U.S. Department
of Transportation

**National Highway
Traffic Safety
Administration**

DOT HS
Draft Report

February 2004

The SISAME-3D Program: Structural Crash Model Extraction and Simulation

Stuart G. Mentzer
Information Systems and Services, Inc.
8601 Georgia Ave., Suite 708
Silver Spring, MD 20910

Prepared for:

National Highway Traffic Safety Administration
400 7th Street, S.W.
Washington, DC 20590

Under Contract No. DTNH22–01–C–07317

This document is available to the public from the National Technical Information Service, Springfield, Virginia 22161.

NOTICE

This document is disseminated under the sponsorship of the Department of Transportation, National Highway Traffic Safety Administration, in the interest of information exchange. The United States Government assumes no liability for the contents or use thereof.

NOTICE

The United States Government does not endorse products or manufacturers. Trade or manufacturers' names are used only because they are considered essential to the objective of this report.

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
DOT HS		

4. Title and Subtitle	5. Report Date
THE SISAME-3D PROGRAM: Structural Crash Model Extraction and Simulation	February 2004
	6. Performing Organization Code
7. Author(s) Stuart G. Mentzer	8. Performing Organization Report No.
9. Performing Organization Name and Address Information Systems and Services, Inc. 8601 Georgia Ave., Suite 708 Silver Spring, MD 20910	10. Work Unit No. (TRAIS)
	11. Contract or Grant No. DTNH22–01–C–07317
12. Sponsoring Agency Name and Address National Highway Traffic Safety Administration 400 7th Street, S.W. Washington, DC 20590	13. Type of Report and Period Covered Draft Report October 2001 – February 2004
	14. Sponsoring Agency Code

15. Supplementary Notes

Randa Radwan Samaha – Technical Representative, Office of Crashworthiness Research/NHTSA, NRD-11

16. Abstract

This report contains the users guide and formulation for the SISAME-3D v.1.1 structural impact modeling program. SISAME-3D extracts lumped-parameter structural models from crash event instrumentation data, and can be used to simulate such models.

SISAME-3D uses a combination of user-specified model configurations and crash event motion and, optionally, load data to find optimal models. The robust global solution is based on the complete set of mass equations of motion and load history targets for the full time span of one or more crash events. Any subset of the load-paths and mass weights can be extracted from sufficiently complete and accurate instrumentation sets. The load-paths are parameterized to obtain a tractable optimization problem. Constraints are used to assure physically meaningful load-paths. Parameter estimates and bounds can be specified to control the solution. A range of static and dynamic load-path types and a range of outputs are available.

The SISAME-3D source code is available from the National Highway Traffic Safety Administration, Office of Crashworthiness Research (NRD-11).

17. Key Words	18. Distribution Statement
Lumped-parameter model, computer simulation, least squares, constrained optimization.	This document is available to the public from the National Technical Information Service, Springfield, Virginia 22161.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	151	

Form DOT F 1700.7 (8-72)

Reproduction of completed page authorized

Contents

1 Introduction **1**

1.1 Overview . . . 1

1.2 History . . . 3

1.3 Capabilities . . . 4

1.4 Future . . . 5

1.5 General Information . . . 5

1.6 Documentation . . . 5

2 SISAME-3D Modeling Concepts **6**

2.1 SISAME-3D Models . . . 6

2.1.1 Structures . . . 6

2.1.2 Elements . . . 7

2.1.3 Model Configurations . . . 11

2.1.4 Degrees of Freedom . . . 12

2.1.5 Coordinate Frames . . . 12

2.2 SISAME-3D Outputs . . . 13

2.3 The Extraction Process . . . 14

3 SISAME-3D Users Guide **15**

3.1 Running SISAME-3D . . . 16

3.2 Model Construction . . . 16

3.2.1 Model Configurations . . . 17

3.2.2 Degrees of Freedom (DOF) . . . 18

3.2.3 Coordinate Frames . . . 18

3.2.4 Generalized Spatial Parameter Tuples . . . 18

3.2.5 Resting Position 19
3.2.6 Initial Position 20
3.2.7 Rotational Position 20
3.2.8 Point and Shape Position 21
3.2.9 Initial Velocity 22
3.2.10 Dimensional Systems 23
3.3 IDs, Names, and References 24
3.3.1 SISAME-3D IDs 24
3.3.2 Component Names and References 25
3.3.3 Element Set References 26
3.3.4 Aspect Names and References 27
3.3.5 Aspect Set References 29
3.3.6 Parameter Names and References 29
3.4 Specifying Parameters 31
3.4.1 Parameter Specifiers 31
3.4.2 Parameter Block Specifiers 32
3.4.3 Parameter Tuple Specifiers 32
3.4.4 Value Specifiers 32
3.4.5 Reference Specifiers 33
3.4.6 Default Reference Specifiers 33
3.4.7 Transform Specifiers 34
3.4.8 Extraction Specifiers 34
3.5 Signal Files 35
3.5.1 Input Signal Files 35
3.5.2 Output Signal Files 36
3.5.3 UDS Signal Files 36
3.6 Event File Overview 37
3.6.1 Event File Template 37

3.6.2 Link Parameter Templates 40
3.6.3 Event File Organization 41
3.6.4 Event File Fields 41
3.6.5 Input Signal File Names 42
3.6.6 Specification Tables 42
3.7 Run Information 44
3.8 Model Structure 47
3.9 Vehicle Structure 49
3.10 Substructure 51
3.11 Mass Element 53
3.12 Fixed Mass Element 57
3.13 Point Aspect 58
3.14 Shape Aspects 60
3.14.1 Box 61
3.14.2 Cylinder 62
3.14.3 Cone 62
3.14.4 Sphere 63
3.14.5 Ellipsoid 63
3.14.6 Plane 64
3.15 Link Element 65
3.16 Static Aspects 67
3.16.1 Universal Static Parameters 67
3.16.2 Elastic Static Aspects 69
3.16.2.1 Static Type LE : Linear Elastic 70
3.16.2.2 Static Type EE : Exponential Elastic 71
3.16.2.3 Static Type AE : Arctangent Elastic 72
3.16.2.4 Static Type BE : Bilinear Elastic 73
3.16.2.5 Static Type SE : Segmented Elastic 75
3.16.3 Inelastic Static Aspects 77

3.16.3.1 Static Type AI : Arctangent Inelastic 78
3.16.3.2 Static Type BI : Bilinear Inelastic 79
3.16.3.3 Static Type SI : Segmented Inelastic 81
3.17 Dynamic Aspects 85
3.17.1 Universal Dynamic Parameters 86
3.17.2 Dampers 87
3.17.2.1 Dynamic Type LD : Linear Damper 88
3.17.2.2 Dynamic Type AD : Arctangent Damper 88
3.17.3 Magnifiers 89
3.17.3.1 Dynamic Type LM : Linear Magnifier 90
3.17.3.2 Dynamic Type AM : Arctangent Magnifier 90
3.18 Load Element 91
3.19 Output Information 94
3.20 Multiple-Event Modeling 98
3.20.1 Event Matching 98
3.20.2 Master File Overview 99
3.20.3 Master File Organization 99
3.20.4 Run Information 101
3.20.5 Event Information 103
3.20.5.1 Event Specifications 103

4 SISAME-3D Formulation 104
4.1 Rigid Mass Equations of Motion 104
4.2 Partitioned Equations of Motion 105
4.3 Mass Target Equations 106
4.4 Load Target Equations 108
4.5 Weighted Target System 109
4.6 Constrained Approximation Problem 110

5 SISAME-3D Implementation **111**

5.1 LCP Solution 111

5.2 Link Force Representation 113

5.2.1 Linear Approximants 113

5.2.2 Domain Transitions 115

5.3 Extraction Iteration 116

5.4 Integral Domain Targets 116

5.5 Supplementary Target Equations 117

5.6 Target Equation Weighting 118

5.7 Extracted Parameter Constraints 119

5.8 Phase I Procedure 120

5.9 Filtered Data Representation 121

5.10 Time Step Selection 121

5.11 Numerical Integration 122

A SISAME-3D Modeling Guidelines **124**

A.1 Understanding Extraction 124

A.2 Instrumentation 125

A.3 Model Configuration 126

A.4 Weight Extraction 126

A.5 Dynamic Extraction 127

A.6 Controlling and Refining Extractions 128

A.7 Assessing the Extracted Model 129

A.8 Multiple-Events 129

A.9 Common Problems 130

A.10 Predictive Simulations 130

References **133**

Index **135**

List of Figures

1.1 SISAME-3D angled offset vehicle impact model. 2

2.1 SISAME-3D Structure levels. 7

3.1 SISAME-3D dimensional systems. 23

3.2 SISAME-3D Master file template. 100

1 Introduction

The SISAME-3D (**S**tructural **I**mpact **S**imulation **A**nd **M**odel **E**xtraction) program is a general purpose tool for extracting and simulating lumped-parameter structural models in large deformation impact events.

The goal of the SISAME-3D methodology is to develop vehicle crash models of modest complexity that can accurately and efficiently predict the structural behavior over a range of events. Such models can be simulated in a variety of single and multiple vehicle crash event configurations providing a capability that is not practical with crash testing. The ability to efficiently study the effects of structural modifications is valuable for vehicle safety research. The ability to generate optimal structure redesigns that both meet crash performance targets and respect limitations on the design provides a powerful vehicle design capability that complements the strengths of finite element modeling.

SISAME-3D can generate optimal models directly from crash test data, bypassing the difficulties in obtaining accurate large deformation models from structural analysis, static testing, and finite element approaches. SISAME-3D can also be used to distill finite element models to more efficient lumped parameter models and to tune those models to better match observed test behavior. The ability to mix defined and extracted parameters, to extract models from multiple events, and the extensive estimate, bound, and referencing controls allows all available information to be used to obtain accurate and robust models. SISAME-3D provides a wide range of features, a self-documenting interface, and extensive input verification, making it suitable for serious modeling projects.

SISAME-3D was developed for the U.S. Department of Transportation, National Highway Traffic Safety Administration, Office of Crashworthiness Research.

This report documents the usage and theory of SISAME-3D. Section 2 presents the basic SISAME-3D modeling concepts. Section 3 contains the SISAME-3D Users Guide. Section 4 presents the SISAME-3D methodology formulation. Appendix A provides guidelines for SISAME-3D modeling.

1.1 Overview

SISAME-3D models are systems of rigid bodies (**Masses**) connected by massless flexible members (**Links**) representing the load-paths in the structure, such as the angled offset vehicle

impact model in Figure 1.1. Links can represent a range of load-paths, from simple axial linear-elastic springs to multi-axial, nonlinear, path-dependent, energy-dissipating structures.

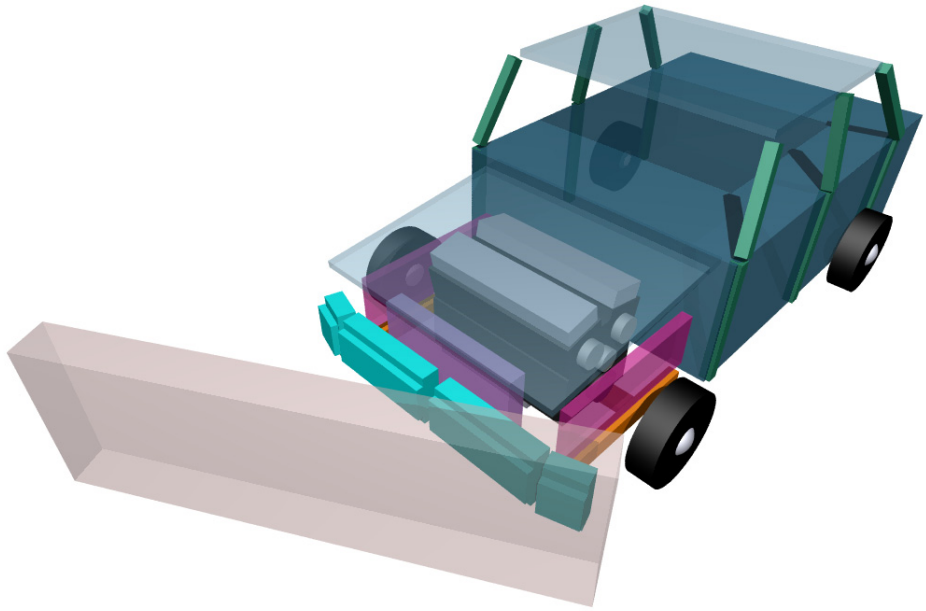


Figure 1.1: SISAME-3D angled offset vehicle impact model.

To simulate a model the Mass weight/inertia properties and initial velocities and the Link attachments and characteristics must be defined. Numerical simulation of the model can then be performed by SISAME-3D to determine the model's behavior.

Extracting a structural model from test or simulation data is intrinsically much more difficult than simulation. In general, extraction is the process of solving for some of the Link characteristics of the model when given some of the Mass motions and/or structural loads. Since realistic vehicle models have more Links than Masses, the instantaneous equations of motion are typically nonsquare and thus a determinate solution for the load-path forces at each time step is not possible. Instead, SISAME-3D performs a constrained least squares solution for the optimal model parameters using the entire time span of the instrument data. This turns the underdetermined instantaneous problem into an overdetermined optimization problem.

This global approach and the use of constraints to eliminate nonphysical behavior make the extraction process fairly robust. If the motions are accurate and the actual system can be accurately represented by the chosen model configuration, then the model will be able to accurately reproduce the original event and to predict events that are sufficiently similar to that event. If, in addition, the data is sufficiently complete and the model configuration is sufficiently accurate that only a small range of models can accurately match the data, then the model will be an accurate representation of the actual structure.

SISAME-3D is able to extract Link characteristics and Mass weight values. The inputs are the model configuration, defined Mass weights, motion histories of some or all of the Masses, and, possibly, inertial or applied load histories corresponding to sets of Masses or Links. The motion and load data are augmented by constraints on the Link characteristics (based on *a priori* knowledge of load-path behavior) and, optionally, user-specified estimate and/or constraint information.

In most cases a basic model extraction relying on SISAME-3D defaults will produce a good model. When greater control over the extraction is desired the user can adjust the target confidence bands and/or add supplementary information such as weight estimates, clearances between components, constraints on load-path forces, or target loads for load-path subsets. SISAME-3D is designed to use available information about a structure to refine its solution in a balanced way.

1.2 History

The historical approach to developing lumped-parameter vehicle crash models has been to use a combination of engineering analysis, experience, and static crush testing to specify the characteristics of the major load-paths. Models developed in this way have achieved only limited success because of the difficulty in capturing the dynamic crush behavior in a static test.

More recently finite element modeling has become the preferred method for structural crash analysis. While a powerful approach for analysis of a well-known structure, finite element models are expensive to develop and simulate and do not always accurately capture the dynamic crash performance. Although future advances should reduce these obstacles to finite element crash analysis, finite element modeling will remain an extremely poor tool for designing a structure to a desired crash performance.

The original approach for extracting models directly from crash test data was the Fiat methodology [2, 3, 6], which was effectively restricted to static models with only as many load-paths as mass elements (square systems), and performed a determinate solution for the load-path forces at each instant that did not necessarily produce physically meaningful structures.

The first extraction method using approximation techniques to allow an arbitrary number of extracted load-paths (nonsquare systems) is presented in the thesis of Dr. W.T. Hollowell [1]. This approach solved for optimal chronological linearizations of the static load-paths with automatic time segmentation, and was successful in identifying the characteristics of nonsquare models.

This approach was modified to the global method used in the uniaxial SISAME [11] program with the aims of providing greater robustness and directly generating simulation-ready mod-

els. SISAME-3D adds three-dimensional dynamics and a range of new capabilities to this framework, providing a unique and powerful impact modeling tool.

1.3 Capabilities

SISAME-3D can be used to model a range of structural impact events. This includes full frontal and offset vehicle impacts into fixed and moving barriers, two vehicle full frontal and offset impacts, and side impacts of a moving barrier or pole into a vehicle.

SISAME-3D can be used to distill small, efficient models from complex finite element (FEM) models for simulation of events sufficiently similar to the FEM-simulated "test" event.

SISAME-3D can serve as a high-level vehicle design complement to finite element analysis. Optimal models can be extracted from safety-enhanced input signals (based on occupant modeling or experience with other vehicles) to provide a direct redesign of the structure without trial-and-error finite element model simulations that can be costly and are likely to miss the optimal structure. During vehicle prototyping this design optimization can be performed using finite element models to provide the input signals and for verification of the redesigned vehicle's performance. The redesign extraction can be limited to a subset of the structure and the tradeoff between the match to the design signals and the change from the current structure can be controlled by the modeler. SISAME-3D can be readily extended to include material or construction cost functions and constraints in selecting the new design.

SISAME-3D can perform weight extractions to help define a model's weight distribution when vehicle tear-down data is incomplete or unavailable.

SISAME-3D provides a large enough set of load-path properties to support a broader range of modeling tasks. SISAME-3D vehicle models can be extended with simple biomechanical models to study the timing, severity, and effect of interaction of occupants with the vehicle structures. SISAME-3D can also be used to perform more general lumped-parameter modeling of impact and non-deformation events.

Multiple-event modeling can be performed to obtain models that are a balanced match to an optionally weighted combination of events having some components in common.

A number of features make SISAME-3D easy to work with. The Event file format is flexible and self-documenting, provides many defaults, and supports powerful referencing relationships between model components and parameters. Model extractions can incorporate any available supplementary information about the structure and the knowledge of the modeler into estimates and bounds on the extracted parameters that refine the model solution. The modeler can also control the relative weighting of the different types of target equations used in the extraction process. Known portions of a model can also be simulated while other portions are being extracted. The filtering specifications for each instrument signal can be

controlled separately. Extraction runs directly generate the signals and measure of fit report needed to assess the extracted model, and the model file needed to simulate it.

The filtering method developed for SISAME-3D modeling has other applications and is available separately in the SimFil program [7, 9].

1.4 Future

SISAME-3D is under ongoing development to extend the 3D modeling capabilities with additional load-path attachment types, contact interactions, constrained joints, and support for new modeling applications.

1.5 General Information

SISAME-3D is driven by self-documenting Event files. SISAME-3D can operate in three dimensional systems: SI, Metric or English. SISAME-3D uses NHTSA UDS format input and output signal files. Extensive error checking is performed on the inputs to assure that the program input requirements are met and that the results are meaningful.

SISAME-3D version 1.1 is written in object-oriented ANSI/ISO standard C++ and is available for a number of computing platforms.

The associated VeCor [10] program can be used to assess and preprocess instrumentation signals prior to modeling. The filtering used in SISAME-3D is based on the SimFil [7, 9] program.

1.6 Documentation

This document applies to SISAME-3D version 1.1. The VeCor program is documented in [10]. The SimFil program is documented in [7, 9].

2 SISAME-3D Modeling Concepts

This section presents the basic terminology, conventions, and concepts used in SISAME-3D modeling. The terminology and conventions are described in Section 2.1. The output types available are described in Section 2.2. The extraction process is discussed in Section 2.3.

2.1 SISAME-3D Models

SISAME-3D models are lumped-parameter structures consisting of rigid **Mass** and **Fixed Mass** Elements connected by massless flexible **Link** Elements corresponding to the load-paths of the model. **Load** Elements can be added to specify driving loads or to group a set of Link loads or Mass inertial loads. Model Elements are contained in **Structures** that can be nested. Structures and Elements are the **Components** of a SISAME-3D model.

Simulation models are models with all Mass weights and load-path characteristics defined. Simulation models contain only simulated or driven Masses, defined Links, and group or driving Loads, as defined below.

Extraction models are models with undefined Mass weights and/or load-path characteristics. Extractions are based on finding the unknown model parameters to best satisfy a number of **target equations** subject to **constraints**. Target equations may include the equations of motion for Masses, equations targeting sums of load-path and/or Mass inertial loads to a given load history, and modeler-supplied estimates for the unknown parameters. Constraints are generated automatically to ensure physically meaningful behavior and the modeler can specify additional constraints on the extracted parameters.

2.1.1 Structures

SISAME-3D **Structures** are composites of model Elements and nested Structures. A Structure's **member** Components are those Components for which it is the immediate **enclosing** Structure. The tree of all member Components and nested Structure Components are a Structure's **contained** Components. Each Structure has an initial coordinate frame and velocity relative to which the initial coordinate frames and velocities of its contained Components are defined, but a Structure is a collection that does not have its own motion history.

A Structure is also used to specify properties that apply to its contained Components, such as a total weight.

There are three levels of SISAME-3D Structures:

- The **Model** encompasses the whole model.
- **Vehicles** can be members of the Model.
- **Substructures** can be members of the Model or a Vehicle.

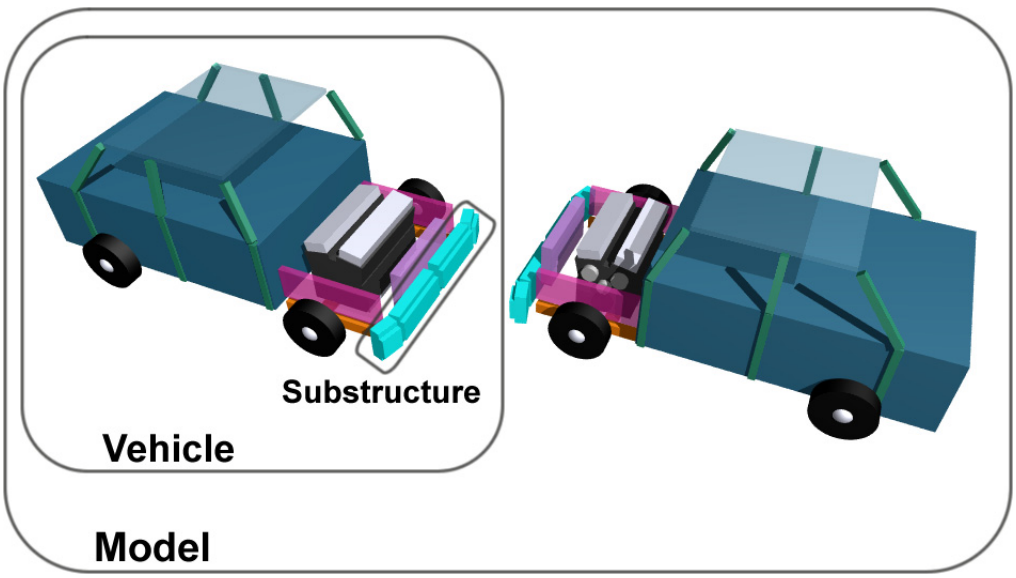
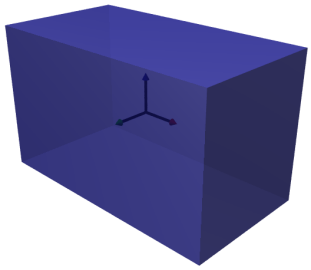


Figure 2.1: SISAME-3D Structure levels.

2.1.2 Elements

SISAME-3D Models can contain Mass, Fixed Mass, Link, and Load **Elements**.

Masses



Masses are rigid bodies that move with the linear and rotational degrees of freedom of the Model.

There are three classes of Masses.

- A **simulated Mass** has motions that are determined by the applied forces, integrating its equations of motion from an initial position and velocity. A simulated Mass can have an associated **comparison signal** to facilitate comparing the simulated motion with a given motion signal.
- A **constant-velocity (c-v) Mass** maintains its initial velocity (linear and angular) throughout the event.
- An **instrumented Mass** has motions that are specified by an associated **motion signal**. Instrumented Masses can be **extracted-weight Masses**, which have their weight determined by the extraction process. Instrumented Masses are sub-classified as target or driven Masses:
  - A **target Mass** is used to generate target equations of motion for the model extraction (see Section 4.3). Target Masses are generally associated with a motion instrument.
  - A **driven Mass** is an instrumented Mass that is not used to generate target equations. Driven Masses are used for Masses that have motions that are either determined by forces external to the Model, such as a test sled with a controlled motion, or are very rough approximations for which target equations are not desired, such as positions taken from test camera films.

Since SISAME-3D can simulate defined substructures of a model while other substructures are being extracted, an extraction run can include Masses of any class. A simulation run can include simulated and/or driven Masses, but no target Masses.

The relevant quantities for each Mass are:

Mass Quantity	Symbol	Formula
acceleration	a	
velocity	v	$\int_t a$
position	p	$\int_t v$
weight (in mass units)	m	
weight (in force units)	w	
inertial force	g	$w\,a$
exerted force	$\tilde{g}$	

where the exerted force is the net force acting on the Mass.

A Mass can have any number of **Points** and **Shapes** (see below).

Fixed Masses

Fixed Masses are rigid bodies fixed to the inertial (ground) reference frame. Any number of Fixed Mass elements can be defined and positioned in a Model. A Link can be attached to a Fixed Mass to represent an impact with a ground-fixed object, such as a rigid barrier. A Fixed Mass can also be used to represent the ground surface.

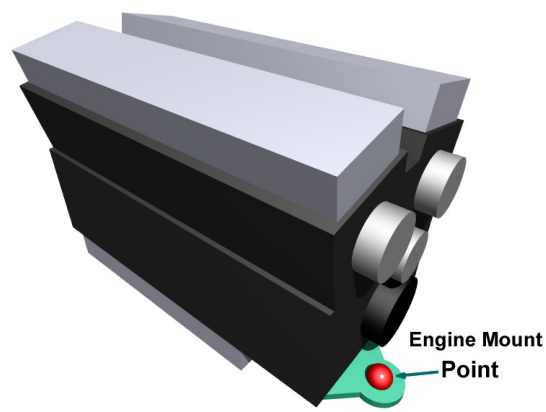
A Fixed Mass can have any number of **Points** and **Shapes** (see below).

Points and Shapes

Mass and Fixed Mass Elements can have any number of Points and Shapes.

A **Point** is a spatial point and coordinate frame at a fixed position on a Mass or Fixed Mass. Points are used as Link attachments, as points where driving Loads are applied, and as instrument locations. Each Mass and Fixed Mass has an implicit Point at its center of gravity (CG).

A **Shape** is a geometric shape at a fixed position on a Mass or Fixed Mass. Shapes control the contact interactions among Masses and Fixed Masses.

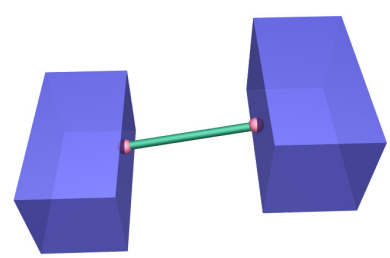


Point and Shape entities are called **Aspects** of their Mass or Fixed Mass.

Links

Links are massless flexible load-carrying members that connect two Mass or Fixed Mass Points.

There are two classes of Links.



- A **defined Link** has fully defined characteristics.
- An **extracted Link** has some unknown characteristics that are determined by model extraction.

An extraction run Model can have both extracted and defined Links. A simulation run Model can have defined Links but no extracted Links.

Each Link is connected to two Mass or Fixed Mass Points, p_1 and p_2. The relative position of p_1 and p_2 determines the Link behavior. Axial forces are generated along the line between p_1 and p_2. Because the Link is massless the net force/torque on the Link with respect to any point is always zero.

When p_1 and p_2 pass through or near the same point the axial force direction can change abruptly: correct positioning of the associated Masses, Fixed Masses, and Points is necessary to avoid unwanted crossovers. If p_1 and p_2 are coincident when the Masses are in their resting positions the Link can only experience tension axially since its length can never be less than its resting length.

Links can have static and dynamic characteristics called **Aspects**.

A **Static Aspect** determines the static force generated by the Link as a function of its deflection history. A **Dynamic Aspects** determines the deflection rate-dependent contribution to the static force that gives the total Link force. A range of elastic (non-dissipating) and inelastic (dissipating) Static Aspect types are available. A Link can have one Static Aspect. A pure static load-path is a Link with no Dynamic Aspect.

A range of Damper and Dynamic Magnifier Dynamic Aspect types are available. Dampers generate a deflection rate-dependent force that is independent of the Link's static force, if any. Magnifiers multiply the static force by a factor that is a function of the deflection rate. A Link can have one Damper and one Magnifier Dynamic Aspect. A pure Damper load-path is a Link with no Static or Magnifier Aspects.

The relevant quantities for each Link are:

Link Quantity	**Symbol**	**Formula**
deflection	x	$d(p_1) - d(p_2)$
relative velocity	r	$v(p_1) - v(p_2)$
static force	f^s	f^s
dynamic magnification	M	$M(r, \sigma(f^s))$
damping force	δ	$\delta(r)$
force	f	$Mf^s + \delta$
energy	e	$\int_x f$

where $\sigma(\cdot)$ is the sign function. The Link quantities use a **compression–positive sign convention**.

Loads

There are three classes of SISAME-3D Loads.

- A **group Load** is a sum of a specified collection of Mass inertia and/or Link loads. A group Load can have associated **comparison signals** to facilitate comparing the collection's loading with a given loading history.
- A **target Load** is a sum of a specified collection of Mass inertia and/or Link loads with associated **target signals**. Target Loads are intended to provide a target for a loading that depends on extracted parameters.
- A **driving Load** has a signal-based loading that is applied to one or more Points.

The relevant quantities for each Load are:

Load Quantity	Symbol	Formula
force	f	
impulse	i	$\int_t f$
integrated impulse	j	$\int_t i$

2.1.3 Model Configurations

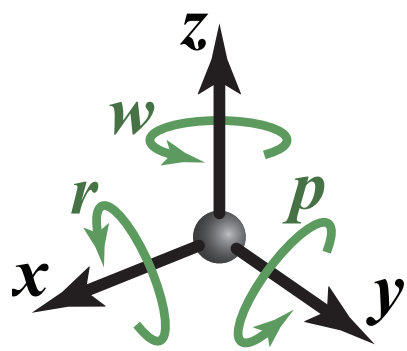
In general, any number and combination of Structures and their Mass, Fixed Mass, Link, and Load Elements is a valid SISAME-3D Model.

Two limitations arise when extraction runs include simulated Masses, allowing known portions of the structure to be included in the extraction process:

- Extracted Links cannot be connected to simulated Masses: the simulated Masses must be connected to the rest of the Model with defined Links.
- Simulated Masses cannot have extracted weight or inertia parameters.

2.1.4 Degrees of Freedom

SISAME-3D Models exist in a right-hand Cartesian coordinate system and can have up to six degrees of freedom (DOF). The translational axes are identified as $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ and the corresponding rotational axes as $\boldsymbol{r}$, $\boldsymbol{p}$, and $\boldsymbol{w}$ (for roll, pitch, and yaw).



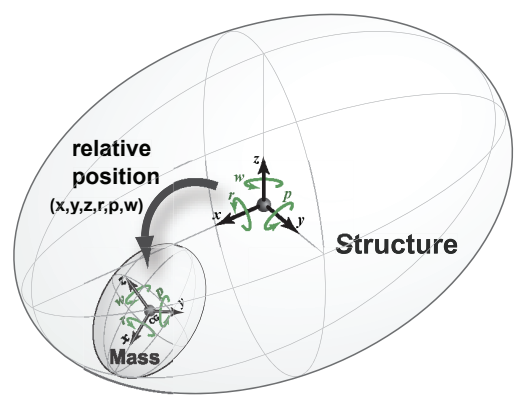
All Components of a Model share a common degrees of freedom axis set. Any meaningful combination of the six axes can be specified, including Models with only rotational degrees of freedom.

2.1.5 Coordinate Frames

SISAME-3D Models are implicitly located in an inertial (ground) coordinate frame and each Structure, Mass, and Fixed Mass has its own local coordinate frame. Coordinate frames are fixed to an **origin** point. The origin of a Mass with weight/inertia mass properties is its **center of gravity (CG)**. The origins of other Components are arbitrary reference points. All coordinate frames have the axes in the Model's degrees of freedom. Structure coordinate frames are used for the relative positioning of member Components. Mass and Fixed Mass coordinate frames are used for the relative positioning of their Points and Shapes. A Mass coordinate frame moves with the Mass during the event time span.

Model Components have resting and initial positions that are specified relative to their enclosing Structure's coordinate frame (the inertial frame for the Model Structure).

Component positions are specified by a **generalized position** tuple ($\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$, $\boldsymbol{r}$, $\boldsymbol{p}$, $\boldsymbol{w}$) of linear and rotational coordinates. SISAME-3D uses the convention that translations ($\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$) are applied first followed by the **Cardan angle** sequence of rotations ($\boldsymbol{r}$, $\boldsymbol{p}$, $\boldsymbol{w}$) (roll, pitch, yaw) about body-fixed axes (the order of rotations is significant). Axes not present in the Model's degrees of freedom are omitted from the tuple sequence, maintaining the order.



Positional tuple entries default to zero, yielding a frame that is coincident with that of the enclosing Structure.

Initial velocities are also specified with relative **generalized velocity** tuples ($\boldsymbol{v_x}$, $\boldsymbol{v_y}$, $\boldsymbol{v_z}$, $\boldsymbol{v_r}$, $\boldsymbol{v_p}$, $\boldsymbol{v_w}$). Initial velocities are relative to those of the coincident point on the enclosing Structure.

Velocity tuple entries default to zero, giving the same rigid body velocity as the enclosing Structure. The use of these relative position and velocity tuples with default entries of zero makes the specification of typical initial conditions simple. The initial velocity for a Structure with all of its contained Components that is initially moving as a rigid body can be specified entirely using the Structure's initial velocity tuple.

2.2 SISAME-3D Outputs

SISAME-3D can generate a range of outputs from a run upon request:

- Time series signal files for selected Masses, Links, and Loads.
- Link X-Y history files, such as force *vs.* deflection histories.
- Link parametric static and dynamic characteristic files for visualization of the Link properties.
- Tabular text files with key time series signals for selected Masses, Links, and Loads.
- A fit measure report showing the quantitative fit to the target and comparison signals.
- A SISAME-3D Event file that provides a simulation-ready model from an extraction run or a processed version of the input Event file from a simulation run.
- A POV-Ray™ model image file for verification and documentation of the model.

SISAME-3D provides both filtered instrument and extracted model signals from extraction runs to help assess the extracted model. The effective extracted outputs show the motions and loads that the extracted model would experience at each instantaneous state of the event. Comparing them to the filtered instrument signals shows the quality of the model's fit in the domain in which SISAME-3D extracts the model. If this comparison is satisfactory the extracted model should be used to resimulate the event to verify that its simulation still tracks the instrument signals satisfactorily.

Because extraction is an approximation process, the forces on a Mass will not precisely sum to its measured $w\,a$ value in extraction run outputs as they would in a simulation. Resimulation of an extracted model will give exact consistency for simulated Masses, but will not exactly match the original motion data.

2.3 The Extraction Process

This section outlines the steps required in the extraction process. The relevant software is indicated for each step.

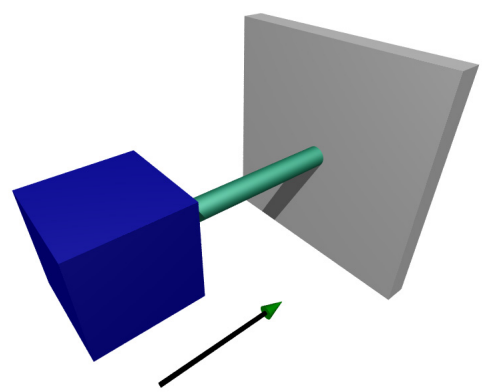
1. Prepare input data.
   (a) Assess/refine test motion and load time series signals. VeCor
   The VeCor program or another approach should be used to assess and refine test-derived instrument signals to assure sufficiently accurate load-path deflections and other quantities important in the extraction process.
   (b) Combine and assign motion signals to the Model's instrumented Masses and load signals to the instrumented Loads.
2. Build extraction model.
   (a) Assign known weights and inertias to target Masses.
   (b) Select filtering for instrumented Masses and Loads.
   (c) [Optional] Extract unknown weights using target Load data. SISAME-3D
   (d) Choose load-paths to include as defined and extracted Links.
   (e) Add known substructures of defined Links and simulated Masses.
   (f) Add any defined Static or Dynamic Parameters to extracted Links.
   (g) Add extracted Link and/or weight specifications including any desired Parameter estimates and/or constraints.
3. Perform extraction run. SISAME-3D
   (a) Request SISAME-3D outputs of filtered and extracted signals for target Masses and Loads, parametric Link representations, the fit measure report, and the extracted model file.
   (b) Assess model by evaluating the fit of target Masses and Loads and examining the extracted Link characteristics. Refine model inputs and re-extract, if desired.
   (c) The sensitivity of the extracted model Parameters can be roughly gauged by repeating the extraction with increased Parameter conditioning (a smaller *ConPC*) and seeing how much the solution changes.
4. Perform resimulation run. SISAME-3D
   (a) Use the extracted model file to resimulate the extraction event and compare with the filtered motions produced from the extraction run to assess the model's baseline simulation accuracy.

3 SISAME-3D Users Guide

SISAME-3D is driven by a self-documenting ASCII text input Event file. Event files are divided into three sections describing the run, the model, and the desired outputs. Each section is made up of a number of specification fields of the form *Tag = Value*.

Here is a very simple SISAME-3D Event file:

```
SISAME-3D Event File

Run Information

  RunID=Run
    DelTOut=.0001  FinTOut=.1

Model Information

  ModelID=Model
    DOF=(x)  DimSys=Metric

    FixedMassID=Wall

    MassID=Mass
      Weight=1000  IniV=50

    LinkID=Link
      Point1=Mass  Point2=Wall
      Sta=LE  S=10000

Output Information

  OutClass=MassTS  Qty=AVP  Mass=Mass
```



This Event file describes a one-dimensional x-axis (`DOF=(x)`) Model with one Mass Element, `Mass`, connected by a Link Element, `Link`, to a fixed `Wall`. The model uses the Metric dimensional system. The `Mass` weighs 1000 kilograms and the `Link` has a linear elastic (`LE`) Static Aspect with a stiffness of 10000 newtons/millimeter. The initial conditions have the `Mass` moving at 50 kilometers/hour. All properties of the system are defined, so this is a **simulation run**. Time series file output of the `Mass` acceleration, velocity, and position are specified.

This Users Guide details the construction and usage of SISAME-3D models.

3.1 Running SISAME-3D

The SISAME-3D program is run using the command line syntax:

`SISAME3D` [*input_file*]

where *input_file* is an optional Event or Master Event file name. An Event input file is used for modeling a single event and a Master Event file is used for multiple-Event modeling. If no Event file is specified on the command line SISAME-3D will prompt for one.

Input file names generally have one of two forms although neither is required:

base.type.extension or *base.type*

where *base* is the base file name, *type* indicates the SISAME-3D file type, such as `ext` or `sim`, and *extension* is a file extension selected to identify all SISAME-3D Event files, such as `sis`, `s3d`, or `sisame3d`. If an *extension* is used it will also be used in any output Event file names.

SISAME-3D begins by processing and checking the Event file. A log file is created having the same name as the Event file name but with a `.log` extension. For multiple-Event runs, a log file is created for the Master file and for each Event file. All warning and error messages go to the log file (error messages are also written to the console). A summary of the run and model specifications is written to the log file. SISAME-3D then performs the extraction or simulation, sending progress messages to the log file and the console. Finally, SISAME-3D generates the requested outputs and writes a summary of the output files generated to the log file.

3.2 Model Construction

A SISAME-3D Event file has three main sections:

- `Run Information` Specifies the parameters controlling the run.
- `Model Information` Specifies the Model Components and their properties.
- `Output Information` Specifies the outputs desired from the run.

The properties within each section are specified by **field specifiers** of the form *Tag* = *Value* where *Tag* is the field tag name and *Value* is the specified value for the field. Fields can have single character, text string, Boolean (true/false), or numeric values depending on the type of field.

Some fields are **ID specifiers** that assign user-selected identifier strings to Components or sub-Component entities of a Model. ID specifiers have tags ending in "`ID`". ID specifiers are described in Section 3.3.

The numeric, Boolean, and character flag specification fields are called **Parameters**. Some Parameters are specified in aggregate groups called Parameter **blocks** and **tuples**.

A Model can include **Structure** Components that contain Mass, Fixed Mass, Link, and Load **Elements** as well as nested Structures. The Structure hierarchy from highest to lowest is: Model, Vehicle, Substructure. A Structure in the Event file contains all of the Model Components from its specifications to the next Structure of equal or higher level.

Masses and Fixed Masses can have **Point** Aspects to provide Link and instrument attachment points and as points where driving Loads are applied, and **Shape** Aspects to define their shapes and control contact interactions. Each Mass and Fixed Mass has an automatically defined Point with ID CG at its origin, which is its center of gravity if it has mass/inertia properties.

Links can have **Static** and **Dynamic** Aspects that provide their loading characteristics.

Model Components are specified following their ID fields, so, for example, `MassID=Mass` precedes the weight and initial velocity specifications in the sample Event file above.

Components and Parameters have names that are used to refer to them outside of their own specifications. Names are constructed by concatenating IDs and Tags, so, for example, the `Weight` Parameter of a Mass with ID `Engine` in a Vehicle with ID `Car` has the name `Car.Engine.Weight` (a *`ModelID.`* prefix is not required). Shorthand names can be used in most contexts. Sections 3.3.2 and 3.3.4 describe names and references in detail.

3.2.1 Model Configurations

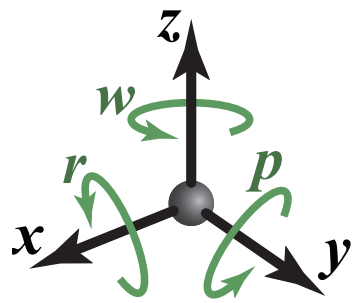
Generally, any number and combination of Structures and Elements is a valid SISAME-3D Model: Model size is limited only by the available system memory.

Model configurations are subject to two restrictions:

- Extracted Links cannot be connected to simulated Masses.
- Simulated Masses cannot have extracted weight or inertia parameters.

3.2.2 Degrees of Freedom (DOF)

Models exist in a right-hand Cartesian coordinate system and can have up to six degrees of freedom (DOF). The translational axes are identified as $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ and the corresponding rotational axes as $\boldsymbol{r}$, $\boldsymbol{p}$, and $\boldsymbol{w}$ (for roll, pitch, and yaw).



A Model's degrees of freedom is specified in the `DOF` field of the Event file. Any meaningful combination of the six axes can be specified.

A Model's *`DOF`* must include at least one axis. The *`DOF`* can have zero, one, or three, but not two, rotational axes. If either linear axis in the plane of a rotational axis is present they must both be. So `(x)`, `(xr)`, `(xyw)`, and `(rpw)` are valid *`DOF`*s but `(xw)` is not.

3.2.3 Coordinate Frames

Models are located in an implicit inertial (ground) coordinate frame. Each Structure, Mass, and Fixed Mass has a local coordinate frame fixed at an **origin** point. The origin of a Mass with weight/inertia mass properties is its **center of gravity (CG)**. The origins of other Components are arbitrary reference points. Points and Shapes have coordinate frames at fixed positions relative to their Mass or Fixed Mass CGs. All coordinate frames have the specified `DOF` axes.

Structures, Masses, and Fixed Masses have **resting positions** at which all Link deflections are, by definition, zero. Structures and Masses can also have **initial positions** at time-zero that are displaced from their resting positions, which can generate non-zero Link deflections and preload. Structures and Masses can be given specified **initial velocities**.

In constructing a Model, positions and velocities are specified as **relative** values that default to zero, simplifying the specification of initial conditions for typical modeling tasks. For example, all Components of a Structure can be given a consistent rigid body initial motion by setting just the Structure's initial velocity. These relative positions and velocities are specified by generalized spatial Parameter tuples.

3.2.4 Generalized Spatial Parameter Tuples

The position and velocity of Structures, Masses, and Fixed Masses are specified by **generalized position** and **generalized velocity** Parameter tuples relative to the position and velocity

of the enclosing Structure. The CG-relative position of Mass and Fixed Mass Points and Shapes are also specified with generalized position tuples.

A **generalized spatial tuple** is a Parameter tuple containing both the linear and rotational coordinates. For example, in a Metric Model `IniP=( x:-500 z:80 w:.2 )` specifies a position of $x = -500$ mm, $z = 80$ mm, and $w = .2$ radians (a rotation of .2 radians about the z-axis). Rotational coordinates in a generalized position are in radians. Rotational coordinates in a generalized velocity are in radians/second.

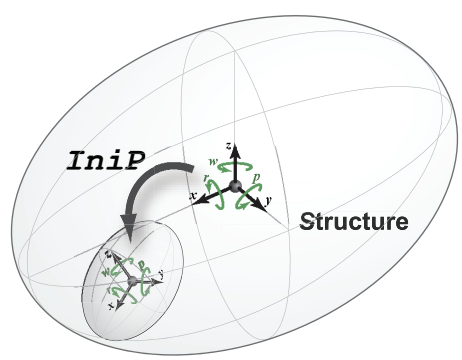
SISAME-3D generalized spatial tuples follow the convention that translations ($\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$) are applied first followed by the **Cardan angle** sequence of rotations ($\boldsymbol{r}$, $\boldsymbol{p}$, $\boldsymbol{w}$) (roll, pitch, yaw) about body-fixed axes. Axes not present in the Model's ***DOF*** are omitted from the tuple sequence, maintaining the order. Values for unspecified ***DOF*** axes default to zero, which gives positions and velocities consistent with the enclosing Component since the tuples are relative to its frame.

Parameter tuples using axis pair keys, such as `xy`, or axis type keys (`l` for linear and `r` for rotational) are also used in SISAME-3D. The full syntax for specifying Parameter tuples is described in Section 3.4.3.

3.2.5 Resting Position

When each Structure, Mass, and Fixed Mass is in its resting position Link deflections are, by definition, zero. This may not be an equilibrium state for the Model since Links can generate non-zero forces at zero deflection and forces due to external Loads and gravity may be present.

The resting position of each Structure, Mass, and Fixed Mass is specified relative to its enclosing Structure by a generalized position tuple tagged `IniP` that contains the body-fixed coordinates required to move from the enclosing Structure frame to the resting position. The Model Structure resting position is specified relative to the inertial frame.



`IniP=( z:-220 p:.8 w:.2 )`

The ***IniP*** coordinates default to zero. If ***IniP*** is not specified the Component's coordinate frame is coincident with that of the enclosing Structure.

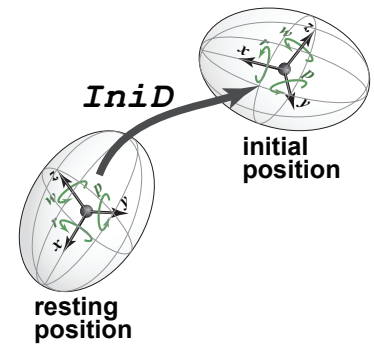
The absolute (inertial frame) resting position of a Component is computed by composing the spatial transforms corresponding to the ***IniP*** tuples of all containing Structures and the Component itself.

3.2.6 Initial Position

Each Structure and Mass can have an initial position at time-zero that is displaced from its resting position. This can preload the Model with non-zero Link deflections and forces.

The initial displaced position of each Structure and Mass is specified relative to its enclosing Structure by a generalized position tuple tagged `IniD` that contains the body-fixed coordinates required to move from the resting position to the initial position.

The *`IniD`* coordinates default to zero. If *`IniD`* is not specified the Component's initial position is the same as its resting position displaced by the containing Structures' *`IniD`* specifications, if any.



`IniD=( x:-90 z:75 p:-.9 )`

If a Structure has a non-zero *`IniD`* but all its contained Components have a default zero *`IniD`* then the Links internal to the Structure will have zero initial deflections even if those that connect outside the Structure do not.

The absolute (inertial frame) initial position of a Component is computed by composing the spatial transforms corresponding to the *`IniP`* and *`IniD`* tuples of all containing Structures and the Component itself. Fixed Masses are fixed at their resting positions and are not affected by the *`IniD`* of their containing Structures.

3.2.7 Rotational Position

The resting and initial rotational position, or orientation, of Structures, Masses, and Fixed Masses is computed by composing the spatial transforms corresponding to their nested *`IniP`* and *`IniD`* relative positions.

Representations for the 3D rotational position cannot distinguish the number of revolutions that may have brought the Component to its final orientation. SISAME-3D treats the resting and initial orientations as having Cardan angles on $(-\pi, \pi]$ radians. There are also two representations for each 3D orientation with different Cardan angle set (flipping the sign of the cosine of the pitch and adding $\pm\pi$ to the roll and yaw angles). For these reasons the rotational positions used and output by SISAME-3D may differ from the user's expectations.

Input rotational position (Cardan angle) signals are automatically adjusted, if necessary, to remove revolutions and/or select the alternate representation to match the internal orientation representation.

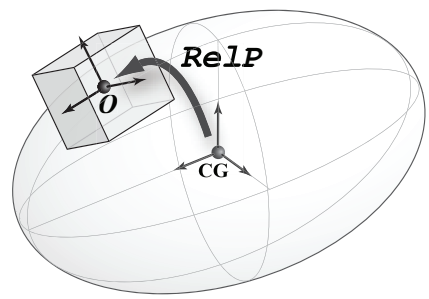
Output rotational position (Cardan angle) signals have the intial rotational positions computed by SISAME-3D but are able to track angles outside of $(-\pi, \pi]$ over the event time span using a heuristic algorithm to avoid signal discontinuities. This algorithm will produce reasonable signals as long as the output time step is not large relative to the revolutional frequency of the motions.

3.2.8 Point and Shape Position

Points and Shapes have coordinate frames at fixed positions relative to their Mass or Fixed Mass CG Point frames.

The position of Points and Shapes are specified relative to its Mass or Fixed Mass CG Point by a generalized position tuple tagged `RelP`.

The absolute (inertial frame) position of a Point or Shape at any time in the event is computed by composing the spatial transform corresponding to its ***RelP*** with the transform corresponding to its Mass or Fixed Mass CG position at that instant.

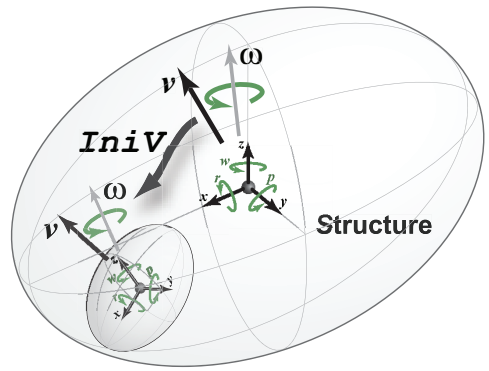


`RelP=( x:65 z:90 p:.1 w:.2 )`

3.2.9 Initial Velocity

Each Structure and Mass has an initial velocity at time-zero.

The initial velocity of each Structure and Mass is specified relative to the velocity of a point on the enclosing Structure that is coincident with its frame origin by a generalized velocity tuple tagged `IniV`. ***`IniV`*** specifies this relative velocity in the local initial position coordinate frame of the Structure or Mass. The Model Structure initial velocity is specified relative to the inertial frame.



`IniV=( x:50 r:1 )`

The ***`IniV`*** coordinates default to zero. If ***`IniV`*** is omitted the Component's initial velocity is consistent with the rigid body initial velocity of the enclosing Structure.

The absolute (inertial frame) initial velocity of a Component is computed by transforming the relative velocity specified in ***`IniV`*** to inertial frame coordinates and adding it to the absolute initial velocity of the coincident point on the enclosing Structure.

3.2.10 Dimensional Systems

SISAME-3D can operate in the three dimensional systems shown in Figure 3.1.

SISAME-3D Dimensional Systems

Data Type	SI	Metric	English
Time	SECONDS	SECONDS	SECONDS
Frequency	HZ	HZ	HZ
Distance	METERS	MILLIMETERS	INCHES
Velocity	METERS/SEC	KILOMETERS/HOUR	MILES/HOUR
Acceleration	METERS/SEC**2	G'S	G'S
Angular Distance	RADIANS	RADIANS	RADIANS
Angular Velocity	RADIANS/SEC	RADIANS/SEC	RADIANS/SEC
Angular Acceleration	RADIANS/SEC**2	RADIANS/SEC**2	RADIANS/SEC**2
Mass	KILOGRAMS	KILOGRAMS	POUNDS
Inertia tensor	KILOGRAM-METERS**2	KILOGRAM-METERS**2	POUND-FEET**2
Force	NEWTONS	NEWTONS	POUNDS
Impulse	NEWTON-SECONDS	NEWTON-SECONDS	POUND-SECONDS
Torque	NEWTON-METERS	NEWTON-METERS	POUND-FEET
Angular Impulse	NEWTON-METER-SECONDS	NEWTON-METER-SECONDS	POUND-FOOT-SECONDS
Energy	JOULES	JOULES	FOOT-POUNDS

Figure 3.1: SISAME-3D dimensional systems.

3.3 IDs, Names, and References

The constituents of a SISAME-3D Model are referred to by name in the Event files. The construction of these names is described below.

3.3.1 SISAME-3D IDs

The Components and Aspects of a SISAME-3D Model and the Run and Event specifications of the SISAME-3D Event and Master files are identified by user-assigned **IDs**. These IDs are used to form the names that appear in references and output file names (see Section 3.3.2 below).

Here are the rules for SISAME-3D IDs:

- IDs can be up to twenty (20) characters long.
- IDs are case-sensitive.
- IDs can include alpha-numeric characters and the four characters `_ - @ %` but may not begin with any of the three characters `- @ %` .
- IDs may not contain internal space, tab, null, newline, or carriage return characters or any of these characters: `\ / | ~ ! # $ ^ & : ; . , " ' ` + * = ? < > [ ] ( ) { }` .
- Event IDs in a Master file must be case-insensitively distinct from other Event IDs.
- Component IDs must be case-insensitively distinct from other member IDs in the same Structure and distinct from all containing Structure and Event IDs. IDs in different Structures and Events can be the same.
- Aspect IDs must be case-insensitively distinct from other Aspect IDs in the same Component and distinct from all containing Component and Event IDs. IDs in different Components can be the same.
- Components or Events with IDs that are the same as a Static or Dynamic Aspect category or type tag (`Sta`, `Dyn`, `Mag`, `Dpr`, `LE`, `EE`, `AE`, `BE`, `SE`, `AI`, `BI`, `SI`, `LD`, `AD`, `LM`, or `AM`) cannot contain an Aspect of that type so such IDs are best avoided.
- IDs make up part of the output signal file names. The rules for IDs are designed to avoid illegal or nonunique file names for common modern computing platforms, but care should be taken in choosing IDs if compatibility with other platforms or storage media (various ISO9660 optical disc formats in particular) is required. File name length limits could also be a factor on some media formats if long IDs are used.

3.3.2 Component Names and References

Component Names

Components are referred to by their **Component names**. The table below shows the symbolic Component names used in this document and how they are constructed in SISAME-3D from the Component IDs. In this section **Component** means any entity in the full Event hierarchy, including the Model Components, the Events, and the Master and Event Run specifications.

Component Names			
Symbolic			**Constructed**
Event Component	*Master Run*	*Run*	`RunID`
	Event	*Event*	`EventID`
	Event Run	*Run*	*Event*`.RunID`
Model Component	*Structure*	*Model*	*Event*`.ModelID`
		Vehicle	*Model*`.VehicleID`
		Substructure	*Structure*`.SubID`
	Element	*Mass*	*Structure*`.MassID`
		Fixed Mass	*Structure*`.FixedMassID`
		Link	*Structure*`.LinkID`
		Load	*Structure*`.LoadID`

The *`EventID`* can generally be omitted unless referring to a Component in another Event. In a single-Event run an Event ID of `Event` can be used if needed to disambiguate a reference to a Run Parameter from a Component with the same ID as the *`RunID`*. In a multiple-Event run an ID of `Events` can be used in front of the Event ID as the ID for the whole Event system if needed to disambiguate the names of Components in different Events. The IDs `Event` and `Events` are best reserved for this purpose.

For example, in a single-Event run the name of a Mass with ID `Engine` that is in a Vehicle with ID `Car` that is in a Model with ID `Model` is `Model.Car.Engine`.

Component names become part of the output signal file names. The *`EventID`* becomes a path name in a multiple-Event run. The Event *`RunID`* is not part of the Component names so distinct SISAME-3D runs should be performed in separate directories.

Component References

Component references are used in SISAME-3D input files to specify Link attachment Masses, reference Links, Load members, output specifications, and as the prefix of Parameter references. In this section **Component** means any entity in the full Event hierarchy, including the Model Components, the Events, and the Master and Event Run specifications. Component references can refer to Components in other Events.

Shorthand Component references are supported, allowing prefix portions of the name to be omitted when the reference resolution algorithm (see below) will match the desired Component.

Component references are resolved by first searching up the containment hierarchy from the local context for a Component with an ID matching the leading ID in the reference in this order:

1. The local context Component.
2. Member Components if the local context is a Composite.
3. For each level up the containment tree from the local context up to the root of the Event system:
   (a) The containing Composite.
   (b) Members of the containing Composite.

Here **Composite** means any Component containing other Components: a Structure, an Event, or the whole Event system. When a match is found the remaining IDs in the reference are checked. If they do not match the name of a Component contained in the leading ID match Component the search continues up the containment hierarchy. Component reference ID matching is case-sensitive.

References are taken to be in the local context in which they occur. Output specifications are in the Model context.

So, for example, within Vehicle `Car1` a Link attachment of `Point1=Engine` suffices as a shorthand reference to the engine in `Car1`, but `Point1=Car2.Engine` is required to reference the engine in Vehicle `Car2`.

3.3.3 Element Set References

Element set references simplify the specification of sets of Elements in Load member and output Element lists. They look like Element names where each part can be an ID, an ID wildcard, or a set of IDs. An Element set reference is of the form:

[*ModelID*.] [*VehicleIDSet* .] [*SubIDSet* .] *ElementIDSet*

where the *ModelID.* , *VehicleIDSet.* , and *SubIDSet.* prefixes can be used to distinguish references from Elements with the same ID in the local context, and where each *IDSet* is of the form:

$$ID \quad \text{or} \quad \{ID1, ID2, \ldots, IDn\} \quad \text{or} \quad ID1, ID2, \ldots, IDn$$

where each *ID* can be a Component ID or one of the wildcards IDs `*` or `[*]` requesting all contained or member Components, respectively. Since Element set references are used for sets of Elements within one Event an *EventID* prefix is not needed.

SISAME-3D constructs an Element set from a set reference according to the following rules:

- Each possible combination of *ID*s is used to generate a potential Element reference that is then is resolved using the search algorithm given in Section 3.3.2 above.
- Valid Elements of the appropriate type are added to the list. Nonexistent Elements or Elements of the wrong type are discarded without triggering a warning or error.
- The wildcard *ID* `*` selects all contained Components of a Structure.
- The wildcard *ID* `[*]` selects all member Components of a Structure.
- A reference that has a wildcard *ID* in the leading *IDSet* is resolved in the local context, *not* by searching up the containment hierarchy, as follows:
  - In an Element context (a Load member list) a leading wildcard selects all contained (`*`) or member (`[*]`) Components of the enclosing Structure.
  - In a Structure context (an output Element list) a leading wildcard selects all contained (`*`) or member (`[*]`) Components of the Structure.

The following are examples of valid Element set references:

```
Car.*      *.{FrontFrame,RearFrame}      {Car1,Car2}.{OccComp,Engine}
```

3.3.4 Aspect Names and References

Components of SISAME-3D Models contain groupings of related characteristics called Aspects. Masses and Fixed Masses can contain Point and Shape Aspects. Links can contain Static and Dynamic Aspects. The construction of both types of Aspect names is shown below.

Shorthand Aspect references are supported, allowing a prefix or all of the *Component* portion of the name to be omitted when the reference resolution rules will match the desired Aspect.

Aspect references are resolved using a search algorithm analogous to that used for Component references (see Section 3.3.2).

Mass and Fixed Mass Aspect Names and References

Masses and Fixed Masses can have Point and Shape Aspects, and these Aspects are referred to by their **Aspect names**. The table below shows the symbolic Aspect names used in this document and how they are constructed.

Mass and Fixed Mass Aspect Names		
Symbolic		**Constructed**
Aspect	*Point*	*Mass*`.PointID`
	Shape	*Mass*`.ShapeID`

`Mass` is the Mass or Fixed Mass name, *`PointID`* is the Point ID, and *`ShapeID`* is the Shape ID. So, for example, a shorthand reference to a Point with ID `Mount` in a Mass with ID `Engine` is `Engine.Mount`.

Link attachments default to the CG Point of the referenced Mass or Fixed Mass if the attachment Point specifier has no *`.PointID`* suffix.

Link Aspect Names and References

Links can have Static and Dynamic Aspects, and these Aspects are referred to by their **Aspect names**. The Static and Dynamic type tags act as the IDs for their Aspects. The table below shows the symbolic Aspect names used in this document and how they are constructed.

Link Aspect Names		
Symbolic		**Constructed**
Aspect	*Static*	*Link*`.Sta`
	Dynamic	*Link*`.Dyn`

`Sta` can be `Sta` or the actual Static Aspect type tag: `LE`, `EE`, `AE`, `BE`, `SE`, `AI`, `BI`, or `SI`. So, for example, shorthand references to a Static `SI` Aspect in a Link with ID `FrontFrame` are `FrontFrame.Sta` and `FrontFrame.SI`.

`Dyn` can be `Dyn` for a Link with a single Dynamic Aspect, `Dpr` for a Damper, `Mag` for a Magnifier, or the actual Dynamic Aspect type tag: `LD`, `AD`, `LM`, or `AM`. So, for example, shorthand references to a Magnifier `LM` Aspect in a Link with ID `FrontFrame` are `FrontFrame.Dyn` (if the Link has no Damper), `FrontFrame.Mag` and `FrontFrame.LM`.

Static and Dynamic Aspect type tags appearing in references are case-sensitive.

3.3.5 Aspect Set References

Aspect set references simplify the specification of sets of Aspects in Load Point member lists. An Aspect set reference is of the form:

$$[\mathit{ElementSet}\,.]\,\mathit{AspectIDSet}$$

where *ElementSet* is an element set reference (see Section 3.3.3), and where *AspectIDSet* is of the form:

$$\mathit{ID} \qquad \text{or} \qquad \{\mathit{ID}1, \mathit{ID}2, \ldots, \mathit{ID}n\} \qquad \text{or} \qquad \mathit{ID}1, \mathit{ID}2, \ldots, \mathit{ID}n$$

where each *ID* is an Aspect ID.

SISAME-3D constructs an Aspect set from a set reference according to the following rules:

- Each possible combination of *ID* s is used to generate a potential Aspect reference that is then is resolved using a search algorithm analogous to that given in Section 3.3.2 above.
- Valid Aspects of the appropriate type are added to the list. Nonexistent Aspects or Aspects of the wrong type are discarded without triggering a warning or error.
- A leading wildcard selects all contained (`*`) or member (`[*]`) Aspects of the context Component.

An example of a valid Aspect set references is: `Engine.{LeftMount,RightMount}`.

3.3.6 Parameter Names and References

SISAME-3D Parameters are referred to by their **Parameter name**. The table below shows how Parameter names are constructed from the name of their containing Event, Run, Component, or Aspect.

Parameter Names	
Symbolic	**Constructed**
Parameter	*Event.Tag*
Parameter	*Run.Tag*
Parameter	*Component.Tag*
Parameter	*Aspect.Tag*

Parameter blocks are arrays of Parameters indexed from 1 to N. Parameter blocks are referred to by their **Parameter block name**, and their sub-blocks and Parameters are referred to by **block references**, constructed from the name of the block's containing Run, Component, or Aspect as shown in the table below.

Parameter Block Names	
Symbolic	**Constructed**
Block	*Run*.*Tag*
Block	*Component*.*Tag*
Block	*Aspect*.*Tag*
Sub-Block	*Block* i:j
Sub-Block	*Block*:j
Sub-Block	*Block* i:
Parameter	*Block* i

Sub-Block references include the block Parameters indexed from i to j. If omitted, the value of i defaults to 1 and the value of j defaults to the upper index, N.

Parameter tuples are sequences of Parameters indexed by a key value. Axis keys (such as `x`), axis pair keys (such as `xy`), and axis type keys(`l` and `r`) are used. Parameter tuples are referred to by their **Parameter tuple name**, and their Parameters are referred to by key-suffixed **tuple references**, constructed as shown in the table below.

Parameter Tuple Names	
Symbolic	**Constructed**
Tuple	*Run*.*Tag*
Tuple	*Component*.*Tag*
Tuple	*Aspect*.*Tag*
Parameter	*Tuple*:`key`

Parameter, Parameter block, and Parameter tuple references are resolved by first resolving the Component or Aspect portion using the algorithms specified in Sections 3.3.2 and 3.3.4 and then identifying the specified Parameter, block, or tuple. Parameter references can refer

to Parameters in other Events. Parameter tags appearing in references are case-insensitive. The *Component*/*Aspect* portion of the reference should unambiguously specify the location of the desired Parameter: the resolution process will not look for other *Component*/*Aspect* matches to find one containing a Parameter with the specified *Tag*.

Shorthand references that use a shorthand Component or Aspect reference (see Sections 3.3.2 and 3.3.4) can be used. For example, references to a Parameter in the same Component or Aspect can omit the *Component* or *Aspect* portion. A Link (non-Aspect) Parameter can be referenced from its Static or Dynamic Aspect using a shorthand reference of the form `LinkID.Tag`.

Transformed references of the form

$$\textit{Reference} * m + c \qquad \text{or} \qquad -\textit{Reference} * m + c$$

are supported in numeric Parameters and Parameter blocks and tuples, where *Reference* can be a *Parameter*, *Block*, *Sub-Block*, or *Tuple* reference (see Section 3.4.7).

Transformed default references of the form

$$\# * m + c \qquad \text{or} \qquad -\# * m + c$$

are supported in numeric Parameters with default values, where the `#` symbol is a placeholder for the default value (see Section 3.4.7).

3.4 Specifying Parameters

Parameters are specified individually or in groupings called blocks and tuples. The rules and options for all of these specifications are presented below.

3.4.1 Parameter Specifiers

Parameters are specified with the syntax

Tag =`Specifier`

where `Specifier` is a Parameter specifier. Depending on the type of Parameter, the specifier could be a defined value, a reference to another Parameter, a reference to the default value, or an extraction specifier.

3.4.2 Parameter Block Specifiers

Parameter blocks are specified with the syntax

Tag = *Specifier* ... *Specifier*

where each *Specifier* can be a Parameter specifier or a single, block, or sub-block Parameter reference (see Section 3.3.6). The specifiers must be separated by whitespace. Multiple lines can be used but a single specifier cannot be split across lines. Some Parameter blocks have automatic values (similar to default values) that are specified by a `#` character.

Repeat counts are supported in automatic and extracted block specifiers using the specifier syntax `#`n and `?`n where n is a positive integer repeat count indicating n Parameters. An extraction specifier repeat count can be followed by an estimate and/or bound specifier (see Section 3.4.8). For example,

```
X=  0  #9
F=  0  ?9( >100 )
```

3.4.3 Parameter Tuple Specifiers

Parameter tuples are specified with the syntax

Tag =(*Specifier* ... *Specifier*)

where each *Specifier* can be a Parameter or reference specifier of the form *key*`:`*val* or a tuple reference (see Section 3.3.6). The parentheses are optional but specifiers must be separated by whitespace. Multiple lines can be used but a single specifier cannot be split across lines.

Parameter specifiers that omit the *key*`:` prefix are assigned the next available key in the canonical order for that key type. Specifiers with inactive keys will generate a warning message and are ignored. References to a tuple Parameter use the specified or next available key rather than the key of the referenced tuple Parameter. Tuple references refer to all of the referenced tuple's Parameters and maintain their keys, so tuple references cannot include a *key*`:` prefix. Tuple references must refer to tuples with the same key set.

3.4.4 Value Specifiers

A defined value specifier is simply a legal value of the appropriate type: real (floating point), integer, Boolean, or character.

Real Example: `Weight=250`

- Trailing decimal points can be omitted
- Scientific notation can be used: `SU=8.2E5`

Integer Example: `MaxIter=100`

Boolean Example: `AnySlope=True`

- True value can be any of { `True T Yes Y On 1` }
- False value can be any of { `False F No N Off 0` }

Character Example: `Class=S`

- Legal character flag values are given in the detailed specifications for each Parameter

3.4.5 Reference Specifiers

A reference specifier contains the name of another Parameter, or, in a Parameter block or tuple, the name of another Parameter block, sub-block, or tuple (see Section 3.3.6). The referenced Parameter can be a defined value, extracted, suppressed, or itself a reference to another Parameter. The type (numeric, Boolean, or character) and value/state of the referenced Parameter must be compatible with the referencing Parameter (*e.g.*, an nonextractable Parameter cannot reference an extracted Parameter). Acyclic reference chains are resolved by SISAME-3D (circular reference chains and self-references are detected as errors).

Numeric Parameter reference specifiers can include a **transform specifier** that modifies the referenced Parameter's value (see Section 3.4.7 below).

A reference to a Parameter with incompatible units will cause a warning message that can be safely ignored if the reference has a transform specifier that provides the necessary units conversion.

A reference to an extracted Parameter sets up a relationship where multiple Model Parameters share a common internal extracted Parameter. The implicit and explicit constraints on all of those Model Parameters must be satisfied by the internal Parameter. SISAME-3D may perform an initial Phase I procedure (see Section 5.8) to obtain Parameter values that satisfy all constraints. If no such values are possible SISAME-3D will terminate and report that no feasible solution exists.

3.4.6 Default Reference Specifiers

A Parameter can be specified with a reference to its own default value with the syntax *Tag* `=#` where the `#` character is a placeholder for the default value. One reason to do this is that it will cause the Parameter specification with the default value to appear in the output Event file.

Default specifiers are also used to specify a **transformed default** for numeric Parameters of the form *Tag* `=#` $*\, m + c$ (see Section 3.4.7).

3.4.7 Transform Specifiers

Transformed references are supported for numeric Parameters and Parameter blocks and tuples and are specified with the syntax

Reference * m + c or − *Reference* * m + c

where *Reference* can be a *Parameter*, *Block*, *Sub-Block*, or *Tuple* reference. For example, `FinTOut=DelTOut*1000`.

Transformed default references are supported for numeric Parameters with default values and are specified with the syntax

* m + c or − # * m + c

where # symbol is a placeholder for the default value. For example, `DelTSim=#*10`.

The transform modifies the referenced or default value by a scale factor of m, and then adds c.

- Either the * m or + c portions can be used alone.
- The unary minus forms on the right are equivalent to replacing m replaced by $-m$.
- Negative values of m or c can be used as in `*-1` or `+-100` (the + is still required before a negative c value).
- Values of m entered in scientific format should not use a + before the exponent to avoid having the exponent treated as the + c portion (*e.g.*, use `1E4` instead of `1E+4`).
- The m and c values should not be enclosed in parentheses and no whitespace should appear within the transform specifier.
- Parameters assigned zero-scaled ($m = 0$) references to extracted Parameters are treated as defined (nonextracted) Parameters with a value of c.

3.4.8 Extraction Specifiers

An extracted Parameter is specified with a question mark. The basic form is simply

Tag =? .

Estimates for and/or bounds on the extracted Parameter's value can be optionally appended in parentheses using the form

Tag =?(*Specifiers*) .

No whitespace should be left between the ? and the (but whitespace can be used to separate the estimate and bound specifiers within the parentheses.

The estimate and bound specifiers are of the form

Estimate: `~Estimate[ConfidenceBand]`

Lower Bound: `>LowerBound`

Upper Bound: `<UpperBound`

where any number of each of these (within the line length limit) can appear in any order. The confidence band for an estimate must be a positive number. The bounds are implemented as $\geq$ and $\leq$ constraints despite the use of the `>` and `<` characters.

The estimates and bounds (but not the confidence bands) can be references or transformed references to other Parameters (see Sections 3.3.6 and 3.4.7).

For example, the specifier

```
S=?( ~FrontFrame.BI.SY*2[1000] <5000 )
```

indicates a stiffness that is estimated to be twice that of the `FrontFrame` Link's yield stiffness with a confidence band of 1000, and is no more than 5000.

To extract equal-valued Parameters use direct *Tag* `=`*Reference* equality references rather than separate *Tag* `=?` extraction specifiers coupled by estimate or bound references.

A sequence of n extracted block Parameters is specified using an entry of the form *Tag* `=?` n or *Tag* `=?` n`(` *Specifiers* `)`. Block references in the *Specifiers* estimates or bounds are supported, and the referenced Parameters are assigned sequentially to each of the n extracted Parameter specifiers (the block reference must contain at least n Parameters). For example, the force block specification `F= 0 ?5( >F1:5 ) ?10` could be used to enforce nonnegative slopes on the first five of fifteen segments.

3.5 Signal Files

SISAME-3D reads files containing motion and load input time series signals and writes files containing the time series, X-Y history, and parametric Link output signals. The signal file support is described in detail below.

3.5.1 Input Signal Files

SISAME-3D reads NHTSA UDS format signal files (see below).

Acceleration, velocity, or position input motion signals can be used. Rotational position signals are Cardan roll, pitch, yaw sequence histories, not simple integrals of the angular velocity axial components.

Force, impulse, or integrated impulse input loading signals can be used. Integrated impulse signals are simple integrals of the axial impulses.

The time step of all input time series signals must be the same but it need not match the output time step, *DelTOut*. The time step should be small enough to capture the full frequency content of interest. The SimFil program [7, 9] can be used to change the time step of a time series with minimal loss of frequency content or distortion.

The time span of all input time series signals must at least extend to the final output time, *FinTOut*. Including a wider signal time span, both before time-zero and beyond *FinTOut*, is beneficial for the SISAME-3D filtered signal representation.

Input signals must have appropriate units for the Model's dimensional system (see Section 3.2.10). UDS files are converted automatically to the correct units by SISAME-3D.

3.5.2 Output Signal Files

SISAME-3D generates NHTSA UDS output signal files (see below).

The output signal time step and final time are controlled by the Run parameters *DelTOut* and *FinTOut*. All output signals begin at time zero.

Output signals are in the appropriate SISAME-3D units for the Model's dimensional system (see Section 3.2.10).

3.5.3 UDS Signal Files

UDS files are portable binary files that can represent time series and X-Y signals. UDS files are transparently converted during read to the local machine numeric floating point and byte order representation.

UDS format input signal files with recognized units are automatically converted to the appropriate standard SISAME-3D units in the Model's dimensional system when they are read. So time series signals with time in milliseconds get converted to seconds and in a Metric Model force signals in pounds get converted to newtons.

The standard file name extension for UDS files is `uds` but this is not required for UDS input signal files.

UDS file documentation and software is available from the NHTSA Research and Development Office of Crashworthiness.

3.6 Event File Overview

A SISAME-3D **Event file** is an ASCII text file that describes a Model and the conditions for an extraction or simulation Event to be computed. In a single-Event run the Event file is provided as the input to SISAME-3D. In a multiple-Event run the Master file specifies an Event file for each Event. An overall template for the Event file is presented in Section 3.6.1. Templates for the Link Static and Dynamic Parameters are presented in the Section 3.6.2. These templates are distributed with SISAME-3D to aid in creating Event files. Descriptive notes and rules on the format and constructing of Event files are presented in the Sections that follow.

3.6.1 Event File Template

```
SISAME-3D Event File

Run Information

  RunID=  Descr=  Class=
    Title=
    MaxIter=  ConPC=  ConPD=  MultPD=  Relax=  IterCon=  ConvTol=
    WtgTime=  WtgFac=
    DelTExt=  DelTSim=  DelTOut=  FinTOut=

Model Information

  ModelID=  Descr=
    DOF=  Gravity=  DimSys=  WeightMag=
    Weight=  IniP=  IniD=  IniV=
    Cutoff=  ZeroSm=  EndSm=
    ConV=  ConP=  ConSS=  ConI=  ConJ=

    FixedMassID=  Descr=
      IniP=  Color=  Transp=
      PointID=  RelP=  Size=  Color=  Transp=
      BoxID=  RelP=  Size=  Color=  Transp=
      CylinderID=  RelP=  Size=  Color=  Transp=
      ConeID=  RelP=  Size=  Color=  Transp=
      SphereID=  RelP=  Size=  Color=  Transp=
      EllipsoidID=  RelP=  Size=  Color=  Transp=
      PlaneID=  RelP=  Size=  Color=  Transp=

  VehicleID=  Descr=  Make=  Model=  Year=
    Weight=  IniP=  IniD=  IniV=
    Cutoff=  ZeroSm=  EndSm=
    ConV=  ConP=  ConSS=  ConI=  ConJ=

  SubID=  Descr=
    Weight=  IniP=  IniD=  IniV=
    Cutoff=  ZeroSm=  EndSm=
    ConV=  ConP=  ConSS=  ConI=  ConJ=

    MassID=  Descr=  Class=
      Weight=  Inertia=  IniP=  IniD=  IniV=  Color=  Transp=
      Cutoff=  ZeroSm=  EndSm=
      ConA=  ConV=  ConP=
      TarSignal=  DrvSignal=  CmpSignal=
      PointID=  RelP=  Size=  Color=  Transp=
      BoxID=  RelP=  Size=  Color=  Transp=
      CylinderID=  RelP=  Size=  Color=  Transp=
      ConeID=  RelP=  Size=  Color=  Transp=
      SphereID=  RelP=  Size=  Color=  Transp=
      EllipsoidID=  RelP=  Size=  Color=  Transp=
      PlaneID=  RelP=  Size=  Color=  Transp=
```

```
    LinkID=  Descr=  Class=
      Point1=  Point2=
      Ref=  XScl=  FScl=  RScl=  DScl=  MScl=
      Sta=
      Dyn=

    LoadID=  Descr=  Class=
      Mass=  Link=  Point=
      Cutoff=  ZeroSm=  EndSm=
      ConF=  ConI=  ConJ=
      TarSignal=  DrvSignal=  CmpSignal=

Output Information

  OutClass=MassTbl  Sub=        Mass=
  OutClass=LinkTbl  Sub=        Link=
  OutClass=LoadTbl  Sub=        Load=
  OutClass=MassTS   Qty=AVPavp  Mass=
  OutClass=LinkTS   Qty=XRSDFE  Link=
  OutClass=LoadTS   Qty=FIJfij  Load=
  OutClass=LinkXY   Qty=SDFE    Link=
  OutClass=LinkPar  Qty=SD      Link=
  OutClass=FitRep   File=
  OutClass=Event    File=
  OutClass=POV      File=

Comments
```

3.6.2 Link Parameter Templates

Static Parameters

```
Sta=LE  S=

Sta=EE  SO=  C=  P=  XMag=

Sta=AE  SO=  FMax=

Sta=BE  SO=  XC=  FC=  SC=  XT=  FT=  ST=

Sta=SE  XiScl=  FiScl=  Symmetric=  ConSS=  AnySlope=
  X=
  F=

Sta=AI  SO=  FMax=

Sta=BI  SO=  FYC=  FYT=  SY=

Sta=SI  SU=  ST=  XSlk=  XiScl=  FiScl=  Symmetric=  ConSS=  AnySlope=
  X=
  F=
```

Universal Static Parameters

```
  OneWay=  XScl=  FScl=  XDel=  FDel=  XFailMin=  XFailMax=
```

Dynamic Parameters

```
Dyn=LD  DSlp=  DSlk=

Dyn=AD  DSlp=  DMax=  DSlk=

Dyn=LM  MSlp=

Dyn=AM  MSlp=  MMax=
```

Universal Dynamic Parameters

```
  OneWay=  XScl=  RScl=  DScl=  MScl=
```

3.6.3 Event File Organization

- The Event file header `SISAME-3D Event File` and the section headings `Run Information`, `Model Information`, and `Output Information` are required and must appear on separate lines in that order.
- The `Run Information` section contains specifications that describe and control the SISAME-3D run. The Run specifications section begins with the `RunID` field.
- The `Model Information` section specifies the Structures and the Mass, Fixed Mass, Link, and Load Elements that make up the Model. The Model specifications section begins with the `ModelID` field and is followed by the Model Component specification sections.
- The `Output Information` section specifies the desired outputs.
- The tags `RunID`, `ModelID`, `VehicleID`, `SubID`, `MassID`, `FixedMassID`, `LinkID`, and `LoadID` are **leading tags** that begin specification sections of the Event file and must be the first tag on a line. The other fields in each section can appear in any order and on as many lines as needed.
- Elements specified above all of the Vehicle or Substructure specifications are members of the Model Structure.
- A **Vehicle** contains all of the Elements and Substructures following its specifications until the next Vehicle specifications, if any.
- A **Substructure** contains all of the Elements following its specifications until the next Substructure or Vehicle specifications, if any.
- Components within a Structure can appear in any order.
- Blank lines can be used to separate portions of the Event file.
- An exclamation point (`!`) character (outside of a quoted string) causes the rest of the line to be treated as a comment.
- `Comment=`*`comment`* fields can be included in any section of the Event file.
- Everything below the optional `Comments` heading below the `Output Information` section is ignored.

3.6.4 Event File Fields

- An Event file contains **field** specifications of the form *`Tag`*`=`*`Value`* where *`Tag`* is the tag string that identifies the field and *`Value`* is the specified value or values for that field.
- The numeric, Boolean, and character flag specification fields are called **Parameters**. Some Parameters are specified in aggregate groups called Parameter **blocks** and **tuples**. Section 3.4 covers the specification of Parameters in detail.
- Fields and Parameter block and tuple specifiers must be separated by **whitespace** (one or more space or tab characters).
- Individual field specifiers cannot be broken over multiple lines but Parameter blocks and tuples can use multiple lines.

- **String** field values that contain `=` or `!` characters must be enclosed within double quote characters (`" "`). Any string value can be enclosed in double quotes if desired.
- A field is considered **unspecified** if it is not present in the Event file or if *Tag* `=` appears but is not followed by a specifier.
- If a field is **repeated** in the same context the last specifier is used.

3.6.5 Input Signal File Names

- Input signal files with relative path names are taken to be relative to the SISAME-3D Event file location, allowing SISAME-3D to be run from a directory other than that containing the Event file.
- Input signal file name paths are automatically translated to the local operating system format, where possible, to allow the same SISAME-3D Event file to be used on different computer platforms. The use of relative, rather than absolute, paths facilitates this portability.

3.6.6 Specification Tables

The specifications for each part of the SISAME-3D Event file are described in the following sections. The specifications are presented in tables having the columns and usage described below.

Tag	Description	Value	Default	N	?

Tag The *Tag* for the field that is used with the syntax *Tag* `=` *Value* to assign a value to the field. Tags are shown with this typeface: `Tag`.

Description A description of the field.

Value Indicates the type and allowable values and/or range for the field. The value type is one of:

$\mathcal{R}$ Real (floating point). Range, if any, is shown, for example $\mathcal{R} \geq 0$.
Floating point (123.456) or scientific (1.23456E2) formats are accepted.
The internal precision is approximately 15 decimal digits.

$[\mathcal{R}]$ Real (floating point) block. Range, if any, is shown, for example $[\mathcal{R} \geq 0]$.

$(\mathcal{R})$ Real (floating point) tuple. Range, if any, is shown, for example $(\mathcal{R} \geq 0)$.

$\mathcal{I}$ Integer. Range, if any, is shown, for example $\mathcal{I} \geq 0$.

$\mathcal{B}$ Boolean (true/false). Accepts any of { `True T Yes Y On 1` } for "True" and any of { `False F No N Off 0` } for "False". Case-insensitive.

$\mathcal{C}$ Character. Accepts single character values from the given list, for example $\mathcal{C}$ = `A`, `B`.

$\mathcal{S}$ Character string. A maximum length of L characters is indicated as $\mathcal{S} \leq L$. A list of accepted strings may be shown, for example $\mathcal{S}$ = `StringA`, `StringB`.

Numeric Parameter blocks marked with the $\nearrow$ symbol must have monotonically increasing values.

The value of a field with tag `Tag` is indicated by *`Tag`*.

Default The default value, if any, or possibly one of the entries:

computed The default value is computed from other Model specifications, as described in the notes for that field.

hierarchy The first specified value from a hierarchy of values is the default value, as described in the notes for that field.

none By default the field is inactive and is assigned no value (unlike a field with a blank entry, which has no default value).

The default value is used if the Parameter value is not specified or implied.

All Parameters with a default entry can be specified using the syntax *Tag* = # to cause the default value to be used and to appear in the output Event file.

Numeric Parameters with a default entry can be specified using the **transformed default** syntax *Tag* = # $* m + c$ (see Section 3.4.7).

N An `N` in this column indicates that the Parameter is **suppressible**. Suppressed Parameters are specified by using a value of `N`. A suppressed Parameter is made inactive, as described in the notes for that Parameter.

? A `?` in this column indicates that the Parameter is **extractable**. Extracted Parameters are specified by using a value of or starting with `?` (see Section 3.4.8) or a reference to an extracted Parameter (see Section 3.3.6).

3.7 Run Information

The `Run Information` section is the first section of the SISAME-3D Event file and contains the specifications that describe and control the SISAME-3D run for the Event. In a multiple-Event run the Master Run extraction control Parameters control the solution and the corresponding Parameters in the Event file are ignored.

Run Specifications

Tag	Description	Value	Default	N	?
`RunID`	Run ID	$\mathcal{S} \leq 20$			
`Descr`	Description	$\mathcal{S}$			
`Class`	Run class	$\mathcal{C} \in$ `S`, `E`, `T`	*see Notes*		
`Title`	Run title	$\mathcal{S}$			
`MaxIter`	Max number of extraction iterations	$\mathcal{I} \geq 0$	50	`N`	
`ConPC`	Parameter conditioning confidence band factor	$\mathcal{R} > 0$	1	`N`	
`ConPD`	Parameter damping confidence band factor	$\mathcal{R} > 0$	1	`N`	
`MultPD`	Parameter damping multiplier	$\mathcal{R} > 0$	0.5	`N`	
`Relax`	Iteration relaxation control	$\mathcal{B}$	`True`		
`IterCon`	Iteration constraint control	$\mathcal{B}$	`True`		
`ConvTol`	Convergence tolerance control factor	$\mathcal{R} > 0$	1		
`WtgTime`	Time-based weighting times	$[\mathcal{R} \geq 0 \ \nearrow]$			
`WtgFac`	Time-based weighting factors	$[\mathcal{R}]$			
`DelTExt`	Extraction simulation time step	$\mathcal{R} > 0$	*computed*		
`DelTSim`	Simulation time step	$\mathcal{R} > 0$	*computed*		
`DelTOut`	Output time step	$\mathcal{R} > 0$			
`FinTOut`	Final output time	$\mathcal{R} \geq 0$			

Run Specifications Notes

- `RunID` is the leading tag for the Run specifications. The *`RunID`* appears in outputs to identify the run.
- The Run *`Class`* defaults to `S` for a simulation run and `E` for an extraction run (when extracted Parameters are present). Setting `Class=T` requests a test run, in which no simulation or extraction is performed, but in which the Event file will be processed and the model summary, model file, and parametric Link outputs can be examined. The model file from a Test run contains diagnostic information such as expanded references.

- *MaxIter* is the maximum number of extraction iterations SISAME-3D can use to converge on a self-consistent solution when Model features require iteration. `MaxIter=0` produces a single pass extraction. `MaxIter=N` suppresses all extraction passes and causes run outputs to use the initial values generated by SISAME-3D for the extracted Parameters.
- *ConPC* is the confidence band factor for the Parameter conditioning target equations. These confidence bands are of the form *ConPC* $\cdot 10^5 \cdot \bar{p}$ where $\bar{p}$ is a baseline magnitude for the type and units of the specific Parameter. A smaller *ConPC* will generally improve numerical conditioning of the extraction computations, possibly at some cost to the solution quality. Setting `ConPC=N` suppresses Parameter conditioning. If the Model has extracted Parameters that are never active during the event, or other singular or ill-conditioned features, Parameter conditioning is required to avoid a singular or ill-conditioned least squares system.
- *ConPD* is a confidence band factor for the Parameter damping target equations, which target iterative solutions to the previous solution to damp large Parameter changes and speed convergence. Parameter damping begins after (*MaxIter*/2) iterations or when solution cycling is detected. These confidence bands are of the form *ConPD* $\cdot 10^3 \cdot \bar{p}$ where $\bar{p}$ is a baseline magnitude for the type and units of the specific Parameter. A smaller *ConPD* will facilitate convergence of difficult iterative extractions. Setting `ConPD=N` suppresses Parameter damping.
- *MultPD* is a multiplier applied cumulatively to *ConPD* on each iteration pass to speed solution convergence (with *MultPD* ≤ 1). A smaller *MultPD* will facilitate convergence of difficult iterative extractions. Setting `MultPD=N` suppresses changes to *ConPD*.
- *Relax* controls whether relaxation is enabled during extraction iterations. Relaxation limits the change in the Parameter solutions between iterations in an effort to speed convergence. By default *Relax* is `True` and SISAME-3D will begin and gradually increase the amount of relaxation after (*MaxIter*/2) iterations. No relaxation occurs if *MaxIter* < 3. Relaxation is also used if solution cycling is detected regardless of *Relax*.
- *IterCon* controls whether iteration constraints are enabled during extraction. Iteration constraints limit the solution change in each iteration to control the error of the linearized approximants used to represent nonlinear Parameter contributions. Iteration constraints prevent solution overshooting/oscillation and may be needed to assure convergence, and thus should generally be enabled. By default *IterCon* is `True` enabling the iteration constraints.
- *ConvTol* is a factor that controls the extraction iteration convergence stopping criteria tolerances. A smaller *ConvTol* causes tighter convergence criteria to be used.
- Time-based weighting can be used to emphasize portions of the event by specifying an equal number of *WtgTime* and *WtgFac* Parameter block values. An unweighted solution is equivalent to a constant weighting value of 1. The *WtgFac* are linearly interpolated and the first and last *WtgFac* are extrapolated as constants out to the time span tails to determine the weight at each time step.
- *DelTExt* is the extraction simulation time step. *DelTExt* has no effect in simulation runs. *DelTExt* can be greater than the output time step. The default *DelTExt* is based on an analysis of the Model and should provide sufficient precision for most applications. The default *DelTExt* is set after the Model is processed so a default-based *DelTExt* cannot be the target of a Parameter reference.
- *DelTSim* is the simulation time step. *DelTSim* has no effect in extraction passes. *DelTSim* is

adjusted, if necessary, to evenly divide the output time step, *DelTOut*. The default *DelTSim* is based on an analysis of the Model and should provide sufficient precision for most applications. The default *DelTSim* is set after the Model is processed so a default-based *DelTSim* cannot be the target of a Parameter reference.

- Run Parameters can omit the *RunID*. prefix when referring to other Run Parameters.

3.8 Model Structure

The `Model Information` section contains the Model specifications followed by the specifications for the Vehicle and Substructure Structures and the Mass, Link, and Load Elements that make up the model. The Model contains all the Components of the model. Each Vehicle contains all the Components following its specifications up to the next Vehicle, if any. Each Substructure contains all the Components following its specifications up to the next Vehicle or Substructure, if any. Elements can appear in any order within a Structure.

The Model specifications apply to the Model as a whole and all of its Components.

Model Specifications

Tag	Description	Value	Default	N	?
`ModelID`	Model ID	$\mathcal{S} \leq 20$			
`Descr`	Description	$\mathcal{S}$			
`DOF`	Degrees of Freedom	$(\mathcal{C}) \subseteq$ (`xyzrpw`)			
`Gravity`	Gravity direction	$\mathcal{S} \in \pm$`z`, $\pm$`x`, $\pm$`y`	*none*		
`DimSys`	Dimensional system	$\mathcal{S} \in$ `SI`, `Metric`, `English`			
`WeightMag`	Weight magnitude (in mass units)	$\mathcal{R} > 0$	$\overline{m}$		
`Weight`	Model weight (in mass units)	$\mathcal{R} > 0$	*computed*		?
`IniP`	Resting position relative to the inertial frame	$(\mathcal{R})$	$(\mathbf{0})$		
`IniD`	Initial displacement from the resting position	$(\mathcal{R})$	$(\mathbf{0})$		
`IniV`	Initial velocity	$(\mathcal{R})$	$(\mathbf{0})$		
`Cutoff`	Default cutoff frequency	$\mathcal{R} \geq 0$			
`ZeroSm`	Default time-zero tail smoothing frequency	$\mathcal{R} > 0$		N	
`EndSm`	Default final time tail smoothing frequency	$\mathcal{R} > 0$		N	
`ConV`	Default velocity confidence band factor	$\mathcal{R} > 0$	1	N	
`ConP`	Default position confidence band factor	$\mathcal{R} > 0$	1	N	
`ConSS`	Default smoothness confidence band factor	$\mathcal{R} > 0$	1	N	
`ConI`	Default impulse confidence band factor	$\mathcal{R} > 0$	1	N	
`ConJ`	Default integrated impulse confidence band factor	$\mathcal{R} > 0$	1	N	

Model Specification Notes

- `ModelID` is the leading tag for the Model specifications.
- *`DOF`* is a character tuple that specifies the Model degrees of freedom. The tuple can be any valid subset of the characters: `xyzrpw`. To be valid a *`DOF`* must include at least one axis and if either linear axis in the plane of a rotational axis is present they must both be. Pure translational and pure rotational *`DOF`* are supported. For example, a 2D x-y planar Model would have `DOF=(xyw)` and a 3D model with all six degrees of freedom would have `DOF=(xyzrpw)`. Every Component in a Model shares the common *`DOF`*.
- *`Gravity`* is the optional gravitational force direction. The specified *`Gravity`* axis must be one of the linear *`DOF`* axes and a sign prefix of `+` or `-` is required. For example, gravity in the negative z axis direction is specified with `Gravity=-z`. The gravitational force is applied in the corresponding inertial frame axis direction to all simulated Masses and the gravitational force contribution is applied to all instrumented Masses with defined or extracted weight.
- *`DimSys`* sets the dimensional system (see Section 3.2.10). *`DimSys`* can be abbreviated to as little as one character.
- *`WeightMag`* is the weight magnitude value (in mass units), $\tilde{m}$, that is converted to $\tilde{w}$ (in force units) for some computations. The *`WeightMag`* is used if there are any instrumented or driven Masses, target Loads, comparison signals, or extracted Parameters in the Model. *`WeightMag`* defaults to the average defined Mass weight in the Model or to 50 kg if there are no defined-weight Masses.
- *`Weight`* is the total Model weight.
  - If the Model does not contain extracted-weight Masses, a specified Model *`Weight`* must equal the weight of its contained Masses and Structures. Specifying the *`Weight`* is optional but provides a check on the weights. Specifying `Weight=#` causes the computed Model weight to appear in the output Event file (transformed defaults are not supported).
  - If the Model contains extracted-weight Masses, specifying a Model *`Weight`* adds a Model weight constraint to the extraction (the specified weight must be at least the defined weight of its Masses and Structures). If the Model *`Weight`* is unspecified or if `Weight=?` or `Weight=#` is specified no total weight constraint is generated and the Model weight is an implicitly extracted value (which cannot be referenced). Specifying `Weight=?` or `Weight=#` causes the extracted Model weight to appear in the output Event file.
- *`IniP`*, *`IniD`*, and *`IniV`* are axis-keyed Parameter tuples that specify the resting and initial generalized position and velocity of the Model relative to the inertial (ground) coordinate frame. The Model resting and initial frames need not coincide with the inertial frame.
- *`Cutoff`*, *`ZeroSm`*, and *`EndSm`* are used in the Mass and Load defaulting hierarchies.
- *`ConV`* and *`ConP`* are used in the Mass defaulting hierarchies.
- *`ConSS`* is used in the Link defaulting hierarchy.
- *`ConI`* and *`ConJ`* are used in the Load defaulting hierarchies.

3.9 Vehicle Structure

A Vehicle includes and its specifications apply to all Substructures, Masses, Links, and Loads that follow it until the next Vehicle or the end of the `Model Information` section.

Vehicle Specifications

Tag	Description	Value	Default	N	?
`VehicleID`	Vehicle ID	$\mathcal{S} \leq 20$			
`Descr`	Description	$\mathcal{S}$			
`Make`	Make of Vehicle	$\mathcal{S}$			
`Model`	Model of Vehicle	$\mathcal{S}$			
`Year`	Model year of Vehicle	$\mathcal{I}$			
`Weight`	Vehicle weight (in mass units)	$\mathcal{R} > 0$	*computed*		?
`IniP`	Resting position relative to the Model frame	$(\mathcal{R})$	$(\mathbf{0})$		
`IniD`	Initial displacement from the resting position	$(\mathcal{R})$	$(\mathbf{0})$		
`IniV`	Initial velocity	$(\mathcal{R})$	$(\mathbf{0})$		
`Cutoff`	Default cutoff frequency	$\mathcal{R} \geq 0$	*hierarchy*		
`ZeroSm`	Default time-zero tail smoothing frequency	$\mathcal{R} > 0$	*hierarchy*	N	
`EndSm`	Default final time tail smoothing frequency	$\mathcal{R} > 0$	*hierarchy*	N	
`ConV`	Default velocity confidence band factor	$\mathcal{R} > 0$	*hierarchy*	N	
`ConP`	Default position confidence band factor	$\mathcal{R} > 0$	*hierarchy*	N	
`ConSS`	Default smoothness confidence band factor	$\mathcal{R} > 0$	*hierarchy*	N	
`ConI`	Default impulse confidence band factor	$\mathcal{R} > 0$	*hierarchy*	N	
`ConJ`	Default integrated impulse confidence band factor	$\mathcal{R} > 0$	*hierarchy*	N	

Vehicle Specification Notes

- `VehicleID` is the leading tag for the Vehicle specifications.
- ***`Weight`*** is the total Vehicle weight.
  - If the Vehicle does not contain extracted-weight Masses, a specified Vehicle ***`Weight`*** must equal the weight of its contained Masses and Substructures. Specifying the ***`Weight`*** is optional but provides a check on the weights. Specifying `Weight=#` causes the computed Vehicle weight to appear in the output Event file (transformed defaults are not supported).
  - If the Vehicle contains extracted-weight Masses, specifying a Vehicle ***`Weight`*** adds a Vehicle weight constraint to the extraction (the specified weight must be at least the defined weight of its Masses and Substructures). If the Vehicle ***`Weight`*** is unspecified or if `Weight=?` or

`Weight=#` is specified no total weight constraint is generated and the Vehicle weight is an implicitly extracted value (which cannot be referenced). Specifying `Weight=?` or `Weight=#` causes the extracted Vehicle weight to appear in the output Event file.

- *`IniP`*, *`IniD`*, and *`IniV`* are axis-keyed Parameter tuples that specify the resting and initial generalized position and velocity of the Vehicle relative to the Model coordinate frame.
- *`Cutoff`*, *`ZeroSm`*, and *`EndSm`* are used in the Mass and Load defaulting hierarchies. They default to the Model's corresponding Parameters.
- *`ConV`* and *`ConP`* are used in the Mass defaulting hierarchies. They default to the Model's corresponding Parameter and then to 1.
- *`ConSS`* is used in the Link defaulting hierarchy. It defaults to the Model's *`ConSS`* and then to 1.
- *`ConI`* and *`ConJ`* are used in the Load defaulting hierarchies. They default to the Model's corresponding Parameter and then to 1.

3.10 Substructure

A Substructure includes and its specifications apply to all Masses, Links, and Loads that follow it until the next Substructure or Vehicle or the end of the `Model Information` section.

Substructure Specifications

Tag	Description	Value	Default	N	?
`SubID`	Substructure ID	$\mathcal{S} \leq 20$			
`Descr`	Description	$\mathcal{S}$			
`Weight`	Substructure weight (in mass units)	$\mathcal{R} > 0$	*computed*		?
`IniP`	Resting position relative to the Structure frame	$(\mathcal{R})$	$(\mathbf{0})$		
`IniD`	Initial displacement from the resting position	$(\mathcal{R})$	$(\mathbf{0})$		
`IniV`	Initial velocity	$(\mathcal{R})$	$(\mathbf{0})$		
`Cutoff`	Default cutoff frequency	$\mathcal{R} \geq 0$	*hierarchy*		
`ZeroSm`	Default time-zero tail smoothing frequency	$\mathcal{R} > 0$	*hierarchy*	N	
`EndSm`	Default final time tail smoothing frequency	$\mathcal{R} > 0$	*hierarchy*	N	
`ConV`	Default velocity confidence band factor	$\mathcal{R} > 0$	*hierarchy*	N	
`ConP`	Default position confidence band factor	$\mathcal{R} > 0$	*hierarchy*	N	
`ConSS`	Default smoothness confidence band factor	$\mathcal{R} > 0$	*hierarchy*	N	
`ConI`	Default impulse confidence band factor	$\mathcal{R} > 0$	*hierarchy*	N	
`ConJ`	Default integrated impulse confidence band factor	$\mathcal{R} > 0$	*hierarchy*	N	

Substructure Specification Notes

- `SubID` is the leading tag for the Substructure specifications.
- *`Weight`* is the total Substructure weight.
  - If the Substructure does not contain extracted-weight Masses, a specified Substructure *`Weight`* must equal the weight of its contained Masses. Specifying the *`Weight`* is optional but provides a check on the Mass weights. Specifying `Weight=#` causes the computed Substructure weight to appear in the output Event file (transformed defaults are not supported).
  - If the Substructure contains extracted-weight Masses, specifying a Substructure *`Weight`* adds a Substructure weight constraint to the extraction (the specified weight must be at least the defined weight of its Masses). If the Substructure *`Weight`* is unspecified or if `Weight=?` is specified no total weight constraint is generated and the Substructure weight is an implicitly extracted value (which cannot be referenced). Specifying `Weight=?` or `Weight=#` causes the extracted Substructure weight to appear in the output Event file.

- *IniP*, *IniD*, and *IniV* are axis-keyed Parameter tuples that specify the resting and initial generalized position and velocity of the Substructure relative to its enclosing Structure's coordinate frame.
- *Cutoff*, *ZeroSm*, and *EndSm* are used in the Mass and Load defaulting hierarchies. They default to the enclosing Structure's corresponding Parameters.
- *ConV* and *ConP* are used in the Mass defaulting hierarchies. They default to the enclosing Structure's corresponding Parameter and then to 1.
- *ConSS* is used in the Link defaulting hierarchy. It defaults to the enclosing Structure's *ConSS* and then to 1.
- *ConI* and *ConJ* are used in the Load defaulting hierarchies. They default to the enclosing Structure's corresponding Parameter and then to 1.

3.11 Mass Element

A Mass is a rigid body that moves within its Structure and the inertial (ground) reference frame. Link and Contact force interactions can occur with a Mass.

Mass Specifications

Tag	Description	Value	Default	N	?
`MassID`	Mass ID	$\mathcal{S} \leq 20$			
`Descr`	Description	$\mathcal{S}$			
`Class`	Class	$\mathcal{C} \in$ `S`, `C`, `T`, `D`, `d`	*see Notes*		
`Weight`	Weight (in mass units)	$\mathcal{R} > 0$	*none*		?
`Inertia`	Inertia tensor	$(\mathcal{R})$	*none*		
`IniP`	Resting position relative to the Structure frame	$(\mathcal{R})$	$(\mathbf{0})$		
`IniD`	Initial displacement from the resting position	$(\mathcal{R})$	$(\mathbf{0})$		
`IniV`	Initial velocity	$(\mathcal{R})$	$(\mathbf{0})$		
`Color`	Default Shape color	$\mathcal{S}$			
`Transp`	Default Shape transparency	$\mathcal{R} \in [0, 1]$	0		
`Cutoff`	Cutoff frequency	$\mathcal{R} \geq 0$	*hierarchy*		
`ZeroSm`	Time-zero tail smoothing frequency	$\mathcal{R} > 0$	*hierarchy*	`N`	
`EndSm`	Final time tail smoothing frequency	$\mathcal{R} > 0$	*hierarchy*	`N`	
`ConA`	Acceleration confidence band	$(\mathcal{R}) > 0$	*see Notes*		
`ConV`	Velocity confidence band factor	$\mathcal{R} > 0$	*hierarchy*	`N`	
`ConP`	Position confidence band factor	$\mathcal{R} > 0$	*hierarchy*	`N`	
`TarSignal`	Target motion signal specifiers	$(\mathcal{S})$	*none*		
`DrvSignal`	Driving motion signal specifiers	$(\mathcal{S})$	*none*		
`CmpSignal`	Comparison motion signal specifiers	$(\mathcal{S})$	*none*		
`PointID`	Point ID	$\mathcal{S} \leq 20$			
`Shape``ID`	Shape ID	$\mathcal{S} \leq 20$			

Mass Specification Notes

- `MassID` is the leading tag for the Mass specifications.
- The Mass *`Class`* specifies the type of Mass.
  - A **simulated Mass** can be specified by `Class=S`. Simulated Masses must have the appropriate *`Weight`* and *`Inertia`* values defined. Simulated Masses can have comparison motion signals

(specified with *CmpSignal*) that are used to generate fit measures but have no effect on the Model's behavior.

- A **constant-velocity (c-v) Mass** can be specified by `Class=C`. C-v Masses may have defined *Weight* or *Inertia*.
- A **target Mass** can be specified by `Class=T`. Signals defining the target Mass motion must be specified in *TarSignal*. Extraction target equations are generated for a target Mass.
- A **driven Mass** can be specified by `Class=D` or `Class=d` (driven Masses with `Class=d` become simulated Masses in the output model). Signals defining the driven Mass motion must be specified in *DrvSignal*. No extraction target equations are generated for a driven Mass.
- If no *TarSignal* or *DrvSignal* is specified the *Class* defaults to `S` (simulated) if the appropriate *Weight* and *Inertia* are specified and to `C` (c-v) if no *Weight* or *Inertia* is specified.
- If *TarSignal* is specified, the *Class* defaults to `T` (target) if the Mass' motion target equations would contain extracted Parameters and to `d` (driven) otherwise.
- If *DrvSignal* is specified, the *Class* defaults to `D` (driven).

- A *Weight* value must be defined for simulated Masses when any linear degrees of freedom are present in the Model. The *Weight* must be defined or extracted for instrumented Masses when any linear degrees of freedom are present in the Model. *Weight* is optional for driven and constant-velocity Masses.
- `Weight=?` specifies an extracted Mass weight value. Extracted weight Masses are usually included in target Loads to provide SISAME-3D the clearest information about the weight. Alternately, they can appear implicitly by having extracted Links to which they connect in target Loads, although this additional coupling of extracted Parameters can lead to a significantly different Model. SISAME-3D can return a zero extracted weight if that provides the optimal fit to the test data although that is not a legal weight for simulation. Estimates and/or bounds may be needed to supplement the test data and refine such weight extractions.
- An *Inertia* tensor Parameter tuple must be specified for simulated Masses when any rotational degrees of freedom are present in the Model. The *Inertia* must be defined or extracted for instrumented Masses when any rotational degrees of freedom are present in the Model. *Inertia* is optional for driven and constant-velocity Masses. *Inertia* is axis-pair keyed tuple, with the canonical key order: `xx`, `yy`, `zz`, `xy`, `yz`, `zx` (the axis order in the keys is not significant).

  The inertia tensor is represented by the symmetric matrix

$$\mathbf{I} = \begin{bmatrix} i_{xx} & i_{xy} & i_{xz} \\ i_{yx} & i_{yy} & i_{yz} \\ i_{zx} & i_{zy} & i_{zz} \end{bmatrix}$$

  where the diagonal elements are the moments of inertia defined as

$$i_{xx} = \int_V (y^2 + z^2)\,\rho\,dV \qquad i_{yy} = \int_V (z^2 + x^2)\,\rho\,dV \qquad i_{zz} = \int_V (x^2 + y^2)\,\rho\,dV$$

  and the off-diagonal elements are the negated products of inertia defined as

$$i_{xy} = i_{yx} = -\int_V x\,y\,\rho\,dV \qquad i_{yz} = i_{zy} = -\int_V y\,z\,\rho\,dV \qquad i_{zx} = i_{xz} = -\int_V z\,x\,\rho\,dV$$

where x, y, and z are the CG-frame-relative coordinates of points on the Mass, V is the volume of the Mass, and ρ is the mass density of the Mass.

The *Inertia* tuple Parameters are the elements of the inertia tensor, so *Inertia:xy*$= i_{xy}$ and so forth. The diagonal *Inertia* values must be positive and consistent with realizable moments. The off-diagonal Parameters default to zero. Inertia values for inactive axis pairs (with no rotation in the corresponding plane) can be specified but are ignored.

- *IniP*, *IniD*, and *IniV* are axis-keyed Parameter tuples that specify the resting and initial generalized position and velocity of the Mass relative to its enclosing Structure's coordinate frame.
  - *IniP* is the generalized resting position of the Mass CG frame relative to the enclosing Structure's coordinate frame. When all Structures and Masses are in their resting positions the Link deflections are, by definition, zero.
  - *IniD* is the generalized initial displacement of the Mass CG frame relative to its resting position.
  - *IniV* is the generalized initial velocity of the Mass CG frame relative to the coincident point in the enclosing Structure's coordinate frame.
- *Color* specifies the default color for any Shapes in the Mass (see Section 3.14). The *Color* can be any POV-Ray color specifier such as `MediumBlue` or `rgb<1,.5,.5>`.
- *Transp* specifies the default transparency for any Shapes and Points in the Mass (see Section 3.14). *Transp* defaults to zero (opaque) and can be any value from zero to one.
- *Cutoff*, *ZeroSm*, and *EndSm* control the input motion signal filtering. See the SimFil documentation [7, 9] for more information on the filtered signal representation used in SISAME-3D.
  - *Cutoff* is the low-pass filtering cutoff frequency. *Cutoff* defaults to the enclosing Structure's *Cutoff*.
  - *ZeroSm* and *EndSm* are the acceleration time-zero and end tail smoothing frequencies. *ZeroSm* and *EndSm* default to the enclosing Structure's corresponding Parameters and then to 2.5∗*Cutoff*. Tail smoothing of either tail can be suppressed by specifying a value of `N`.
- *ConA* is an RMS confidence band magnitude tuple for the acceleration of target and comparison Masses, and of driven Masses with defined weights. *ConA* is keyed by axis type, `l` for linear and `r` for rotational. *ConA:l* defaults to $10\,\mathrm{g}\sqrt{\tilde{w}/w_i}$ (10 g for unspecified and extracted-weight Masses), where w_i is the weight of the Mass and $\tilde{w}$ is the weight magnitude. *ConA:r* defaults to $100\ \mathrm{radians/s^2}\sqrt{\tilde{I}/I_i}$, where I_i is a representative inertia magnitude for the Mass and $\tilde{I}$ is a representative inertia magnitude for the Model. These scaled geometric means are designed to balance the solution weighting among low and and high weight/inertia Masses.
- *ConV* and *ConP* are confidence band factors for the velocities and positions of target and comparison Masses. The weighting of these target equations is based on *ConA*. *ConV* defaults to the enclosing Structure's *ConV* and then to 1. *ConP* defaults to the enclosing Structure's *ConP* and then to 1. *ConV* and/or *ConP* can be set to `N` to suppress velocity and/or position domain targeting (this will usually speed up the extraction). Decreasing *ConV* and/or *ConP* will generally improve the extracted model's velocity and/or position fit but at some cost to the acceleration fit.

- *`TarSignal`*, *`DrvSignal`*, and *`CmpSignal`* specify the target, driving, or comparison input motion signals.
  - Only one type of signal specifier can be used in a Mass: target, driving, or comparison. The signal type must be consistent with any specified Mass Class.
  - Signal specifiers have the form *axis*`:`*file_name*.
  - Multiple signals can be specified following each `Signal` tag or each signal can be specified in a separately tagged field.
  - Motion signals must be in the inertial reference frame and positions taken with respect to the inertial frame origin.
  - Motion signals for all DOF axes must be specified if any are.
  - Acceleration, velocity, and position time series signals are supported. Position and velocity signals must have time-zero values consistent with the Mass initial conditions to avoid signal distortion.
  - Rotational position domain signals must be Cardan body-fixed roll, pitch, yaw sequence histories (the type output by SISAME-3D). If rotational position signals are used for any axis they must be used for all rotational axes in that Mass.
- *`PointID`* is the leading tag for the specifications of a Mass Point. Any number of Points can be specified for each Mass following the general Mass specifications. (See Section 3.13.)
- Specifiers of the form *`Shape`*`ID` are leading tags for the specifications of a Mass Shape. *`Shape`* is one of `Box`, `Cylinder`, `Cone`, `Sphere`, `Ellipsoid`, or `Plane`. Any number of Shapes can be specified for each Mass following the general Mass specifications. (See Section 3.14.)

3.12 Fixed Mass Element

A Fixed Mass is a rigid body fixed to the inertial (ground) reference frame. Link and Contact force interactions can occur with a Fixed Mass. A Fixed Mass is similar to a c-v Mass with zero absolute velocity but its position is not affected by the *IniD* initial displacements of containing Structures.

Fixed Mass Specifications

Tag	Description	Value	Default	N	?
`FixedMassID`	Fixed Mass ID	$\mathcal{S} \leq 20$			
`Descr`	Description	$\mathcal{S}$			
`IniP`	Position relative to the Structure frame	$(\mathcal{R})$	$(\mathbf{0})$		
`Color`	Default Shape color	$\mathcal{S}$			
`Transp`	Default Shape transparency	$\mathcal{R} \in [0, 1]$	0		
`PointID`	Point ID	$\mathcal{S} \leq 20$			
Shape`ID`	Shape ID	$\mathcal{S} \leq 20$			

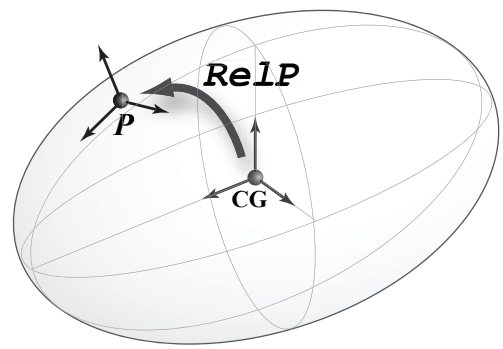
Fixed Mass Specification Notes

- `FixedMassID` is the leading tag for the Fixed Mass specifications.
- *IniP* is an axis-keyed Parameter tuple that specifies the generalized position of the Fixed Mass relative to its enclosing Structure's coordinate frame. The Fixed Mass position is not displaced by any containing Structures' *IniD* displacements but its absolute position is dependent on their *IniP* positions.
- *Color* specifies the default color for any Shapes in the Mass (see Section 3.14). The *Color* can be any POV-Ray color specifier such as `MediumBlue` or `rgb<1,.5,.5>`.
- *Transp* specifies the default transparency for any Shapes and Points in the Mass (see Section 3.14). *Transp* defaults to zero (opaque) and can be any value from zero to one.
- *PointID* is the leading tag for the specifications of a Mass Point. Any number of Points can be specified for each Mass following the general Mass specifications. (See Section 3.13.)
- Specifiers of the form *Shape*`ID` are leading tags for the specifications of a Mass Shape. *Shape* is one of `Box`, `Cylinder`, `Cone`, `Sphere`, `Ellipsoid`, or `Plane`. Any number of Shapes can be specified for each Mass following the general Mass specifications. (See Section 3.14.)

3.13 Point Aspect

A Point is a spatial point with a coordinate frame on a rigid Mass or Fixed Mass. Points serve as Link and instrument attachment points and as points where driving Loads are applied. Every Mass and Fixed Mass has an implicit Point with ID `CG` at its center of gravity (or origin for a Fixed Mass or a Mass with no associated weight) and with the same frame as the Mass or Fixed Mass. Points are fixed at a position and orientation relative to the CG frame. Each Mass or Fixed Mass may have any number of Points.

A Point is positioned relative to the CG frame of its Mass or Fixed Mass via the *`RelP`* Parameter tuple. Points that serve only as Link attachments are not affected by any rotational orientation different from that of the CG but Points with associated instruments may require a different orientation from that of the CG.



Point Specifications

Tag	Description	Value	Default	N	?
`PointID`	Point ID	$\mathcal{S} \leq 20$			
`RelP`	CG-relative generalized position	$(\mathcal{R})$	$(\mathbf{0})$		
`Size`	Point size	$(\mathcal{R})$	*see Notes*		
`Color`	Point color	$\mathcal{S}$	*see Notes*		
`Transp`	Point transparency	$\mathcal{R} \in [0, 1]$	0		

Point Specification Notes

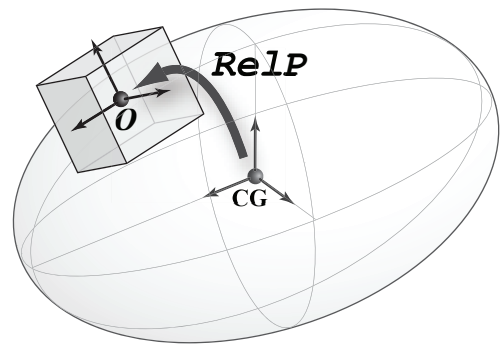
- `PointID` is the leading tag for the Point specifications.
- Any number of Points can be specified following the general specifications of a Mass or FixedMass (Point and Shape specifications can be interspersed).
- *`RelP`* is an axis-keyed Parameter tuple that specifies the generalized position of a Point's coordinate frame relative to its containing Mass or Fixed Mass coordinate frame. The Point is part of the rigid Mass or Fixed Mass and remains in that CG-relative position. *`RelP`* defaults to the zero Parameter tuple, which gives a Point frame equal to the CG frame.
- *`Size`* is a Parameter tuple indicating the radius of the sphere used to represent the Point in model images, for example `Size=( r:125 )`. Only the `r` specifier is allowed in a Point *`Size`*. The default Point *`Size`* is based on the size of the containing Mass or Fixed Mass.

- *`Color`* specifies the Point color. The *`Color`* can be any POV-Ray color specifier such as `MediumBlue` or `rgb<1,.5,.5>`. The *`Color`* of CG Points defaults to the containing Mass or Fixed Mass *`Color`* and then to gray. The *`Color`* of other Points defaults to a bright green.
- *`Transp`* specifies the Point transparency. *`Transp`* defaults to the containing Mass or Fixed Mass *`Transp`* and then to zero (opaque) and can be any value from zero to one.
- The CG Point of each Mass and Fixed Mass is not explicitly specified and thus their size, color, and transparency cannot be altered.

3.14 Shape Aspects

A Shape is a geometric shape with a coordinate frame that is part of a rigid Mass or Fixed Mass. Shapes are used in Contact interactions and in model image file outputs. Shapes are fixed at a position and orientation relative to the CG frame. Shape color and transparency can be set for Each Mass or Fixed Mass may have any number of Shapes.

A Shape is positioned relative to the CG frame of its Mass or Fixed Mass via the *RelP* Parameter tuple. Shape dimensions and directional axes are specified via the *Size* Parameter tuple, as described below for each Shape type.



Shape Specifications

Tag	Description	Value	Default	N	?
`BoxID`	Box ID	$\mathcal{S} \leq 20$			
`CylinderID`	Cylinder ID	$\mathcal{S} \leq 20$			
`ConeID`	Cone ID	$\mathcal{S} \leq 20$			
`SphereID`	Sphere ID	$\mathcal{S} \leq 20$			
`EllipsoidID`	Ellipsoid ID	$\mathcal{S} \leq 20$			
`PlaneID`	Plane ID	$\mathcal{S} \leq 20$			
`RelP`	CG-relative generalized position	$(\mathcal{R})$	$(\mathbf{0})$		
`Size`	Shape size	$(\mathcal{R})$			
`Color`	Shape color	$\mathcal{S}$	*see Notes*		
`Transp`	Shape transparency	$\mathcal{R} \in [0, 1]$	0		

Shape Specification Notes

- `BoxID`, `CylinderID`, `ConeID`, `SphereID`, `EllipsoidID`, and `PlaneID` are the leading tags for the Shape specifications.
- Any number of Shapes can be specified following the general specifications of a Mass or FixedMass (Shape and Point specifications can be interspersed).
- *RelP* is an axis-keyed Parameter tuple that specifies the generalized position of a Shape's coordinate frame relative to its containing Mass or Fixed Mass coordinate frame. The Shape is part of the rigid Mass or Fixed Mass and remains in that CG-relative position. *RelP* defaults to the zero Parameter tuple, which gives a Shape frame equal to the CG frame.

- *Size* is a Parameter tuple indicating the size and orientation of the Shape relative to its *RelP* frame position. *Size* may include non-DOF axis dimensions that will appear in the model image but do not affect its behavior (*RelP* ignores any non-DOF axis positioning specifiers). The *Size* specifications for each type of Shape are detailed below.

- *Color* specifies the Shape color. The *Color* can be any POV-Ray color specifier such as `MediumBlue` or `rgb<1,.5,.5>`. *Color* defaults to the containing Mass or Fixed Mass *Color* and then to a Shape-specific default color.

- *Transp* specifies the Shape transparency. *Transp* defaults to the containing Mass or Fixed Mass *Transp* and then to zero (opaque) and can be any value from zero to one.

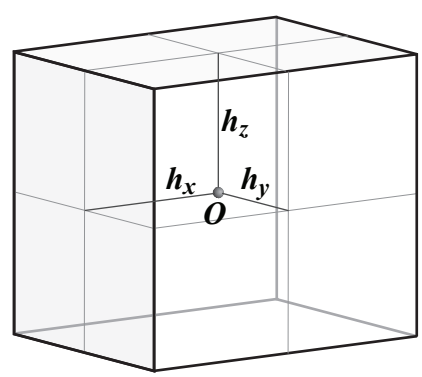
3.14.1 Box

A Box is a hyperrectangle with its coordinate frame origin at its center and axes parallel to its sides.

Box *Size* dimensions are the half-widths of each side. If the *Size* includes a dimension of the form `r:`$\boldsymbol{h}$ then $\boldsymbol{h}$ is the default dimension for the unspecified axes. So a cube is obtained by specifying only `r:`$\boldsymbol{h}$.

Examples:

```
Size=( r:200 )
Size=( x:500 y:200 z:300 )
```



`Size=( r:`$\boldsymbol{h}$ `x:`$\boldsymbol{h_x}$ `y:`$\boldsymbol{h_y}$ `z:`$\boldsymbol{h_z}$ `)`

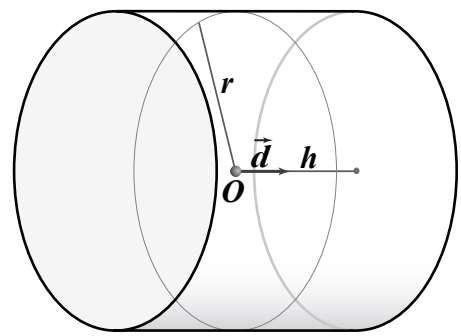
3.14.2 Cylinder

A Cylinder has its coordinate frame origin at its center with one directional axis normal to the circular sections.

Cylinder *Size* dimensions are the radius of the circular sections, `r:`***r***, and the half-width of the length along the directional axis, ***d*** `:`***h***. Any of the three linear axes can be designated as the directional axis.

Example:

```
Size=( r:250 y:80 )
```



`Size=( r:`***r*** ***d*** `:`***h*** `)`
where ***d*** is one of `x`, `y`, or `z`

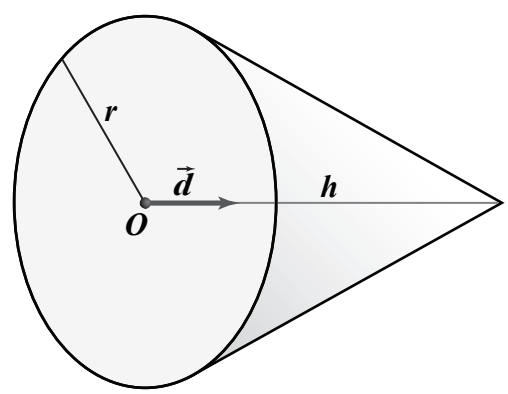
3.14.3 Cone

A Cone has its coordinate frame origin at the center of its circular base with one directional axis normal to the base and pointing towards the tip.

Cone *Size* dimensions are the radius of the circular base, `r:`***r***, and the length from base to tip along the directional axis, ***d*** `:`***h***. Any of the three linear axes can be designated as the directional axis.

Example:

```
Size=( r:100 x:50 )
```



`Size=( r:`***r*** ***d*** `:`***h*** `)`
where ***d*** is one of `x`, `y`, or `z`

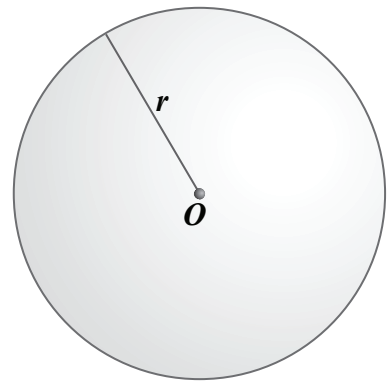
3.14.4 Sphere

A Sphere has its coordinate frame origin at its center.

Sphere *`Size`* has a single dimension of the radius, `r:`***r***.

Example:

```
Size=( r:125 )
```



`Size=( r:`***r*** `)`

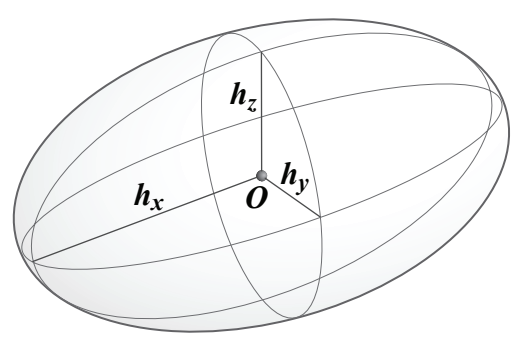
3.14.5 Ellipsoid

An Ellipsoid has its coordinate frame origin at its center with axes coincident with its principal axes.

Ellipsoid *`Size`* dimensions are the half-lengths along each principal axis. If the *`Size`* includes a dimension of the form `r:`***h*** then ***h*** is the default dimension for the unspecified axes.

Examples:

```
Size=( r:200 z:100 )
Size=( x:500 y:200 z:300 )
```



`Size=( r:`$\boldsymbol{h}$ `x:`$\boldsymbol{h}_x$ `y:`$\boldsymbol{h}_y$ `z:`$\boldsymbol{h}_z$ `)`

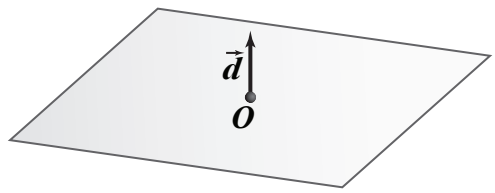
3.14.6 Plane

A Plane is an infinite plane with a coordinate frame centered at a point on the plane with a directional axis outwardly normal to the plane. (A zero-thickness Box can provide a finite plane.)

Plane *`Size`* has a single linear axis specifier. The sign of the value indicates the outward normal direction along the axis (the magnitude of the value has no effect).

Example:

```
Size=( z:+1 )
```



`Size=(` *`d`*`:+1 )`
`Size=(` *`d`*`:-1 )`
where *`d`* is one of `x`, `y`, or `z`

3.15 Link Element

A Link is a massless flexible load-carrying connection between Points on two Masses or Fixed Masses.

Link Specifications

Tag	Description	Value	Default	N	?
`LinkID`	Link ID	$\mathcal{S} \leq 20$			
`Descr`	Description	$\mathcal{S}$			
`Class`	Class	$\mathcal{C} \in$ `D`,`E`	*see Notes*		
`Point1`	Attachment Point #1	$\mathcal{S}$			
`Point2`	Attachment Point #2	$\mathcal{S}$			
`Ref`	Reference Link	$\mathcal{S}$	*none*		
`XScl`	Deflection scale factor ($X_\otimes$)	$\mathcal{R} > 0$	1		
`FScl`	Static force scale factor ($F_\otimes$)	$\mathcal{R} \geq 0$	1		
`RScl`	Relative velocity scale factor ($R_\otimes$)	$\mathcal{R} > 0$	1		
`DScl`	Damping force scale factor ($D_\otimes$)	$\mathcal{R} \geq 0$	1		
`MScl`	Dynamic Magnifier scale factor ($M_\otimes$)	$\mathcal{R} \geq 0$	1		
`Sta`	Static Aspect type	$\mathcal{S} \in$ `LE`, `EE`, `AE`, `BE`, `SE`, `AI`, `BI`, `SI`	*none*		
`Dyn`	Dynamic Aspect type	$\mathcal{S} \in$ `LD`, `AD`, `LM`, `AM`	*none*		

Link Specification Notes

- `LinkID` is the leading tag for the Link specifications.
- The Link *`Class`* defaults to `D` for defined Links and `E` for extracted Links. The *`Class`* is uniquely determined by default so explicit specification of the Link *`Class`* is optional.
- *`Point1`* and *`Point2`* are references to the Mass or FixedMass Points to which the Link is attached. If a Point reference doesn't specify a Point ID then the CG Point of the Mass or FixedMass is used by default. Only the relative position of the two Points affects the Link behavior so either assignment of the two Points to *`Point1`* and *`Point2`* is equivalent. Note that extracted Links cannot be connected to simulated Masses.
- **Inactive Links** can be specified for use as references by either omitting the *`Point1`* and *`Point2`* fields or by specifying the same Mass for both fields. Inactive Links cannot have any extracted Parameters and should not use automatic tail deflections. Inactive Links are counted in the defined Link tallies. Parametric outputs are available for inactive Links.

- *Ref* identifies a Link whose Static and Dynamic Aspects this Link's Aspects, if not specified, will refer to. Scaling relative to the referenced Link can be applied by using Link transform scale factors. Acyclic chains of `Ref` references are supported. Note that the referenced Link's deflection range determines any automatic tail deflections.
- **Link transforms** provide an easy way to modify Static and Dynamic Aspect characteristics. **Link-level** scale factor transforms $X_{\otimes}$, $F_{\otimes}$, $R_{\otimes}$, $D_{\otimes}$, and $M_{\otimes}$ (those specified before the Static and Dynamic Aspect specifications) apply to Static and Dynamic Aspect Parameters specified or in a referenced *Ref* Link. Additional transform Parameters are available in the Static (see Section 3.16.1) and Dynamic (see Section 3.17.1) Aspect specifications.
  - Transforms are applied to the affected **transforming Parameters** with defined values or referencing Parameters in other Links. Transforms are *not* applied to Parameters specified as extracted (*Tag* `=?(` *Specifiers* `)`) (so their estimates and bounds are not transformed) or to Parameters referencing other Parameters in the same Link.
  - Scale factor transforms are incorporated into the value of affected transforming Parameters and do not appear in the output Event file.
  - Transforms are incorporated before references to a transforming Parameter are resolved.
- `Sta` is the leading tag for the Static Aspect specifications, if any. The Static Aspect specifications must follow the general Link specifications. The Static Aspect types are described in detail in Section 3.16.
- `Dyn` is the leading tag for a Dynamic Aspect specification, if any. Dynamic Aspect specifications must follow the general Link specifications. A Link can have up to one Damper and one Magnifier Aspect `Dyn` specifier. The Dynamic Aspect types are described in detail in Section 3.17.

3.16 Static Aspects

SISAME-3D provides eight Link Static Aspect types, including five elastic and three inelastic (energy dissipating) types. Each Static Aspect has a two-character type tag, such as `LE`, that is specified in the Link's `Sta` field to select the type and serves as its ID in references.

A compression–positive sign convention is used for the Static Aspects. The static forces are denoted by f^S. The static stiffnesses are denoted by s^S.

Section 3.16.2 describes the elastic Static Aspect types and Section 3.16.3 describes the inelastic Static Aspect types.

3.16.1 Universal Static Parameters

A number of universal Static Parameters are available for all of the Static Aspect types. These Parameters can be included anywhere within the Static Aspect specifications (after the `Sta` field). The effects of these Parameters are not illustrated in the Static Aspect figures that follow.

The `OneWay` Parameter allows specification of compression-only or tension-only Static Aspects. A compression-only Static Aspect (`OneWay=C`) generates only nonnegative static forces. A tension-only (`OneWay=T`) Static Aspect generates only nonpositive static forces.

Static transforms provide an easy way to modify Static Aspect characteristics. **Scale factor** transforms $X_\otimes$ and $F_\otimes$ scale the Static characteristic along the deflection and force axes. **Offset** transforms X_Δ and F_Δ add offset values to the Static characteristic deflections and forces.

Static failure can be specified to occur if the deflection reaches or goes below a given $X^{\text{Fail}}_{\text{Min}}$ value or if it reaches or goes above a given $X^{\text{Fail}}_{\text{Max}}$ value. A failed Static Aspect generates zero static force for the remainder of the event.

Universal Static Parameters

Tag	Description	Value	Default	N	?
`OneWay`	One-way Static Aspect flag	$\mathcal{C} \in$ `C`, `T`	*none*	`N`	
`XScl`	Deflection scale factor ($X_\otimes$)	$\mathcal{R} > 0$	1		
`FScl`	Force scale factor ($F_\otimes$)	$\mathcal{R} \geq 0$	1		
`XDel`	Deflection offset (X_Δ)	$\mathcal{R}$	0		
`FDel`	Force offset (F_Δ)	$\mathcal{R}$	0		?
`XFailMin`	Tension failure deflection ($X^{\text{Fail}}_{\text{Min}}$)	$\mathcal{R}$	*none*	`N`	
`XFailMax`	Compression failure deflection ($X^{\text{Fail}}_{\text{Max}}$)	$\mathcal{R}$	*none*	`N`	

Universal Static Parameter Notes

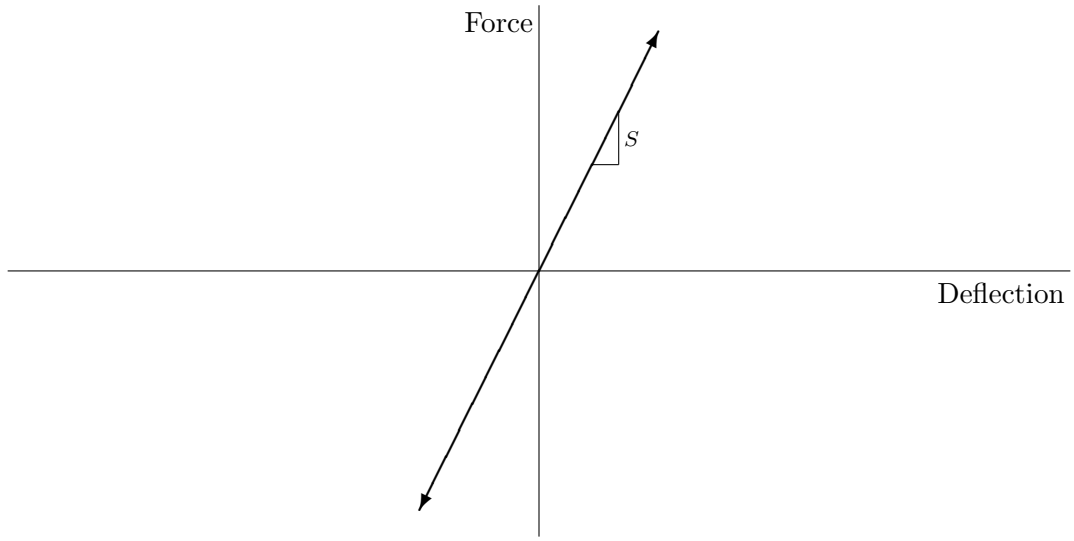
- Extracted one-way Static Aspects can hamper the solution convergence. Reducing the `ConPC`, `ConPD`, `MultPD`, and/or (for segmented Static Aspects) `ConSS` values may improve convergence in such cases.
- Transforms are applied to the affected **transforming Parameters** with defined values or referencing Parameters in other Links. Transforms are *not* applied to Parameters specified as extracted (*Tag* `=?(` *Specifiers* `)`) or to Parameters referencing other Parameters in the same Link.
- Scale factor transforms are incorporated into the value of affected transforming Parameters and do not appear in the output Event file.
- Transforms are incorporated before references to a transforming Parameter are resolved.
- X_Δ and F_Δ are transformed by Link-level (but not Static) scale factor transforms and are not incorporated into the value of transforming Static Parameters.
- The effect of X_Δ and F_Δ on an inelastic Static Aspect is to offset the initial Static characteristic.
- The combination of a one-way Static Aspect with $X_\Delta \neq 0$ and/or $F_\Delta \neq 0$ can cause some Static Parameters to be inactive in the first pass solution. The use of Parameter bounds to assure initial activity may be beneficial.
- References to Static scale factors should include the `Sta.` portion to distinguish them from references to corresponding Link-level scale factors.
- $X^{\text{Fail}}_{\text{Min}}$ and $X^{\text{Fail}}_{\text{Max}}$ are scaled by both the Static and Link-level $X_\otimes$ values, if any, but the failure deflections are not affected by the value of X_Δ.

3.16.2 Elastic Static Aspects

SISAME-3D provides five elastic Static Aspect types. Each of the elastic Static Aspect types has a static force that is a true (non-path-dependent) function of deflection.

The properties and Parameters for each elastic Static Aspect type are described below.

3.16.2.1 Static Type LE : Linear Elastic



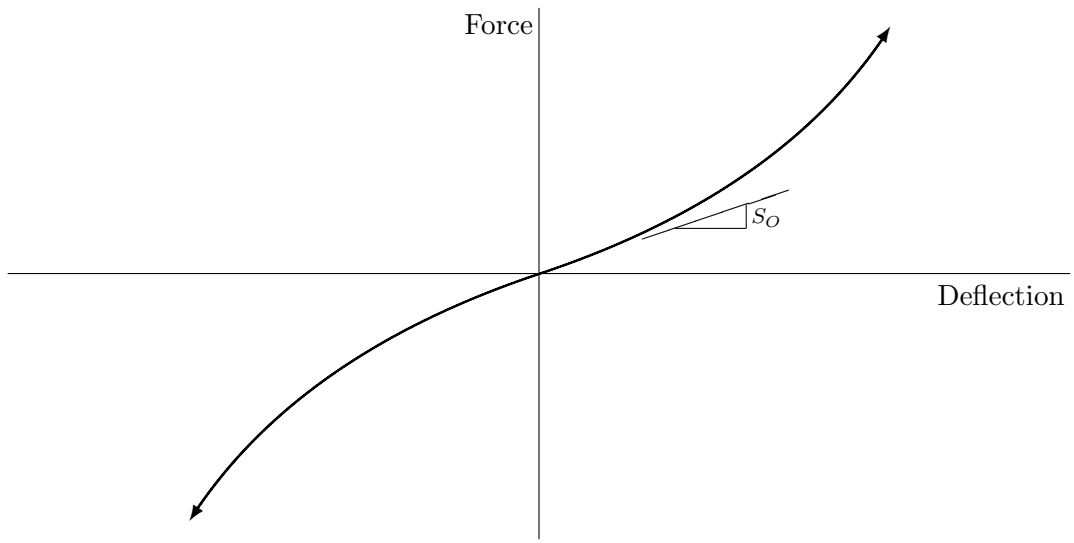
Static type LE is linear elastic with **force** and **stiffness**

$$f^S = Sx \qquad\qquad s^S = S\,.$$

Static Type LE Parameters

Tag	Description	Value	Default	N	?
S	Stiffness (S)	$\mathcal{R} \geq 0$			?

3.16.2.2 Static Type EE : Exponential Elastic



Static type EE is exponential elastic with **force** and **stiffness**

$$f^S \;=\; S_O\, x \;+\; C\, \sigma(x)\, |x|^P \qquad\qquad s^S \;=\; S_O \;+\; C\, P\, |x|^{P-1}$$

where S_O is the stiffness at the origin, $\sigma(x) = \pm 1$ is the sign of x, and C and P are the constant and exponent of the exponential term, respectively. If $C > 0$ and $P > 1$ the stiffness increases with $|x|$.

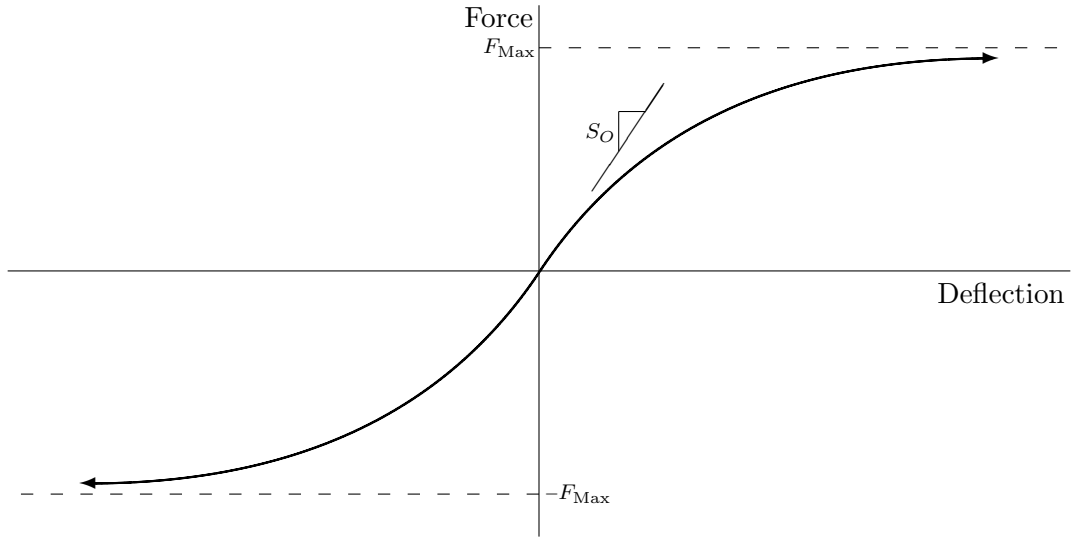
Static Type EE Parameters

Tag	Description	Value	Default	N	?
SO	Origin stiffness (S_O)	$\mathcal{R} \geq 0$			?
C	Constant (C)	$\mathcal{R} \geq 0$			?
P	Exponent (P)	$\mathcal{R} \geq 1$			?
XMag	Deflection magnitude estimate	$\mathcal{R} \geq 0$	0		

Static Type EE Notes

- Extraction of the exponent P can be hindered by the inherent exponential volatility. Despite the default presence of iteration constraints, convergence may benefit from the use of bounds and/or estimates on P.
- *XMag* is used to bound the stiffness for simulation time step selection if the Link is attached to a simulated Mass. Increasing *XMag* can force a small time step. A warning is generated if *XMag* is less than the actual deflection magnitude.
- Deflection scaling ($X_\otimes$) cannot be used with an extracted exponent P unless C is a non-reference extracted Parameter or a reference to another Parameter in the same Link.

3.16.2.3 Static Type AE : Arctangent Elastic



Static type AE is arctangent elastic with **origin stiffness** S_O and asymptotic **maximum force** magnitude F_{Max}.

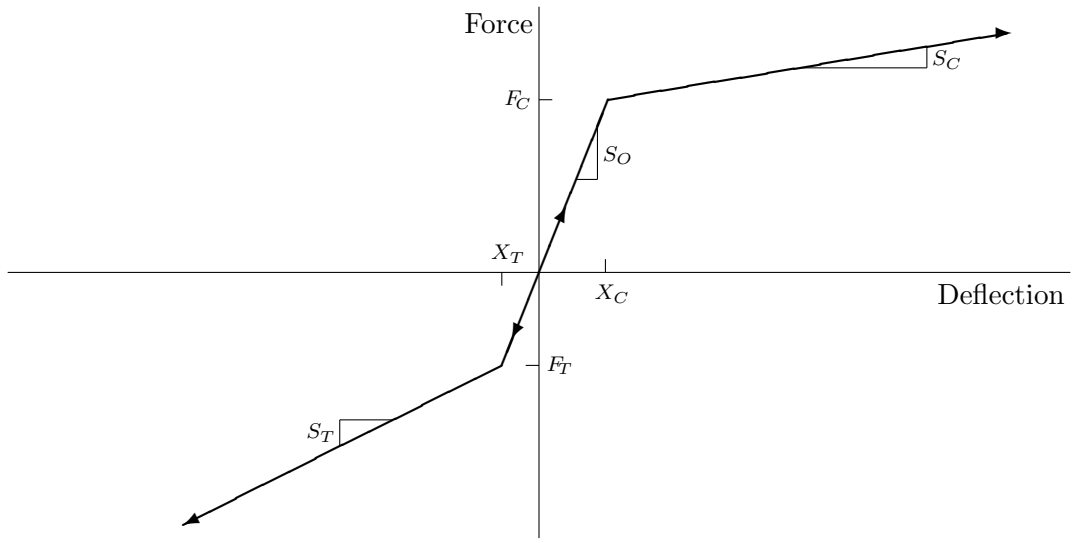
The **force** and **stiffness** are

$$f^S = \left(\frac{2}{\pi} F_{\text{Max}}\right) \text{Arctan}\left(\frac{\pi}{2} \frac{S_O}{F_{\text{Max}}} x\right) \qquad s^S = \frac{S_O}{1 + \left(\frac{\pi}{2} \frac{S_O}{F_{\text{Max}}} x\right)^2} .$$

Static Type AE Parameters

Tag	Description	Value	Default	N	?
SO	Origin stiffness (S_O)	$\mathcal{R} \geq 0$			?
FMax	Asymptotic maximum force magnitude (F_{Max})	$\mathcal{R} \geq 0$			?

3.16.2.4 Static Type BE : Bilinear Elastic



Static type BE is bilinear elastic.

The origin line **force** and **stiffness** are

$$f_O^S = S_O\, x \qquad\qquad s_O^S = S_O\,.$$

The compression line **force** and **stiffness** are

$$f_C^S = F_C + [x - X_C]\, S_C \qquad\qquad s_C^S = S_C\,.$$

The tension line **force** and **stiffness** are

$$f_T^S = F_T + [x - X_T]\, S_T \qquad\qquad s_T^S = S_T\,.$$

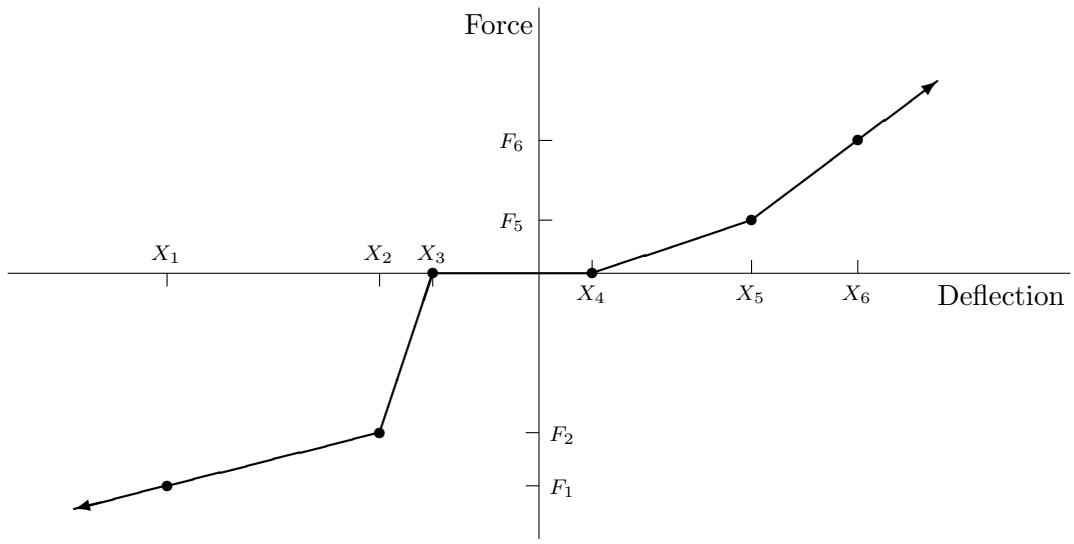
Static Type BE Parameters

Tag	Description	Value	Default	N	?
SO	Origin stiffness (S_O)	$\mathcal{R} \geq 0$			?
XC	Compression breakpoint deflection (X_C)	$\mathcal{R} \geq 0$	$-X_T$	N	?
FC	Compression breakpoint force (F_C)	$\mathcal{R} \geq 0$	$-F_T$	N	?
SC	Compression stiffness (S_C)	$\mathcal{R} \geq 0$	S_T	N	?
XT	Tension breakpoint deflection (X_T)	$\mathcal{R} \leq 0$	$-X_C$	N	?
FT	Tension breakpoint force (F_T)	$\mathcal{R} \leq 0$	$-F_C$	N	?
ST	Tension stiffness (S_T)	$\mathcal{R} \geq 0$	S_C	N	?

Static Type BE Notes

- Static BE Parameters are related by $F_C = S_O X_C$ and $F_T = S_O X_T$, which imply:
  - $S_O = 0 \implies F_C = F_T = 0$.
  - $X_C = 0 \implies F_C = 0$ and $X_T = 0 \implies F_T = 0$.
  - $F_C = 0 \implies S_O = 0$ and/or $X_C = 0$ and $F_T = 0 \implies S_O = 0$ and/or $X_T = 0$.
- Breakpoints can be specified using either the deflection or force Parameter but not both. Unspecified (implicit) breakpoint Parameters are determined by the above relationships. If neither deflection nor force Parameters are specified for a breakpoint they are determined by the defaults from the opposite breakpoint.
- Deflection Parameters must be used if $S_O = 0$ is specified or to allow extraction of $S_O = 0$. Deflection Parameters are also recommended if an extracted S_O may be small.
- Force Parameters can be used with known breakpoint deflections by specifying `FC=SO*`X_C or `FT=SO*`X_T, or alternatively, `SO=FC*`$1/X_C$ or `SO=FT*`$1/X_T$.
- Implicit breakpoint Parameters can be referenced if their implicit value is defined or is linear in the explicit extracted Parameters.
- Breakpoint Parameters without an explicit or implicit value are assigned the defaults shown.
- The compression line can be suppressed by using `XC=N` or `FC=N` or `SC=N`. The tension line can be suppressed using by `XC=N` or `FT=N` or `ST=N`. For BE Aspects with extracted breakpoints, suppressing the compression or tension lines is more efficient than specifying `SC=SO` or `ST=SO`.
- $F_T \neq -F_C$ and/or $S_T \neq S_C$ can be used to model asymmetric compression–tension behavior.

3.16.2.5 Static Type SE : Segmented Elastic



Static type SE is segmented elastic with a force defined by linear segments between specified deflection–force points.

The **force** is defined by the points (X_i, F_i) for $i = 1, \ldots, N$ where X_i is increasing with i. The outermost segments are extrapolated to define the force at deflections outside the range of the points. **Preload** occurs if the force at $x = 0$ is nonzero.

Symmetric Static type SE is a variant with the force for positive deflections defined by the points (X_i, F_i) where $X_i + X_\Delta \geq 0$ and the symmetric force for negative deflections defined by the points $(-X_i, -F_i)$.

Static Type SE Parameters

Tag	Description	Value	Default	N	?
`XiScl`	Deflection block scale factor ($X_{i\otimes}$)	$\mathcal{R} > 0$	1		
`FiScl`	Force block scale factor ($F_{i\otimes}$)	$\mathcal{R} \geq 0$	1		
`ConSS`	Segment smoothness confidence band factor	$\mathcal{R} > 0$	*hierarchy*	`N`	
`AnySlope`	Extracted segment slope control	$\mathcal{B}$	`False`		
`Symmetric`	Symmetric SE variant selector	$\mathcal{B}$	`False`		
`X`	Deflection (X_i) block	$[\mathcal{R} \nearrow]$			
`F`	Force (F_i) block	$[\mathcal{R}]$			?

Static Type SE Notes

- $X_{i\otimes}$ and $F_{i\otimes}$ can be used to scale the X_i and F_i block values before applying any Link transforms.
- *ConSS* is the confidence band factor for extracted segment smoothness target equations. These equations target no slope change between adjacent segments, so reducing *ConSS* tends to smooth the Static characteristic. The segment smoothness confidence band is of the form *ConSS*$\cdot 5000(\tilde{w}/\tilde{x})$ where $\tilde{w}$ is the weight magnitude (in force units) and $\tilde{x}$ is a unit-invariant baseline deflection. *ConSS* defaults to the enclosing Structure's *ConSS* and then to 1. Segment smoothness targets can be suppressed by using `ConSS=N`. `ConSS` is suppressed in Aspects defined by a `Ref` reference. *ConSS* is not affected by Link transforms.
- *AnySlope* controls the allowable sign of extracted segment slopes. A value of `True` allows unconstrained slope signs. By default the extracted segment (those with extracted forces at either end) slopes are constrained to be nonnegative. If `AnySlope=True` is used segment slopes can still be constrained to be nonnegative by using specifiers of the form `?( >F`i `)` for the entry for F_{i+1}, or `?`n`( >F`i`: )` for a section of n forces starting at F_{i+1}.
- An equal number of X_i and F_i values are specified following the `X` and `F` tags. These are block Parameters and both single and block reference entries are supported (see Section 3.3.6).
- `X` block deflections can be selected automatically by entering `#` instead of a value. For Links between instrumented Masses SISAME-3D can use the actual deflection range to set automatic X_1 or X_N tail deflections. For symmetric SE Aspects the automatic tail deflection values are $X_1 = 0$ and $X_N = \max(|x|)$. Sequences of automatic deflections are evenly spaced. Automatic deflection values are affected by Link transforms. The form `#`n can be used to indicate n successive automatic deflections.
- The form `?`n can be used to indicate n successive extracted forces.
- The (X_i, F_i) points are offset by (X_Δ, F_Δ). For symmetric SE Aspects the $(-X_i, -F_i)$ negative deflection points are offset by $(-X_\Delta, -F_\Delta)$ to preserve symmetry.
- For symmetric SE Aspects $X_1 + X_\Delta \geq 0$ is required. Note that if $X_1 + X_\Delta = 0$ but $F_1 + F_\Delta \neq 0$ there will be a force discontinuity at the origin. A segment smoothness target is applied to an extracted origin segment between $(X_1 + X_\Delta, F_1 + F_\Delta)$ and $(-(X_1 + X_\Delta), -(F_1 + F_\Delta))$ only if $X_1 + X_\Delta > 0$.
- Negative segment slopes are allowed although they deviate from idealized physical lumped-parameter behavior. They may be useful for modeling load-paths of greater size and/or complexity that are actually composites of multiple load-paths when using a coarse model discretization. The `AnySlope` Parameter can be used to allow negative extracted segment slopes. Negative-sloped tail segments are extrapolated across the zero force line, which will give nonphysical behavior. SISAME-3D issues a warning if negative slopes are obtained for an extracted tail segment.

3.16.3 Inelastic Static Aspects

SISAME-3D provides three inelastic Static Aspect types. Inelastic Static Aspects provide a static force that is dependent on the current and past deflections (path-dependent).

Yielding (inelastic deformation) occurs during increasing deflection traversal of a compression boundary or decreasing deflection traversal of a tension boundary.

Unloading occurs when the deflection reverses direction while on a compression or tension boundary. The force follows a linear elastic unloading line from the last boundary point. The unloading path, consisting of the unloading line and any slack or reloading (in the opposite direction) line behaves linear elastically and is traversed until a boundary is reached or a new domain entered (such as one-way clipping at zero force).

For bidirectionally yielding Aspects the opposite boundary shifts as necessary so that the unloading path intersects it: the compression boundary shifts by negative deflections and the tension boundary shifts by positive deflections. The boundary shifts create an expanding energy envelope.

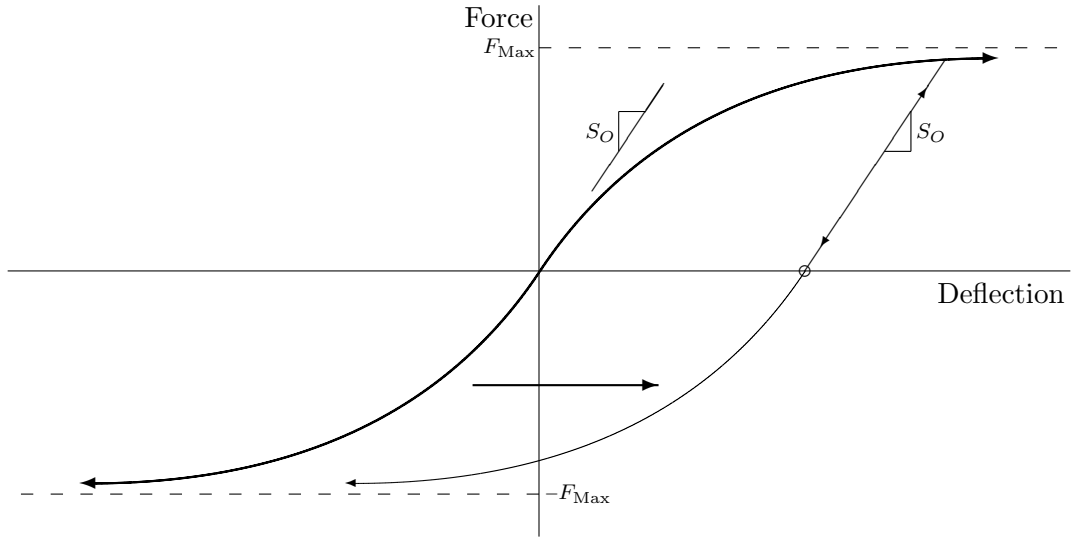
One-way Aspects do not yield (thus the unloading path and opposite boundary do not shift) while the force is clipped at zero.

Energy is dissipated through cyclic hysteresis during clockwise traversals in the deflection–force plane.

If the initial deflection–force point differs from the Aspect's natural origin the Aspect behaves as if the Link's deflection had started at the origin and moved continuously to the initial deflection. (Some assumption is necessary as these Aspects are path-dependent.)

The properties and Parameters for each inelastic Static Aspect type are described below.

3.16.3.1 Static Type AI : Arctangent Inelastic



Static type AI is arctangent inelastic with **origin and unloading stiffness** S_O and asymptotic **maximum force** magnitude F_{Max}.

The initial **boundary force** and **stiffness** are

$$f^S = \left(\frac{2}{\pi} F_{\text{Max}}\right) \text{Arctan}\left(\frac{\pi}{2} \frac{S_O}{F_{\text{Max}}} x\right) \qquad s^S = \frac{S_O}{1 + \left(\frac{\pi}{2} \frac{S_O}{F_{\text{Max}}} x\right)^2} .$$

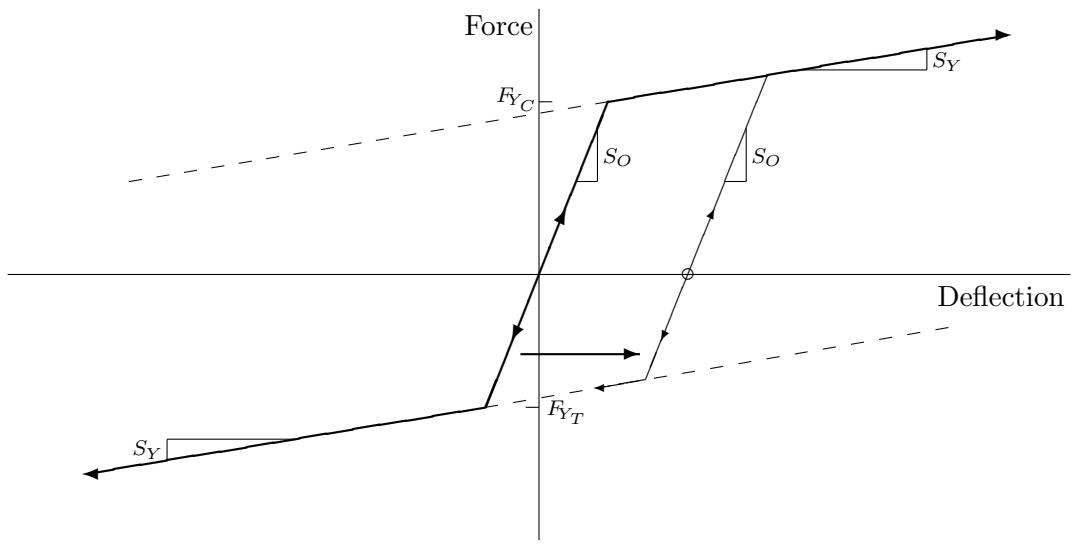
Yielding occurs during traversal of the compression or tension boundaries. The opposite boundary shifts as necessary so that the unloading line intersects it.

Unloading traverses an elastic line with slope S_O from the last boundary point until either boundary is reached.

Static Type AI Parameters

Tag	Description	Value	Default	N	?
`SO`	Origin and unloading stiffness (S_O)	$\mathcal{R} \geq 0$			?
`FMax`	Asymptotic maximum force magnitude (F_{Max})	$\mathcal{R} \geq 0$			?

3.16.3.2 Static Type BI : Bilinear Inelastic



Static type BI is bilinear inelastic with **origin and unloading stiffness** S_O, initial **compression yield force** F_{Y_C}, initial **tension yield force** F_{Y_T}, and **yielding stiffness** of S_Y.

The origin line **force** and **stiffness** are

$$f_O^S \;=\; S_O\, x \qquad\qquad s_O^S \;=\; S_O\,.$$

The compression boundary **force** and **stiffness** are

$$f_C^S \;=\; F_{Y_C} \;+\; [x - X_{Y_C}]\, S_Y \qquad\qquad s_C^S \;=\; S_Y\,.$$

The tension boundary **force** and **stiffness** are

$$f_T^S \;=\; F_{Y_T} \;+\; [x - X_{Y_T}]\, S_Y \qquad\qquad s_T^S \;=\; S_Y\,.$$

Yielding occurs during traversal of the compression or tension boundaries. The opposite boundary shifts as necessary so that the unloading line intersects it.

Unloading traverses an elastic line with slope S_O from the last boundary point until either boundary is reached.

Static Type BI Parameters

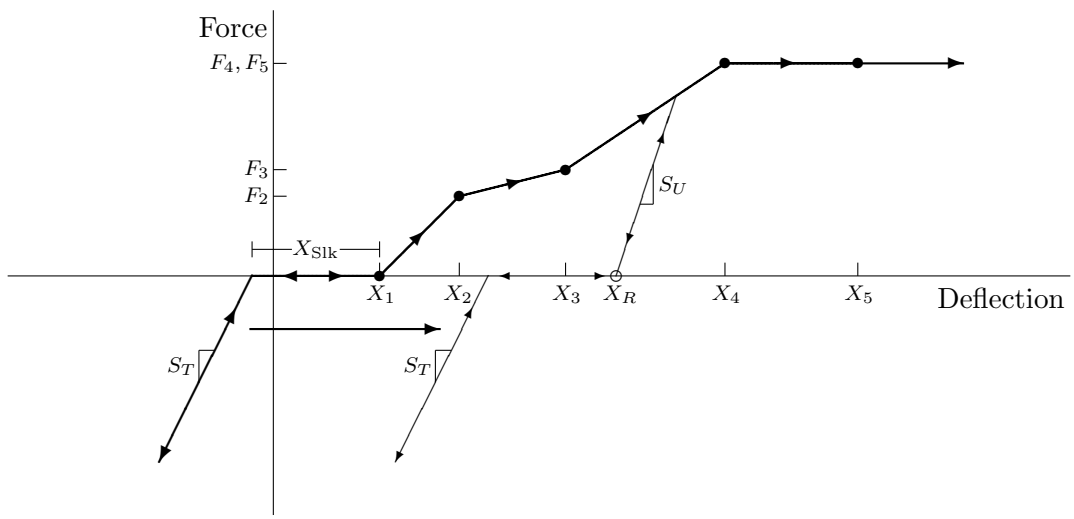
Tag	Description	Value	Default	N	?
SO	Origin and unloading stiffness (S_O)	$\mathcal{R} \geq S_Y$			?
FYC	Compression yield force (F_{Y_C})	$\mathcal{R} \geq 0$	$-F_{Y_T}$	N	?
FYT	Tension yield force (F_{Y_T})	$\mathcal{R} \leq 0$	$-F_{Y_C}$	N	?
SY	Yield stiffness (S_Y)	$S_O \geq \mathcal{R} \geq 0$		N	?

Static Type BI Notes

- If *SO* $= 0$ the force is constant for all deflections and the other BI Parameters are inactive and are suppressed by default.
- Yield deflection values, X_{Y_C} or X_{Y_T}, can be implicitly defined by specifying `FYC=SO*`X_{Y_C} or `FYT=SO*`X_{Y_T}.
- Yielding in either direction can be suppressed using `FYC=N` or `FYT=N`.
- Specifying `SY=N` suppresses yielding in both directions (giving linear elastic behavior) and suppresses the inactive F_{Y_C} and F_{Y_T} by default.
- In One-way BI Aspects with $S_Y > 0$ both F_{Y_C} and F_{Y_T} can contribute to the behavior at large deflections where unloading may reach the opposite boundary before reaching zero force. Suppressing the opposite boundary (using `FYT=N` or `FYC=N`) assures unloading continuously to zero force from all deflections.
- $F_{Y_T} \neq -F_{Y_C}$ can be used to model asymmetric compression–tension behavior.

3.16.3.3 Static Type SI : Segmented Inelastic

Nonsymmetric Static Type SI



Nonsymmetric Static type SI is segmented inelastic with a compression boundary defined by linear segments between specified deflection–force points.

The **compression boundary** is defined by the points (X_i, F_i) for $i = 1, \ldots, N$ where X_i is increasing with i and $F_i + F_\Delta \geq 0$. The last compression segment is extrapolated to define the boundary for $x > X_N + X_\Delta$.

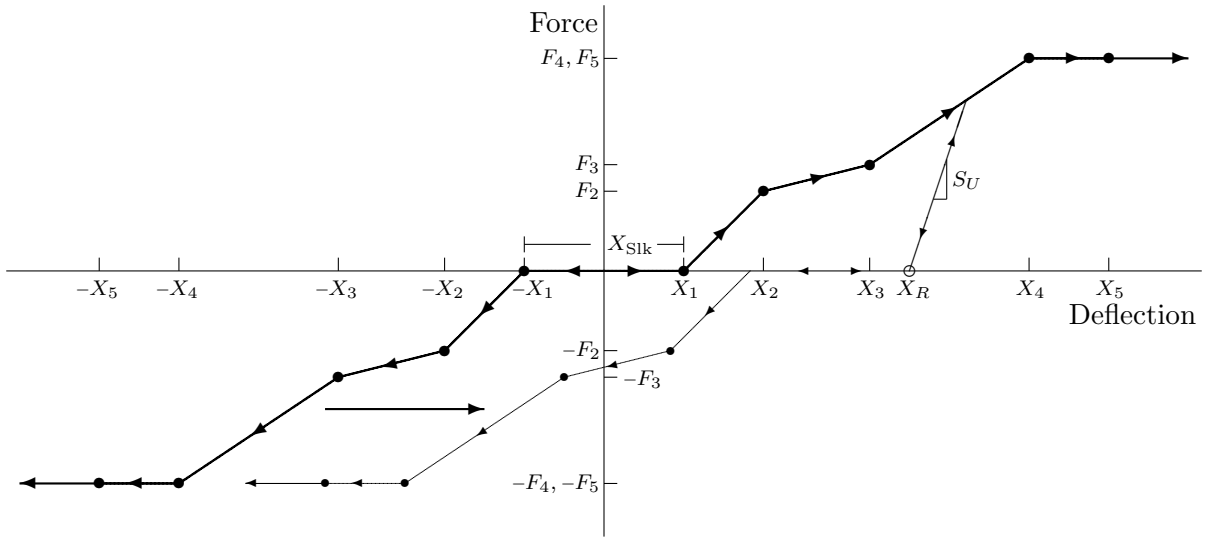
Yielding occurs during increasing deflection traversal of the compression boundary.

Unloading traverses an elastic line with slope S_U from the last boundary point down to zero force. If the deflection increases again to the last boundary deflection yielding continues on the compression boundary.

Slack (zero force) occurs for a distance of X_{Slk} below the point, X_R, where the unloading line intersects the zero force axis.

Tension traverses an elastic line with slope S_T for all deflections below $X_R - X_{\text{Slk}}$.

Symmetric Static Type SI



Symmetric Static type SI is segmented inelastic with compression and tension boundaries defined by linear segments between deflection–force points.

The initial **compression boundary** is defined by the points (X_i, F_i) for $i = 1, \ldots, N$ where X_i is increasing with i, $X_i + X_\Delta \geq 0$, and $F_i + F_\Delta \geq 0$. The symmetric initial **tension boundary** is defined by the points $(-X_i, -F_i)$. The last segments are extrapolated to define the boundary for deflections outside the boundary point range.

Yielding occurs during traversal of the compression or tension boundaries. The opposite boundary shifts as necessary so that the unloading path intersects it.

Unloading traverses an elastic line with slope S_U from the last boundary point to zero force. If the deflection again reaches the last boundary deflection yielding continues on the boundary.

Slack (zero force) occurs for the (implied) slack distance, X_{Slk}, between the point, X_R, where the current unloading line intersects the zero force axis and the point where **reloading** towards the opposite boundary along a line with slope S_U begins.

Static Type SI Parameters

Tag	Description	Value	Default	N	?
`SU`	Unloading slope (S_U)	$\mathcal{R} \geq 0$			?
`ST`	Tension slope (S_T) for nonsymmetric SI	$\mathcal{R} \geq 0$	S_U		?
`XSlk`	Slack (X_{Slk}) for nonsymmetric SI	$\mathcal{R} \geq 0$	0		?
`XiScl`	Deflection block scale factor ($X_{i\otimes}$)	$\mathcal{R} > 0$	1		
`FiScl`	Force block scale factor ($F_{i\otimes}$)	$\mathcal{R} \geq 0$	1		
`ConSS`	Segment smoothness confidence band factor	$\mathcal{R} > 0$	*hierarchy*	N	
`AnySlope`	Extracted segment slope control	$\mathcal{B}$	`False`		
`Symmetric`	Symmetric SI variant selector	$\mathcal{B}$	`False`		
`X`	Boundary deflection (X_i) block	$[\mathcal{R} \nearrow]$			
`F`	Boundary force (F_i) block	$[\mathcal{R}]$			?

Static Type SI Notes

- If S_U is less than any of the boundary segment slopes it is possible for energy to be created in the deflection–force plane if the unloading line crosses outside of the boundary. SISAME-3D issues a warning if an SI Aspect is defined with this property. Extracted S_U values are constrained to be at least as large as any segment slope.
- S_T applies only to nonsymmetric SI Aspects. For symmetric SI Aspects the unloading slope, S_U, is used for unloading/reloading in both compression and tension.
- X_{Slk} can be specified only for nonsymmetric SI Aspects. For symmetric SI Aspects the **implied slack** value is $\max(2(X_1 - ({}^{F_1}/_{S_U})), 0)$.
- $X_{i\otimes}$ and $F_{i\otimes}$ can be used to scale the X_i and F_i block values before applying any Link transforms and without affecting other SI Parameters.
- *`ConSS`* is the confidence band factor for extracted segment smoothness target equations. These equations target no slope change between adjacent segments, so reducing *`ConSS`* tends to smooth the Static characteristic. The segment smoothness confidence band is of the form *`ConSS`*$\cdot 5000({}^{\tilde{w}}/_{\tilde{x}})$ where $\tilde{w}$ is the weight magnitude (in force units) and $\tilde{x}$ is a unit-invariant baseline deflection. *`ConSS`* defaults to the enclosing Structure's *`ConSS`* and then to 1. Segment smoothness targets can be suppressed by using `ConSS=N`. `ConSS` is suppressed in Aspects defined by a `Ref` reference. *`ConSS`* is not affected by Link transforms.
- *`AnySlope`* controls the allowable sign of extracted segment slopes. A value of `True` allows unconstrained slope signs. By default the extracted segment (those with extracted forces at either end) slopes are constrained to be nonnegative. If `AnySlope=True` is used segment slopes can still be constrained to be nonnegative by using specifiers of the form `?( >F`i `)` for the entry for F_{i+1}, or `?`n`( >F`i`: )` for a section of n forces starting at F_{i+1}.
- An equal number of X_i and F_i values are specified following the `X` and `F` tags. These are block Parameters and both single and block reference entries are supported (see Section 3.3.6).

- `X` block deflections can be selected automatically by entering `#` instead of a value. For Links between instrumented Masses SISAME-3D can use the actual deflection range to set automatic X_1 or X_N tail deflections. For symmetric SI Aspects the automatic tail deflection values are $X_1 = 0$ and $X_N = \max(|x|)$. Sequences of automatic deflections are evenly spaced. Automatic deflection values are affected by Link transforms. The form `#`n can be used to indicate n successive automatic deflections.
- $F_i + F_\Delta \geq 0$ is required for defined and extracted forces.
- Extracted forces are constrained to keep segment slopes below S_U.
- The form `?`n can be used to indicate n successive extracted forces.
- The (X_i, F_i) compression boundary points are offset by (X_Δ, F_Δ). For symmetric SI Aspects the $(-X_i, -F_i)$ tension boundary points are offset by $(-X_\Delta, -F_\Delta)$ to preserve symmetry.
- If $F_1 + F_\Delta > 0$ the unloading line from $(\hat{X}_1 + X_\Delta, F_1 + F_\Delta)$, where $\hat{X}_1$ is the current shifted compression X_1 value, is traversed between the zero force axis and the compression boundary when $X_R < x < \hat{X}_1$. For symmetric SI Aspects the tension boundary is treated symmetrically. (In the degenerate $S_U = 0$ case the transition to the compression boundary occurs discontinuously at $\hat{X}_1 + X_\Delta$ and subsequent unloading occurs on a zero slope line.)
- Negative segment slopes are allowed although they deviate from idealized physical lumped-parameter behavior. They may be useful for modeling load-paths of greater size and/or complexity that are actually composites of multiple load-paths when using a coarse model discretization. And a steep negative slope may be appropriate to represent a buckling event. The `AnySlope` Parameter can be used to allow negative extracted segment slopes. Negative-sloped last segments are extrapolated to their zero force crossing and remain at zero force beyond that deflection (yielding continues while at zero force). SISAME-3D issues a warning if negative slopes are obtained for an extracted last segment.

3.17 Dynamic Aspects

SISAME-3D provides four Link Dynamic Aspect types, including two Damper and two Magnifier types. A Link can have up to one Damper of either type and one Magnifier of either type. Each Dynamic Aspect has a two-character type tag, such as `AM`, that is specified in the Link's `Dyn` field to select the type and serves as its ID in references.

The two Damper types provide a force that is a function of the Link's relative velocity. The damping force is independent of the associated static force and can be used with no Static Aspect to obtain a pure damping Element.

The two Magnifier types scale the static force by a magnification factor that is a function of the Link's relative velocity.

A compression–positive sign convention is used for the Dynamic Aspects.

The Dampers are described in Section 3.17.2 and the Magnifiers are described in Section 3.17.3.

3.17.1 Universal Dynamic Parameters

A number of universal Dynamic Parameters are available for all of the Dynamic Aspect types. These Parameters can be included anywhere within the Dynamic Aspect specifications (after the `Dyn` field). The effects of these Parameters are not illustrated in the Dynamic Aspect figures that follow.

The `OneWay` Parameter allows specification of compression-only or tension-only Dynamic Aspects. A compression-only Dynamic Aspect (`OneWay=C`) is inactive when the relative velocity is negative. A tension-only Dynamic Aspect (`OneWay=T`) is inactive when the relative velocity is positive.

Dynamic transforms provide an easy way to modify Dynamic Aspect characteristics. **Scale factor** transforms $X_\otimes$, $R_\otimes$, $D_\otimes$, and $M_\otimes$ scale the Dynamic characteristic along the corresponding axes.

Universal Dynamic Parameters

Tag	Description	Value	Default	N	?
`OneWay`	One-way Dynamic Aspect flag	$\mathcal{C} \in$ `C`, `T`	*none*	`N`	
`XScl`	Deflection scale factor ($X_\otimes$)	$\mathcal{R} > 0$	1		
`RScl`	Relative velocity scale factor ($R_\otimes$)	$\mathcal{R} > 0$	1		
`DScl`	Damping force scale factor ($D_\otimes$)	$\mathcal{R} \geq 0$	1		
`MScl`	Dynamic Magnifier scale factor ($M_\otimes$)	$\mathcal{R} \geq 0$	1		

Universal Dynamic Parameter Notes

- Transforms are applied to the affected **transforming Parameters** with defined values or referencing Parameters in other Links. Transforms are *not* applied to Parameters specified as extracted (*`Tag`*`=?(` *`Specifiers`* `)`) or to Parameters referencing other Parameters in the same Link.
- Scale factor transforms are incorporated into the value of affected transforming Parameters and do not appear in the output Event file.
- Transforms are incorporated before references to a transforming Parameter are resolved.
- $M_\otimes$ scales the parametric contribution to the dynamic magnification that is added to one to get the magnification: M_{Slp} and, for Dynamic type AM, $(M_{\mathrm{Max}} - 1)$ are scaled by $M_\otimes$.
- $X_\otimes$ affects only the damping slack deflection, D_{Slk}.
- $D_\otimes$ has no effect on Dynamic Magnifiers.
- $M_\otimes$ has no effect on Dampers.
- References to the Dynamic scale factors should include the *`Dyn.`* portion to distinguish them from references to corresponding Link-level scale factors.

3.17.2 Dampers

Dampers generate a force that is a function of a Link's relative velocity. Dampers act independently of and in parallel with a Link's Static Aspect, if any.

A Damper slack, D_{Slk}, can be specified to prevent damping at deflections less than D_{Slk}.

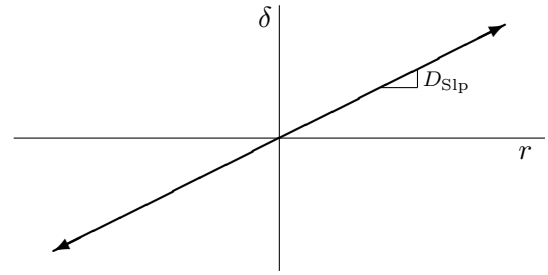
A compression-only (`OneWay=C`) Damper generates zero force at negative relative velocities. A tension-only (`OneWay=T`) Damper generates zero force at positive relative velocities.

The damping function and a picture of the Damper characteristic are shown below for both Damper types. Both of the damping functions are of the form $f = f^s + \delta(r)$.

The damping slope, D_{Slp}, has units of force/velocity using the force and velocity units of the Model's dimensional system.

3.17.2.1 Dynamic Type LD : Linear Damper

$$f \;=\; f^S \;+\; D_{\text{Slp}}\, r$$

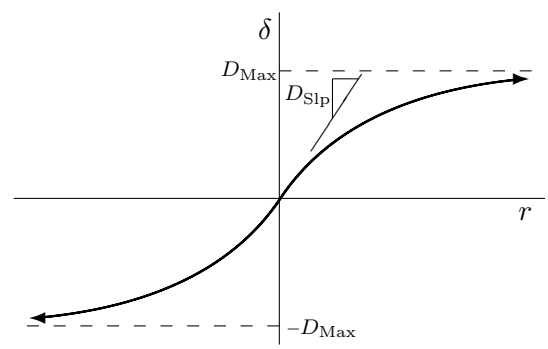


Dynamic Type LD Parameters

Tag	Description	Value	Default	N	?
`DSlp`	Damping slope (D_{Slp})	$\mathcal{R} \geq 0$			?
`DSlk`	Damping slack (D_{Slk})	$\mathcal{R}$	*none*	`N`	

3.17.2.2 Dynamic Type AD : Arctangent Damper

$$f \;=\; f^S \;+\; \left(\frac{2}{\pi} D_{\text{Max}}\right) \text{Arctan}\left(\frac{\pi}{2}\,\frac{D_{\text{Slp}}}{D_{\text{Max}}}\, r\right)$$



Dynamic Type AD Parameters

Tag	Description	Value	Default	N	?
`DSlp`	Damping slope at $r = 0$ (D_{Slp})	$\mathcal{R} \geq 0$			?
`DMax`	Maximum damping force magnitude (D_{Max})	$\mathcal{R} \geq 0$			?
`DSlk`	Damping slack (D_{Slk})	$\mathcal{R}$	*none*	`N`	

3.17.3 Magnifiers

Magnifiers scale the static force based on a function of the relative velocity. This is used to approximately compensate for various inertial lag strain rate effects. These effects arise from the model discretization error (the actual load-paths are not massless) and from using load-paths that do not model transverse motions [12]. Transverse motions are particularly significant during buckling/collapse events.

The Magnifiers appear complicated because of the $\sigma(r f^s)$ exponent ($\sigma(\cdot)$ is the sign function). The exponent is applied to a term that is at least 1 and depends only on the relative velocity magnitude. If the static force and the relative velocity have the same sign, then the Magnifier effect increases the force magnitude, and so the static force is multiplied by the term. If they have opposite signs then the Magnifier effect decreases the force magnitude, and so the static force is divided by the term. This is consistent with the intuition that the dynamic effects should resist the motion and dissipate energy.

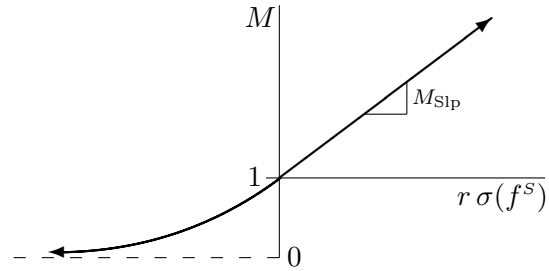
A compression-only (`OneWay=C`) Magnifier has a factor of one at negative relative velocities. A tension-only (`OneWay=T`) Magnifier has a factor of one at positive relative velocities.

Applying a Magnifier to extracted Static Aspects which may have zero force crossings determined by extracted Parameters requires that SISAME-3D iterate for the model, since the sign of the force affects the magnification factor, and may hamper the solution convergence. Reducing the ***ConPC***, ***ConPD***, ***MultPD***, and/or (for segmented Static Aspects) ***ConSS*** values may improve convergence in such cases.

The Magnifier functions and picture are shown below for both Magnifier types. Both of the Magnifiers are of the form $f = M(r, \sigma(f^s))\, f^s$.

3.17.3.1 Dynamic Type LM : Linear Magnifier

$$f \;=\; [1 + M_{\text{Slp}}\,|r|\,]^{\sigma(rf^S)}\,f^S$$

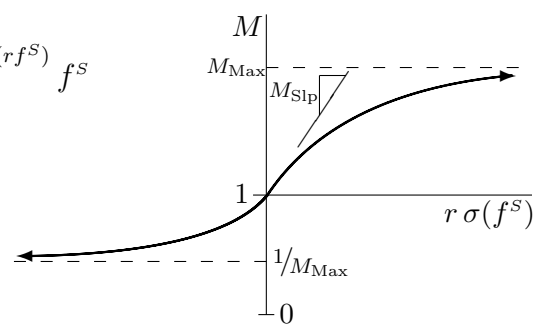


Dynamic Type LM Parameters

Tag	Description	Value	Default	N	?
`MSlp`	Magnifier slope (M_{Slp})	$\mathcal{R} \geq 0$			?

3.17.3.2 Dynamic Type AM : Arctangent Magnifier

$$f \;=\; \Big[1 + \Big(\tfrac{2}{\pi}(M_{\text{Max}} - 1)\Big)\,\text{Arctan}\Big(\tfrac{\pi}{2}\,\tfrac{M_{\text{Slp}}}{M_{\text{Max}}-1}\,|r|\,\Big)\Big]^{\sigma(rf^S)}\,f^S$$



Dynamic Type AM Parameters

Tag	Description	Value	Default	N	?
`MSlp`	Magnifier slope at $r = 0$ (M_{Slp})	$\mathcal{R} \geq 0$			?
`MMax`	Maximum magnification factor (M_{Max})	$\mathcal{R} \geq 1$			?

3.18 Load Element

A Load represents either a collection of Mass inertia and Link loads, or a driving load applied to each of a collection of Masses.

Load Specifications

Tag	Description	Value	Default	N	?
`LoadID`	Load ID	$\mathcal{S} \leq 20$			
`Descr`	Description	$\mathcal{S}$			
`Class`	Class	$\mathcal{C} \in$ `G`, `T`, `D`	*see Notes*		
`Mass`	Load Mass members	$\mathcal{S}$	*none*		
`Link`	Load Link members	$\mathcal{S}$	*none*		
`Point`	Driving Load application Point members	$\mathcal{S}$	*none*		
`Cutoff`	Cutoff frequency	$\mathcal{R} \geq 0$	*hierarchy*		
`ZeroSm`	Time-zero tail smoothing frequency	$\mathcal{R} > 0$	*hierarchy*	`N`	
`EndSm`	Final time tail smoothing frequency	$\mathcal{R} > 0$	*hierarchy*	`N`	
`ConF`	Force confidence band	$(\mathcal{R}) > 0$	*see Notes*		
`ConI`	Impulse confidence band factor	$\mathcal{R} > 0$	*hierarchy*	`N`	
`ConJ`	Integrated impulse confidence band factor	$\mathcal{R} > 0$	*hierarchy*	`N`	
`TarSignal`	Target loading signal specifiers	$(\mathcal{S})$	*none*		
`DrvSignal`	Driving loading signal specifiers	$(\mathcal{S})$	*none*		
`CmpSignal`	Comparison loading signal specifiers	$(\mathcal{S})$	*none*		

Load Specification Notes

- `LoadID` is the leading tag for the Load specifications.
- The Load *`Class`* specifies the type of Load.
  - A **group Load** can be specified by `Class=G`. A group Load has Mass and/or Link members. Group Loads can have comparison loading signals (specified with *`CmpSignal`*) that are used to generate fit measures but have no effect on the Model's behavior.
  - A **target Load** can be specified by `Class=T`. A target Load has Mass and/or Link members and target loading signal(s) (specified with *`TarSignal`*). The target loading signals are used as a target for the corresponding group Load in the extraction process. Target Loads are not carried over to the extracted model file.
  - A **driving Load** can be specified by `Class=D`. A driving Load has one or more Points specified to indicate where to apply the loading. Signals defining the driving loading must be specified

in *DrvSignal*. Driving Loads applied to instrumented, constant-velocity, or Fixed Masses only affect their applied loading and, for instrumented Masses, their effective motion.

  - If *CmpSignal* is specified or if no signals specifications are present the *Class* defaults to `G` (group).
  - If *TarSignal* is specified, the *Class* defaults to `T` (target).
  - If *DrvSignal* or Points are specified, the *Class* defaults to `D` (driving).

- A Mass contributes its inertial (motion-based) loading to group and target Loads, not the applied loading on the Mass. Load Mass members must have specified (defined or extracted) weights and inertia tensors as appropriate for the Model DOF.
- Group and target Load signals are in the inertial reference frame and taken with respect to the inertial frame origin. Linear forces will generate torques about the inertial origin that must be taken into account when targeting rotational axis loadings.
- *Mass*, *Link*, and *Point* specify the members of the Load.
  - *Mass* and *Link* lists support Element set references (see Section 3.3.3). *Point* lists support Aspect set references (see Section 3.3.5). Expanded member lists are generated in test run (Run `Class=T`) output Event files for verification.
  - A *Point* specifier that omits a Point ID defaults to the CG Point of the referenced Mass.
  - Signs can be specified for the Mass, Link, and Point contributions by placing `(+)` or `(-)` in the member lists with surrounding whitespace. The signs default to `(+)`. Specified signs apply to all members that follow until the next sign specifier, if any. For example, `Mass= (+) LeftFrameAssy.* (-) RightFrameAssy.*`.
  - The default `(+)` contribution for a Mass is its inertial loading, which has the sign of the *applied* load to generate its motion. A negative contribution has the sign of the load *exerted* by the Mass. The signs of all Masses in a Load are usually the same, but opposite signs could be used to obtain the difference between the inertial loading of two similar or symmetric substructures.
  - The default `(+)` contribution for a Link is the loading applied to its *Point2*. Use `(-)` to obtain the loading applied to its *Point1*.
  - In driving Loads the sign can be used to flip the sign of the applied loading. This can be used to apply opposite actuator-like forces to two Points.
- *Cutoff*, *ZeroSm*, and *EndSm* control the input loading signal filtering. See the SimFil documentation [7, 9] for more information on the filtered signal representation used in SISAME-3D.
  - *Cutoff* is the low-pass filtering cutoff frequency. *Cutoff* defaults to the enclosing Structure's *Cutoff*.
  - *ZeroSm* and *EndSm* are the force time-zero and end tail smoothing frequencies. *ZeroSm* and *EndSm* default to the enclosing Structure's corresponding Parameters and then to 2.5∗*Cutoff*. Tail smoothing of either tail can be suppressed by specifying a value of `N`.
- *ConF* is an RMS confidence band magnitude tuple for the force of target and comparison Loads. *ConF* is keyed by axis type, `l` for linear and `r` for rotational. *ConF:l* defaults to $10\,w$, where w is the Load's representative weight in force units. *ConF:r* defaults to $10\,w * 1\,\mathrm{m}$, which is equivalent to the *ConF:l* applied at a 1 meter moment arm. For a Load with Mass members, w is the sum of each Mass' weight or, for extracted or unspecified weight Masses, the Model's average specified

Mass weight (or the weight magnitude, $\tilde{w}$, if no such average weight is available) in force units. For a Load with only Link members w is the Model's $\tilde{w}$ in force units.

- *ConI* and *ConJ* are confidence band factors for the impulses and integrated impulses of target and comparison Loads. The weighting of these target equations is based on *ConF*. *ConI* and *ConJ* default to the enclosing Structure's corresponding Parameter and then to 1. *ConI* and/or *ConJ* can be set to `N` to suppress impulse and/or integrated impulse domain targeting (this will usually speed up the extraction). Decreasing *ConI* and/or *ConJ* will generally improve the extracted model's impulse and/or integrated impulse fit but at some cost to the force fit.
- *TarSignal*, *DrvSignal*, and *CmpSignal* specify the target, driving, or comparison input loading signals.
  - Only one type of signal specifier can be used in a Load: target, driving, or comparison. The signal type must be consistent with any specified Load Class.
  - Signal specifiers have the form *axis*`:`*file_name*. A specifier of the form *axis*`:0` can be used to specify a zero loading for an axis.
  - Multiple signals can be specified following each `Signal` tag or each signal can be specified in a separately tagged field.
  - Group and target loading signals must be in the inertial reference frame and with respect to the inertial frame origin.
  - Driving loading signals must be in inertial frame coordinates and are applied at and with respect to each driving Load Point's origin.
  - Loading signals for all or a subset of the DOF axes can be specified.
  - Force/torque, impulse, and integrated impulse time series signals are supported.

3.19 Output Information

Output Specifications

Tag	Description	Value	Default	N	?
`OutClass`	Output class	*see below*			
`Sub`	Subsampling rate	$\mathcal{I} \geq 1$	1		
`Qty`	Output quantity flags	*see below*			
`Mass`	Mass set list	$\mathcal{S}$	*none*		
`Link`	Link set list	$\mathcal{S}$	*none*		
`Load`	Load set list	$\mathcal{S}$	*none*		
`File`	File name for the output	$\mathcal{S}$	*see Notes*		

The `Output Information` template is repeated here with descriptions of each output class:

```
Output Information
  OutClass=MassTbl  Sub=        Mass=        Mass Tabular
  OutClass=LinkTbl  Sub=        Link=        Link Tabular
  OutClass=LoadTbl  Sub=        Load=        Load Tabular
  OutClass=MassTS   Qty=AVPavp  Mass=        Mass Time Series
  OutClass=LinkTS   Qty=XRSDFE  Link=        Link Time Series
  OutClass=LoadTS   Qty=FIJfij  Load=        Load Time Series
  OutClass=LinkXY   Qty=SDFE    Link=        Link Y vs. X History
  OutClass=LinkPar  Qty=SD      Link=        Link Parametric
  OutClass=FitRep   File=                    Fit Measure Report
  OutClass=Event    File=                    Event specifications
  OutClass=POV      File=                    POV-Ray model image
```

Output Specification Notes

- `OutClass` is the leading tag for an output specification.
- Output specifications can appear in any order.
- SISAME-3D scans the output specifications to check for errors before running the model.
- Test run (Run `Class=T`) outputs are limited to parametric Link, Event, and POV-Ray files. Extracted Link parametric files are generated using the initial values assigned to the extracted Parameters.
- Extraction runs perform a final simulation pass only if outputs other than parametric Link or model file are requested.
- Mass and Load outputs are in the inertial frame orientation and taken with respect to the inertial frame origin.

- Tabular outputs are text files with a column formatted display the relevant time series using the specified subsampling rate, if any. Tabular files are given names of the form: *Element*`.`*Axis*`.tbl.txt` .
- A Mass *`Qty`* contains any subset of the letters { `A`, `V`, `P`, `a`, `v`, `p` }. These request acceleration (A), velocity (V), and position (P) outputs. The upper case letters request the actual motion of the Mass. For a simulated Mass with a comparison motion, the lower case letters request the (filtered) comparison motions. For an instrumented Mass, the lower case letters requests the effective motions: the effective acceleration is computed from the net Link and driving Load force acting on the Mass' weight and intertia, and this is integrated to obtain the effective velocity and position. For target Masses, the actual and effective motions can be compared to assess the fit in the solution domain.
- A Link *`Qty`* contains any subset of the letters { `X`, `R`, `S`, `D`, `F`, `E` }, which request deflection, relative velocity, static force, dynamic force, force (total), and energy output, respectively. Only `S`, `D`, `F`, and `E` can be used in `LinkXY` outputs. Only `S` and `D` can be used in `LinkPar` outputs. Link dynamic force history outputs include the contributions of both Damper and Magnifier Aspects if both are present.
- A Load *`Qty`* contains any subset of the letters { `F`, `I`, `J`, `f`, `i`, `j` }. These request force (F), impulse (I), and integrated impulse (J) outputs. The upper case letters request group outputs for a group or target Load. The lower case letters request target, driving, or comparison (filtered) signal outputs. For a target Load the group outputs include the effective force of the extracted Components, and these can be compared to the filtered target signals to assess the solution fit in the solution domain. There are no group signals associated with a driving Load.
- The *`Mass`*, *`Link`*, and *`Load`* fields contain lists of Element and Element set references (see Sections 3.3.2 and 3.3.3). The local context for these references is the Model Structure.
- Multiple *`Qty`* and Element specifiers can be used within a single `OutClass` specifier. Each *`Qty`* applies to the subsequent Elements specified until the next *`Qty`*, if any.
- Extracted Link force and energy outputs (excluding parametric output) are determined by driving the extracted Links with the instrumented Mass motions, not by a resimulation of the model.
- `LinkPar` Static output files are in the form of static force vs. deflection points that give a representation of the Static Aspect characteristics. Failure deflections are not shown in these files.
- `LinkPar` Dynamic output files are in the form of damping force or magnification factor vs. relative velocity points that give a representation of the Dynamic Aspect characteristics. Dynamic Magnifier outputs show the $\sigma(f^S) > 0$ case.
- A `LinkPar` *`Qty`* of `D` produces files for both Damper and Magnifier Dynamic Aspects for Links with two Dynamic Aspects.
- The `LinkPar` *`Qty`* defaults to `S` if not specified.
- The `FitRep` request produces a fit measure report file showing the unweighted and weighted RMS difference between target Mass and Load filtered instrument and effective signals. Fit measures are also shown for non-target Masses and Loads that have comparison files, and for driven Masses with defined weight/inertia and adjacent Links or driving Loads. The fit reports do not include measures of fit to Parameter conditioning, Parameter damping, Static segment

smoothness, and user-specified estimate targets. The `FitRep` output file name defaults to the name of the input Event file but with a `.fit` extension.

- The `Event` request produces an output SISAME-3D Event file.
  - The output `Event` file name defaults to the same name as the input file but with an added `.out` type extension.
  - `Event` files contain *processed* versions of the Links, incorporating any transforms into the Link Parameters where appropriate.
  - The model file output from a test run (Run `Class=T`) is the processed input model including the extracted Parameter specifications. To obtain a model file with the initial solution Parameter values perform an extraction run with `MaxIter=N`.
  - Computed or extracted numeric values are formatted with about seven decimal digits of precision in output Event files.
  - Comments are not carried over to the output Event file.
- The `POV` request produces a POV-Ray image file of the Model that can be used to verify and document the Model configuration.
  - The POV file name defaults to the name of the input Event file but with a `.pov` extension.
  - The POV file shows the Masses and Fixed Masses in their resting positions. Their Shapes and Points appear with their default or specified size and color. The CG Points of Masses and Fixed Masses are shown even if they have no specified Shapes or Points. Links are shown as thin cylinders connecting their attachment Points. The Link cylinder is given a size based on the size of the Points and a default color.
  - POV files for 1D and 2D models are created to show an orthographic projection from an appropriate camera position. If the Model Shapes have non-DOF dimensions other meaningful views can be generated by altering the POV file. Non-zero Shape transparencies may be necessary to see some of the Shapes.
  - POV files for pure rotational models (with no linear DOFs) can be generated but will have all Shapes centered at the same origin point. Non-zero Shape transparencies may be necessary to see some of the Shapes.
  - Sections of the POV-Ray file are easily removed to show a subset of the Model parts. For example, the Links or Points can be removed to show only the Shapes. Or one portion of the Model can be isolated to create a detail view of a substructure.
  - The camera and lighting values in the POV file can be modified to create model images from other angles and close-up images.
  - The POV files z-axis coordinates are sign-reversed relative to the SISAME-3D Model to adjust to POV-Ray's left-hand coordinate system. POV-Ray also defines rotations about a fixed coordinate system, so rotations in the POV file will differ, in general, from the body-fixed rotations used in SISAME-3D.
- Output signal files are given names the form: *Element*.*Axis*.*DataType*.*FileType*, where *DataType* is a three character code for the data type as shown in the table below, such as `ats` for the acceleration time series of a simulated Mass, and *FileType* is the file type extension, such as `uds` for NHTSA UDS files. The log file shows the file names used for each output signal.

DataType Code Character Positions

	1		2		3	
Mass	a	acceleration	t	time series	s	simulated
	v	velocity			i	instrumented
	p	position			e	effective
					c	comparison
Link	x	deflection	t	time series	d	defined
	r	relative velocity	x	vs. deflection	e	extracted
	s	static force				
	d	dynamic force				
	f	force				
	e	energy				
Parametric Link	d	damper	p	parametric	d	defined
	m	magnifier			e	extracted
Load	f	force	t	time series	g	group
	i	impulse			i	instrumented
	j	integrated impulse			d	driving
					c	comparison

3.20 Multiple-Event Modeling

SISAME-3D supports multiple-Event modeling to provide extraction of robust models that are a balanced match to a set of Events having some Model Components in common. Events for different impact modes can be used to generate a multi-modal model. Events for different impact speeds can be used to obtain a better partition of the loads over extracted static and dynamic Parameters.

Different types of impacts involving different Vehicle combinations and speeds can be used in a multiple-Event run. Structures can be partially matched if they have some different Components: SISAME-3D automatically finds and equates the matching Model Components (see Section 3.20.1 below).

Multiple-Event modeling is conducted by providing SISAME-3D a Master input file that specifies the extraction controls and the Event files for each Event. The relative emphasis on the Events can be controlled by weighting Parameters.

3.20.1 Event Matching

SISAME-3D finds the matching Structures, Masses, and Links of each Event's Model and equates their corresponding extracted Parameters. The Component ID and essential structural Parameters must agree for a match to occur. Other Parameters, such as those describing the Event (e.g., initial velocity), filtering, and target confidence bands need not agree between matched Components.

Structure matching requirements are fairly loose. Matching Vehicles must have the same ID but can have different weights, makes, models, model years, and Components. Matching Substructures must have the same ID but can have different weights and Components. Matching Structures with different weight values that contain matching extracted-weight Masses may not have feasible weight solutions. Within matching Structures Mass and Link matching is performed.

Mass matching requires only that the IDs and weight Parameters match. Masses of different Class (Simulated, Target, or Driven) can match each other.

Link matching requires that the IDs and the Static and Dynamic behavioral Parameters match (Parameters not affecting the Link simulation behavior, such as `ConSS`, need not match). The IDs of the attachment Points can differ between the Events (a warning message will be produced). Inactive Links are not matched.

Links are **profile matched** before references are resolved to support some referencing capabilities that depend on the Link deflection and relative velocity ranges. This is a preliminary match that looks only at the Link IDs and the types of Static and Dynamic Aspects. If two

Links are profile matched but fail to fully match after references are resolved an error occurs. If the match is not desired one of the Link IDs can be changed to prevent profile matching. If the match is desired then the Link specifications must be adjusted to agree.

Static SE and SI Aspects in matched Links that have automatic tail deflections will automatically use a common deflection range that is based on the union of the ranges for all of the matched Events.

Fixed Mass and Load Elements are not matched since they cannot contain extracted Parameters.

Parameter matching requires that both Parameters have the same value (after any transforms are applied) or that both are extracted or both suppressed. Parameters that reference an extracted Parameter can match reference or non-reference extracted Parameters, and if both are references the reference targets need not be matching Parameters or even present in both Events. Two extracted Parameters that were not linked within their Event can become linked by the Event matching process if they are related by a reference in a matched Event. Bounds and estimates need not agree between matching extracted Parameters but conflicting bounds will cause the extraction to terminate with a "No feasible solutions" error.

The Event Matching Report in the Master log file should be checked to assure the desired Components have been matched. Only name-matched Components within each Model are shown in the Event Matching Report. Comments document differences between matched Components or reasons that Components with matching IDs were not matched. Any undesired matches can be suppressed by changing the Component's ID in one of the Events.

3.20.2 Master File Overview

Figure 3.2 presents a template for the SISAME-3D Master file. This template is distributed with SISAME-3D to aid in creating Master files.

3.20.3 Master File Organization

- The file header `SISAME-3D Master File` and the section headings `Run Information` and `Event Information` are required and must appear on separate lines in that order.
- The `Run Information` section contains specifications that describe and control the SISAME-3D run. The Run specifications section begins with the `RunID` field. The Run Parameters in the Master file control the extraction: the corresponding Parameters in the Event files are ignored in a multiple-Event run.
- The `Event Information` section specifies the Events to be modeled. Each Event specification begins with an `EventID` field.

```
SISAME-3D Master File


Run Information

  RunID=  Descr=  Class=
    Title=
    MaxIter=  ConPC=  ConPD=  MultPD=  Relax=  IterCon=  ConvTol=


Event Information

  EventID=  Descr=  File=  Wtg=


Comments
```

Figure 3.2: SISAME-3D Master file template.

- The tags `RunID` and `EventID` are **leading tags** that begin specification sections of the Master file and must be the first tag on a line. The other fields in each section can appear in any order and on as many lines as needed. The *`RunID`* and the *`EventID`*s must be case insensitively distinct.
- Any number of Events can be specified.
- Blank lines can be used to separate portions of the Master file.
- An exclamation point (`!`) character (outside of a quoted string) causes the rest of the line to be treated as a comment.
- `Comment=`*comment* fields can be included in any section of the Master file.
- Everything below the optional `Comments` heading is ignored.
- The format and rules for Master file fields are the same as those for Event files (see Section 3.6.4).

3.20.4 Run Information

The `Run Information` section is the first section of the SISAME-3D Master file and contains the specifications that describe and control the multiple-Event run.

Master Run Specifications

Tag	Description	Value	Default	N
`RunID`	Run ID	$\mathcal{S} \leq 20$		
`Descr`	Description	$\mathcal{S}$		
`Class`	Run class	$\mathcal{C} =$ `S`, `E`, `T`	*see Notes*	
`Title`	Run title	$\mathcal{S}$		
`MaxIter`	Max number of extraction iterations	$\mathcal{I} \geq 0$	50	`N`
`ConPC`	Parameter conditioning confidence band factor	$\mathcal{R} > 0$	1	`N`
`ConPD`	Parameter damping confidence band factor	$\mathcal{R} > 0$	1	`N`
`MultPD`	Parameter damping multiplier	$\mathcal{R} > 0$	0.5	`N`
`Relax`	Iteration relaxation control	$\mathcal{B}$	`True`	
`IterCon`	Iteration constraint control	$\mathcal{B}$	`True`	
`ConvTol`	Convergence tolerance control factor	$\mathcal{R} > 0$	1	

Master Run Specifications Notes

- `RunID` is the leading tag for the Run specifications. The *`RunID`* appears in outputs to identify the run.
- The Run *`Class`* defaults to `S` for a simulation run and `E` for an extraction run (when extracted Parameters are present). Setting `Class=T` requests a test run, in which no simulation or extraction is performed, but in which the Master and Event files will be processed and the model summary, model file, and parametric Link outputs can be examined.
- *`MaxIter`* is the maximum number of extraction iterations SISAME-3D can use to converge on a self-consistent solution when Model features require iteration. `MaxIter=0` produces a single pass extraction. `MaxIter=N` suppresses all extraction passes and causes run outputs to use the initial values generated by SISAME-3D for the extracted Parameters.
- *`ConPC`* is the confidence band factor for the Parameter conditioning target equations. These confidence bands are of the form *`ConPC`* $\cdot 10^5 \cdot \bar{p}$ where $\bar{p}$ is a baseline magnitude for the type and units of the specific Parameter. A smaller *`ConPC`* will generally improve numerical conditioning of the extraction computations, possibly at some cost to the solution quality. Setting `ConPC=N` suppresses Parameter conditioning. If the Event Models have extracted Parameters that are never active, or other singular or ill-conditioned features, Parameter conditioning is required to avoid a singular or ill-conditioned least squares system.

- *ConPD* is a confidence band factor for the Parameter damping target equations, which target iterative solutions to the previous solution to damp large Parameter changes and speed convergence. Parameter damping begins after ($MaxIter/2$) iterations or when solution cycling is detected. These confidence bands are of the form $ConPD \cdot 10^3 \cdot \bar{p}$ where $\bar{p}$ is a baseline magnitude for the type and units of the specific Parameter. A smaller *ConPD* will facilitate convergence of difficult iterative extractions. Setting `ConPD=N` suppresses Parameter damping.
- *MultPD* is a multiplier applied cumulatively to *ConPD* on each iteration pass to speed solution convergence (with $MultPD \leq 1$). A smaller *MultPD* will facilitate convergence of difficult iterative extractions. Setting `MultPD=N` suppresses changes to *ConPD*.
- *Relax* controls whether relaxation is enabled during extraction iterations. Relaxation limits the change in the Parameter solutions between iterations in an effort to speed convergence. By default *Relax* is `True` and SISAME-3D will begin and gradually increase the amount of relaxation after ($MaxIter/2$) iterations. No relaxation occurs if $MaxIter < 3$. Relaxation is also used if solution cycling is detected regardless of *Relax*.
- *IterCon* controls whether iteration constraints are enabled during extraction. Iteration constraints limit the solution change in each iteration to control the error of the linearized approximants used to represent nonlinear Parameter contributions. Iteration constraints prevent solution overshooting/oscillation and may be needed to assure convergence, and thus should generally be enabled. By default *IterCon* is `True` enabling the iteration constraints.
- *ConvTol* is a factor that controls the extraction iteration convergence stopping criteria tolerances. A smaller *ConvTol* causes tighter convergence criteria to be used.
- The Master Run Parameters control the multiple-Event run: the corresponding Run Parameters in the Event files are ignored.

3.20.5 Event Information

The `Event Information` section contains the specifications for each Event.

3.20.5.1 Event Specifications

Event Specifications

Tag	Description	Value	Default	N
`EventID`	Event ID	$\mathcal{S} \leq 20$		
`Descr`	Description	$\mathcal{S}$		
`File`	SISAME-3D Event file	$\mathcal{S}$		
`Wtg`	Event weighting factor	$\mathcal{R} > 0$	1	

Event Specification Notes

- `EventID` is the leading tag for the Event specifications.
- SISAME-3D Event files with relative path names are taken to be relative to the SISAME-3D Master file location, allowing SISAME-3D to be run from a directory other than that containing the Master file.
- SISAME-3D Event file name paths are automatically translated to the local operating system format, where possible, to allow the same SISAME-3D Master file to be used on different computer platforms. The use of relative, rather than absolute, paths facilitates this portability.
- The SISAME-3D Event files need not be located in their Event's *`EventID`* output directory but relative driving Load and driven Mass signal file paths in the output model files remain relative to the SISAME-3D Event file path and thus such models cannot be simulated from the Event output directory.
- *`Wtg`* is the Event weighting factor. Only the relative, not absolute, weighting of each Event determines its emphasis in the extraction.

4 SISAME-3D Formulation

The formulation of the SISAME-3D modeling methodology is presented below. First, the target equations that are the basis of model extraction are presented. This is followed by a description of the construction of the least squares problem for the optimal model. The solution of this optimization problem and other aspects of the implementation are described in Section 5.

4.1 Rigid Mass Equations of Motion

The Newton-Euler equations of motion for a rigid body (Mass) equate the applied and inertial generalized forces in inertial reference frame coordinates taken with respect to the body's center of gravity (CG) at each instant [13]. The **generalized force** is the vector of linear and rotational (torque) forces corresponding to the degrees of freedom. Letting $\mathbf{f}^{(A)}$ be the generalized **applied force** and $\mathbf{f}^{(I)}$ be the generalized **inertial force**, the generalized equations of motion for a rigid body, at their simplest, are given by

$$\mathbf{f}^{(A)} = \mathbf{f}^{(I)} \tag{4.1}$$

In SISAME-3D, the applied force on a Mass can include Link, driving Load, and gravity forces

$$\mathbf{f}^{(A)} = \mathbf{\Phi}^{(L)}\,\mathbf{f}^{(L)} + \mathbf{\Phi}^{(D)}\,\mathbf{f}^{(D)} + \mathbf{f}^{(G)} \tag{4.2}$$

where (L) is used for Links, (D) for driving Loads, and (G) for gravity, and the $\mathbf{\Phi}$ are instantaneous **influence matrices** that convert the Link and Load forces applied at various points on the body to the equivalent CG force. The influence matrices change over time as the points move relative to the CG with respect to the inertial frame and the geometry of the Links is altered by the relative movement of their attachment points. The gravity force acts as a constant linear force on the body's CG point so it needs no influence matrix.

The inertial force of a rigid body is given by

$$\mathbf{f}^{(I)} = \begin{pmatrix} m\,\mathbf{a} \\ \mathbf{I}\,\boldsymbol{\alpha} + \boldsymbol{\omega} \times \mathbf{I}\,\boldsymbol{\omega} \end{pmatrix} = \frac{d}{dt}\begin{pmatrix} m\,\mathbf{v} \\ \mathbf{I}\,\boldsymbol{\omega} \end{pmatrix} \tag{4.3}$$

where m is the body's mass, $\mathbf{I}$ is its inertia tensor, $\mathbf{a}$ is its linear acceleration vector, $\mathbf{v}$ is its linear velocity vector, $\boldsymbol{\alpha}$ is its angular acceleration vector, and $\boldsymbol{\omega}$ is its angular velocity

vector. The second representation is the time derivative of the generalized momentum. The inertia tensor and all motion vectors are taken with respect to an inertial frame at the body CG. Although the inertia tensor with respect to a body-fixed frame is constant, the inertia tensor changes with respect to the inertial frame as the body rotates.

4.2 Partitioned Equations of Motion

SISAME-3D Models can include simulated, target, driven, and constant-velocity (c-v) Masses. Driven and constant-velocity Mass motions can affect extractions and simulations by any Links connecting them to other Masses and by any target Loads in which they are members, but their equations of motion are not used for extraction or simulation.

The SISAME-3D extraction process is best explained by forming partitioned generalized equations of motion that distinguish simulated and target Masses and defined and extracted Links. The equations of motion for all of the simulated and target Masses in a Model can be partitioned into

$$\begin{pmatrix} \mathbf{f}_S^{(A)} \\ \mathbf{f}_T^{(A)} \end{pmatrix} = \begin{bmatrix} \mathbf{\Phi}_{SD}^{(L)} & \mathbf{\Phi}_{SE}^{(L)} \\ \mathbf{\Phi}_{TD}^{(L)} & \mathbf{\Phi}_{TE}^{(L)} \end{bmatrix} \begin{pmatrix} \mathbf{f}_D^{(L)} \\ \mathbf{f}_E^{(L)} \end{pmatrix} + \begin{pmatrix} \mathbf{\Phi}_S^{(D)} \, \mathbf{f}_S^{(D)} \\ \mathbf{\Phi}_T^{(D)} \, \mathbf{f}_T^{(D)} \end{pmatrix} + \begin{pmatrix} \mathbf{f}_S^{(G)} \\ \mathbf{f}_T^{(G)} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_S^{(I)} \\ \mathbf{f}_T^{(I)} \end{pmatrix} \tag{4.4}$$

where $\mathbf{f}^{(A)}$ and $\mathbf{f}^{(I)}$ now include all the Masses. The upper portion corresponds to the simulated Masses ($\square_S$) and the lower portion to the target Masses ($\square_T$), and the left half of $\mathbf{\Phi}^{(L)}$ corresponds to the defined Links ($\square_D$) and the right half to the extracted Links ($\square_E$).

Considering the equations of motion for the simulated Masses, at each instant, given generalized positions and velocities of all Masses, all the constituents of the applied force are computable except the extracted Link force, $\mathbf{f}_E^{(L)}$: the global approach used in SISAME-3D does not produce the force in the extracted Links until after an extraction process that considers the entire event time span. For tractability the contribution of $\mathbf{f}_E^{(L)}$ to the simulated Mass applied force is eliminated by requiring $\mathbf{\Phi}_{SE}^{(L)} = \mathbf{0}$, which is equivalent to the rule that extracted Links cannot be connected to simulated Masses. The generalized inertial force velocity/momentum state is also known since simulated Masses must have defined mass and inertia tensors. The simulated Mass equations of motion can then be numerically integrated in an extraction run just as they are during simulation.

The generalized equations of motion for the target Masses contain a mix of known and unknown quantities at each instant. The defined Link driving Load, and gravity forces are known as are the generalized motion vectors in the inertial force. The generalized extracted Link force and any extracted mass or inertia tensor values are unknown.

If the generalized target Mass equations are rewritten as

$$\mathbf{\Phi}_{TE}^{(L)} \, \mathbf{f}_E^{(L)} \;=\; \mathbf{f}_T^{(I)} \;-\; \mathbf{\Phi}_{TD}^{(L)} \, \mathbf{f}_D^{(L)} \;-\; \mathbf{\Phi}_T^{(D)} \, \mathbf{f}_T^{(D)} \;-\; \mathbf{f}_T^{(G)} \tag{4.5}$$

it can be seen that if the mass and inertia tensor are defined this would express the instantaneous extracted Link force in terms of known quantities. If $\mathbf{\Phi}_{TE}^{(L)}$ was square and non-singular this system could be solved for the generalized extracted Link force at each instant. This method was used in the Fiat methodology [2, 3, 6] but there are serious drawbacks with such an approach. To get a square, non-singular $\mathbf{\Phi}_{TE}^{(L)}$ requires an equal number of extracted Links and target Masses in a loop-free configuration, which is not appropriate for modeling structures with multiple load-paths such as vehicles. Also, the resulting Link deflection–force histories are not likely to correspond to physically meaningful load-paths.

Instead, the SISAME-3D extraction process combines the target Mass equations of motion for the entire event into a single system that can then be solved for the optimal Model. This allows Models to have any configuration, including those with more extracted Links than target Masses. The extracted Link properties must be properly parameterized to assure a tractable computation that produces physically meaningful, simulation-ready Models: the parameterization and solution of this system of equations is explored in the following sections.

4.3 Mass Target Equations

The generalized instantaneous Link force can be expressed as a combination of static and dynamic contributions

$$\mathbf{f}^{(L)} \;=\; \mathbf{D}\,\mathbf{s}^{(L)} \,+\, \boldsymbol{\delta} \tag{4.6}$$

where $\mathbf{D}$ is the diagonal matrix of generalized dynamic magnifiers, $\mathbf{s}^{(L)}$ is the generalized static Link force, and $\boldsymbol{\delta}$ is the generalized damping force vector. To obtain an efficiently solvable extraction problem, the Link force is then parameterized so that the instantaneous Link force is linear in the Parameters (see Section 5.2) giving

$$\mathbf{f}^{(L)} \;=\; \mathbf{\Pi}^{(L)}\,\mathbf{p}^{(L)} \tag{4.7}$$

where $\mathbf{\Pi}^{(L)}$ is the generalized Link Parameter force influence matrix and $\mathbf{p}^{(L)}$ is the vector of static and dynamic Link Parameters. Using this parameterized Link force, (4.2) can be expressed as

$$\mathbf{f}^{(A)} \;=\; \mathbf{\Psi}^{(L)}\,\mathbf{p}^{(L)} \,+\, \mathbf{\Phi}^{(D)}\,\mathbf{f}^{(D)} \,+\, \mathbf{f}^{(G)} \tag{4.8}$$

where $\mathbf{\Psi}^{(L)} = \mathbf{\Phi}^{(L)}\,\mathbf{\Pi}^{(L)}$ is the Link Parameter Mass force influence matrix.

For simulated Masses all of the values are known at each instant (recall $\mathbf{\Phi}_{SE}^{(L)} = \mathbf{0}$) so this can be used to compute the generalized applied force needed to numerically integrate their motions.

For target Masses there can be contributions to the generalized applied force from both defined and extracted parameters, so

$$\mathbf{f}_{T}^{(A)} \;=\; \mathbf{\Psi}_{TD}^{(L)}\,\mathbf{p}_{D}^{(L)} \,+\, \mathbf{\Psi}_{TE}^{(L)}\,\mathbf{p}_{E}^{(L)} \,+\, \mathbf{\Phi}_{T}^{(D)}\,\mathbf{f}_{T}^{(D)} \,+\, \mathbf{f}_{T}^{(G)} \tag{4.9}$$

where $\mathbf{\Psi}^{(L)}$ has been partitioned to separate the defined and extracted Link Parameters. Allowing for extracted mass and inertia tensor, the generalized target Mass inertial force can also be parameterized as

$$\mathbf{f}_T^{(I)} \;=\; \mathbf{\Phi}_{TD}^{(I)}\,\mathbf{p}_D^{(I)} \;+\; \mathbf{\Phi}_{TE}^{(I)}\,\mathbf{p}_E^{(I)} \tag{4.10}$$

where the $\mathbf{\Phi}^{(I)}$ are the instantaneous inertial Parameter force influence matrices and the $\mathbf{p}^{(I)}$ are the inertial (mass and inertia) Parameters.

The instantaneous generalized target Mass equations of motion can then be expressed as

$$\mathbf{f}_T^{(A)} \;=\; \mathbf{\Psi}_{TD}^{(L)}\,\mathbf{p}_D^{(L)} \;+\; \mathbf{\Psi}_{TE}^{(L)}\,\mathbf{p}_E^{(L)} \;+\; \mathbf{\Phi}_T^{(D)}\,\mathbf{f}_T^{(D)} \;+\; \mathbf{f}_T^{(G)} \;=\; \mathbf{\Phi}_{TD}^{(I)}\,\mathbf{p}_D^{(I)} \;+\; \mathbf{\Phi}_{TE}^{(I)}\,\mathbf{p}_E^{(I)} \;=\; \mathbf{f}_T^{(I)} \tag{4.11}$$

or

$$\mathbf{\Psi}_{TE}^{(L)}\,\mathbf{p}_E^{(L)} \;-\; \mathbf{\Phi}_{TE}^{(I)}\,\mathbf{p}_E^{(I)} \;=\; \mathbf{\Phi}_{TD}^{(I)}\,\mathbf{p}_D^{(I)} \;-\; \mathbf{\Psi}_{TD}^{(L)}\,\mathbf{p}_D^{(L)} \;-\; \mathbf{\Phi}_T^{(D)}\,\mathbf{f}_T^{(D)} \;-\; \mathbf{f}_T^{(G)} \tag{4.12}$$

where the extracted Parameter contributions have been grouped on the left hand side.

Letting

$$\mathbf{\Gamma}_T^{(M)} \;=\; \begin{bmatrix} \mathbf{\Psi}_{TE}^{(L)} & -\mathbf{\Phi}_{TE}^{(I)} \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{\gamma}_T^{(M)} \;=\; \mathbf{\Phi}_{TD}^{(I)}\,\mathbf{p}_D^{(I)} \;-\; \mathbf{\Psi}_{TD}^{(L)}\,\mathbf{p}_D^{(L)} \;-\; \mathbf{\Phi}_T^{(D)}\,\mathbf{f}_T^{(D)} \;-\; \mathbf{f}_T^{(G)}$$

the instantaneous target Mass equations can be expressed

$$\mathbf{\Gamma}_T^{(M)}\,\mathbf{p}_E \;=\; \boldsymbol{\gamma}_T^{(M)} \tag{4.13}$$

where $\mathbf{p}_E = [\,\mathbf{p}_E^{(L)} \;\; \mathbf{p}_E^{(I)}\,]^T$.

The generalized target Mass equations of motion for each time step will be combined into a large **Mass target system**, which can be solved for the extracted Parameters, $\mathbf{p}_E$, that best satisfy the set of equations of motion. Since this becomes an overdetermined system that is solved approximately, the instantaneous target system is better expressed

$$\mathbf{\Gamma}_T^{(M)}\,\mathbf{p}_E \;\approx\; \boldsymbol{\gamma}_T^{(M)}\,. \tag{4.14}$$

Other equations are usually included before this is solved for the optimal Parameters.

A solution, $\mathbf{p}_E$, that provides a good fit to these generalized force–acceleration domain target equations does not assure that the resulting Model's simulations will not drift in the generalized velocity and position domains, since the acceleration errors may not have near-zero first and second integrals. The Model's simulation accuracy can be improved by including target systems for the generalized velocity and position domains. This is achieved by target equations that are the first and second integral over time of (4.14)

$$\left[\int_t \mathbf{\Gamma}_T^{(M)}\right]\mathbf{p}_E \;\approx\; \int_t \boldsymbol{\gamma}_T^{(M)} \qquad \text{and} \qquad \left[\int_t\!\int_t \mathbf{\Gamma}_T^{(M)}\right]\mathbf{p}_E \;\approx\; \int_t\!\int_t \boldsymbol{\gamma}_T^{(M)}\,. \tag{4.15}$$

These generalized **velocity and position target systems** are actually in generalized impulse and integrated impulse units since the acceleration domain target system is in units of

generalized force. Note that the rotational position target equations are targeting the integral of the angular velocity, which is not a valid representation of the orientation, when multiple rotational degrees of freedom are present: the use of these targets is efficient and has the desired effect of reducing errors in the orientation.

Extending (4.14) with the generalized velocity and position target systems gives the full Mass target system

$$\hat{\mathbf{\Gamma}}_T^{(M)}\,\mathbf{p}_E \;\approx\; \hat{\boldsymbol{\gamma}}_T^{(M)}\,. \tag{4.16}$$

4.4 Load Target Equations

Additional target equations are created for each target Load at each instant. Target Loads can include generalized Mass inertial and Link forces, each of which is linear in the Model Parameters. The target Load equations are of the form

$$\mathbf{f}^{(\mathcal{L})} = \mathbf{\Phi}^{(\mathcal{L})}\,\mathbf{p} \;\approx\; \boldsymbol{f}^{(\mathcal{L})} \tag{4.17}$$

where $\mathbf{f}^{(\mathcal{L})}$ is the vector of generalized target Load contribution forces, $\mathbf{\Phi}^{(\mathcal{L})}$ is the Parameter target Load influence matrix, and $\boldsymbol{f}^{(\mathcal{L})}$ is the target Load signal force vector at that instant.

Grouping the extracted Parameters on the left gives the target Load system

$$\mathbf{\Phi}_T^{(\mathcal{L})}\,\mathbf{p}_E \;\approx\; \boldsymbol{f}^{(\mathcal{L})} \;-\; \mathbf{\Phi}_D^{(\mathcal{L})}\,\mathbf{p}_D \tag{4.18}$$

and letting

$$\mathbf{\Gamma}^{(\mathcal{L})} \;=\; [\,\mathbf{\Phi}_T^{(\mathcal{L})}\,] \qquad \text{and} \qquad \boldsymbol{\gamma}^{(\mathcal{L})} \;=\; \boldsymbol{f}^{(\mathcal{L})} \;-\; \mathbf{\Phi}_D^{(\mathcal{L})}\,\mathbf{p}_D$$

the instantaneous target Load equations can be expressed

$$\mathbf{\Gamma}^{(\mathcal{L})}\,\mathbf{p}_E \;=\; \boldsymbol{\gamma}^{(\mathcal{L})}\,. \tag{4.19}$$

As with the target Mass velocity and position target equations, it is generally beneficial to include target equations that are the first and second integral over time of (4.19)

$$\left[\int_t \mathbf{\Gamma}^{(\mathcal{L})}\right]\mathbf{p}_E \;\approx\; \int_t \boldsymbol{\gamma}^{(\mathcal{L})} \qquad \text{and} \qquad \left[\int_t\!\int_t \mathbf{\Gamma}^{(\mathcal{L})}\right]\mathbf{p}_E \;\approx\; \int_t\!\int_t \boldsymbol{\gamma}^{(\mathcal{L})}\,. \tag{4.20}$$

Extending (4.19) with the impulse and integrated impulse target systems gives the full Load target system

$$\hat{\mathbf{\Gamma}}^{(\mathcal{L})}\,\mathbf{p}_E \;\approx\; \hat{\boldsymbol{\gamma}}^{(\mathcal{L})}\,. \tag{4.21}$$

4.5 Weighted Target System

Combining the rows of the Mass and Load target equations together, the instantaneous target system can be expressed as

$$\mathbf{\Gamma}\,\mathbf{p}_E \;\approx\; \boldsymbol{\gamma}\,. \tag{4.22}$$

Supplementary target equations (see Section 5.5) are included, for example to specify estimated values for some of the Parameters. Let these other target equations be denoted by $\mathbf{\Psi}\,\mathbf{p}_E \approx \boldsymbol{\psi}$.

If time steps $0, \ldots, N$ are used to generate the instantaneous target equations, the complete target system is

$$\begin{bmatrix} \mathbf{\Gamma}^{(0)} \\ \mathbf{\Gamma}^{(1)} \\ \vdots \\ \mathbf{\Gamma}^{(N)} \\ \mathbf{\Psi} \end{bmatrix} \mathbf{p}_E \;\approx\; \begin{bmatrix} \boldsymbol{\gamma}^{(0)} \\ \boldsymbol{\gamma}^{(1)} \\ \vdots \\ \boldsymbol{\gamma}^{(N)} \\ \boldsymbol{\psi} \end{bmatrix} \tag{4.23}$$

which is written more concisely as

$$\mathbf{\Upsilon}\,\mathbf{p}_E \;\approx\; \boldsymbol{\upsilon}\,. \tag{4.24}$$

In practice $\mathbf{\Upsilon}$ has many more rows than columns so the Parameters are overdetermined (barring singularity) and the target equations can only be approximately satisfied.

To obtain a reasonable approximate solution, the target equations are weighted so that an error of one unit in each target equation is of equal importance. This requires pre-multiplying by a weighting matrix, $\mathbf{\Omega}$.

$$\mathbf{\Omega}\,\mathbf{\Upsilon}\,\mathbf{p}_E \;\approx\; \mathbf{\Omega}\,\boldsymbol{\upsilon} \tag{4.25}$$

or

$$\mathbf{T}\,\mathbf{p}_E \;\approx\; \boldsymbol{\tau} \tag{4.26}$$

with the natural correspondence. $\mathbf{T}$ is the weighted target matrix and $\boldsymbol{\tau}$ is the weighted target vector. In practice $\mathbf{\Omega}$ is diagonal; no coupling between the equation errors is considered.

The complete weighted target system is an overdetermined set of equations, the optimal solution of which gives the desired model. By considering all of the target information together, SISAME-3D can find the globally optimal model Parameters, making the most use of the often limited information available.

To solve for the unknown Parameters SISAME-3D first time-steps through the event, numerically integrating for the motions of simulated Masses and accumulating the target equation information, and then the target information is solved, subject to constraints on the Parameters, for the optimal model. The structure and solution of this optimization problem is treated below.

4.6 Constrained Approximation Problem

To optimally solve an overdetermined system first requires selecting the error norm to be minimized. SISAME-3D uses the least squares (L_2) error norm for the weighted target equations as the basis for the optimization. The choice of this error measure is driven primarily by the fact that least squares problems are tractable and readily solved. Alternatives such as an L_1 error norm may also be practical.

The L_2 error of the target equations is

$$\begin{aligned} e &= \|\mathbf{T}\,\mathbf{p}_E - \boldsymbol{\tau}\|_2^2 \\ &= \mathbf{p}_E^T\,\mathbf{T}^T\,\mathbf{T}\,\mathbf{p}_E \;-\; 2\,\mathbf{p}_E^T\,\mathbf{T}^T\,\boldsymbol{\tau} \;+\; \boldsymbol{\tau}^T\boldsymbol{\tau}\,. \end{aligned} \tag{4.27}$$

Letting

$$\mathbf{G} \;=\; \mathbf{T}^T\,\mathbf{T} \qquad \text{and} \qquad \mathbf{h} \;=\; \mathbf{T}^T\boldsymbol{\tau} \qquad \text{and} \qquad b \;=\; \boldsymbol{\tau}^T\boldsymbol{\tau}$$

this becomes

$$e \;=\; \mathbf{p}_E^T\,\mathbf{G}\,\mathbf{p}_E \;-\; 2\,\mathbf{p}_E^T\mathbf{h} \;+\; b \tag{4.28}$$

where $\mathbf{G}$ is symmetric, positive semidefinite with dimensions $P \times P$.

The gradient of the error is

$$\nabla e \;=\; \mathbf{2}\,\mathbf{G}\,\mathbf{p}_E - \mathbf{2}\,\mathbf{h} \tag{4.29}$$

where $\mathbf{2}$ is the diagonal matrix with entries of 2. The optimal *unconstrained* Parameters are obtained by setting $\nabla e = \mathbf{0}$, giving $\mathbf{p}_E = \mathbf{G}^{-1}\,\mathbf{h}$. The system $\nabla e = \mathbf{0}$ is known as the **normal equations**.

Constraints are used in SISAME-3D to enforce valid behavior of the Links and to satisfy Parameter bounds and specified Structure weights for the extracted Mass weights. Suppose the equality constraints are given by $\mathbf{C}_=\,\mathbf{p}_E = \mathbf{c}_=$ and the inequality constraints by $\mathbf{C}\,\mathbf{p}_E \le \mathbf{c}$. To incorporate the constraints in the solution process the Lagrangian is formed

$$\ell \;=\; e \;+\; \boldsymbol{\lambda}^T\,(\mathbf{C}_=\,\mathbf{p}_E - \mathbf{c}_=) \;+\; \boldsymbol{\mu}^T\,(\mathbf{C}\,\mathbf{p}_E - \mathbf{c}) \tag{4.30}$$

where $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are the vectors of Lagrange multipliers associated with the equality and inequality constraints, respectively. The necessary and sufficient conditions [5] for the globally optimal Parameters are

$$\begin{aligned} \nabla_{\mathbf{p}_E}\,\ell \;=\; \nabla e \;+\; \mathbf{C}_=^T\,\boldsymbol{\lambda} \;+\; \mathbf{C}^T\,\boldsymbol{\mu} &= \mathbf{0} \\ \mathbf{C}_=\,\mathbf{p}_E &= \mathbf{c}_= \\ \mathbf{C}\,\mathbf{p}_E &\le \mathbf{c} \\ \boldsymbol{\mu} &\ge \mathbf{0} \\ \boldsymbol{\mu}^T\,(\mathbf{C}\,\mathbf{p}_E - \mathbf{c}) &= 0\,. \end{aligned} \tag{4.31}$$

This is a Linear Complementary Programming (LCP) [14] problem that can be solved by techniques based on the simplex method for Linear Programming (see Section 5.1).

5 SISAME-3D Implementation

5.1 LCP Solution

The Linear Complementary Programming (LCP) optimality conditions (4.31) can be solved directly by standard LU factorization techniques. The instantaneous contributions to the $\mathbf{G}$ matrix and $\mathbf{h}$ vector can be accumulated during the time stepping procedure, so the very large $\mathbf{T}$ matrix is never explicitly formed.

Finding the optimal solution is made difficult by the need to determine which inequality constraints are active equalities at the optimum point. For activity set A the active inequality constraints become $\mathbf{C}_A\mathbf{p}_E = \mathbf{c}_A$ and the inactive constraint Lagrange multipliers are set to $\boldsymbol{\mu}_I = 0$ where the inactivity set is I. The optimality conditions for the tentative activity set A reduce to

$$\begin{bmatrix} \mathbf{G} & \mathbf{C}_=^T & \mathbf{C}_A^T \\ \mathbf{C}_= & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_A & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{pmatrix} \mathbf{p}_E \\ \mathbf{2\lambda} \\ \mathbf{2\mu}_A \end{pmatrix} = \begin{pmatrix} \mathbf{h} \\ \mathbf{c}_= \\ \mathbf{c}_A \end{pmatrix} \tag{5.1}$$

with the additional conditions

$$\begin{aligned} \mathbf{C}_I\,\mathbf{p}_E &\leq \mathbf{c}_I \\ \boldsymbol{\mu}_A &\geq \mathbf{0}\,. \end{aligned} \tag{5.2}$$

Thus an activity set is optimal if the inactive constraints are satisfied and the active multipliers are nonnegative. These are primal and dual feasibility requirements, respectively. Testing for optimality is performed by solving the system in (5.1) and checking the conditions in (5.2).

Procedures for finding the (globally) optimal solution are similar to the simplex method for Linear Programming [14] except that the number of active constraints is not fixed at the number of variables. The method used in SISAME-3D pivots between active constraint sets while maintaining a primal feasible solution that is (barring degeneracy) strictly improving. At each stage a feasible point on the current active constraint set is maintained. If an inactive constraint is violated by the solution of (5.1), the current working feasible point can be moved towards the solution until the first inactive constraint becomes a tight equality. It is not hard to show that the error measure at the new feasible point is reduced. If the

solution violates no inactive constraints the feasible point can be updated to this new point. An active inequality constraint with a negative multiplier can also be dropped from the active set at any stage. It can be shown that the next solution of (5.1) will have a smaller error measure and will not violate the dropped constraint. When the solution satisfies both conditions in (5.2) the global optimum has been found.

While much more efficient than exhaustively trying all possible active sets, this process can still require a large number of pivots in models of modest complexity. Efficient computation of the $\mathbf{p}_E$ and $\boldsymbol{\mu}_A$ values is therefore critical to the efficiency of the extraction process.

SISAME-3D uses a partial LU factorization from the previous pivot in computing the solution. The partial factorization for active set A is

$$\begin{bmatrix} \mathbf{P\,S} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \left[\begin{array}{ccc|c} \mathbf{G} & \mathbf{C}_=^T & \mathbf{C}_A^T & \mathbf{h} \\ \mathbf{C}_= & \mathbf{0} & \mathbf{0} & \mathbf{c}_= \\ \mathbf{C}_A & \mathbf{0} & \mathbf{0} & \mathbf{c}_A \end{array}\right] = \begin{bmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{Y}_A & \mathbf{I} \end{bmatrix} \left[\begin{array}{cc|c} \mathbf{U} & \hat{\mathbf{C}}_A^T & \hat{\mathbf{h}} \\ \mathbf{0} & \mathbf{Z}_A & \hat{\mathbf{c}}_A \end{array}\right] \tag{5.3}$$

where $\mathbf{PS}$ is a permutation and scaling matrix to allow for row scaling and partial pivoting during the reduction of $\mathbf{G}$ and the equality constraints ($\mathbf{PS}$, $\mathbf{L}$, and $\mathbf{U}$ have dimensions of P plus the number of equality constraints), $\mathbf{L}$ is unit lower triangular, $\mathbf{U}$ is upper triangular, and $\mathbf{Z}_A$ is not reduced to triangular form. The LU portion includes the equality constraint rows and columns since these do not change during the pivoting process and $\mathbf{G}$ may be singular without them in some cases. Since $\mathbf{G}$ is symmetric positive-definite, in theory a more efficient Cholesky factorization that avoids partial pivoting and scaling could be used. However, the condition number of $\mathbf{G}$ is often not small enough to make this a safe method, and the efficiency of this factorization is not critical since it is performed only once in each iteration.

Note that solving the partially factored system involves solving $\mathbf{2Z}_A\boldsymbol{\mu}_A = \hat{\mathbf{c}}_A$ then back-substitution on

$$\mathbf{U}\begin{pmatrix} \mathbf{p}_E \\ \mathbf{2\lambda} \end{pmatrix} = \hat{\mathbf{h}} - \mathbf{2}\hat{\mathbf{C}}_A^T\boldsymbol{\mu}_A \tag{5.4}$$

which is considerably less work than solving the full system from scratch. The active constraint pivoting scheme used in SISAME-3D attempts to remove a constraint with a negative Lagrange multiplier after adding each constraint to A. The removed constraint is selected based only on the values of $\boldsymbol{\mu}_A$, so back-substitution for $\mathbf{p}_E$ is avoided completely in a constraint removal pass.

The partially factored system must be efficiently updated for the addition or removal of a constraint from the active set. Such update formulas are not difficult to develop. Since $\mathbf{Z}_A$ is not row reduced, no cumulative error occurs when $\mathbf{Z}_A$ is updated, and the solution for $\boldsymbol{\mu}_A$ can be performed with row scaling and partial pivoting. These accuracy benefits come at the cost of lower efficiency than methods that update a factorization of $\mathbf{Z}_A$.

5.2 Link Force Representation

At each instant the SISAME-3D formulation requires a linear representation for the Link forces in terms of the extracted Static and Dynamic Parameters (see Sections 4.3 and 4.4). In some cases this is straightforward.

A Link with a linear elastic Static LE Aspect and a linear LD Damper has a force that linear in the Parameters S and D_{Slp}:

$$f \;=\; S\,x \;+\; D_{\text{Slp}}\,r\,.$$

A segmented elastic Static SE Aspect has the static force representation

$$f^S \;=\; \left(\tfrac{X_{i+1}-x}{X_{i+1}-X_i}\right) F_i \;+\; \left(\tfrac{x-X_i}{X_{i+1}-X_i}\right) F_{i+1}$$

where the $[X_i, X_{i+1}]$ segment is active at x, which is linear since the X_i values are defined.

Other Static and Dynamic types and combinations can have nonlinear extracted Parameter contributions. Linearized approximants are employed to address these nonlinearities (see Section 5.2.1). Some Static and Dynamic types can have domain transitions that depend on extracted Parameters and must be estimated from the previous solution (see Section 5.2.2). The use of approximants or estimates requires SISAME-3D to use an iterative procedure to obtain a solution (see Section 5.3).

5.2.1 Linear Approximants

Nonlinear Link force expressions are linearized in SISAME-3D using first order Taylor expansions about the previous Parameter solution. This yields approximants of the form

$$\overline{f}(\mathbf{p}_E) \;=\; f(\widetilde{\mathbf{p}_E}) \;+\; \nabla f(\mathbf{p}_E)\,(\mathbf{p}_E - \widetilde{\mathbf{p}_E})$$

where $\overline{f}$ is the approximant for f and $\widetilde{\mathbf{p}_E}$ is the previous solution for the Parameter vector $\mathbf{p}_E$. Some examples of the approximants are given below.

The Static EE type has static force

$$f^S \;=\; S_O\,x \;+\; C\,\sigma(x)\,|x|^P$$

which is nonlinear if P is extracted. The EE approximant is

$$\overline{f^S} \;=\; [x]\,S_O \;+\; \left[\sigma(x)\,|x|^{\widetilde{P}}\right] C \;+\; \left[\widetilde{C}\sigma(x)\,|x|^{\widetilde{P}} \ln|x|\right] P \;-\; \left[\widetilde{C}\sigma(x)\,|x|^{\widetilde{P}} \ln|x|\widetilde{P}\right]\,.$$

The Static AE type has static force

$$f^S \;=\; \left(\tfrac{2}{\pi}\,F_{\text{Max}}\right) \text{Arctan}\left(\tfrac{\pi}{2}\,\tfrac{S_O}{F_{\text{Max}}}\,x\right)$$

which is nonlinear unless the value of $\frac{S_O}{F_{\text{Max}}}$ is defined. Letting $\mathcal{X} = \frac{\pi}{2}\frac{S_O}{F_{\text{Max}}}\,x$ the AE approximant is

$$\overline{f^S} \;=\; \left[\tfrac{x}{1+\widetilde{\mathcal{X}}^2}\right] S_O \;+\; \left[\tfrac{2}{\pi}\left(\text{Arctan}\left(\widetilde{\mathcal{X}}\right) - \tfrac{\widetilde{\mathcal{X}}}{1+\widetilde{\mathcal{X}}^2}\right)\right] F_{\text{Max}}\,.$$

The Static AI type boundary force approximant is similar with an additional term for the effect of boundary shifts.

The Static BE type has a linear origin line static force of $f^S = S_O\,x$ and compression and tension line static forces of the form

$$f^S_\square \;=\; F_\square \;+\; [x - X_\square]S_\square$$

where $\square$ is C or T, which is nonlinear if both $X_\square$ and $S_\square$ are extracted. If $X_\square$ is the explicit Parameter then, using $F_\square = S_O X_\square$, the BE approximant is

$$\overline{f^S_\square} \;=\; \left[\widetilde{X}_\square\right] S_O \;+\; \left[\widetilde{S}_O - \widetilde{S}_\square\right] X_\square \;+\; \left[x - \widetilde{X}_\square\right] S_\square \;-\; \left[\left(\widetilde{S}_O - \widetilde{S}_\square\right)\widetilde{X}_\square\right]\,.$$

If $F_\square$ is the explicit Parameter and $S_O > 0$ the BE approximant is

$$\overline{f^S_\square} \;=\; \left[\tfrac{\widetilde{F}_\square\widetilde{S}_\square}{\widetilde{S}_O^2}\right] S_O \;+\; \left[1 - \tfrac{\widetilde{S}_\square}{\widetilde{S}_O}\right] F_\square \;+\; \left[x - \widetilde{X}_\square\right] S_\square$$

(when $\widetilde{S}_O = 0$ a different approximant is used). The Static BI type boundary force approximant is similar to the explicit $F_\square$ Parameter case.

The nonsymmetric Static SI type has the tension line static force

$$f^S_T \;=\; S_T\left(x - \left(X_B - \tfrac{F_B}{S_U} - X_{\text{Slk}}\right)\right)$$

where (X_B, F_B) is the compression boundary point that the unloading path connects to, which has two nonlinear terms. The SI tension line approximant is

$$\overline{f^S_T} \;=\; \left[\tfrac{\widetilde{S_T}}{\widetilde{S_U}}\right] F_B \;-\; \left[\tfrac{\widetilde{F_B}\widetilde{S_T}}{\widetilde{S_U^2}}\right] S_U \;+\; \left[x - \left(X_B - \tfrac{\widetilde{F_B}}{\widetilde{S_U}} - \widetilde{X_{\text{Slk}}}\right)\right] S_T \;+\; \left[\widetilde{S_T}\right] X_{\text{Slk}} \;-\; \left[\widetilde{X_{\text{Slk}}}\widetilde{S_T}\right]\,.$$

The Dynamic AD Damper type has force

$$f \;=\; f^S \;+\; \left(\tfrac{2}{\pi}D_{\text{Max}}\right)\text{Arctan}\left(\tfrac{\pi}{2}\tfrac{D_{\text{Slp}}}{D_{\text{Max}}}r\right)$$

which is nonlinear unless the value of $\frac{D_{\text{Slp}}}{D_{\text{Max}}}$ is defined. Letting $\mathcal{R} = \frac{\pi}{2}\frac{D_{\text{Slp}}}{D_{\text{Max}}}\,r$ the AD approximant is

$$\overline{f} \;=\; f^S \;+\; \left[\tfrac{r}{1+\widetilde{\mathcal{R}}^2}\right] D_{\text{Slp}} \;+\; \left[\tfrac{2}{\pi}\left(\text{Arctan}\left(\widetilde{\mathcal{R}}\right) - \tfrac{\widetilde{\mathcal{R}}}{1+\widetilde{\mathcal{R}}^2}\right)\right] D_{\text{Max}}\,.$$

Extracted Dynamic Magnifiers require approximants because their Parameters appear in a product/ratio with the (possibly extracted) Static Parameters and because their Parameters appear in the denominator for the $\sigma(rf^S) = -1$ case.

The Dynamic LM Magnifier type has force

$$f \;=\; [1 + M_{\text{Slp}}\,|r|\,]^{\sigma(rf^S)}\, f^S$$

which is nonlinear if M_{Slp} is extracted and either f^S is extracted or $\sigma(rf^S) = -1$. Letting $\mathcal{M} = M_{\text{Slp}}\,|r|$ the LM approximants are

$$\overline{f}|_{\sigma=1} \;=\; \left[1 + \widetilde{\mathcal{M}}\right] f^S \;+\; \left[\widetilde{f^S}|r|\right] M_{\text{Slp}} \;-\; \left[\widetilde{f^S}\widetilde{\mathcal{M}}\right]$$

$$\overline{f}|_{\sigma=-1} \;=\; \left[\frac{1}{1+\widetilde{\mathcal{M}}}\right] f^S \;-\; \left[\frac{\widetilde{f^S}|r|}{(1+\widetilde{\mathcal{M}})^2}\right] M_{\text{Slp}} \;+\; \left[\frac{\widetilde{f^S}\widetilde{\mathcal{M}}}{(1+\widetilde{\mathcal{M}})^2}\right]$$

The Dynamic AM Magnifier type has force

$$f \;=\; \left[1 + \left(\tfrac{2}{\pi}(M_{\text{Max}} - 1)\right) \text{Arctan}\left(\tfrac{\pi}{2}\,\tfrac{M_{\text{Slp}}}{M_{\text{Max}}-1}\,|r|\,\right)\right]^{\sigma(rf^S)} f^S$$

which is nonlinear unless the value of $\frac{M_{\text{Slp}}}{M_{\text{Max}}-1}$ is defined or if M_{Slp} or M_{Max} is extracted and either f^S is extracted or $\sigma(rf^S) = -1$. Letting $\mathcal{M} = \frac{2}{\pi}(M_{\text{Max}} - 1)$ and $\mathcal{R} = \frac{\pi}{2}\frac{M_{\text{Slp}}}{M_{\text{Max}}-1}\,|r| = \frac{M_{\text{Slp}}}{\mathcal{M}}\,|r|$ and $\mathcal{A} = \text{Arctan}(\mathcal{R})$ the AM approximants are

$$\overline{f}|_{\sigma=1} \;=\; \left[1 + \widetilde{\mathcal{M}}\widetilde{\mathcal{A}}\right] f^S \;+\; \left[\widetilde{f^S}\frac{|r|}{1+\widetilde{\mathcal{R}}^2}\right] M_{\text{Slp}} \;+\; \left[\widetilde{f^S}\frac{2}{\pi}\left(\widetilde{\mathcal{A}} - \frac{\widetilde{\mathcal{R}}}{1+\widetilde{\mathcal{R}}^2}\right)\right](M_{\text{Max}} - 1) \;-\; \left[\widetilde{f^S}\widetilde{\mathcal{M}}\widetilde{\mathcal{A}}\right]$$

$$\overline{f}|_{\sigma=-1} \;=\; \left[\frac{1}{1+\widetilde{\mathcal{M}}\widetilde{\mathcal{A}}}\right] f^S \;-\; \left[\frac{\widetilde{f^S}\frac{|r|}{1+\widetilde{\mathcal{R}}^2}}{(1+\widetilde{\mathcal{M}}\widetilde{\mathcal{A}})^2}\right] M_{\text{Slp}} \;-\; \left[\frac{\widetilde{f^S}\frac{2}{\pi}\left(\widetilde{\mathcal{A}} - \frac{\widetilde{\mathcal{R}}}{1+\widetilde{\mathcal{R}}^2}\right)}{(1+\widetilde{\mathcal{M}}\widetilde{\mathcal{A}})^2}\right](M_{\text{Max}} - 1) \;+\; \left[\frac{\widetilde{f^S}\widetilde{\mathcal{M}}\widetilde{\mathcal{A}}}{(1+\widetilde{\mathcal{M}}\widetilde{\mathcal{A}})^2}\right]$$

The approximants are, in general, inexact representations of the Link force (since $\mathbf{p}_E \neq \widetilde{\mathbf{p}_E}$) so an iterative procedure (see Section 5.3) is used to converge on a self-consistent solution: as $(\mathbf{p}_E - \widetilde{\mathbf{p}_E}) \to 0$ the approximation error goes to zero.

To control the approximation error and facilitate robust convergence SISAME-3D uses **iteration constraints** that limit the change to $\mathbf{p}_E$ in each iteration. Constraints specific to each approximant form are implemented to achieve a balance between convergence speed and robustness.

5.2.2 Domain Transitions

Many of the Static types can have domain transitions that depend on extracted Parameters, including

- Inelastic Static transitions between the unloading line and the opposite boundary or slack domains.

- One-way clipping transitions.
- Static BE and BI transitions between the origin line and compression/tension lines/boundaries.
- Static SI transitions between unloading, slack, and reloading domains.

The Dynamic Magnifier exponent, $\sigma(rf^s)$, depends on the sign of the static force, which can depend on extracted Static Parameters.

When domain transitions depend on extracted Parameters SISAME-3D uses the previous solution to select the active domain at each instant. This inexact domain identification requires the use of an iterative procedure (see Section 5.3) to converge on a self-consistent solution.

5.3 Extraction Iteration

When approximants (see Section 5.2.1) are used or when domain transitions depend on extracted Parameters (see Section 5.2.2) an iterative procedure is used to seek a self-consistent (fixed point) solution. The iteration starts with an initial solution (generated automatically) and uses this solution to determine the static force sign and other state information necessary to set up a tractable approximation problem that tends to improve the solution. The resulting solution is used in the same way, and the iterations continue until convergence yields a self-consistent solution.

SISAME-3D automatically detects and reports the presence of features requiring iterative solutions. Most real-world models will contain such features.

SISAME-3D uses an automatic **relaxation** process to speed convergence as the maximum number of iterations is approached or when the iterations are cycling (usually near a self-consistent solution). During relaxation the solution changes by a diminishing fraction of the distance from the last solution to the newly extracted solution.

Despite the lack of guaranteed optimality for iteratively obtained models, trials indicate that optimal or near-optimal models are obtained in almost all cases. With an appropriate initial feasible solution, it is not hard to show that, for any one feature, the iterations generally tend to improve successive solutions. Nevertheless, the fewer the "difficult" features the faster the iterations can be expected to converge, and the more likely the resulting model will be at least near-optimal.

5.4 Integral Domain Targets

The velocity and position domain Mass target equations of (4.15) and the impulse and integrated impulse domain Load target equations of (4.20) are implemented by numerical

integration. In particular, the influence matrix integrals

$$\int_t \mathbf{\Gamma}_T^{(M)} \quad \text{and} \quad \int_t\int_t \mathbf{\Gamma}_T^{(M)} \quad \text{and} \quad \int_t \mathbf{\Gamma}^{(\mathcal{L})} \quad \text{and} \quad \int_t\int_t \mathbf{\Gamma}^{(\mathcal{L})}$$

must be accumulated to form the target system at each time step.

In practice, a scaled sum is used to approximate the integral domain target equations: a higher-order integration method would be inappropriate given the slope discontinuities in some of the Static Aspect characteristics and would require storing $K_E \times P_S$ matrices for multiple time steps.

The integrated influence matrices include Parameter contributions from time-zero onward, yielding target systems that are less sparse than the acceleration and force domain target systems. The computational effort to accumulate the integral domain target systems and this decrease in sparsity causes some slowdown of the extraction process. Since the integrated signals are smoother than the acceleration and force signals, SISAME-3D applies the integral domain targets only at every eight time step to reduce the computational burden. Integral domain targeting can be suppressed entirely by setting the Mass *ConV* and *ConP* and the Load *ConI* and *ConJ* Parameters to `N`, but this is not recommended for most modeling applications.

5.5 Supplementary Target Equations

The extracted Link solution can be refined by the introduction of supplementary target equations. These might be used to condition the least squares problem in cases that are inherently ill-conditioned, such as models with many load-path loops. Or target equations might be used to provide estimates for some of the Parameters to assist SISAME-3D in partitioning the loads among multiple load-paths. In general, providing any additional information known about the Parameters helps in narrowing the sometimes large space of near-optimal models.

SISAME-3D uses a lightly weighted diagonal **Parameter conditioning** target system of the form

$$\mathbf{I}\,\mathbf{p}_E \;\approx\; \mathbf{0} \tag{5.5}$$

to prevent singularity when there are Parameters that are never active in the given motion data or, for example, when two extracted Links are connected in parallel to the same Masses. This also improves the condition of extractions that are otherwise ill-conditioned.

If SISAME-3D detects pairs of Parameters, $\mathbf{p}_i$ and $\mathbf{p}_j$, with identical contributions to the least squares system (4.22) weighted equations of the form $\mathbf{p}_i \approx \mathbf{p}_j$ are included in a **Parameter equivalence conditioning** target system of the form

$$\mathbf{E}\,\mathbf{p}_E \;\approx\; \mathbf{0} \tag{5.6}$$

to improve the condition of the extraction in the presence of these singularities.

SISAME-3D also uses a diagonal **Parameter damping** target system of the form

$$\mathbf{I}\,\mathbf{p}_E \;\approx\; \mathbf{p}_f \tag{5.7}$$

where $\mathbf{p}_f$ is the current feasible solution. This target system tends to damp changes in the Parameters and thus assists in the final solution convergence.

Another lightly weighted target system is used to make SISAME-3D prefer smooth segmented Static characteristics. For each junction between two segments, either of which include at least one extracted force, there is a target equation of the form

$$\tfrac{\Delta_i}{\Delta_i^{(2)}}\,F_{i-1} \;-\; F_i \;+\; \tfrac{\Delta_i}{\Delta_i^{(2)}}\,F_{i+1} \;\approx\; 0 \tag{5.8}$$

where $\Delta_i \doteq X_{i+1} - X_i$ and $\Delta_i^{(2)} \doteq X_{i+1} - X_{i-1}$. The target error is the amount (in force units) that F_i differs from the value at X_i of the line between F_{i-1} and F_{i+1}.

All of these supplementary target equations are included in the $\mathbf{\Psi}\,\mathbf{p}_E \approx \boldsymbol{\psi}$ system of (4.23).

5.6 Target Equation Weighting

The target equations are weighted by the matrix $\mathbf{\Omega}$. The weights are needed to assure that the least squares solution balances the errors with respect to their significance and not their numerical size. SISAME-3D uses a diagonal $\mathbf{\Omega}$, a common simplifying approach although it treats the errors as independent random variables, which they are not.

For truly independent errors, weights can be derived from confidence bands. Fixing a common level of confidence (*e.g.*, 90%), a band of $\pm\Delta$ can be assigned to each equation. The corresponding weight would be $1/\Delta$. This approach is used to set the weights for the supplementary target equations.

To select weights for the Mass acceleration and Load force target equations that compensate for the number of time steps and the dependencies among errors at nearby time points, a confidence band of $\pm\Delta_{\text{RMS}}$ on the root mean square error for each equation is used. The corresponding weight is

$$\frac{1}{\Delta_{\text{RMS}}\sqrt{N+1}}$$

where N+1 is the total number of target time steps, $0, \ldots, N$. The modeler can set the Δ_{RMS} for each target Mass (*ConA*) and each target Load (*ConF*).

Mass velocity and position targets are weighted by

$$\frac{1}{\texttt{ConA}\sqrt{N+1}}\,\frac{\eta_M}{\texttt{ConV}\,(N+1)} \qquad \text{and} \qquad \frac{1}{\texttt{ConA}\sqrt{N+1}}\,\frac{\eta_M}{\texttt{ConP}\,(N+1)^2}$$

respectively, where η_M includes compensation for applying these targets at a subset of the time steps and a heuristic factor selected so that the default `ConV` and `ConP` values of 1 give a reasonable balance between the acceleration, velocity, and position target weighting.

Similarly, Load impulse and integrated impulse targets are weighted by

$$\frac{1}{\texttt{ConF}\sqrt{N+1}}\,\frac{\eta_L}{\texttt{ConI}\,(N+1)} \qquad \text{and} \qquad \frac{1}{\texttt{ConF}\sqrt{N+1}}\,\frac{\eta_L}{\texttt{ConJ}\,(N+1)^2}$$

respectively, where η_L includes compensation for applying these targets at a subset of the time steps and a heuristic factor selected so that the default `ConI` and `ConJ` values of 1 give a reasonable balance between the force, impulse, and integrated impulse target weighting.

5.7 Extracted Parameter Constraints

Constraints are used in SISAME-3D to ensure physically meaningful extracted components, to meet user-specified extracted Parameter bounds (see Section 3.4.8), and to control approximant errors (see Section 5.2.1). The SISAME-3D solution procedure supports linear equality and inequality constraints.

Automatically generated constraints ensure

- Extracted Mass weights are nonnegative and meet any specified Structure weights.
- Extracted Static and Dynamic Parameters meet the requirements of their types.

For example, an extracted nonsymmetric Static SI Aspect has constraints

- $S_U \;\geq\; 0$
- $S_T \;\geq\; 0$
- $X_{\text{Slk}} \;\geq\; 0$
- $F_i \;+\; F_\Delta \;\geq\; 0$ for $i = 1, \ldots, N$
- $F_i \;\geq\; F_{i-1}$ for $i = 2, \ldots, N$ ensuring nonnegative segment slopes (unless `AnySlope=True`)
- $F_i \;-\; F_{i-1} \;\leq\; (X_i - X_{i-1})\, S_U$ for $i = 2, \ldots, N$ ensuring segment slopes are less than S_U

User-specified extracted Parameter bounds allow partial Parameter information to be exploited to reduce the space of feasible solutions. (Constraints increase the computational effort more than target equations so extracted Parameter estimates should be preferred when acceptable.)

SISAME-3D automatically eliminates redundant constraints for efficiency.

5.8 Phase I Procedure

The extraction process requires an initial feasible solution from which to begin the optimization (see Section 5.1). SISAME-3D constructs an initial solution that satisfies the inherent constraints on model component behavior and the user-specified numeric Parameter bounds. References between extracted Parameters may cause the constructed solution to be infeasible. In this case a Phase I procedure is automatically activated to obtain a feasible solution.

The Phase I procedure can also be triggered in iterations subsequent to the first pass if ill-conditioning causes the solution to drift sufficiently far from precise feasibility.

For the Phase I procedure an additional Parameter, p_0, is introduced, the error function is augmented with $\frac{1}{2}Mp_0^2 + Mp_0$ for some large value of M, and the constraint $p_0 \geq 0$ is included. The existing constraints are also modified to

$$\mathbf{C}_{=}\,\mathbf{p}_E + \mathbf{s}\,p_0 = \mathbf{c}_{=} \qquad \text{and} \qquad \mathbf{C}\,\mathbf{p}_E + \mathbf{u}\,p_0 \leq \mathbf{c}\,.$$

The values of $\mathbf{s}$ and $\mathbf{u}$ and an initial feasible solution to this augmented problem are constructed as follows, starting with an arbitrary Parameter vector $\mathbf{p}_E$. The elements of $\mathbf{u}$ are zero for the original constraints that are not violated by $\mathbf{p}_E$ and -1 for those that are. The initial value of p_0 is simply any value large enough so that all the violated inequality constraints, with the additional $-p_0$ term, are satisfied. Then $\mathbf{s}$ is set to whatever is needed to satisfy the augmented equality constraints with this p_0.

In Phase I the optimality equations of (5.1) become

$$\begin{bmatrix} M & \mathbf{0} & \mathbf{s}^T & \mathbf{u}_A^T \\ \mathbf{0} & \mathbf{G} & \mathbf{C}_{=}^T & \mathbf{C}_A^T \\ \mathbf{s} & \mathbf{C}_{=} & \mathbf{0} & \mathbf{0} \\ \mathbf{u}_A & \mathbf{C}_A & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{pmatrix} p_0 \\ \mathbf{p}_E \\ \mathbf{2\lambda} \\ \mathbf{2\mu}_A \end{pmatrix} = \begin{pmatrix} -M \\ \mathbf{h} \\ \mathbf{c}_{=} \\ \mathbf{c}_A \end{pmatrix}. \tag{5.9}$$

If M is large enough and a feasible solution exists, the solution to this augmented problem will have $p_0 = 0$ and thus will be feasible for the original problem. If $p_0 \neq 0$ SISAME-3D will increase M automatically and repeat the Phase I solution. If M becomes so large that ill-conditioning is severe, SISAME-3D terminates and reports that no feasible solution exists.

This Phase I procedure involves minimal additional storage and has the benefit of producing the desired solution at the same time since, with $p_0 = 0$, the solution for $\mathbf{p}_E$ will minimize the original error function. In practice the Phase I solution for $\mathbf{p}_E$ is used as the initial feasible solution in a verification solution of the unaugmented problem to eliminate any numerical distortion that the large M value may have introduced.

5.9 Filtered Data Representation

The instrumented motion and load signals are transformed to a special Discrete Fourier Transform (DFT) representation for reconstruction of the filtered motions at each simulation time step. A baseline function with the correct acceleration, velocity, and position tail values is removed before the DFT is computed. This facilitates a DFT-based filter that simultaneously approximates the motion in all three domains without drift errors in any of them. The complete description of this filtering approach can be found in the documentation for the SimFil program [7, 9].

The filtering in SISAME-3D is based on the SimFil filter with the following provisos:

- SISAME-3D uses a DFT final time of up to 1.1 times the final output time (*FinTOut*) depending on the final time of the input signals. This can improve the end tail representation in some situations.
- SISAME-3D uses a tail smoothing window length constant of $\rho = 1$ giving shorter tail smoothing window than SimFil's $\rho = 2$.
- SISAME-3D uses default tail smoothing frequencies of 2.5 times the cutoff frequency.

5.10 Time Step Selection

The time steps used during extraction and simulation affect the accuracy of a number of SISAME-3D's computations. The simulated Mass motion state is determined from the current defined Link state by integrating the equations of motion from the previous time step. The rotational position of Masses of every class is based on an integration of the quaternion representation. The extraction simulation time step determines the sampling frequency with which target equations are generated.

SISAME-3D selects default extraction and simulation time steps based on the Model's vibratory, filtering, and revolutional frequencies, damping characteristics, and the output and DFT timing. The default time steps should provide sufficient accuracy for most applications.

The highest revolutional frequency of the simulated Masses is not known before the simulation is performed so SISAME-3D considers only the initial angular velocities. If the revolutional frequency of simulated Masses increases sufficiently over the event time span, after completing the simulation SISAME-3D may issue a warning to the log file that suggests a time step that is smaller than the time step(s) used. In this case it is advisable to specify sufficiently small time step(s) explicitly and repeat the run.

Default and user-specified time steps are also adjusted for additional criteria:

- The simulation time step must divide the output time step.

- The extraction simulation time step must be a multiple of the simulation time step.
- When input signals are specified a simulation time step with a sufficiently large common divisor with the DFT final time is chosen, when possible, to allow a fast DFT reconstruction algorithm to be used.

5.11 Numerical Integration

The simulated Mass motions are numerically integrated using a low-order predictor–corrector method. Higher order methods are not appropriate because of the slope discontinuities that may be present in the Static Link characteristics. SISAME-3D uses a predictor step followed by two corrector steps. The combination is a second order method for the velocity and a third order method for the position.

The predictor step is based on the state of the simulated Masses at the previous simulation time step. The predictor step at the ith time step for each simulated Mass is

$$\begin{aligned} v_i^* &= v_{i-1} + \Delta a_{i-1} && \text{(Euler)} \\ p_i^* &= p_{i-1} + \tfrac{\Delta}{2}\left(v_{i-1} + v_i^*\right) && \text{(Trapezoid)} \end{aligned} \tag{5.10}$$

Given the velocities and positions, the predicted force in the defined Links $\mathbf{f}_D^*$ is computed and the predicted simulated Mass accelerations, a_i^* are obtained from

$$\mathbf{a}_S^* = \mathbf{W}_S^{-1}\,\mathbf{\Phi}_{SD}\,\mathbf{f}_D^* \tag{5.11}$$

The first corrector step is

$$\begin{aligned} v_i^{**} &= v_{i-1} + \tfrac{\Delta}{2}\left(a_{i-1} + a_i^*\right) && \text{(Trapezoid)} \\ p_i^{**} &= p_{i-1} + \Delta v_{i-1} + \tfrac{\Delta^2}{6}\left(2a_{i-1} + a_i^*\right) && \text{(Parabolic)} \end{aligned} \tag{5.12}$$

The Link force $\mathbf{f}_D^{**}$ is computed and the accelerations, a_i^{**}, are obtained from

$$\mathbf{a}_S^{**} = \mathbf{W}_S^{-1}\,\mathbf{\Phi}_{SD}\,\mathbf{f}_D^{**} \tag{5.13}$$

The second corrector step has the same form as the first

$$\begin{aligned} v_i &= v_{i-1} + \tfrac{\Delta}{2}\left(a_{i-1} + a_i^{**}\right) && \text{(Trapezoid)} \\ p_i &= p_{i-1} + \Delta v_{i-1} + \tfrac{\Delta^2}{6}\left(2a_{i-1} + a_i^{**}\right) && \text{(Parabolic)} \end{aligned} \tag{5.14}$$

The Link force $\mathbf{f}_D$ is computed and the accelerations, a_i, are obtained from

$$\mathbf{a}_S = \mathbf{W}_S^{-1}\,\mathbf{\Phi}_{SD}\,\mathbf{f}_D \tag{5.15}$$

Appendices

A SISAME-3D Modeling Guidelines

SISAME-3D can be applied to a range of predictive, parametric, and design modeling tasks, but model extraction is a fundamentally different process than simulation and requires different skills from the modeler. Some basic SISAME-3D modeling notes and guidelines are presented below.

A.1 Understanding Extraction

- A number of factors influence the extraction process and can limit the accuracy of a model. Accurate instrument data for all the significant Mass motions are necessary because the computations SISAME-3D performs are inherently sensitive to motion data errors. The Mass weights, Link configuration, and Static and Dynamic Aspect types must also be appropriate. A model's degree-of-freedom constraints and the detail of its discretization of the structure will limit its applicability to a range of events.
- SISAME-3D can produce predictively useful models from a set of accelerometer signals with little knowledge of the structure. Without supplementary information about the structure or additional instrument data the model produced may not accurately capture the load distribution in the structure. The load-path "loops" needed to model vehicle structures make the instantaneous load distribution indeterminate given the accelerations of the Masses. Even though SISAME-3D considers the entire span of the event in its extractions, some indeterminacy may remain, and there may be a large space of near-optimal models that can reproduce the acceleration signals with almost equal accuracy. All of these models may produce predictively accurate motions in the range of events of interest. A small amount of supplementary information, or extraction from more than one event, may suffice to remove the indeterminacy and yield an accurate structure.
- To obtain the most complete and accurate models, instrument locations should, ideally, capture all significant masses that will undergo distinct motions.
- Simulations of events more severe than the extraction event require extrapolated load-path behavior and are generally of lower predictive accuracy.
- Simulations of events involving significantly different failure modes than the extraction event are generally of lower predictive accuracy.

A.2 Instrumentation

- Careful assessment and refinement of the test instrument data using VeCor or a comparable tool is vital to the extraction of quality models, especially for more complex models. The load-path deflections are typically computed from the difference between doubly integrated accelerometer signals making them sensitive to even modest signal errors.
- If possible, it is best to provide SISAME-3D velocity and impulse domain signals. This avoids either double integration or double differentiation of the signals and will allow signals with a larger time step to be used without loss of accuracy.
- When acceleration and force domain signals are used a small time step may be required to accurately extract the Parameters of stiff, high frequency parts of the Model. A time step of $\leq 1/100f$, where f is the highest vibrational frequency of interest in Hz, may be needed. Some guidance as to a sufficient input signal time step can be obtained by seeing the (usually conservative) time step SISAME-3D selects for simulation of the resulting model. For typical crash modeling applications a signal time step of 10^{-4} seconds or less should be adequate.
- Cardan angle time series input signals can be used to define the rotational position of Masses. If they are used for any of the rotational axes for a Mass they must be used for all of the rotational axes: Cardan angles are not the simple integrals of angular velocity and must be used together to compute the angular velocity and acceleration. A fairly small input time step may be required to obtain accurate rotational motions from Cardan signals. SISAME-3D will automatically adjust Cardan input signals so that the time-zero Cardan angles to lie on $(-\pi, \pi]$. It may also adjust the Cardan signals to match the internal orientation representation: there are two representations for each orientation.
- The filtering method uses a portion of the motion data outside the extraction time span, if available, for acceleration/force tail smoothing. The best results will be obtained if the entire available span of the unfiltered motion/loading signals are input to SISAME-3D.
- With test-derived instrumentation signals some tail smoothing is usually beneficial. Start with the default tail smoothing for the selected cutoff frequency and adjust as needed.
- With simulation-derived input signals tail smoothing may not be necessary and can be suppressed. But if position input signals are used (velocity signals are suggested) some relatively high frequency tail smoothing may be helpful in removing acceleration tail signal noise resulting from double differentiation.
- When the measured motions for a Mass are known to be of poor quality even after preprocessing the emphasis on that Mass' target equations can be reduced by increasing its *ConA* confidence band(s) (see Section 3.11). In cases of severely degraded motion it may be best to suppress the Mass target equations by setting the Mass `Class=d`. There is a tradeoff involved in deemphasizing a Mass motion: if the resulting model poorly fits the Mass position then the adjacent Link deflections will be inaccurate and thus the resimulated motions may drift from the measured motions.
- If the motions for a Mass node of interest are unavailable it may be possible to use an artificial motion to allow extraction of the desired model configuration. Another test may be available with the desired instrumentation – a model generated from that test could be used to "scale" to the correct impact velocity if necessary. A finite element model may be available from which

the motion can be generated. Or a model could first be generated without the Mass and then the Mass could be inserted into the Model by splitting the necessary Links (using *XScl* values summing to one for each Link split) and, possibly, adding estimated Links for other load-paths to the Mass. Iterations of first simulating just the inserted Mass (with the other Masses driven by their instrument signals) followed by extraction of the whole model may be helpful in finding a self-consistent model with realistic behavior for the inserted Mass.

A.3 Model Configuration

- When the set of active load-paths is not completely known additional load-paths can be included in preliminary extractions to help identify the important load-paths. Nonintuitive load-paths are sometimes important to the model because the composite nature of the Masses and load-paths can obscure their interconnections and interactions. A common mistake is to attempt to channel a number of loads through a Mass when load-paths that bypass the Mass are also active.
- Load-paths representing large or composite structures in coarser models may benefit from the use of `AnySlope=True` and will generally benefit from larger dynamic contributions. This will give less physically-idealized models that may be more predictively accurate for the range of events of interest.
- More complex load-path behaviors than those provided directly by SISAME-3D Static and Dynamic Aspect types can be obtained by combining multiple Links in parallel, or in series connected by low-weight Mass nodes.
- In keeping with the nature of lumped-parameter modeling, SISAME-3D is intended for relatively low frequency extractions. Extractions at high frequencies can greatly increase the computation time, usually with little or no benefit to model quality. For vehicle crash test modeling applications, a 0.1–0.2 second time span and a 100 Hz or less cutoff frequency are generally appropriate.
- Stiffer Links and higher cutoff frequencies cause SISAME-3D to select a smaller default simulation time step, thus slowing the extraction and/or simulation.

A.4 Weight Extraction

- Mass weights can be extracted in a separate **weight extraction** run containing only the Masses and the target Loads containing them.
- The simplest weight extractions use a single target Load that has all the Masses as members. For barrier impacts this target Load is the total barrier force. For vehicle–to–vehicle impacts a zero target Load is used for the set of all Masses in both Vehicles.
- The Masses in a barrier target Load may need a negative sign contribution or the barrier signals sign-reversed, depending on the orientation of the structure and the sign of the barrier loading signals.

- Weight extractions without weight estimates or bounds provide an assessment of the completeness of the motion instrumentation and assist in the selection of model extraction cutoff frequencies. If significant structure masses experiencing distinct motions are not captured by the instrumentation the Load match will be of lower accuracy, and a lower frequency model extraction will be appropriate.

- Weight extractions with a single target Load can have many near-optimal solutions for the Mass weights. Weight estimates and/or bounds based on engineering knowledge should be used to obtain a more robust model weight distribution.

- Defined Model, Vehicle, or Substructure weights are enforced by SISAME-3D via equality constraints on the Structures' extracted Mass weights.

- Impulse and integrated impulse domain targeting has an effect on the weight extraction solution. Since the velocity and position signal tails are fixed in the DFT time span used in SISAME-3D's filtered representation, the impulse and integrated impulse match in a weight extraction cannot usually be greatly altered by the weight solution. Too much emphasis on the match in these domains can lead to a degraded force domain solution. The *ConI* and *ConJ* Load Parameters can be used to control this tradeoff (see Section 3.18).

- Combining weight and load-path extraction in the same SISAME-3D run provides some coupling of the Mass weights and Links Parameters, which may or may not be desirable. SISAME-3D can find a somewhat better fit to the cumulative data but it may excessively skew the Link solution in doing so if there is not sufficient information to clearly imply a unique model. Getting the right balance between Load and Mass targeting emphasis may require adjusting the relative values of *ConF* and *ConA*. This type of run also allows the inertial force of Masses to be split over multiple target Loads containing the extracted Links attached to the extracted-weight Masses.

A.5 Dynamic Extraction

- Simultaneous extraction of both Dynamic and Static Link Aspects is supported. A crash data set may not contain sufficient information to clearly distinguish the dynamic and static contributions, in which case estimates and/or bounds on the extracted Dynamic Parameters may improve the quality of the model. Multiple-Event extractions with different speeds events provide a better basis from which to obtain robust Dynamic Parameters.

- Simultaneous extraction of both Damper and Magnifier Dynamic Aspects on the same Link may not yield an accurate partition of the dynamic contribution without supplementary estimates or bounds on some of the Dynamic Parameters.

- Dynamic Magnifier effects can partially compensate for the inertial lag not captured in a coarsely discretized model. This can cause larger dynamic contributions to be extracted for larger, composite load-paths.

A.6 Controlling and Refining Extractions

- Extractions can be refined by improving the data preparation or altering the model configuration, Mass weights, defined Static and Dynamic load-path Parameters, filtering frequencies, or target confidence bands. Extractions can also be improved by using supplementary information such as additional defined Parameters, additional target data, or estimates and/or bounds for extracted Parameters.
- Experience has shown that a poor extraction fit is less likely to be caused by the Mass weight distribution or dynamic effects than it is to the data quality and preparation or the lack of key load-paths.
- The use of Parameter estimates and bounds is valuable when partial information about the extracted structures is known. This can help SISAME-3D identify the physical structure more accurately when the instrument data alone has many near-optimal models of the chosen configuration.
- Target equations used in the constrained, least squares solution procedure for the extracted Parameters are weighted to balance the various errors. Weighting is based on confidence bands on the target data that should be assigned so that target equation errors are within the band with some uniform certainty level, for example 90%. Some target equations, such as the target Mass inertial force and target Load matches, are repeated at each simulation time step and are weighted with respect to a root mean square (RMS) error over the full time span. In these cases, the confidence bands should also be selected with respect to the RMS error.
- Target Loads are particularly valuable for better partitioning loads in the "looped" load-path models that are required for vehicle modeling. The closed loops of extracted Links in such models make an instantaneous force solution problem singular, and in practice the Mass motions alone may not strongly identify a unique force partition within the structure. Providing a target Load for one load-path in each loop would completely remove the singularities. Extraction from multiple events is another approach to improving the structure identification.
- Certain features of extraction models do not allow direct solution for the optimal model, and SISAME-3D uses a fixed-point iteration procedure to attempt to converge on a self-consistent that, in almost all cases, will be a near-optimal model.
- For **faster extractions** a larger than default *DelTSim* can be used. Smaller than default *MaxIter*, *ConPD*, and *MultPD* values and a larger than default *ConvTol* value can reduce the number of iterations. Decreasing the *Cutoff* value(s) will speed each iteration. Setting `ConV=N` and `ConP=N` will speed up each iteration but may worsen the velocity and position domain fits and the drift of the resimulation.
- For **better extracted models** a default or smaller *DelTSim* should be used. Larger than default *MaxIter*, *ConPD*, and *MultPD* may be helpful by allowing more iterations. A smaller than default *ConvTol* may slightly improve the solution. The *ConA*, *ConV*, *ConP*, *ConF*, *ConI*, and *ConJ* confidence band values can be adjusted to better balance the fit among the target domains.
- Extracted Static SE and SI Aspects can be "smoothed" by decreasing the *ConSS*. This may be particularly helpful for `AnySlope=True` Aspects.

A.7 Assessing the Extracted Model

- The model is extracted to best fit the filtered target signals, so the filtered outputs produced by SISAME-3D are the best basis for comparison with model-derived outputs. The output sampling rate may not be sufficient for accurate integration so obtain acceleration, velocity, and position outputs directly from the SISAME-3D extraction run, as needed.
- The effective motion outputs are based on the accelerations that would be produced by the extracted and defined Link forces if driven through the extraction event deflection histories. Comparing these to the filtered instrumented motions shows how closely the SISAME-3D model matches the extraction event in the domain that the solution is performed and without the cumulative drift of resimulation. If this fit is acceptable the extracted model should be exercised in a resimulation run to assess its baseline accuracy before conducting predictive simulations of other events. A poor resimulation match indicates that the data, configuration, weights, defined static and dynamic behavior, target confidence bands, and/or discretization assumptions need to be reexamined.
- Extracted Link force and energy outputs, group Load outputs, and motions are based on the Links being driven by the instrumented Mass motions. The extracted model is an optimal but inexact fit to the model/data combination, so the forces from an extraction run do not sum exactly to the instrumented Mass inertial forces.
- Even small deflection errors can lead to a poor motion fit prior to and during inelastic load-path unloading. Further refining the instrument data can improve the extraction results.

A.8 Multiple-Events

- The same dimensional system must be used in each Event file.
- Any extraction control Parameters in the Event files are ignored: only those in the Master file are active.
- Using a Master file for multiple simulations or for extraction Events without common Components yields the same results as separate single-Event runs.
- SISAME-3D is fairly relaxed in Event matching between Components with the same ID and matching containing Structures. To avoid an unwanted Component match either change the Component's ID or put the Component in a Substructure not present in the other Model(s). To avoid an unwanted match of a Structure and all its contained Components change the Structure's ID.
- Multiple-Event extraction iterations may perform more slowly and with poorer convergence behavior than for typical single-Event extractions because of a more complex solution space. Additional Parameter damping via smaller *ConPD* and *MultPD* values may speed the convergence.
- The default equal Event weighting will not generally yield the same fit measures for all Events since the target equation magnitudes vary between Events. Generally, more severe impacts have larger magnitude target coefficients and thus tend to produce larger fit measures.

- The Event weighting factor, *Wtg*, must be positive. To obtain the equivalent effect of a zero-weighted Event for a SISAME-3D run simply comment out the Event with a leading `!`.

A.9 Common Problems

- SISAME-3D catches numerous model specification errors and issues detailed messages to assist in correcting a problem. The log file should be checked for warnings even when a run completes without error.
- The output Event file from a Test run (Run `Class=T`) contains expanded references and other diagnostic information that may be useful in resolving model specification problems.
- It is easy to mis-configure the components of a complex 3D model. POV-Ray model image files are a valuable tool for verifying model configuration and positioning.
- If SISAME-3D reports that an ill-conditioned least squares system occurred during an extraction possible causes include:
  - Revered positive–negative Mass attachments on a Link with a nonsymmetric Static SI Aspect such that the Link experiences primarily tension instead of compression (a warning about this condition is generated). Symmetric SI Aspects eliminate the possibility of reversed attachments and are appropriate when yielding may occur in either direction.
  - Inactive extracted Parameters, possibly due to model configuration errors.
  - Attempting to extract Static and Dynamic Parameters on load-paths where the data does not sufficiently distinguish the static and dynamic contributions.
  - Inherent ill-conditioning in the model–data combination.

  Decreasing *ConPC* can improve the conditioning, but the extracted model should be scrutinized. In many cases the ill-conditioning is inherent in the extraction problem and will not seriously degrade the resulting model.

A.10 Predictive Simulations

- Simulating extracted models in less severe conditions than the extraction event may not exercise some of the extracted Parameters, using, in effect, a less complex model.
- Simulating extracted models in more severe conditions than the extraction event may entail significant extrapolation of the extracted Links, thus reducing the accuracy of the simulation.
- Validation extractions using data generated by simulating analytic models are also subject to limitations which can distort the extracted Links and, usually to a lesser extent, reduce the predictive accuracy of the extracted model. Some of the causes of these problems are:
  - Extracted Links with different deflection breakpoints than the corresponding simulated Link.
  - Cutoff frequencies low enough to eliminate or distort model features (particularly "sharp" features like failures).

  - Signal tail smoothing (alters the input signal tail values). `ZeroSm=N` and `EndSm=N` should be used with analytically generated signals.
  - Link Parameters that are inactive during the simulated event (so that constraints and conditioning determine their extracted values).
- For **faster simulations** a larger *`DelTSim`* than default can be used (up to a factor of 100 can generally be used without serious loss of precision).
- Very small *`DelTSim`* values can cause cumulative numeric drift errors that degrade the simulation accuracy.
- Artificially large load-path stiffness Parameters, such as S_U or S_T in Static SI Aspects, can be extracted in some cases and will lead to a small default *`DelTSim`* value in subsequent simulations of the extracted model. Controlling these Parameters (see Section 3.4.8) usually has little cost to the overall fit of the model and can greatly speed the simulations.

References

[1] Hollowell, W.T. (1986). *Adaptive Time Domain, Constrained System Identification of Nonlinear Structures.* Doctoral thesis, University of Virginia.

[2] Larsen, S. and Shaw, L.M. (1980 Revision). *Technical Justification and User's Manual for Fiat Automotive Structural Crashworthiness Computer Program A (Revision No. 3).* Dynamic Science Inc. Report No. 4890–76–51.

[3] Larsen, S. and Shaw, L.M. (1980 Revision). *Technical Justification and User's Manual for Fiat Automotive Structural Crashworthiness Computer Program B (Revision No. 3).* Dynamic Science Inc. Report No. 4890–76–52.

[4] Lawson, C.L. and Hanson, R.J. (1974). *Solving Least Squares Problems.* Prentice–Hall, Inc., Englewood Cliffs, N.J.

[5] Luenberger, D.G. (1973). *Introduction to Linear and Nonlinear Programming.* Addison–Wesley, Reading, Mass.

[6] Mentzer, S.G. (1982). *Analysis and Enhancement of the Fiat Methodology for Vehicle Crash Modeling and Simulation.* Automated Sciences Group, Inc. Report No. ASG–TR–82–18.

[7] Mentzer, S.G. (1991). *The SIMFIL Program: Simultaneous Filtering.* U.S. Department of Transportation Report No. DOT HS 807 731.

[8] Mentzer, S.G. *et al* (1992). *The SISAME Methodology for Extraction of Optimal Structural Crash Models.* SAE Report No. 920358.

[9] Mentzer, S.G. (1995). *The SimFil Program v.2: Simultaneous Filtering.* U.S. Department of Transportation Draft Report.

[10] Mentzer, S.G. (1995). *The VeCor Program.* U.S. Department of Transportation Draft Report.

[11] Mentzer, S.G. (2002). *The SISAME Program: Structural Crash Model Extraction and Simulation.* U.S. Department of Transportation Draft Report.

[12] Prasad, P. and Padgaonkar, A.J. (1981). *Static–to–Dynamic Amplification Factors for Use in Lumped-Mass Vehicle Crash Models.* SAE Report No. 810475.

[13] Shabana, A.A. (2001). *Computational Dynamics.* Wiley, New York.

[14] Shapiro, J.F. (1979). *Mathematical Programming: Structures and Algorithms.* Wiley, New York.

Index

A

Acceleration, 8
Angle
  Cardan, 12, 19, 20, 125
Applications, 4, 124–131
Approximant, 113
Aspect, 10
  Approximant, 113
  Dynamic, *see* Dynamic Aspect
  ID, 24
  Name, 27
  Point, 8, 9
  Reference, 27
    Set, 29
    Shorthand, 27
  Shape, 8, 9, 58, 60
  Static, *see* Static Aspect
  Transform, 66
Axes, 12, 18

B

Box, 61
  ID, 60

C

Capabilities, 4
  Design, 4
  FEM distillation, 4
  Multiple-Event modeling, 4, 98
  Weight extraction, 4
Cardan angles, 12, 19, 20, 125
Center of gravity, 9, 12, 18
CG, 9, 12, 18, 58
Color, 55, 57, 59, 61
Component, 6
  ID, 24
  Name, 25
  Reference, 26
    Shorthand, 25, 26
Cone, 62
  ID, 60
Constrained approximation problem, 110
Constraints, *see* Solution, Constraints
Coordinate frame, 12, 18
  Ground, 12, 18
  Inertial, 12, 18
  Origin, 12, 18
Coordinate system, 94
Cylinder, 62
  ID, 60

D

Damper, 10, 87–88
  AD, 88
  LD, 88
Damping force, 10
Data file, *see* Signal, File
Deflection, 10
Degrees of freedom, 12, 18, 48
Dimensional systems, 23
DOF, 12, 18, 48
Dynamic Aspect, 10, 85–90
  Approximant, 113
  Dampers, 10, 87–88
    AD, 88
    LD, 88
  Domain transitions, 115
  Extraction, 127
  Magnifiers, 10, 89–90, 127
    AM, 90
    LM, 90
  Name, 28
  One-way, 86
  Reference, 28
  Transform, 86
  Universal Parameters, 86
Dynamic magnification, 10

E

Element, 7

Fixed Mass, 57
ID, 24
Link, 6, 9, 65
Load, 11, 91
Mass, 6, 7, 53
Fixed, 9
Model, 41
Name, 25
Reference, 26
Set, 26–27
Shorthand, 25, 26
Ellipsoid, 63
ID, 60
Enclosing Structure, 6
Energy, 10
Dissipation, 77
Equations of motion, 104–106
Errors, 130
Event
ID, 24–25, 103
Matching, 98
Specifications, 103
Weighting, 103
Event file, 16, 37
Dynamic Parameter template, 40
Example, 15
Field, 41
Boolean, 43
Character, 43
Extractable, 43
Extracted, 43
Integer, 43
Real, 42
Repeat, 42
Suppressed, 43
Suppressible, 43
Tag, 15, 42
Value, 43
Fixed Mass specifications, 57
Leading tag, 41
Link specifications, 65
Link template, 40
Load specifications, 91
Mass specifications, 53
Model Information, 47
Model specifications, 47
Organization, 41
Output Information, 94
Parameter, 17, 41
Point specifications, 58
Run Information, 44
Shape specifications, 60
Specifications, 42
Legend, 42
Static Parameter template, 40
Substructure specifications, 51
Template, 37
Vehicle specifications, 49
Executing SISAME-3D, 16
Exerted force, 8
Extracted Parameter, *see* Parameter, Extracted
Extraction, 2, 6, 124
Assessment, 129
Controlling, 128
Design, 4
Implementation, 111–120
No feasible solution, 33
Parameter specifiers, 33
Process, 14
Refining, 128
Resimulation, 13, 14
Specifiers, 34
Time step, 45, 121
Weight and load-path, 127

F

Failure
Static, 67
Fiat methodology, 3, 106
Field, 16, 41
Parameter, 17, 41
Tag, 15
Filtering, 5, 121
Finite element modeling, 4, 125
Fit measure report, 13, 95
Fixed Mass, 9, 57
Aspect

Name, 28
Reference, 28
CG, 58
ID, 57
Orientation, 57
Position, 57
Specifications, 57
Force, 10, 11, 104
Damping, 10
Dynamic, 10
Exerted, 8
Inertia, 8
Magnifier, 10
Static, 10
Formulation, 104–110

G

Generalized force, 104
Generalized momentum, 105
Generalized vector, 104
Ground frame, 12
Guidelines, 124–131

I

ID, 17, 24
Aspect, 24
Component, 24
Element, 24
Event, 24
Link, 24
Load, 24
Mass, 24
Substructure, 24
Vehicle, 24
Wildcard, 27
Ill-conditioned least squares system, 130
Impulse, 11
Inactive Link, 65
Inertia
Moments, 54
Products, 54
Tensor, 54
Inertia force, 8
inertial frame, 12
Influence matrix, 104
Instrumentation, 125
Accuracy, 124
Additional, 124
Artificial, 125
File, 5
Output, 13
Filtering, 125
Location, 124
Preparation, 14, 125, 129
Quality, 125
Synthesized, 125
Integrated impulse, 11, 36

L

LCP, 110, 111
Limitations, 11, 17
Linear approximant, 113
Link, 6, 9, 65
Aspect
Name, 28
Reference, 28
Class, 9
Connection, 10
Damping force, 10
Defined, 9
Deflection, 10
Dynamic Aspect, *see* Dynamic Aspect
Dynamic magnification, 10
Energy, 10
Extracted, 9, 11, 17
Force, 10
Approximant, 113
Damping, 10
Dynamic, 10
Magnifier, 10
Representation, 113
Static, 10
ID, 24, 65
Inactive, 65
Name, 25
Parameter
Extraction, 113
Template, 40

Quantities, 10
Reference, 26
Relative velocity, 10
Sign convention, 10, 67, 85
Specifications, 65
Static Aspect, *see* Static Aspect
Static force, 10
Transform, 66
Load, 11, 91
Class, 11
Driving, 11, 91
Force, 11
Group, 11, 91
ID, 24, 91
Impulse, 11
Integrated impulse, 11
Name, 25
Quantities, 11
Reference, 26
Specifications, 91
Target, 11, 91, 126
Equations, 108
Log file, 16, 130

M

Magnifier, 10, 89–90, 127
AM, 90
LM, 90
Mass, 6, 7, 53
Acceleration, 8
Aspect
Name, 28
Reference, 28
C-V, 8
CG, 58
Class, 7
Constant velocity, 54
Constant-velocity, 8
Driven, 8, 54
Exerted force, 8
Fixed, 9
ID, 24, 53
Inertia force, 8
Instrumented, 8
Name, 25
Orientation, 55
Point, 56, 57
Position, 8, 55
Quantities, 8
Reference, 26
Shape, 56, 57
Simulated, 8, 11, 17, 53
Specifications, 53
Target, 8, 54
Equations, 106–108
Velocity, 8, 55
Weight, 8, 126
Extraction, 4, 11, 17, 126
Magnitude, 48
Master file, 16, 99
Event Information, 103
Event specifications, 103
Leading tag, 100
Organization, 99
Run Information, 101
Template, 100
Model, 1, 6, 7, 47
Axes, 12, 18
Component, *see* Component
Configuration, 11, 17, 126
Construction, 16
Degrees of freedom, 12, 18
Dimensional systems, 23
Element, *see* Element
Extraction, *see* Extraction
ID, 48
Orientation, 48
Output, 13
Parameter, *see* Parameter
Position, 48
Simulation, *see* Simulation
Size, 11, 17
Specifications, 47
Structure, *see* Structure
Substructure, *see* Substructure
Vehicle, *see* Vehicle
Velocity, 48
Weight, 48, 110, 127

Modeling
  Concepts, 6–14
Momentum, 105
Multiple-Event, 4, 98
  Usage, 129

N

Name, 24–31
  Aspect, 27
  Component, 25
  Element, 25
  Link, 25
  Load, 25
  Mass, 25
  Parameter, 29
  Substructure, 25
  Vehicle, 25
Numerical integration, 122
Numerical simulation, 121–122

O

One-way
  Dynamic Aspect, 86
  Static Aspect, 67
Orientation, 12, 19, 20
  Fixed Mass, 57
  Mass, 55
  Model, 48
  Output signal, 94
  Point, 58
  Shape, 60
  Substructure, 52
  Vehicle, 50
Output, 13, 94
  Effective, 13
  Extracted, 13
  Fit measure report, 13, 95
  Instrument, 13
  Specifications, 94
Output signal
  Files, 24, 25, 96
  Orientation, 94

P

Parameter, 17, 41
Block, 17, 30, 41, 42
  Extracted, 35
  Monotonic, 43
  Name, 29
  Reference, 29, 32, 35
  Specifier, 32
Boolean, 43
Character, 43
Conditioning, 14, 45, 95, 101, 117, 130
  Equivalence, 117
Damping, 45, 95, 102, 118, 128
Default, 43
  Transform, 34, 43
Extractable, 43
Extracted, 43
  Bound, 34
  Constraints, 119
  Estimate, 34
  Reference, 33
  Specifier, 34
Integer, 43
Name, 29
Real, 42
  Block, 42
  Tuple, 42
Reference, 29, 33
  Default, 33
  Extracted, 33
  Shorthand, 31
  Transform, 31, 33, 34
  Units, 33
Specifier, 31
  Extraction, 34
  Reference, 33
  Transform, 34
  Value, 32
Suppressed, 43
Suppressible, 43
Tag, 15, 42
  Leading, 41, 100
Transforming, 66, 68, 86
Tuple, 17, 30, 41, 42
  Name, 29
  Reference, 29, 32

Spatial, 18
Specifier, 32
Value, 42, 43
Phase I procedure, 33, 120
Plane, 64
ID, 60
Point, 8, 9, 56–58
CG, 58
Color, 59
ID, 58
Name, 28
Orientation, 58
Position, 58
Reference, 28
Size, 58
Specifications, 58
Transparency, 59
Position, 8
Generalized, 12, 18
Initial, 12, 18, 20
Resting, 12, 18, 19
Rotational, 20
POV-Ray, 13
Problems
Common, 130

R

Reference, 24–31
Ambiguous, 25
Aspect, 27
Shorthand, 27
Block, 29
Component, 25–26
Resolution, 26
Shorthand, 25, 26
Element, 25–26
Resolution, 26
Shorthand, 25, 26
Link, 26
Load, 26
Mass, 26
Parameter, 29
Shorthand, 31
Resolution, 26
Set, 26–27, 29
Shorthand, 25–27, 31
Substructure, 26
Tuple, 29
Vehicle, 26
Relative velocity, 10
Run
ID, 44, 101
Running SISAME-3D, 16

S

Shape, 8, 9, 56, 57, 60
Box, 61
Color, 55, 57, 61
Cone, 62
Cylinder, 62
Ellipsoid, 63
Name, 28
Orientation, 60
Plane, 64
Position, 60
Reference, 28
Size, 61
Specifications, 60
Sphere, 63
Transparency, 55, 57, 61
Shorthand references, 25–27
Signal
Comparison, 8, 11
File, 5, 35
Input, 35, 42
Output, 13, 36
UDS, 5, 35, 36, 96
Input, 35
Motion, 8
Output, 13, 36
Specifier, 56, 93
Target, 11
SimFil, 5, 36, 55, 92, 121
Simulation, 6, 121–122, 130
Numerical integration, 122
Time step, 45, 121
SISAME, 3
SISAME-3D

Applications, 124–131
Dimensional systems, 23
Event file, *see* Event file
Executing, 16
Filtering, 121
Formulation, 104–110
Guidelines, 124–131
History, 3
Implementation, 111–122
Model, *see* Model
Modeling, *see* Modeling
Overview, 1
Running, 16
Users Guide, 15–103
Solution
Computations, 109
Constraints, 35, 110–112, 119
Iteration, 45, 102, 115
Weight, 48, 49, 51, 127
Convergence, 45, 68, 89, 101, 102, 116, 118, 128
Direct, 111
Ill-conditioned, 130
Implementation, 111–120
Iteration, 45, 101, 102, 116, 128
Constraints, 45, 102, 115
No feasible ∼, 33
Optimality conditions, 110
Parameter, *see* Parameter
Phase I procedure, 33, 120
Pivoting, 112
Quality, 128
Relaxation, 45, 102, 116
Speed, 128
Sphere, 63
ID, 60
Static Aspect, 10, 67–84
Approximant, 113
Domain transitions, 115
Elastic, 69–76
AE, 72
BE, 73
EE, 71
LE, 70
SE, 75
Extraction, 113
Failure, 67
Inelastic, 77–84
AI, 78
BI, 79
SI, 81
Unloading, 77
Yielding, 77
Name, 28
One-way, 67
Reference, 28
Segment smoothness, 96, 118
Transform, 67
Universal Parameters, 67
Static force, 10
Structure, 6, 17
Contained, 6
Enclosing, 6
Member, 6
Model, 6, 7, 47
Substructure, 6, 7, 51
Vehicle, 6, 7, 49
Weight, 110, 127
Substructure, 6, 7, 41, 51
ID, 24, 51
Name, 25
Orientation, 52
Position, 52
Reference, 26
Specifications, 51
Velocity, 52
Weight, 51, 110, 127
Suppressed Parameter, 43
Suppressible Parameter, 43

T

Target
Equations
Acceleration, 107
Confidence bands, 3, 35, 45, 55, 76, 83, 92, 93, 101, 102, 118, 128
Force, 108
Impulse, 108, 116

Integrated Impulse, 108, 116
Load, 108
Mass, 106–108
Position, 107, 116
Supplementary, 117
Velocity, 107, 116
Weighted, 109
Weighting, 118
Integral, 116
Load, 11
Mass, 8
Signal, 11
Test run, 44, 101
Time step
Extraction, 45, 121
Simulation, 45, 121
Time-based weighting, 45
Transform
Dynamic, 86
Link, 66
Parameter
Default, 34, 43
Reference, 33, 34
Static, 67
Transparency, 55, 57, 59, 61

U

UDS file, 5, 35, 36, 96
Users Guide, 15–103

V

VeCor, 5, 14, 125
Vector
Generalized, 18
Vehicle, 6, 7, 41, 49
Design, 4
ID, 24, 49
Impacts, 4
Name, 25
Orientation, 50
Position, 50
Reference, 26
Specifications, 49
Velocity, 50
Weight, 49, 110, 127
Velocity, 8
Generalized, 12, 18
Initial, 18, 22

W

Warnings, 130
Weight, 8
Extraction, 4, 11, 17, 126
Magnitude, 48
Weighted target system, 109
Weighting, 103