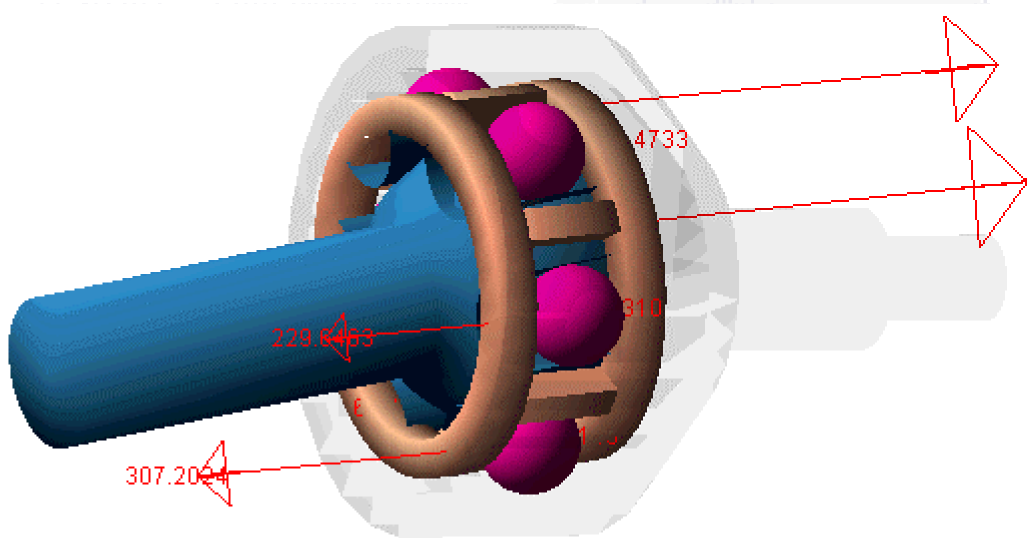
Paul Klevann
Dana Corporation

John Park
Mechanical Dynamics, Inc.

ABSTRACT

This paper presents an ADAMS-based CAE tool for automated model generation and standardized dynamic simulation of double offset type constant velocity (CV) joints for automotive driveline applications. Its model generation is performed either by graphical input panels or by a model file in neutral file format. Analysis results are viewed in both animation pictures and plots.

ADAMS



Double Offset CV Joint Analysis

2000 ADAMS International User's Conference, Orlando, FL

John Park -- MDI, Central Region

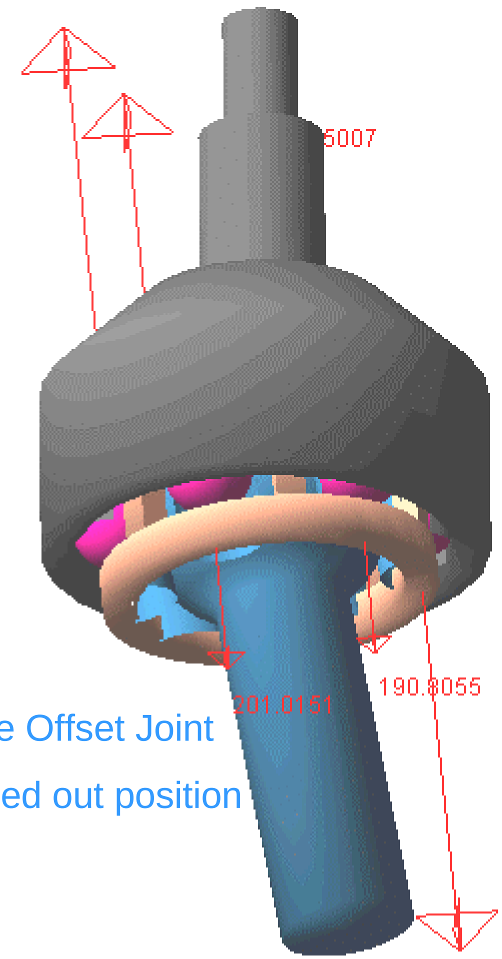
Paul Klevann -- Dana, Technical Resource Park

Mechanical Dynamics

ADAMS

Table of Contents

- Introduction
  - What is CV Joint?
  - Construction of CV Joints?
  - How CV Joints work?
  - Types of CV Joints?
- Double Offset CV Joint Model
  - DOJ Program Structure
  - Automated Model Generation
  - Sample Simulation Results
- Appendix A
  - Sample files for DOJ model
- Appendix B
  - Guidelines for any CV joint-related consulting project



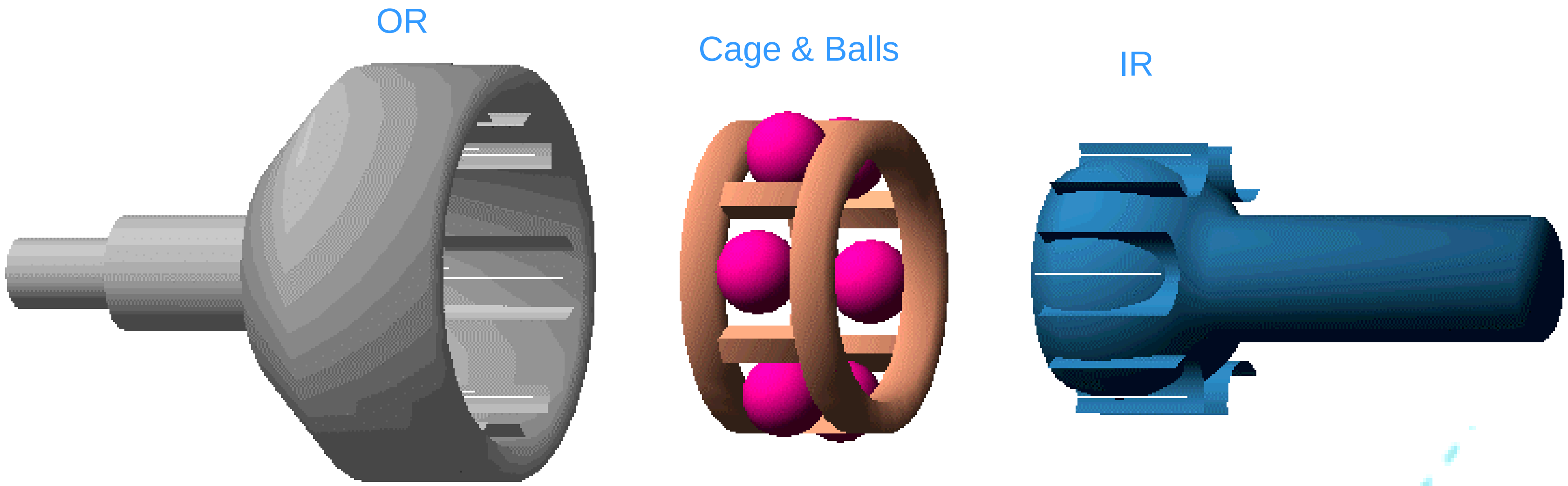
Double Offset Joint
at plunged out position

What is CV Joint?

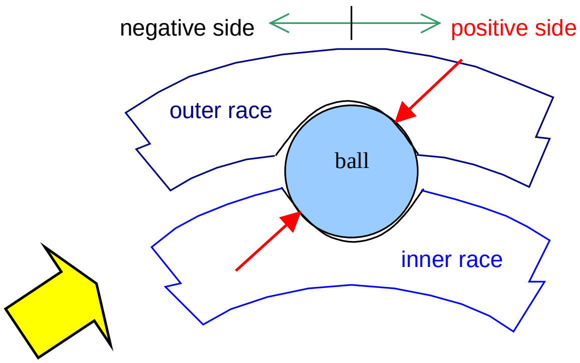
- **CV joint is a driveline component whose output speed equals to the input speed at any joint angle**
  - **Examples of CV joints** ➨ Rzeppa joint, Cross Groove joint, Double Offset joint
  - **Examples of non-CV joints** ➨ Universal joints (Cardan joints), Double Cardan joints
- **CV joints allow the driveline to have**
  - **Articulation DOF** ➨ allows joint angle
  - **Axial Plunge DOF** (some free type CVJ) ➨ allows end motion

Construction of the CV Joints?

- Outer Race (OR)
- Inner Race (IR)
- Cage (Retainer)
- Balls



How do CV Joints work?



- Torque is transmitted via couple of steel balls squeezed between the IR and OR grooves
- The balls are steered (located) to the bisecting angle plane by intersecting IR and OR groove centerlines
- The groove centerline intersection is achieved by the use of axially offset curved grooves or inclined straight grooves

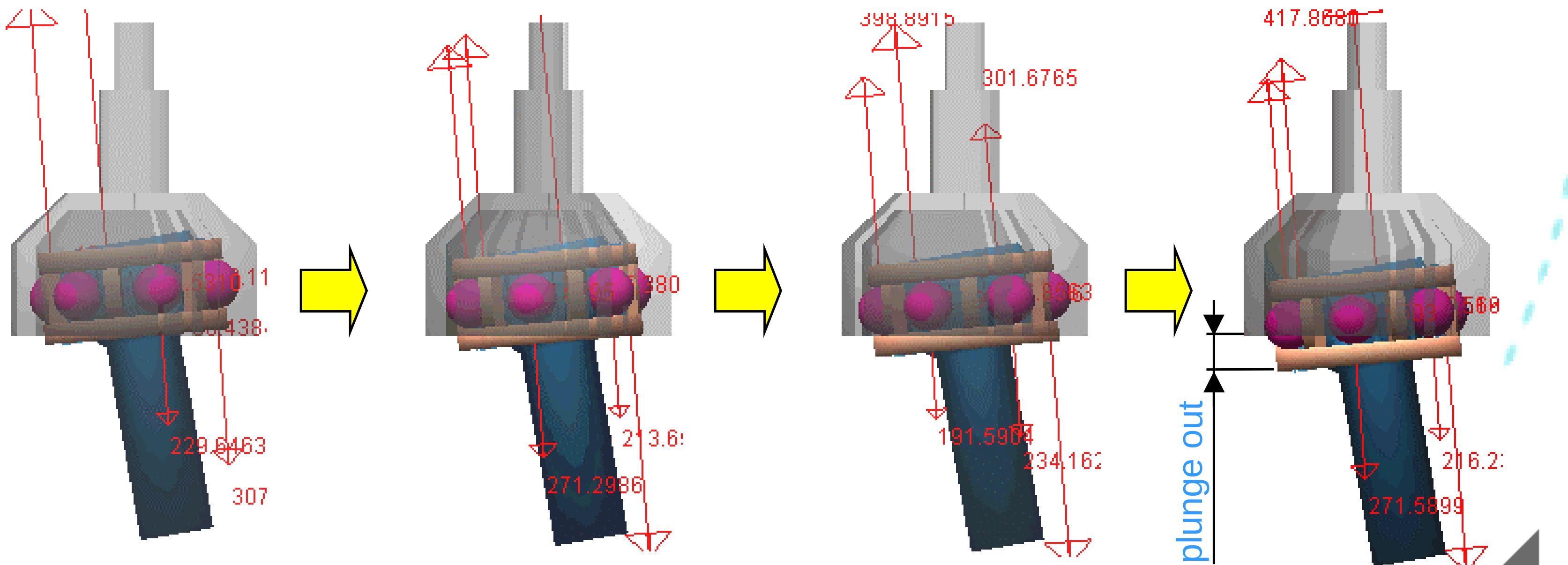
Types of CV Joints?
(based on ball steering mechanism)

- **Rzeppa** CV Joint ➨ axially offset curved OR and IR grooves
- **Undercut Free** Joint ➨ a variant of Rzeppa CV joint of which grooves are partially straight
- **Cross Groove** CV Joint ➨ laterally inclined straight grooves
- **Double Offset** CV Joint ➨ axially offset cage spherical surfaces
- **Weiss** CV Joint ➨ radially inclined straight grooves
- **Tripod** CV Joint ➨ straight OR grooves + spherical IR rollers

Types of CV Joints?

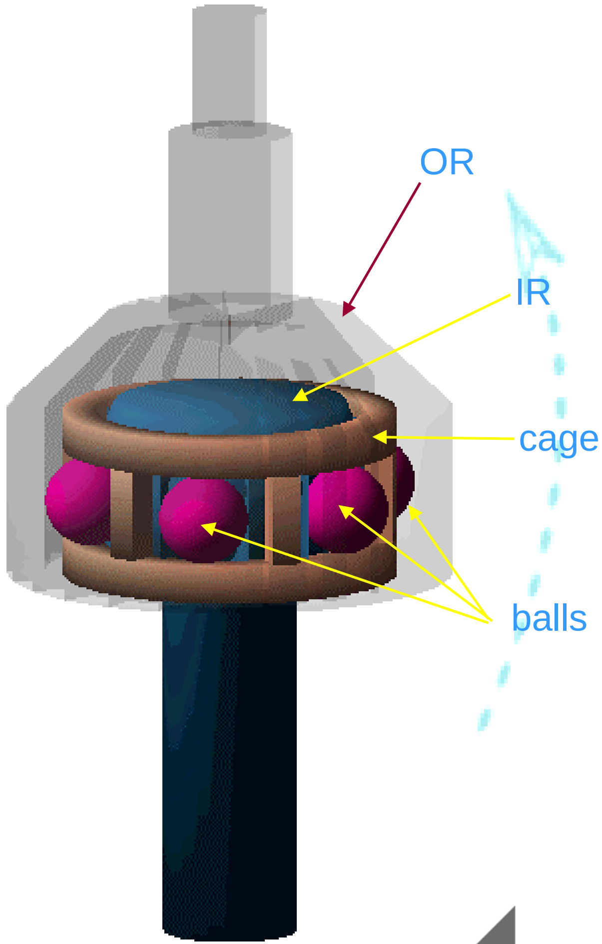
(based on plunge capability)

- **Fixed** Type Joints ➨ Rzeppa, Undercut free, Weiss
- **Free** Type Joints ➨ Cross Groove, Tripod
- **Fixed** or **Free** ➨ Double offset



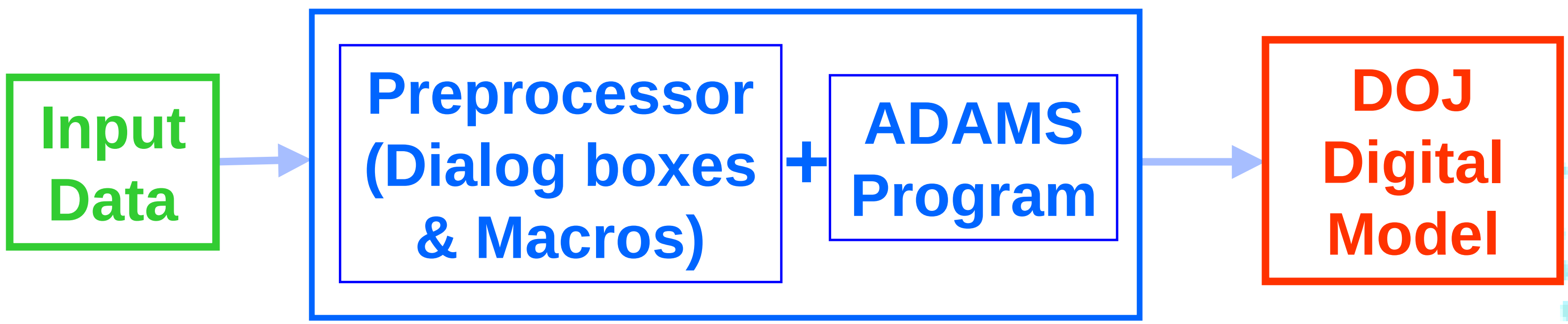
Double Offset CV Joint

- **OR** with axially straight grooves & inner spherical surface
- **IR** with axially straight grooves & outer spherical surface
- **Cage** with axially offset inner & outer spherical surfaces
- **Balls**

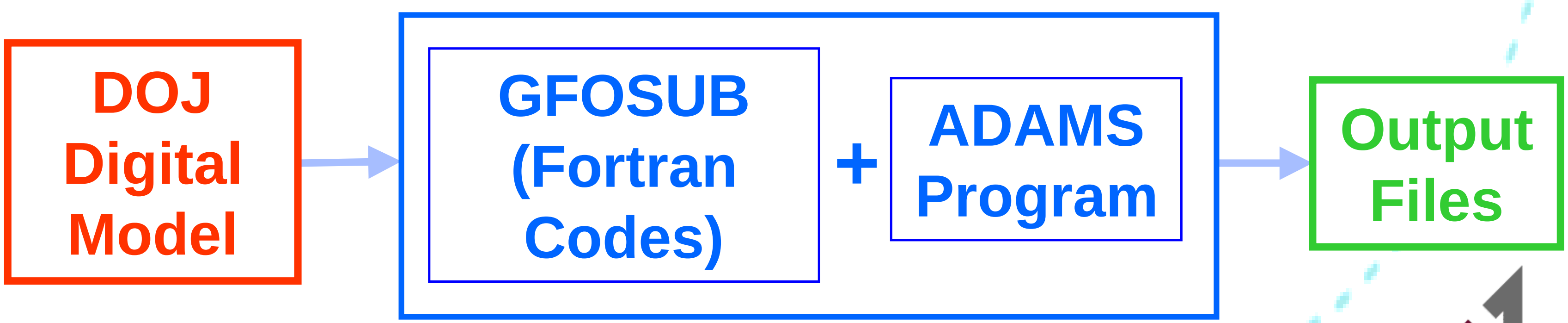


Structure of DOJ Model

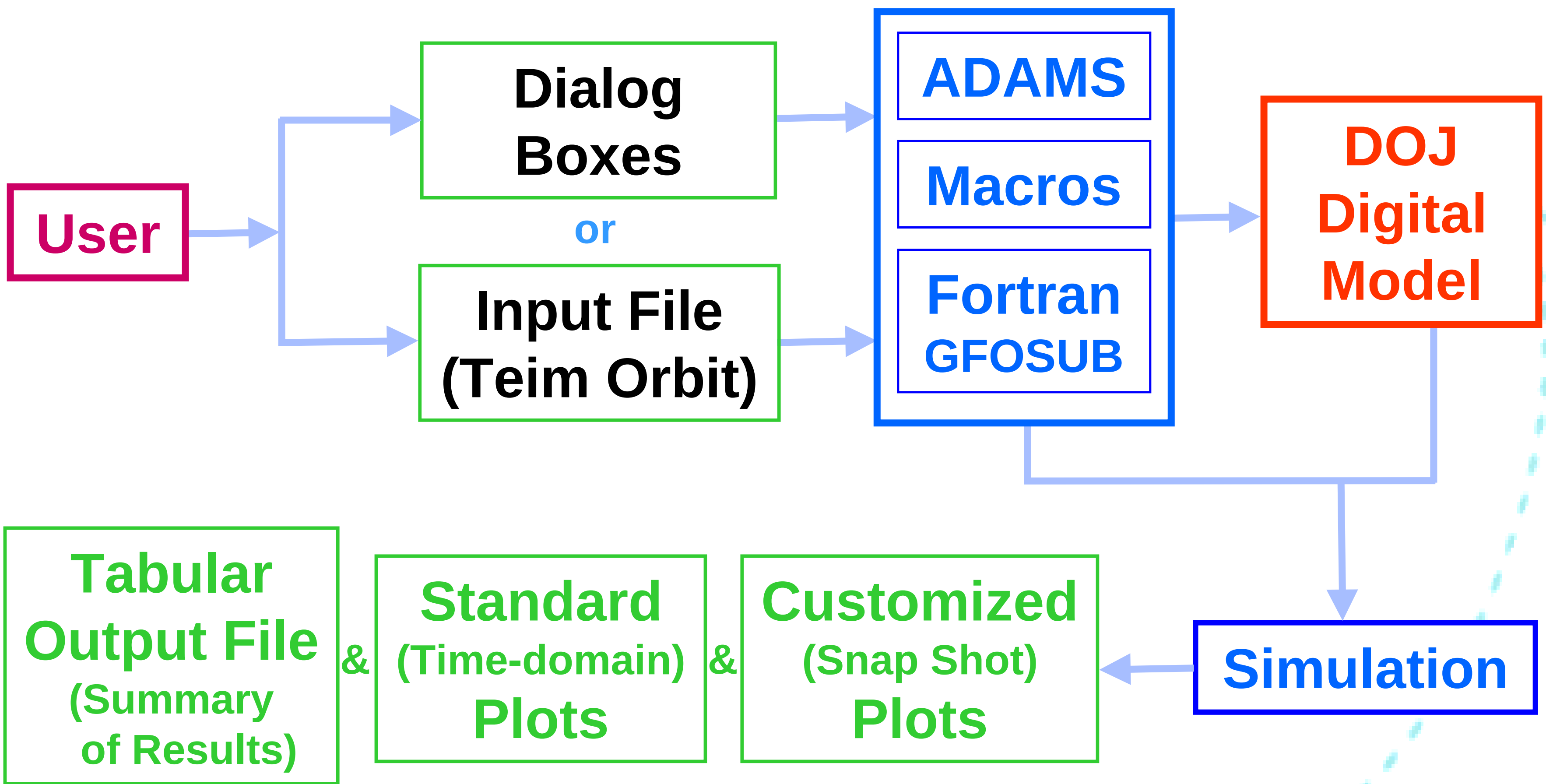
- **Model Generation:**



- **Simulation:**



Model Creation & Simulation Procedure

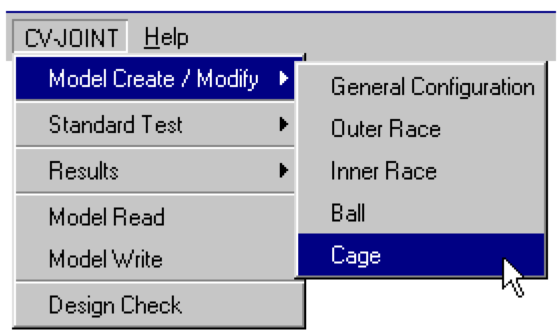


DOJ Model Generation

Step 0 -- Customized Menu

Customized ADAMS/View Ensures That The User Follows The “Correct” Path In Model Creation Via Sequentially Activated Buttons:

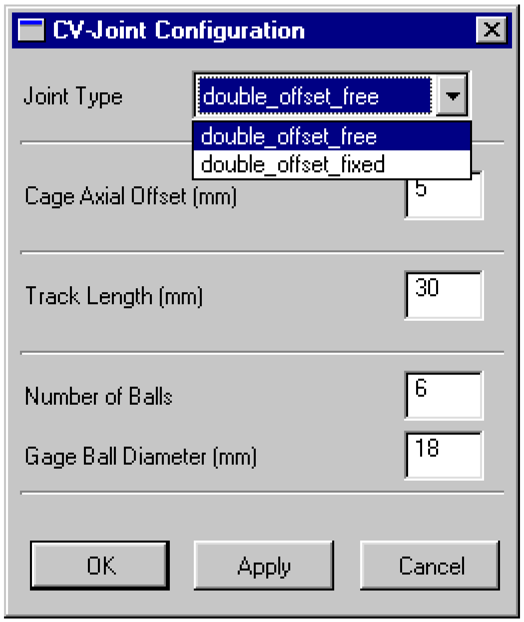
Customized DOJ menu



DOJ Model Generation

Step 1 -- Configuration

Overall configuration dialog box

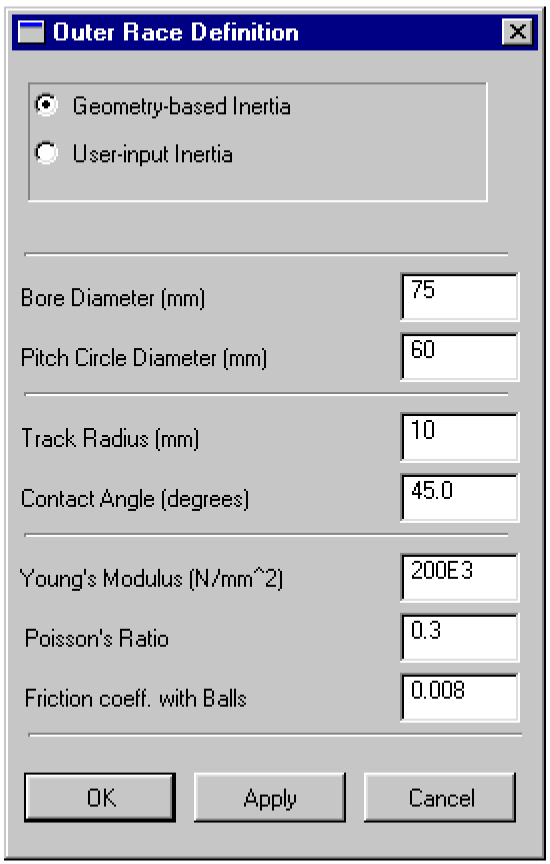


- Joint Type Options:
  - **Fixed Type** ➨ Joint Articulation Only
  - **Free Type** ➨ Allows Axial Plunging
- Any Number of Balls (Grooves)
- Any Amount of Offset

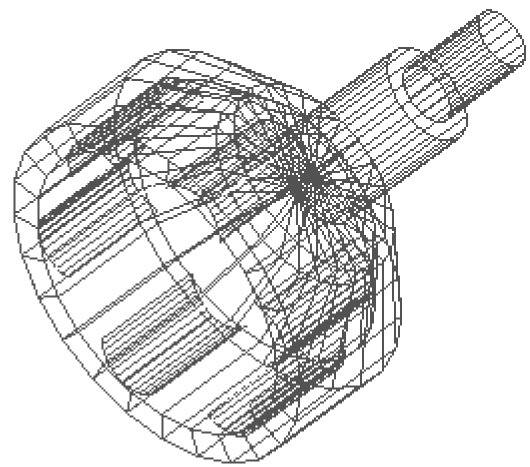
ADAMS

DOJ Model Generation

Step 2 -- Create Outer Race



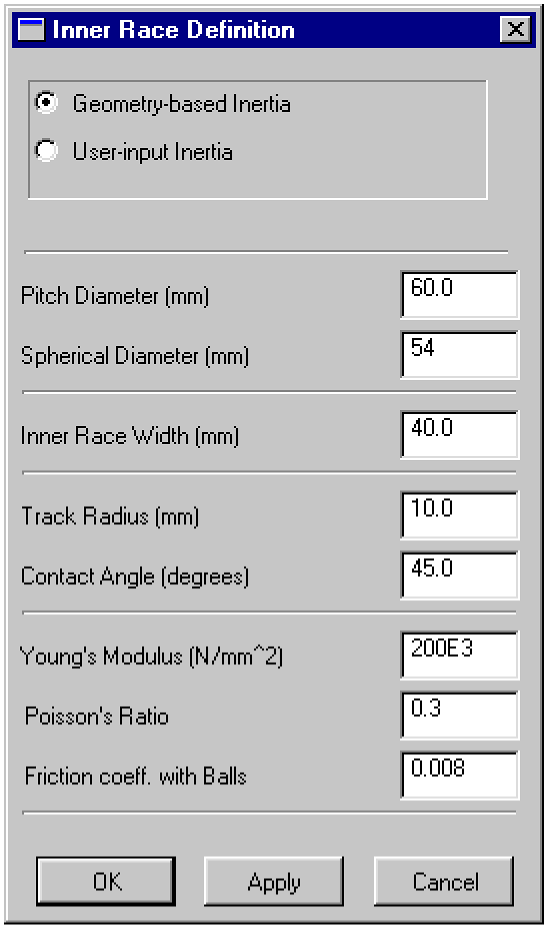
OR dialog box



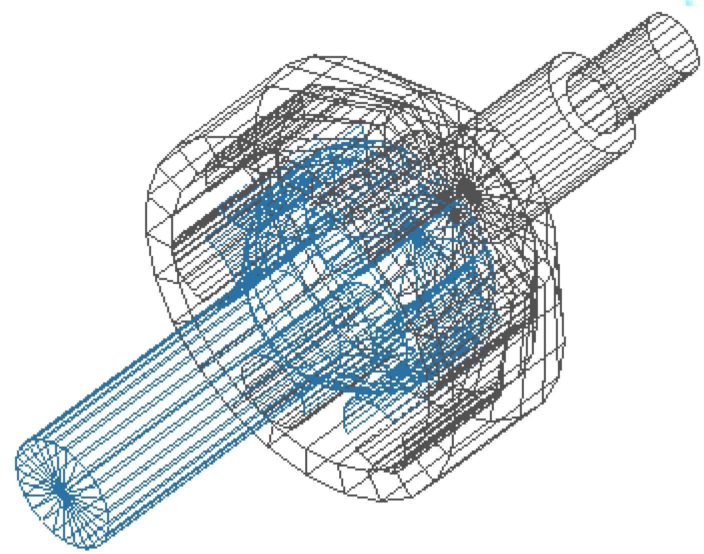
OR

DOJ Model Generation

Step 3 -- Create Inner Race



IR dialog box

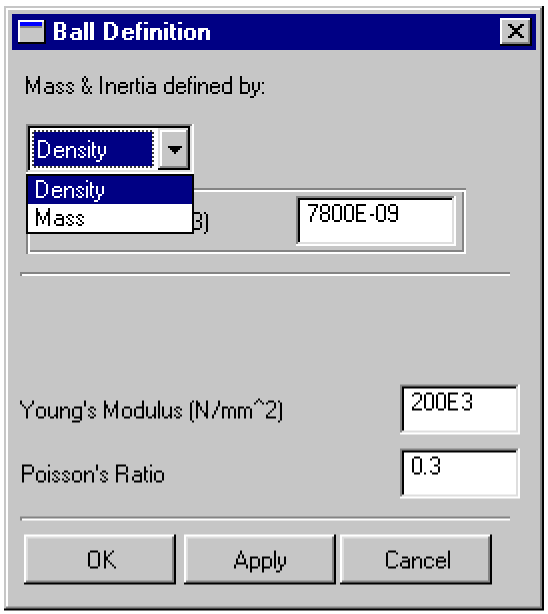


OR + IR

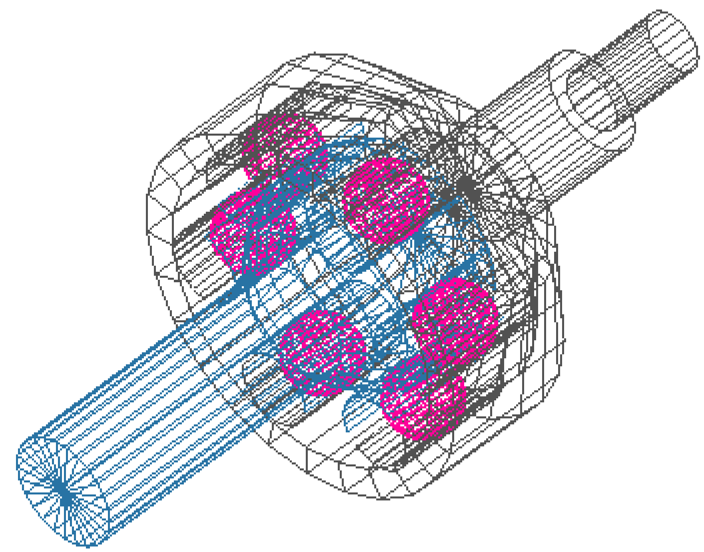
Mechanical Dynamics

DOJ Model Generation

Step 4 -- Create Balls



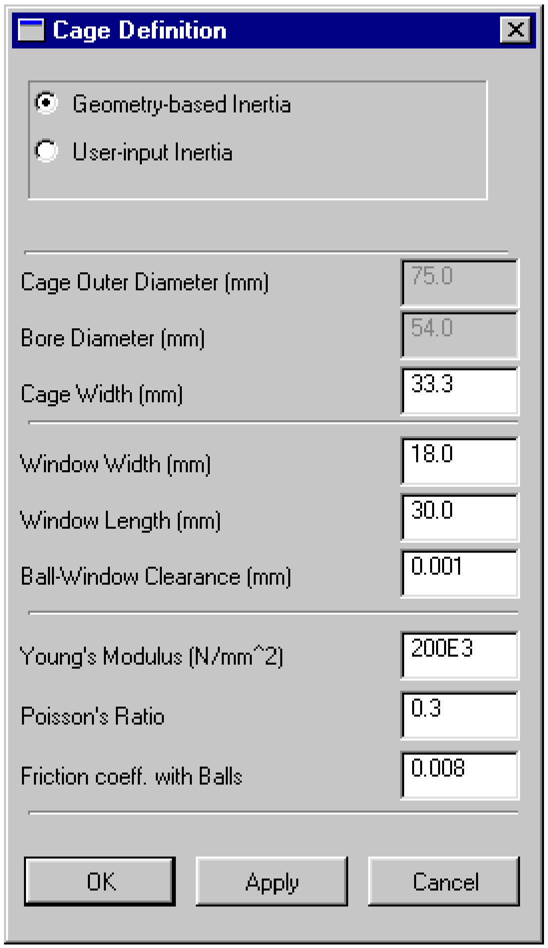
Ball dialog box



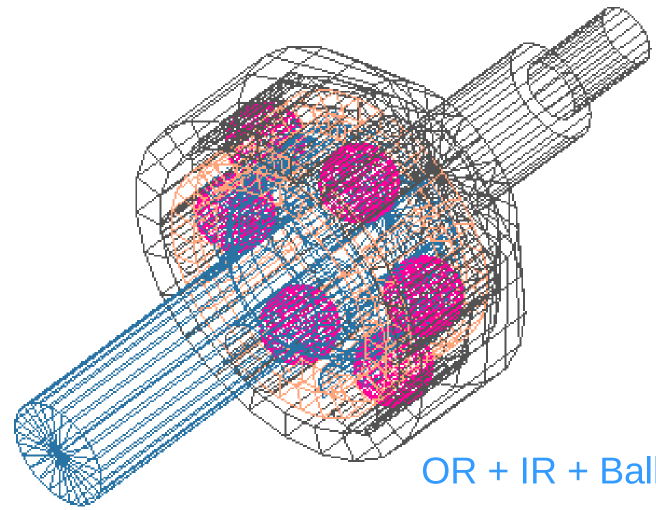
OR + IR + Balls

DOJ Model Generation

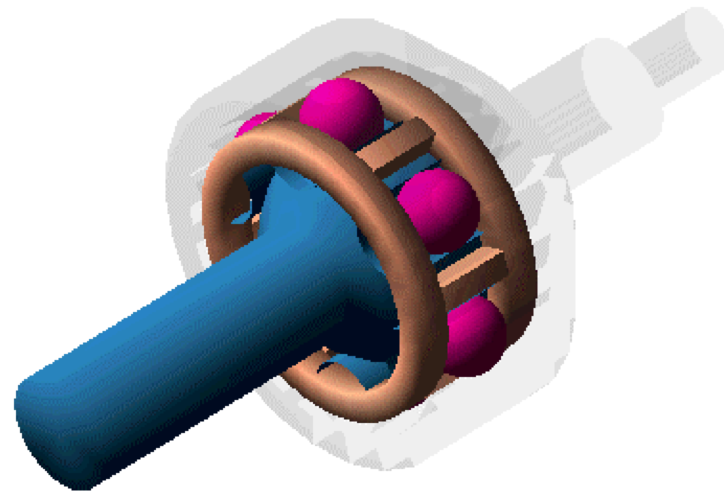
Final Step 5 -- Create Cage (Retainer)



Cage dialog box



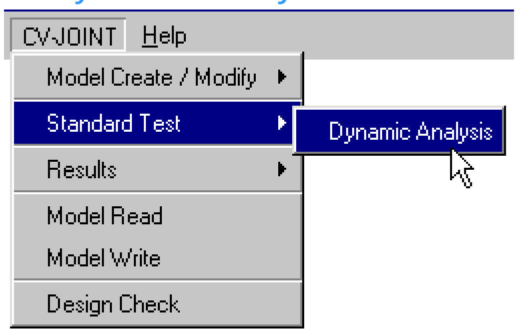
OR + IR + Balls + Cage



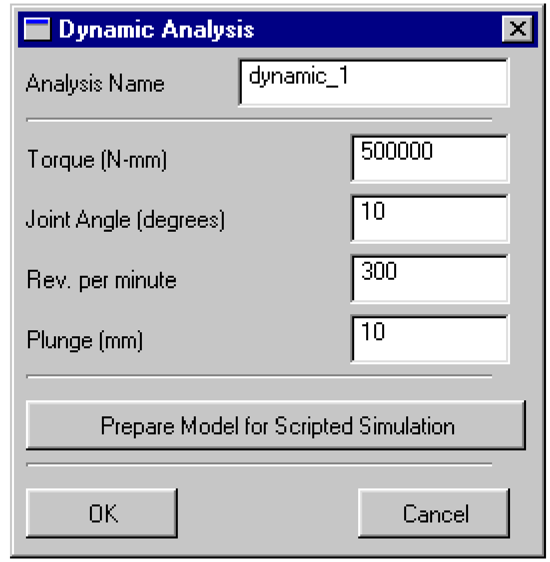
ADAMS

Dynamic Simulation

Dynamic Analysis Menu

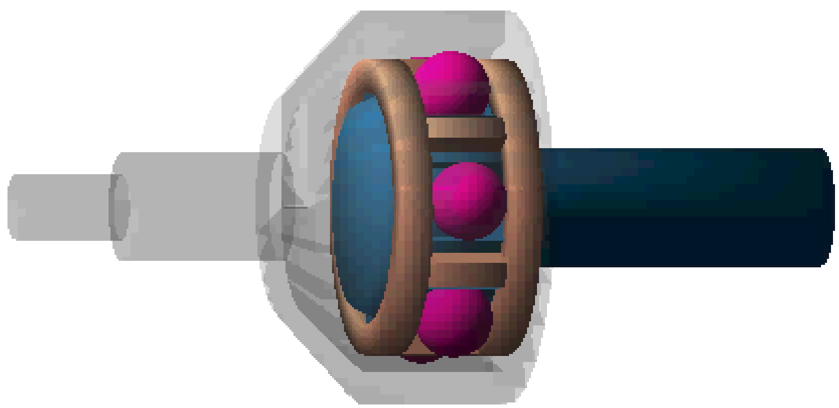


Simulation Dialog Box



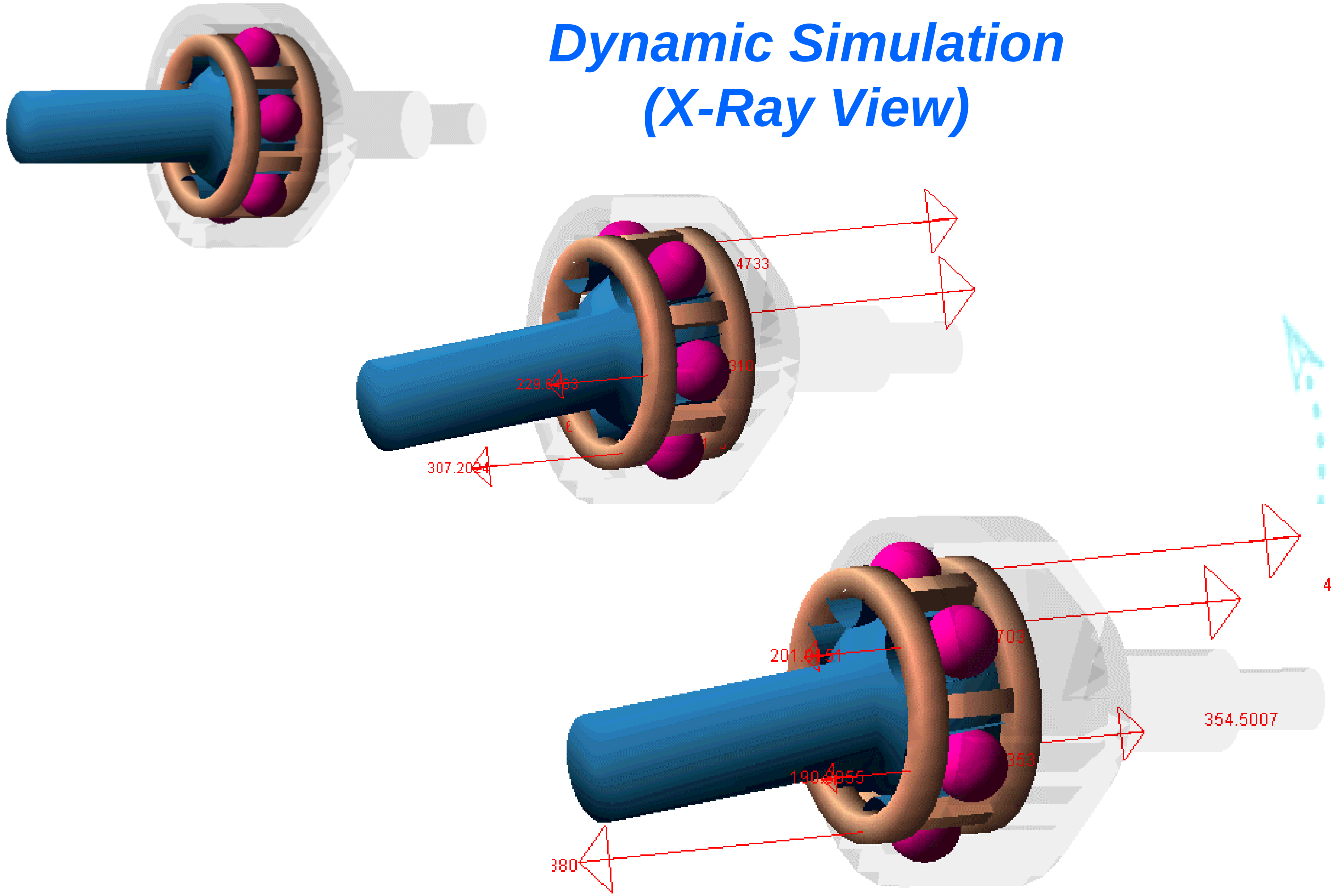
Simulation Results

(Animation Movie)



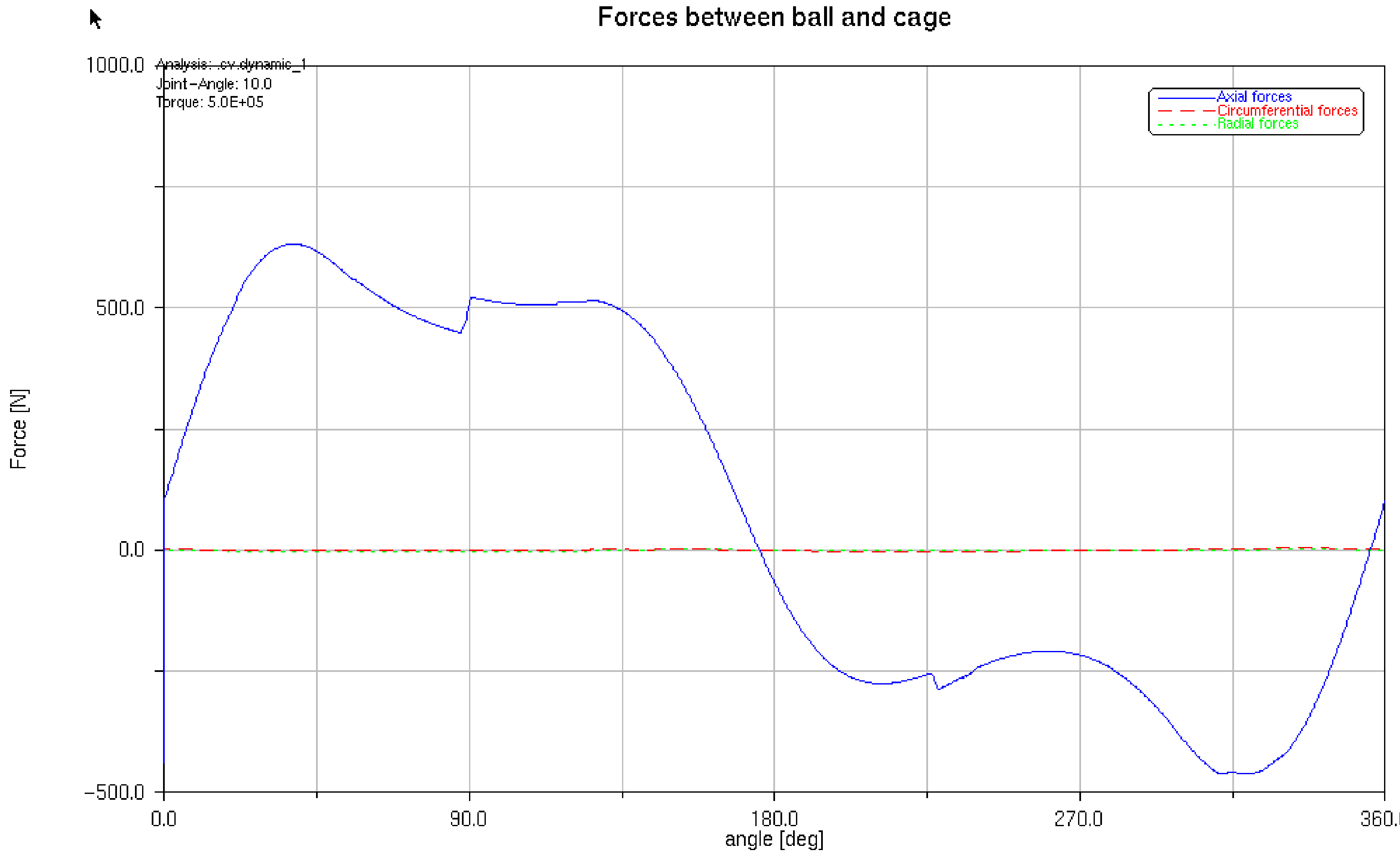
Mechanical Dynamics

Dynamic Simulation
(X-Ray View)



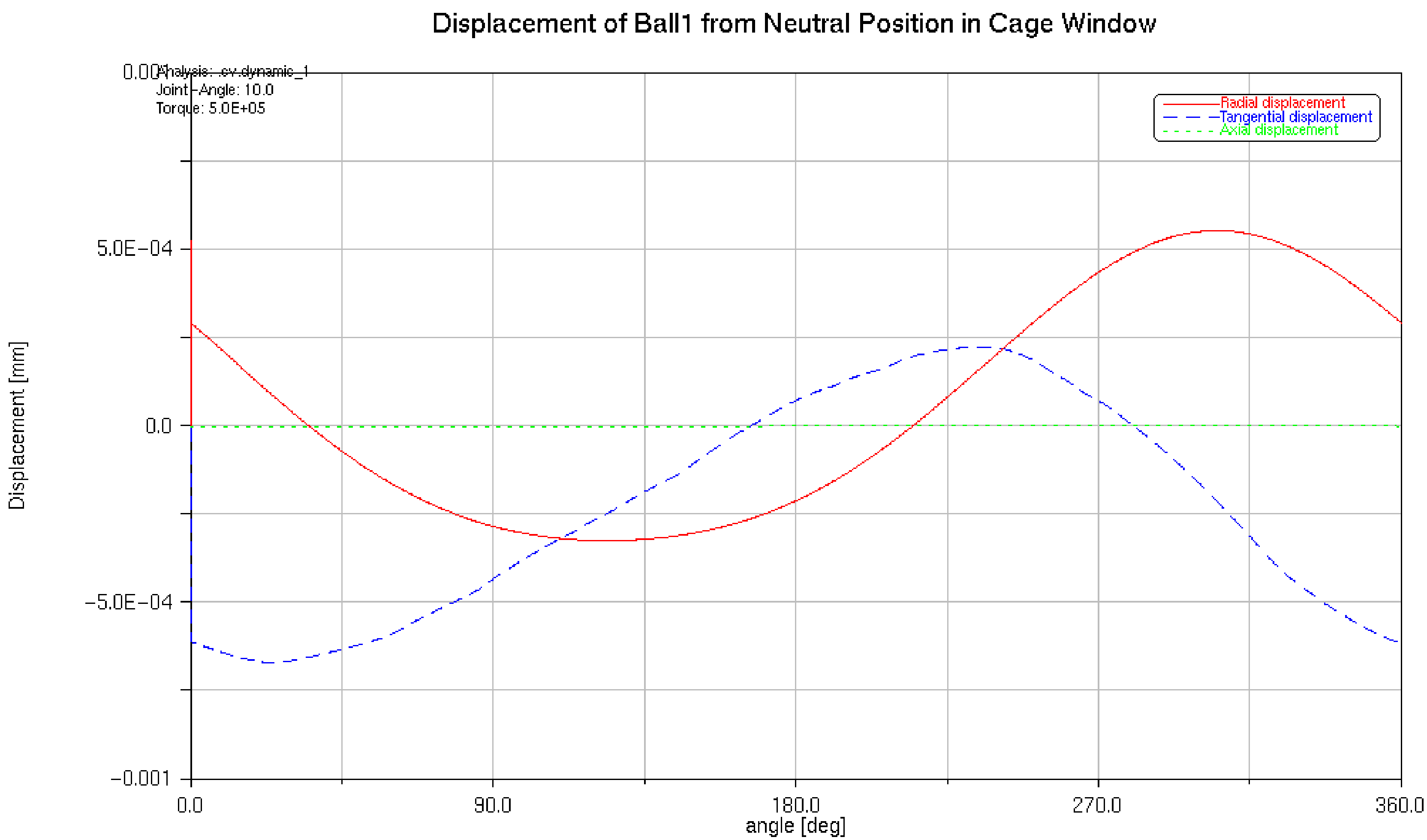
(Sample Result Plots)

double offset joint
ball-cage contact forces



(Sample Result Plots)

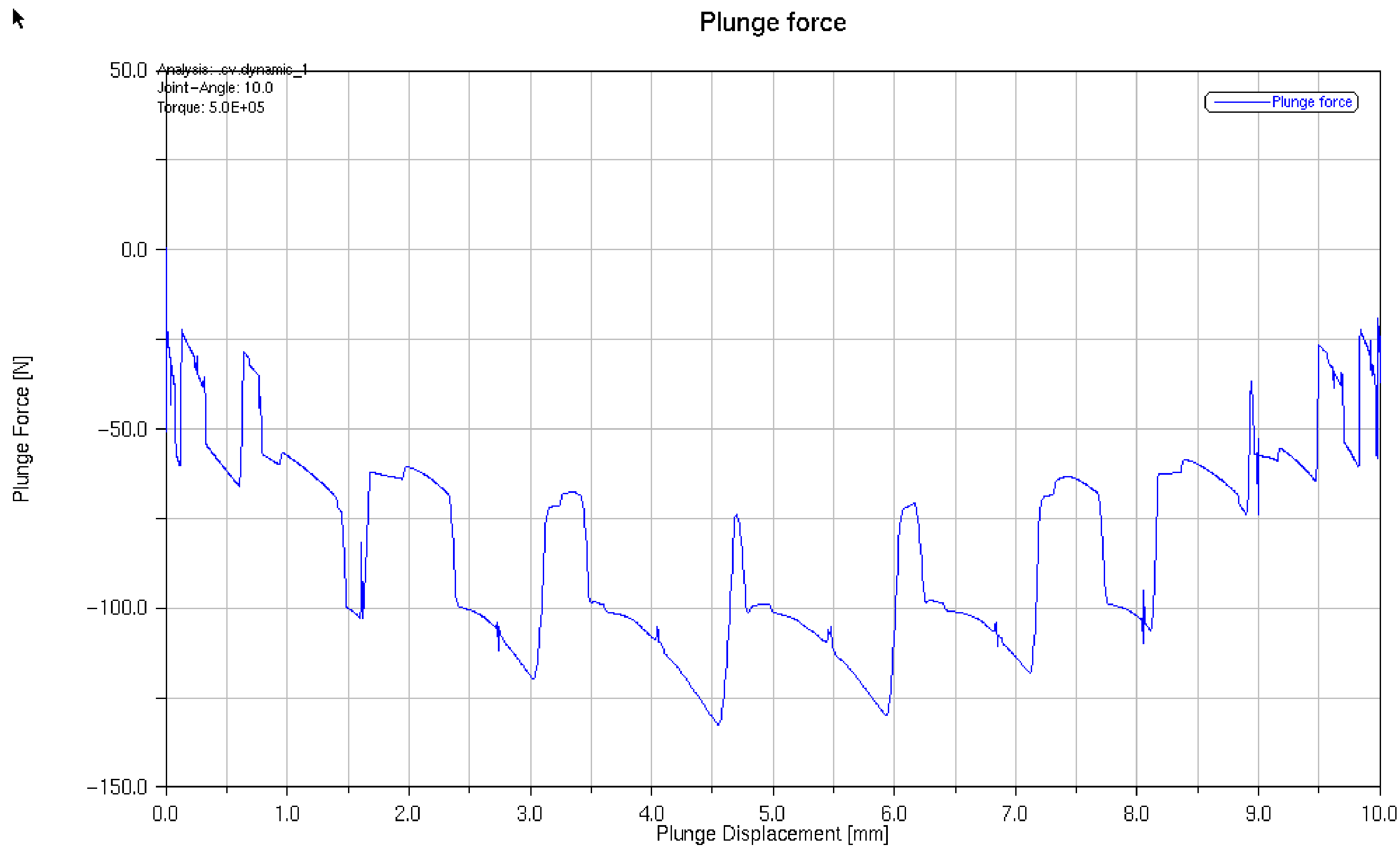
double offset joint
ball-cage relative displacement



Mechanical Dynamics

(Sample Result Plots)

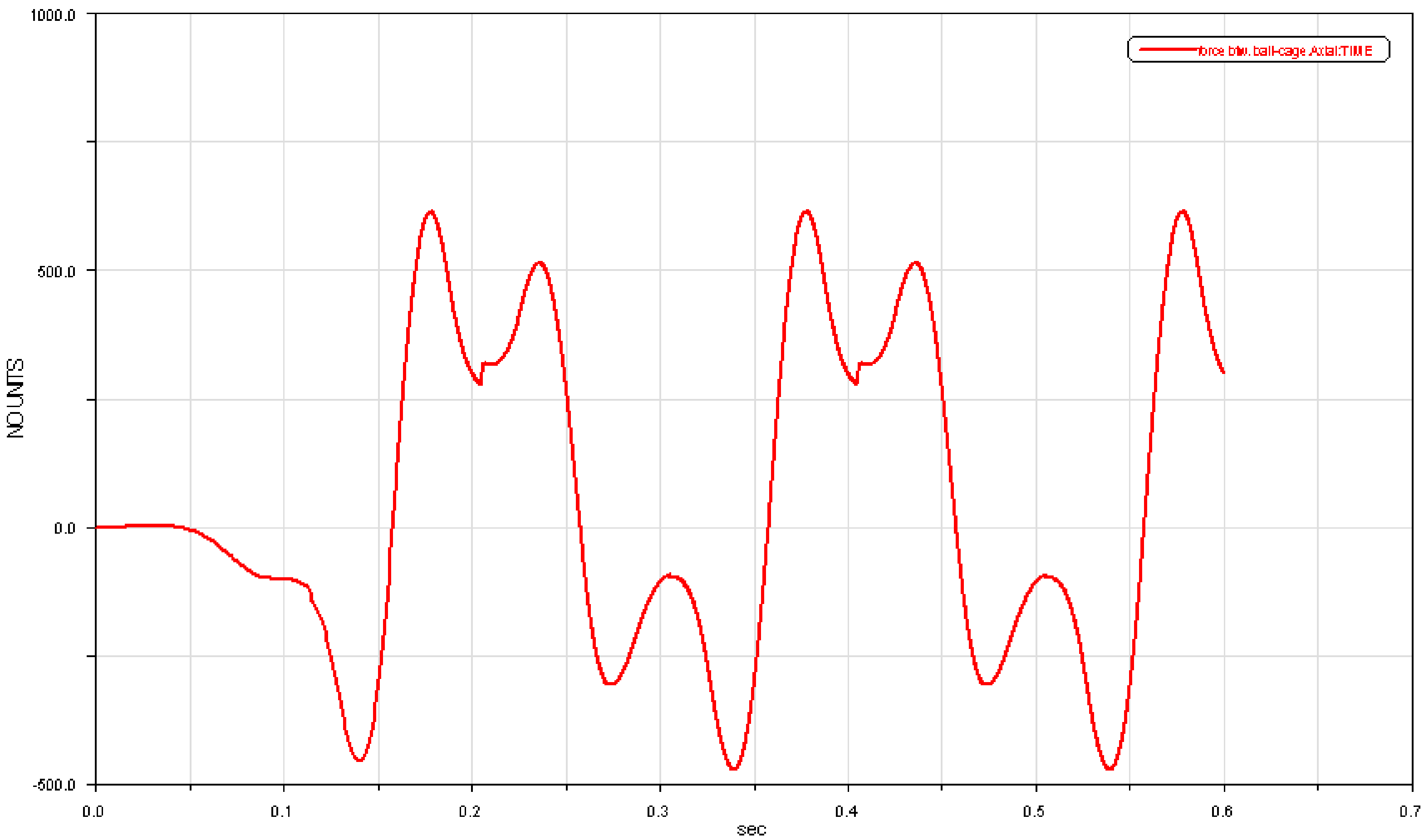
double offset joint
plunge force



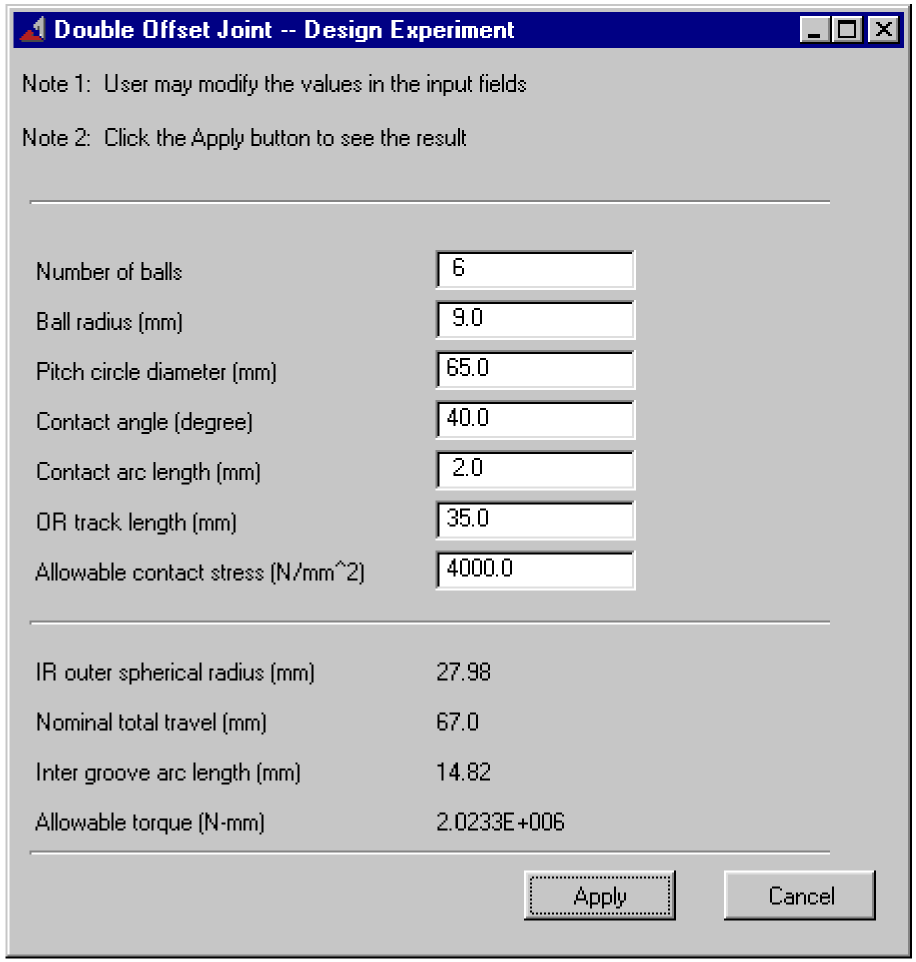
Mechanical Dynamics

(Sample Result Plots)

double offset joint
time-domain plot



Design Experiment Capability



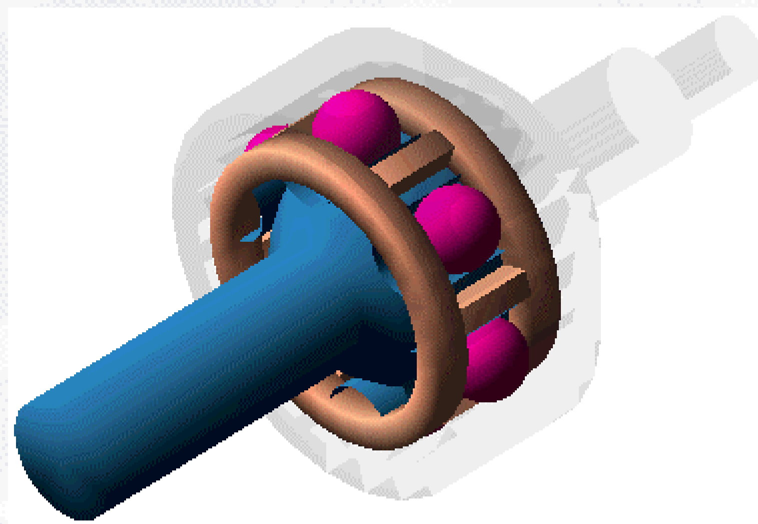
Mechanical Dynamics

Conclusion

- General CV-Joint Modeling
  - **Fully Parametric** Model Facilitates DOE & Optimization Studies
  - Executes Predefined **Dynamic Simulation** (Joint Angle & Plunge)
  - Provides **Design Experiment** Capability
- Convenient Pre- & Post-Processing
  - **Teim-Orbit Neutral File** For Model Storage, Alteration & Retrieval
  - **Results Manager** Enables Output File Storage & Removal
  - **Tabular Output File** for Summary of Model Spec & Results
  - **Plots** Are Available in Both Time-domain & Joint-Angle-Domain
  - Provide **Detailed Information** Including **Hertzian Contact Stress**

ADAMS

Appendix A:
Example Files for DOJ Model



double offset joint

Mechanical Dynamics

(Example of Macro Code)

```
!***********************************************************
! Description: This macro creates and defines the inner race
!              of double offset joint
! Modified: John Park (8/10/99)
!***********************************************************
!END_OF_PARAMETERS
!
!---------------------IR geometry array------------------------------!
!
var set var=.cv.pcri  real=(.cv.cv_pcdi/2)
var set var=.cv.cv_rxi real=( (.cv.cv_rti-.cv.cv_grb)*cos(.cv.cv_gammai) )
var set var=.cv.cv_rzi real=( (.cv.cv_rti-.cv.cv_grb)*sin(.cv.cv_gammai) )

var set var=.cv.track_max_angle_ir  real=75
var set var=.cv.track_zero_angle_ir real=(1+asin(.cv.cv_rzi/.cv.cv_rti))
var set var=.cv.d_ang_ir real=((.cv.track_max_angle_ir-.cv.track_zero_angle_ir)/7.0)

data_element create matrix full matrix_name= .cv.groov_ir row=16 col=3 input_order = by_row units=length &
values=(.cv.cv_rxi+.cv.cv_rti*cos(180-.cv.track_max_angle_ir)),(-cv_rzi+.cv.cv_rti*sin(180-.cv.track_max_angle_ir)),0, &
    (.cv.cv_rxi+.cv.cv_rti*cos(180-(75-1*.cv.d_ang_ir))),(-cv_rzi+.cv.cv_rti*sin(180-(75-1*.cv.d_ang_ir))),0, &
    (.cv.cv_rxi+.cv.cv_rti*cos(180-(75-2*.cv.d_ang_ir))),(-cv_rzi+.cv.cv_rti*sin(180-(75-2*.cv.d_ang_ir))),0, &
    (.cv.cv_rxi+.cv.cv_rti*cos(180-(75-3*.cv.d_ang_ir))),(-cv_rzi+.cv.cv_rti*sin(180-(75-3*.cv.d_ang_ir))),0, &
    (.cv.cv_rxi+.cv.cv_rti*cos(180-(75-4*.cv.d_ang_ir))),(-cv_rzi+.cv.cv_rti*sin(180-(75-4*.cv.d_ang_ir))),0, &
    (.cv.cv_rxi+.cv.cv_rti*cos(180-(75-5*.cv.d_ang_ir))),(-cv_rzi+.cv.cv_rti*sin(180-(75-5*.cv.d_ang_ir))),0, &
    (.cv.cv_rxi+.cv.cv_rti*cos(180-(75-6*.cv.d_ang_ir))),(-cv_rzi+.cv.cv_rti*sin(180-(75-6*.cv.d_ang_ir))),0, &
    (.cv.cv_rxi+.cv.cv_rti*cos(180-.cv.track_zero_angle_ir)),(-cv_rzi+.cv.cv_rti*sin(180-.cv.track_zero_angle_ir)),0, &
    (.cv.cv_rxi+.cv.cv_rti*cos(180+.cv.track_zero_angle_ir)),(cv_rzi+.cv.cv_rti*sin(180+.cv.track_zero_angle_ir)),0, &
    (.cv.cv_rxi+.cv.cv_rti*cos(180+(75-6*.cv.d_ang_ir))),(cv_rzi+.cv.cv_rti*sin(180+(75-6*.cv.d_ang_ir))),0, &
    (.cv.cv_rxi+.cv.cv_rti*cos(180+(75-5*.cv.d_ang_ir))),(cv_rzi+.cv.cv_rti*sin(180+(75-5*.cv.d_ang_ir))),0, &
    (.cv.cv_rxi+.cv.cv_rti*cos(180+(75-4*.cv.d_ang_ir))),(cv_rzi+.cv.cv_rti*sin(180+(75-4*.cv.d_ang_ir))),0, &
    (.cv.cv_rxi+.cv.cv_rti*cos(180+(75-3*.cv.d_ang_ir))),(cv_rzi+.cv.cv_rti*sin(180+(75-3*.cv.d_ang_ir))),0, &
    (.cv.cv_rxi+.cv.cv_rti*cos(180+(75-2*.cv.d_ang_ir))),(cv_rzi+.cv.cv_rti*sin(180+(75-2*.cv.d_ang_ir))),0, &
    (.cv.cv_rxi+.cv.cv_rti*cos(180+(75-1*.cv.d_ang_ir))),(cv_rzi+.cv.cv_rti*sin(180+(75-1*.cv.d_ang_ir))),0, &
    (.cv.cv_rxi+.cv.cv_rti*cos(180+.cv.track_max_angle_ir)),(cv_rzi+.cv.cv_rti*sin(180+.cv.track_max_angle_ir)),0
```


Mechanical Dynamics

(Example of Fortran Code)

```
c-----------------------------------------------------------------------
      SUBROUTINE GFOSUB (ID,TIME,PAR,NPAR,DFLAG,IFLAG,RESULT)

      IMPLICIT           NONE
      INTEGER            GFONUM, ISTAT
      INTEGER            ID, NPAR
      LOGICAL            DFLAG, IFLAG, ERRFLG
      DOUBLE PRECISION   RESULT(6), TIME, PAR(*),X
      INTEGER            NNN,I
c-----------------------------------------------------------------------
c  Author:       John Park  7/29/99
c  Description:  This subroutine calculates the contact force
c                between a ball and the cage in the double offset
c                joint.
c-----------------------------------------------------------------------
c     Define local variables:

      NNN=NINT(PAR(1))

c-----------------------------------------------------------------------
c     Branching out:

c-----Outer race force-----

      IF(NNN.EQ.1000)then
          CALL ORACE (ID,TIME,PAR,NPAR,DFLAG,IFLAG,RESULT)

c-----Inner race force-----

      ELSEIF(NNN.EQ.2000) then
          CALL IRACE (ID,TIME,PAR,NPAR,DFLAG,IFLAG,RESULT)

c-----Cage force-----

      ELSEIF(NNN.EQ.3000) then
          CALL CAGE (ID,TIME,PAR,NPAR,DFLAG,IFLAG,RESULT)
```

ADAMS

(Example of Teim Orbit Input File)

```
$------------------------------------------------------------------CONFIGURATION
[CONFIGURATION]
 JOINT_TYPE  =  'double_offset_free'
 CAGE_AXIAL_OFFSET  =  '5.0'
 OUTER_RACE_TRACK  =  'gothic_arch'
 INNER_RACE_TRACK  =  'gothic_arch'
 BALL_NUMBER  =  6
 GAGE_BALL_DIAMETER  =  18.0
 TRACK_LENGTH  =  30.0
$---------------------------------------------------------------------OUTER_RACE
[OUTER_RACE]
 MASS_AND_INERTIA_DEFINED_BY  =  'Geometry'
 YOUNGS_MODULUS  =  200000.0
 POISSONS_RATIO  =  0.3
 PITCH_DIAMETER  =  60.0
 BORE_DIAMETER  =  75.0
 TRACK_RADIUS  =  10.0
 CONTACT_ANGLE  =  45.0
 COEFFICIENT_OF_FRICTION  =  0.008
$---------------------------------------------------------------------INNER_RACE
[INNER_RACE]
 MASS_AND_INERTIA_DEFINED_BY  =  'Geometry'
 YOUNGS_MODULUS  =  200000.0
 POISSONS_RATIO  =  0.3
 PITCH_DIAMETER  =  60.0
 SPHERICAL_DIAMETER  =  54.0
 WIDTH  =  40.0
 TRACK_RADIUS  =  10.0
 CONTACT_ANGLE  =  45.0
 COEFFICIENT_OF_FRICTION  =  0.008
$---------------------------------------------------------------------------BALL
[BALL]
```

(Example of Tabular Output File)

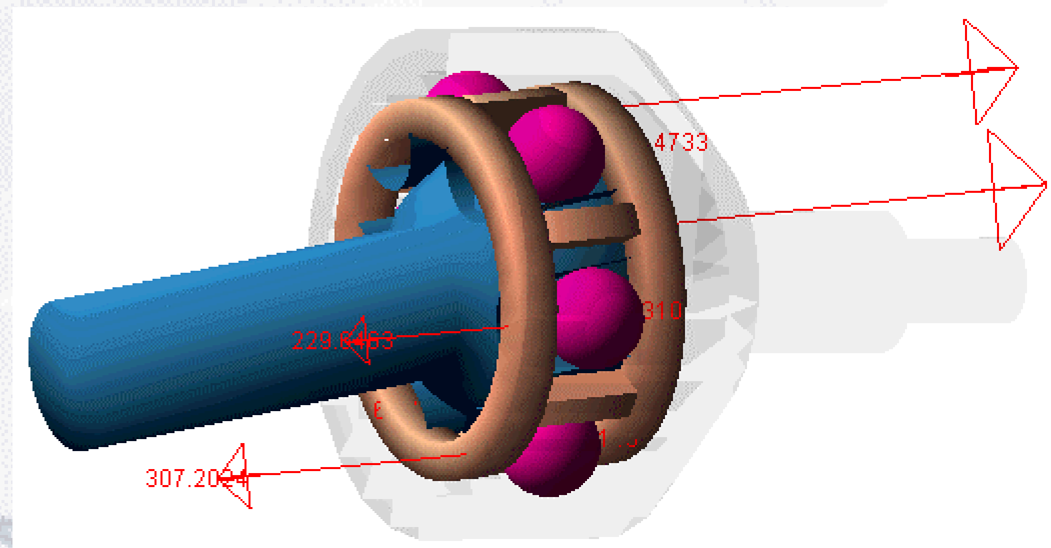
```
    CV-Joint virtual Prototype
-------------------------------
[Test_Setup]
Analysis =                    'dynamic_1'
Analysis_Type =               'dynamic'
Applied_Torque =              5.0E+05
Joint_Angle =                 10.0
Rpm =                                        300.0
---------------------------------------

Configuration
---------------------------------------
Joint-Type:                   double_offset_free
Cage Axial Offset:            5.0
OR Tracktype:                 gothic_arch
IR Tracktype:                 gothic_arch
Number of Balls:                             6
Ball Diameter:                18.0
Track Length:                 30.0

                   omitted

Static Force
---------------------------------------
Max force of ball on outer track:      6069.
Max force of ball on inner track:      6477.
Max force of ball on cage window:         0.

Stresses
---------------------------------------
Max cage web tensile stress [MPa]:       4.26
Hertzian compressive stress (IR) [MPa]  :    7869.68
Deformation of Hertzian stress (IR) [mm]:    0.08042995
Max Subsurface Shear Stress (IR) [MPa]  :    2526.50
Depth of max subsurface stress (IR) [mm]:    0.34000100
Distance of contact point to IR track edge:  0.00
Ellipse semiaxis:   0.8955
Ellipse semiaxis:   0.5834
Effective contact ellipse area IR:   1.6412
% Contact ellipse area lost:         0.0
Radius in contact-crossection of IR-track R2P:    10.0000
KD in Hertz theory:     4.6552
COS(tau) in Hertz theory:     0.3103
```

Appendix B:
Guidelines for Any CVJ Consulting Projects

For inquiry to any CVJ-related consulting projects, contact:

John Janevic, Manager, Central Region, MDI

e-mail: jjane@adams.com
phone: (734) 913-2513



Mechanical Dynamics

Other Available ADAMS CV Joint Models

- Available Baseline Models are :
  - **Rzeppa CV Joint Model**
  - **Undercut Free CV Joint Model**
  - **Cross Groove CV Joint Model**
  - **Double Offset CV Joint Model**
  - **Tripod CV Joint Model**
- Possible Future Additions are:
  - **Universal Joint (Cardan Joint) Model**
  - **Double Cardan Joint Model**
  - **Sliding Spline Joint Model**
  - **Ball Spline Joint Model**

Future Enhancements to ADAMS CV- & U-Joint Models

- MDI Can Add Any Joint Types:
  - Univeral Joint (Cardan Joint)
  - Double Cardan Joint
  - Others (per User specification)
- Integrated CV-Joint With Full-Vehicle ADAMS Models
  - Inclusion with High-Fidelity Driveline Models To Predict CV-Joint Component Behavior Under Conditions Such As (Typical Ride, Handling, Durability Events)
  - ADAMS, ADAMS/Car, ADAMS/Pre, or ADAMS/Driveline

System Requirements for ADAMS CV Joint Models

- **Software**
  - Bare Minimum: ADAMS/View + ADAMS/Solver
  - Other Helpful Modules: ADAMS/Exchange, ADAMS/Linear
  - Possible Future Needs: ADAMS/Car, ADAMS/Driveline
- **Hardware**
  - UNIX or Windows NT workstation with recent OS
  - Typical RAM and disk drive capacity required less than that of FEA workstations
  - FORTRAN compiler (optional)

Typical Implementation Plan

- Purpose
  - Fully Implement **Virtual Prototyping** In CV-Joint Engineering Process
- Key Components
  - **Install** CV-Joint Models At User sites
  - Provide Customized **Training** At User Facilities to develop both standard users and expert users
  - **Maintenance** & Regular **Enhancement** Program

Typical Training Plan

- Goal:
  - Promote Effective Use Of The CV-Joint Package on Both User & Expert Level.
  - The **User** will be able to run the models.
  - The **Expert** will be able to effectively modify any or all of the CV-Joint model and associated code.
- Key Components:
  - **Basic ADAMS Training** Course at User Facilities or MDI facilities in Michigan
  - **CV-Joint Specific Training** Course at User Facilities
  - **Advanced ADAMS Training** Courses at User Facilities or MDI facilities in Michigan

Typical Maintenance & Regular Enhancement Program

- A one year program to support and enhance the CV-Joint models as implemented at User Site.
- Required to accompany Mechanical Dynamics non-commercial software
- Assures User:
  - Upgrades For Compatibility With Future Versions of ADAMS
  - Available consultants at Mechanical Dynamics in Michigan to provide rapid response to questions and issues.