



U.S. Department
of Transportation

**National Highway
Traffic Safety
Administration**

**DOT HS
Draft Report**

November 2007

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

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.

1. Report No. DOT HS		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle THE SISAME-3D PROGRAM: Structural Crash Model Extraction and Simulation				5. Report Date November 2007	
				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 7 th Street, S.W. Washington, DC 20590				13. Type of Report and Period Covered Draft Report October 2001 – November 2007	
				14. Sponsoring Agency Code	
15. Supplementary Notes Office of Crashworthiness Research/NHTSA, NRD-11					
16. Abstract This report contains the users guide and formulation for the SISAME-3D v.1.6 structural impact modeling program. SISAME-3D extracts lumped-parameter structural models from simulation and test-derived impact event 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 Lumped-parameter model, computer simulation, least squares, constrained optimization.			18. Distribution Statement This document is available to the public from the National Technical Information Service, Springfield, Virginia 22161.		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of pages 144	22. Price

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

- 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 12
 - 2.1.4 Degrees of Freedom 13
 - 2.1.5 Coordinate Frames 13
 - 2.2 SISAME-3D Outputs 14
 - 2.3 The Extraction Process 15

- 3 SISAME-3D Users Guide** **16**
 - 3.1 Running SISAME-3D 17
 - 3.2 Model Construction 17
 - 3.2.1 Model Configurations 18
 - 3.2.2 Degrees of Freedom (DOF) 19
 - 3.2.3 Coordinate Frames 19
 - 3.2.4 Generalized Spatial Parameter Tuples 19

3.2.5	Resting Position	20
3.2.6	Initial Position	21
3.2.7	Rotational Position	21
3.2.8	Point and Shape Position	22
3.2.9	Initial Velocity	22
3.2.10	Dimensional Systems	23
3.3	Identifiers, Names, and References	24
3.3.1	Identifiers	24
3.3.2	Names	25
3.3.2.1	Parameter Block Naming	25
3.3.2.2	Parameter Tuple Naming	26
3.3.3	References	26
3.3.3.1	Reference Resolution	26
3.3.3.2	Shorthand References	27
3.3.3.3	Link Mode Aspect References	27
3.3.3.4	Parameter References	28
3.3.3.5	Set References	28
3.4	Specifying Parameters	29
3.4.1	Parameter Specifiers	29
3.4.2	Parameter Block Specifiers	29
3.4.3	Parameter Tuple Specifiers	30
3.4.4	Defined Value Specifiers	30
3.4.5	Reference Specifiers	31
3.4.6	Default Specifiers	31
3.4.7	Expression Specifiers	32
3.4.8	Extraction Specifiers	33
3.5	Signal Files	34
3.5.1	Input Signal Files	34
3.5.2	Output Signal Files	34

3.5.3	UDS Signal Files	34
3.6	Event File Overview	35
3.6.1	Event File Template	35
3.6.2	Link Parameter Templates	38
3.6.3	Event File Organization	39
3.6.4	Event File Fields	39
3.6.5	Input Signal File Names	40
3.6.6	Specification Tables	40
3.7	Run Information	42
3.8	Model Structure	45
3.9	Vehicle Structure	47
3.10	Substructure	49
3.11	Mass Element	51
3.12	Fixed Mass Element	55
3.13	Point Aspect	56
3.14	Shape Aspects	58
3.14.1	Box	59
3.14.2	Cylinder	59
3.14.3	Cone	60
3.14.4	Sphere	60
3.14.5	Ellipsoid	61
3.14.6	Plane	61
3.15	Link Element	62
3.16	Static Aspects	65
3.16.1	Universal Static Parameters	65
3.16.2	Elastic Static Aspects	67
3.16.2.1	Static Type LE : Linear Elastic	68
3.16.2.2	Static Type EE : Exponential Elastic	69
3.16.2.3	Static Type AE : Arctangent Elastic	70

3.16.2.4	Static Type BE : Bilinear Elastic	71
3.16.2.5	Static Type SE : Segmented Elastic	73
3.16.3	Inelastic Static Aspects	75
3.16.3.1	Static Type AI : Arctangent Inelastic	76
3.16.3.2	Static Type BI : Bilinear Inelastic	77
3.16.3.3	Static Type SI : Segmented Inelastic	79
3.17	Dynamic Aspects	84
3.17.1	Universal Dynamic Parameters	85
3.17.2	Dampers	86
3.17.2.1	Dynamic Type LD : Linear Damper	87
3.17.2.2	Dynamic Type AD : Arctangent Damper	87
3.17.3	Magnifiers	88
3.17.3.1	Dynamic Type LM : Linear Magnifier	89
3.17.3.2	Dynamic Type AM : Arctangent Magnifier	89
3.18	Load Element	90
3.19	Output Information	93
3.20	Multiple-Event Modeling	99
3.20.1	Event Matching	99
3.20.2	Master File Overview	100
3.20.3	Master File Organization	100
3.20.4	Run Information	102
3.20.5	Event Information	104
3.20.5.1	Event Specifications	104
4	SISAME-3D Formulation	105
4.1	Rigid Mass Equations of Motion	105
4.2	Partitioned Equations of Motion	106
4.3	Mass Target Equations	107
4.4	Load Target Equations	109
4.5	Weighted Target System	110
4.6	Constrained Approximation Problem	111

5	SISAME-3D Implementation	112
5.1	LCP Solution	112
5.2	Link Force Representation	114
5.2.1	Linear Approximants	114
5.2.2	Domain Transitions	116
5.3	Extraction Iteration	117
5.4	Integral Domain Targets	117
5.5	Supplementary Target Equations	118
5.6	Target Equation Weighting	119
5.7	Extracted Parameter Constraints	120
5.8	Phase I Procedure	121
5.9	Filtered Data Representation	122
5.10	Time Step Selection	122
5.11	Numerical Integration	123
A	SISAME-3D Modeling Guidelines	125
A.1	Understanding Extraction	125
A.2	Instrumentation	126
A.3	Model Configuration	127
A.4	Weight Extraction	127
A.5	Dynamic Extraction	128
A.6	Controlling and Refining Extractions	129
A.7	Assessing the Extracted Model	130
A.8	Multiple-Events	130
A.9	Common Problems	131
A.10	Predictive Simulations	131
	References	133

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.	101

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 knowledge about the structures to be used to obtain more 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 with a range of static and dynamic characteristics.

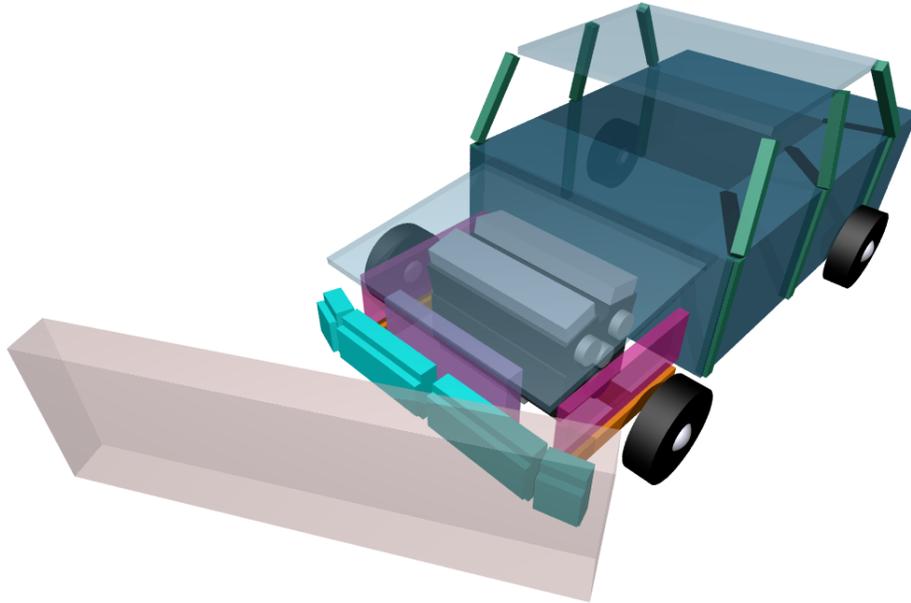


Figure 1.1: SISAME-3D angled offset vehicle impact model.

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 and inertia properties, motion histories of some or all of the Masses, and, optionally, inertial or applied load histories for 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 parameter bounds and estimates, clearances between components, constraints on load-path forces, or target loads for load-path subsets. SISAME-3D is designed to use the 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, 7], 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 [12] program with the aims of providing greater robustness and directly generating simulation-ready models. 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 input files are flexible and self-documenting, provides many defaults, and supports powerful referencing relationships between model components and parameters. Model extractions can incorporate 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 [8, 10].

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, animation outputs, and support for new modeling applications.

1.5 General Information

SISAME-3D is driven by self-documenting input 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 is written in object-oriented ANSI/ISO standard C++ and is available for a number of computing platforms.

The associated VeCor [11] program can be used to assess and preprocess instrumentation signals prior to modeling. The filtering used in SISAME-3D is based on the SimFil [8, 10] program.

This document applies to SISAME-3D version 1.6. The VeCor program is documented in [11]. The SimFil program is documented in [8, 10].

Information on current application versions and documentation can be obtained at <http://www-nrd.nhtsa.dot.gov/sisame/>.

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 **Link** Elements corresponding to the load-paths of the model. **Load** Elements can be added to specify driving or actuator loads or to represent a set of Link loads or Mass inertial loads. Mass, Fixed Mass, Link, and Load **Elements** are contained in **Structures**. Structures can contain other Structures. Structures and Elements are the **Components** of a SISAME-3D model.

Simulation models are models with all Mass weight and inertia properties 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 some undefined Mass weight or inertia properties and/or undefined Link static or dynamic 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.

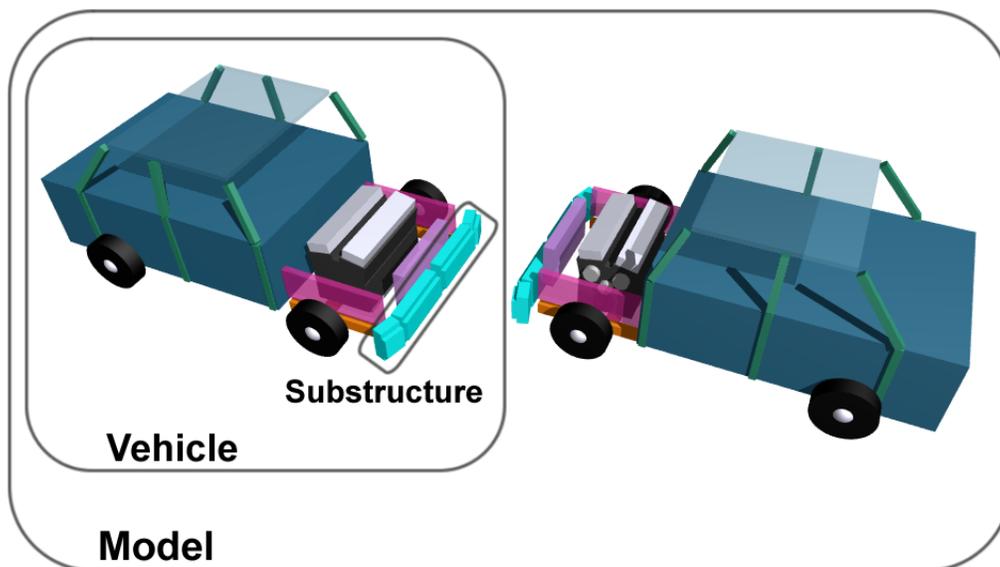


Figure 2.1: SISAME-3D Structure levels.

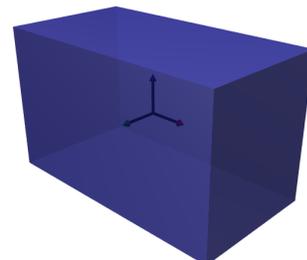
2.1.2 Elements

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).

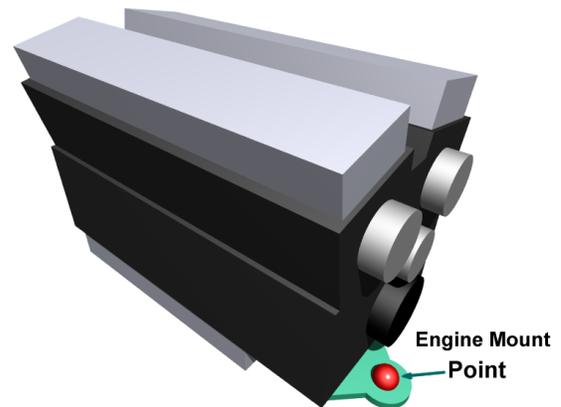
Points and Shapes

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).

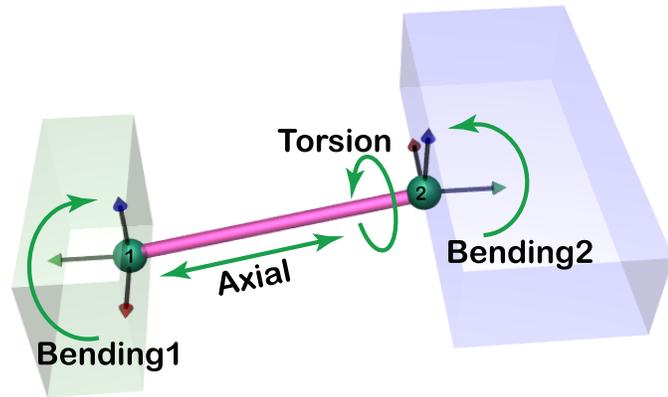
A **Shape** is a geometric shape at a fixed position on a Mass or Fixed Mass. Shapes control the contact interactions among Masses and Fixed Masses.

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. A Link is an idealization of a straight-line connector between two Points with spherical joints at each Point. The Link can compress and extend along the axis between the Points and the Masses at each end can rotate in all directions relative to the Link end Points. In this idealization there is no internal Link deformation so beam-like bending and shear are not modeled: the Bending Modes can be used to model internal bending and shear.



Each Link is connected to two Mass or Fixed Mass Points, p_1 and p_2 . The relative position and orientation of the coordinate frames at p_1 and p_2 determine the Link behavior. Because the Link is massless the net force/torque on the Link with respect to any point is always zero. A Link can have independent loading properties specified for any of the **Modes** of motion shown, allowing Links to represent a range of connection types.

- **Axial** Mode forces are generated along the line between p_1 and p_2 .
- **Torsion** Mode torques occur about the line between p_1 and p_2 .
- **Bending1** and **Bending2** Mode torques occur about a normal to the line between p_1 and p_2 at p_1 and p_2 , respectively. Forces occur at p_1 and p_2 to balance the torque such that the net force/torque on the Link is zero.
- **Rotation** Mode torques depend only on the relative orientation of p_1 and p_2 (and thus only on the orientation of the two Mass or Fixed Mass bodies).

When p_1 and p_2 pass through or near the same point the Axial Mode 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 Axial Link Mode can only experience tension since its length will never be less than its zero resting length.

Link Modes can have static and dynamic characteristics called **Aspects**.

A **Static Aspect** determines the static force generated by the Link Mode 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 Mode force. A range of elastic (non-dissipating) and inelastic (dissipating) Static Aspect types are available. Each Link Mode can have one Static 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 Mode's static force, if any. The instantaneous rotational damping can act about a different axis than the static torque for the Bending and Rotation Modes. Magnifiers multiply the static force by a factor that is a function of the deflection rate. Each Link Mode can have one Damper and one Magnifier Dynamic Aspect. A pure Damper Link Mode has no Static or Magnifier Aspects.

The Bending and Rotation Modes do not have a meaningful tension–compression distinction so symmetric static and dynamic characteristics should be used.

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.

The relevant quantities for each Link are:

Link Quantity	Symbol	Formula
deflection	x	<i>Mode-dependent</i>
relative velocity	r	<i>Mode-dependent</i>
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 the force and motion values are angular quantities for rotational Link Modes, and 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.

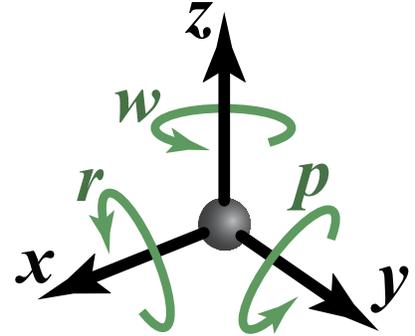
Two limitations arise when extraction runs include simulated Masses, allowing known portions of the structure to be included in the extraction process:

- 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 x , y , and z and the corresponding rotational axes as r , p , and 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.

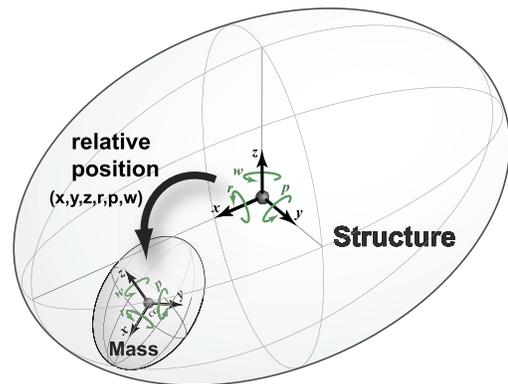


2.1.5 Coordinate Frames

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 (x, y, z, r, p, w) of linear and rotational coordinates. SISAME-3D uses the convention that translations (x, y, z) are applied first followed by the **Cardan angle** sequence of rotations (r, p, 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 $(v_x, v_y, v_z, v_r, v_p, v_w)$. 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.
- POV-RayTM model image files at specified time steps for generation of animations and verification and documentation of the model configuration.

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 *wa* 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 base-line 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

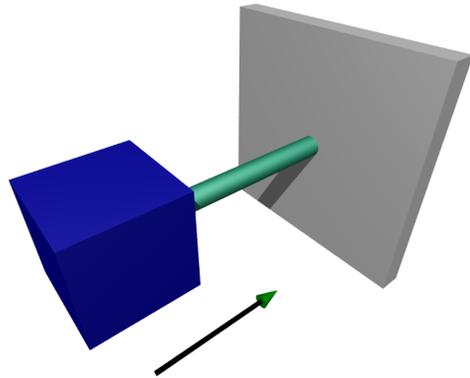
  MassID=Mass
  Weight=1000  IniV=50

  LinkID=Link
  Point1=Mass  Point2=Wall
  Mode=Axial
  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 input 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 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 Model or Run specification entities. 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 **Points** to provide Link and instrument attachment points and as points where driving Loads are applied, and **Shapes** to define their shapes and extents. 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 have **Modes** that specify the loading characteristics for each mode of linear and rotational Link motion and each Mode can have **Static** and **Dynamic** characteristics that provide its 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.

The sub-properties of Model Components including Mass Points and Shapes, Link Modes, and Link Static and Dynamic characteristics are called **Aspects** of their Components.

Components and Parameters have names that are used to refer to them outside of their own specifications. Names are constructed by dot-concatenating Component identifiers and Parameter 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 3.3.3 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.

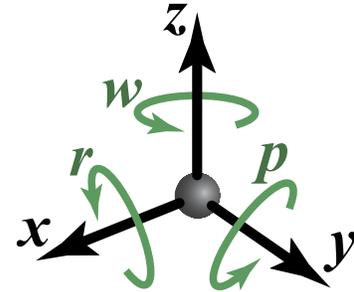
Model configurations are subject to two restrictions:

- 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 x , y , and z and the corresponding rotational axes as r , p , and 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 DOFs 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 $\text{IniP}=(x:-500\ z:80\ w:.2)$ 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 (x, y, z) are applied first followed by the **Cardan angle** sequence of rotations (r, p, 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 (1 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.

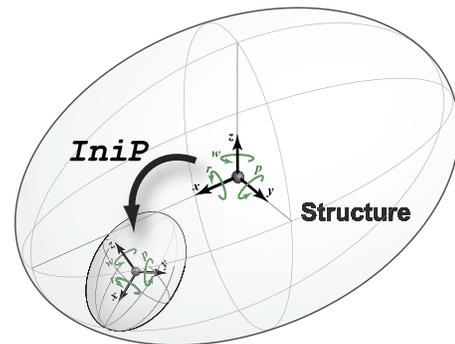
3.2.5 Resting Position

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, 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.

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.



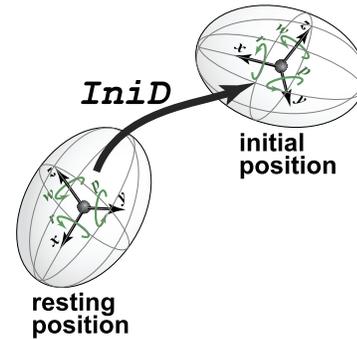
$\text{IniP}=(z:-220\ p:.8\ w:.2)$

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, *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.

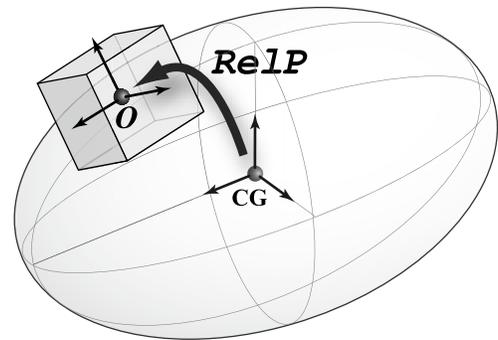
Output rotational position (Cardan angle) signals have the initial rotational positions computed by SISAME-3D but are able to track angles outside of $(-\pi, \pi]$ over the event time span using a heuristic algorithm to avoid signal discontinuities. This algorithm will produce reasonable signals as long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



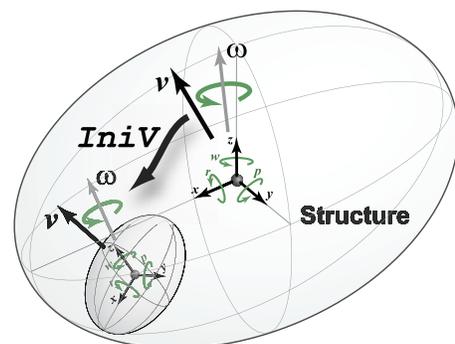
RelP=(x:65 z:90 p:.1 w:.2)

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.

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.



IniV=(x:50 r:1)

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 Identifiers, Names, and References

The constituents of a SISAME-3D Model are given identifiers that are unique within their scope. Full names for each entity are formed from the sequence of containing entity identifiers. Names are used in SISAME-3D input file references and in output file names. Identifiers and the construction of names is described below.

3.3.1 Identifiers

Each Component of a SISAME-3D Model and all of the Aspects and Parameters it contains has an **identifier** that is either a user-assigned **ID** or a predefined Aspect or Parameter **Tag**. Each Run and, for multiple-Event runs, each Event, also have identifiers, as do their Parameters.

Here are the rules for SISAME-3D identifiers:

- Identifiers can be up to twenty (20) characters long.
- Identifiers can include alpha-numeric characters and the characters `_ + - @ %` but may not begin with any of the characters `+ - @ %`.
- Identifiers may not contain internal space, tab, null, newline, or carriage return characters or any of these characters: `\ / | ~ ! # $ ^ & : ; . , " ' ' * = ? < > [] () { } .`
- Identifiers are case-sensitive.
- Identifiers must be case-insensitively distinct from the identifier of any sibling member entities of the containing entity.
- Identifiers must be distinct from the identifiers of all containing entity identifiers up the containment tree.
- Identifiers in different branches of the containment tree can be the same, so, for example, two Vehicles can contain Masses with the ID Engine.
- Components with IDs that are the same as Link Mode Tag (`Axial`, `Torsion`, `Bending1`, `Bending2`, or `Rotation`) or a Static or Dynamic Link 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.
- Identifiers make up parts of the output signal file names. The rules for identifiers 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 Names

The **name** of each Component, Aspect, Parameter, Parameter block, Parameter tuple, or specification entity is simply the name of its containing entity followed by a dot and its own identifier. Full names begin with the Event ID, which is simply **Event** in single-Event runs.

Letting the ID of an Entity be denoted by *EntityID*, full names take on forms such as:

EventID.RunID.DeltaOut

EventID.ModelID.VehicleID.MassID.PointID.Size

EventID.ModelID.VehicleID.LinkID.Torsion

Notes

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

3.3.2.1 Parameter Block Naming

Parameter blocks are arrays of Parameters indexed from 1 to N . Parameter blocks names follow the general scheme and their sub-blocks and Parameters have names constructed as shown in the table below

Parameter Block Names	
Block	<i>Entity.Block</i>
Sub-block from i to j	<i>Entity.Block $i:j$</i>
Sub-block from 1 to j	<i>Entity.Block :j</i>
Sub-block from i to N	<i>Entity.Block $i:$</i>
Parameter with index i	<i>Entity.Block i</i>

where *Entity* is the name of the containing entity and *Block* is the Parameter block Tag.

For example,

Event.Model.Car.LFFrame.Torsion.SI.X5:9

is the name of the sub-block from index 5 to 9 of a deflection Parameter block in a SI Static Aspect of the Torsion Mode of a LFFrame Link in a Car Vehicle.

3.3.2.2 Parameter Tuple Naming

Parameter tuples are sequences of Parameters indexed by a key value. For example, axis keys (such as \mathbf{x}), axis pair keys (such as \mathbf{xy}), and axis type keys (\mathbf{l} and \mathbf{r}) are used. Parameter tuple names follow the general scheme and their Parameters have names constructed as shown in the table below

Parameter Tuple Names	
Tuple	<i>Entity.Tuple</i>
Parameter keyed by <i>key</i>	<i>Entity.Tuple:key</i>

where *Entity* is the name of the containing entity and *Tuple* is the Parameter tuple Tag.

For example,

`Event.Model.Car.Engine.IniV:x`

is the name of the x-axis initial velocity Parameter of an `Engine` Mass in a `Car` Vehicle.

3.3.3 References

References are used in the SISAME-3D input files to create associations between Components and Aspects, such as the attachment Points of a Link, and relationships between Parameters. The hierarchical naming system supports references between entities within an Event as well as between Events.

A reference can be specified as the full name of the target entity. **Shorthand references** that omit prefix portions that are common between the reference context and the target entity are also supported.

3.3.3.1 Reference Resolution

The leading part of a reference containing any Event, Component, and Aspect identifiers is called the **locational reference**. Locational references are resolved by first searching up the containment hierarchy from the local context for an entity matching the locational reference's leading identifier in this order:

1. The local context entity itself.

2. Contained locational entities, if any.
3. For each level up the containment tree from the local context up to the root of the Event system:
 - (a) The containing entity.
 - (b) Contained locational entities of the containing entity.

When a match to the leading identifier is found the remaining identifiers in the locational reference are checked. If they do not match the name of a locational entity the search continues, moving up one level in the containment hierarchy.

Notes

- Locational reference identifier matching is case-sensitive.
- Output specifications are considered to be in the Model context.

3.3.3.2 Shorthand References

The reference resolution algorithm allows shorthand references to “nearby” entities to omit prefix portions of the names. So, for example, within Vehicle **CarA** a Link attachment of **Point1=Engine.XMount** suffices as a shorthand reference to the engine mount in **CarA**, but **Point1=CarB.Engine.XMount** is required to reference the engine mount in **CarB**.

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.

3.3.3.3 Link Mode Aspect References

Static Link Aspect names can use the specific Aspect Tag or the category Tag **Sta**, as in:

Event.Model.Vehicle.Link.Torsion.Sta

Dynamic Link Aspect names can use the specific Aspect Tag or the category Tags **Dpr** for a Damper and **Mag** for a Magnifier, as in:

Event.Model.Vehicle.Link.Axial.Dpr

Event.Model.Vehicle.Link.Bending1.Mag

The category Tag **Dyn** can be used for Link Modes with a single Dynamic Aspect, as in:

Event.Model.Vehicle.Link.Bending2.Dyn

3.3.3.4 Parameter References

Parameter references appear in the specifications for Parameters and Parameter blocks and tuples. These can be simple references to a single Parameter or expressions involving multiple references (see Sections 3.4.5 and 3.4.7).

Parameter, Parameter block, and Parameter tuple references are resolved by first resolving the locational portion of the reference using the algorithm specified in Section 3.3.3.1 and then identifying the specified Parameter, block, or tuple. A shorthand Parameter reference may not include a locational reference, in which case the location resolves to the local context where the reference appears. The locational portion of the reference should unambiguously specify the location of the desired Parameter: the resolution process will not look for multiple locational matches until it finds one matching the reference's Parameter, block, or tuple identifier.

Notes

- Parameter Tag matching is case-insensitive.

3.3.3.5 Set References

Set references simplify the specification of sets of Components and Aspects in places such as Load member and output Element lists. Set references are like names where each part can be an identifier, a wildcard, or a set of these. A set reference is of the form:

$$IdentifierList IdentifierList$$

where each *IdentifierList* is of the form:

$$Identifier \quad \text{or} \quad Identifier1, \dots, IdentifierN$$

where optional braces ($\{\}$) can enclose each *IdentifierList* and where each *Identifier* can be an entity identifier or one of the wildcards IDs $*$ or $[*]$ requesting all contained or member entities of the appropriate type, respectively.

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

- Each possible combination of *Identifier* s is used to generate a reference that is then resolved using the search algorithm given in Section 3.3.3.1 above.
- Entities of the appropriate type are added to the list. Nonexistent entities and entities of the wrong type are discarded without triggering a warning or error.

- The wildcard ID `*` selects all contained entities.
- The wildcard ID `[*]` selects all member entities.
- A leading wildcard ID is resolved in the local context where the reference occurs, *not* by searching up the containment hierarchy.

These are examples of set references:

```
Car.*          {LFWheelAssy,RFWheelAssy}.[*]          Car1,Car2.OccComp,Engine
```

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, *Specifier* can be a defined value, a reference to another Parameter or the Parameter's default value, an expression, or an extraction specifier.

3.4.2 Parameter Block Specifiers

Parameter blocks are specified with the syntax

$$Tag = Specifier \dots Specifier$$

where each *Specifier* can be a Parameter specifier or a single, block, or sub-block Parameter reference or expression (see Sections 3.3.2.1 and 3.3.3.4). 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 default values that are specified by a `#` character.

Repeat counts are supported in block specifiers to indicate the number of Parameters to generate for a specifier using any of the forms:

```

n(expression)
n{reference}
n#      or      #n
n?      or      n?(...)    or      ?n    or      ?n(...)

```

where n is a positive integer repeat count. An explicit repeat count is not required when (sub-)block references appear within the specifier's expression but if present is checked against the number of Parameters implied by the expression. An extraction specifier can include an estimate and/or bound specifier (see Section 3.4.8). Extraction specifiers that generate multiple Parameters need an explicit repeat count even if the estimate and/or bound specifiers include (sub-)block references. For example,

```

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

```

A repeat count must be consistent with the size of any block references within any of the estimate and bound specifiers.

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 or expression (see Sections 3.3.2.2 and 3.3.3.4). 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.

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

3.4.4 Defined Value Specifiers

A defined value specifier is simply a legal constant or expression 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 3.3.3.4). 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.*, a 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).

A reference to an extracted Parameter sets up a relationship where multiple Model Parameters depend on the same internal extracted Parameters. The implicit and explicit constraints on all of those Model Parameters must be satisfied by the shared internal Parameters. 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.

Reference specifiers should be wrapped in `{}` braces if they have identifiers containing `+` or `-` or if the reference is a named expression constant, such as `Pi`, or a literal for that Parameter type. SISAME-3D tries to be smart about distinguishing literal constants from references but they can be indistinguishable if literal constants are used as identifiers: wrapping references in `{}` safely disambiguates them.

Reference specifiers can appear in expressions (see Section 3.4.7).

3.4.6 Default Specifiers

Within a Parameter's specifier the default value can be referred to using the `#` character. Specifying a Parameter as its own default with the syntax `Tag=#` has no effect on the computation but will cause the default value to appear in the output Event file. Default specifiers can also be used in expression specifiers (see Section 3.4.7).

3.4.7 Expression Specifiers

Parameter specifiers can be expressions that may include literal and named constants, functions, and references. Expressions can include any number and nesting of subexpressions. Extractable Parameter specifiers can include references to extracted Parameters. Numeric Parameter expressions can contain the operators `+` `-` `*` `/` `^` as well as parentheses and supported functions. Expressions can be used in Parameter block and tuple contexts: each block or tuple reference within an expression must refer to the same number of elements or an error is generated.

- The operators have their usual meaning and precedence. The `^` operator is used for exponentiation and binds from right-to-left, so `2^3^4` means `2^(3^4)`.
- Division in an Integer Parameter context performs truncation as in common programming languages, not rounding to the nearest integer.
- The only named constant is `Pi = π` for use in real Parameter specifiers. So, for example, a rotation of 90 degrees can be specified as `Pi/2`.
- The functions currently supported are:
 - `rad(degrees)`
 - `acos(cosine)`
 - `asin(sine)`
 - `atan(tangent)`
 - `cos(radians)`
 - `sin(radians)`
 - `tan(radians)`
 - `sqrt(value)`

So, for example, a roll (x-axis) rotation of 30 degrees can be specified as `r:rad(30)`.

- Reference specifiers should be wrapped in `{}` braces if needed for disambiguation (see Section 3.4.5).
- The `#` default specifier can be used within expressions.
- Expressions can be nonlinear in terms of defined Parameters.
- Expressions can be linear in the underlying extracted Parameters. (Implicitly extractable Parameters can be nonlinear in extracted Parameters.)
- Expressions containing multiple Parameter references have their state set in this priority order:
 - Suppressed if any referenced Parameter is suppressed
 - Inactive if any referenced Parameter is inactive
 - Extracted if any referenced Parameter is extracted
 - Defined if all referenced Parameters are defined
- Tuple Parameter specifier expressions should not contain whitespace other than inside of parenthesized subexpressions or `{}` braces.

These are examples of expression specifiers:

2*237.5 (Engine.Weight*1.5)+50 #*1.1

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 constants, references to other Parameters, or expressions (see Sections 3.3.3.4 and 3.4.7).

For example, the specifier

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

indicates a stiffness that is estimated to be twice that of the FrontFrame Link's axial 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, *De1TOut*. The time step should be small enough to capture the full frequency content of interest. The *SimFil* program [8, 10] 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.

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 *De1TOut* 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 (see Section 3.2.10) 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 UDS INIVEL field is intended to be the scalar initial velocity of the whole vehicle or component and may not be set correctly in all UDS files, so INIVEL is not comparable to the (vector) *IniP* value and is not checked against it when UDS signal files are read.

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=
  Size= Color= Transp=
  Ref= XSc1= FScl= RSc1= DScl= MScl=
Mode=
  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=XRSFDE  Link=
OutClass=LoadTS   Qty=FIJfij  Load=
OutClass=LinkXY   Qty=SFDE    Link=
OutClass=LinkPar  Qty=SD      Link=
OutClass=FitRep   File=
OutClass=Event    File=
OutClass=POV      Motion= Camera= LookAt= Angle= Time= 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 Var= XiScl= FiScl= ConSS= AnySlope=
X=
F=

Sta=AI SO= FMax=

Sta=BI SO= FYC= FYT= SY=

Sta=SI Var= SU= ST= SC= XSlk= XiScl= FiScl= ConSS= AnySlope=
X=
F=

Universal Static Parameters

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

Dynamic Parameters

Dyn=LD DS1p= DS1k=

Dyn=AD DS1p= DMax= DS1k=

Dyn=LM MS1p=

Dyn=AM MS1p= 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 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 $\mathcal{C} = \text{A, B}$.

\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 $\mathcal{S} = \text{StringA, StringB}$.

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

The value of a field with Tag name `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 value 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 value entry can be specified based on the default value by using expressions containing `#` (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 3.3.3.4).

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	$S \leq 20$			
Descr	Description	S			
Class	Run class	$C \in S, E, T$	<i>see Notes</i>		
Title	Run title	S			
MaxIter	Max number of extraction iterations	$I \geq 0$	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	$\mathcal{R} > 0$	<i>computed</i>		
DelTSim	Simulation time step	$\mathcal{R} > 0$	<i>computed</i>		
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 $MultPD \leq 1$). 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.
- 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.

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	$S \leq 20$			
Descr	Description	S			
DOF	Degrees of Freedom	$(\mathcal{C}) \subseteq (\text{xyzrpw})$			
Gravity	Gravity direction	$S \in \pm z, \pm x, \pm y$	<i>none</i>		
DimSys	Dimensional system	$S \in \text{SI, Metric, English}$			
WeightMag	Weight magnitude (in mass units)	$\mathcal{R} > 0$	\bar{m}		
Weight	Model weight (in mass units)	$\mathcal{R} > 0$	<i>computed</i>		?
IniP	Resting position relative to the inertial frame	(\mathcal{R})	(0)		
IniD	Initial displacement from the resting position	(\mathcal{R})	(0)		
IniV	Initial velocity	(\mathcal{R})	(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 in this case but provides a check on the weights. Specifying *Weight*=# causes the computed Model weight to appear in the output Event file.
 - 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. Specifying *Weight*=? or *Weight*=# causes the extracted Model weight to appear in the output Event file.
 - An expression specified for *Weight* can contain references but must resolve to a defined value or an expression that is linear in the extracted Parameters so that the corresponding equality constraint can be generated. Default-based expressions other than *Weight*=# would be self-inconsistent and are not allowed.
- *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	$S \leq 20$			
Descr	Description	S			
Make	Make of Vehicle	S			
Model	Model of Vehicle	S			
Year	Model year of Vehicle	I			
Weight	Vehicle weight (in mass units)	$\mathcal{R} > 0$	<i>computed</i>		?
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	$\mathcal{R} \geq 0$	<i>hierarchy</i>		
ZeroSm	Default time-zero tail smoothing frequency	$\mathcal{R} > 0$	<i>hierarchy</i>	N	
EndSm	Default final time tail smoothing frequency	$\mathcal{R} > 0$	<i>hierarchy</i>	N	
ConV	Default velocity confidence band factor	$(\mathcal{R}) > 0$	<i>hierarchy</i>	N	
ConP	Default position confidence band factor	$(\mathcal{R}) > 0$	<i>hierarchy</i>	N	
ConSS	Default smoothness confidence band factor	$\mathcal{R} > 0$	<i>hierarchy</i>	N	
ConI	Default impulse confidence band factor	$(\mathcal{R}) > 0$	<i>hierarchy</i>	N	
ConJ	Default integrated impulse confidence band factor	$(\mathcal{R}) > 0$	<i>hierarchy</i>	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 Structures. Specifying the **Weight** is optional in this case but provides a check on the weights. Specifying **Weight=#** causes the computed Vehicle weight to appear in the output Event file.
 - 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 Structures). 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. Specifying `Weight=?` or `Weight=#` causes the extracted Vehicle weight to appear in the output Event file.

- An expression specified for `Weight` can contain references but must resolve to a defined value or an expression that is linear in the extracted Parameters so that the corresponding equality constraint can be generated. Default-based expressions other than `Weight=#` would be self-inconsistent and are not allowed.
- `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 Parameters 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 Parameters 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	$S \leq 20$			
Descr	Description	S			
Weight	Substructure weight (in mass units)	$\mathcal{R} > 0$	<i>computed</i>		?
IniP	Resting position relative to the Structure frame	(\mathcal{R})	(0)		
IniD	Initial displacement from the resting position	(\mathcal{R})	(0)		
IniV	Initial velocity	(\mathcal{R})	(0)		
Cutoff	Default cutoff frequency	$\mathcal{R} \geq 0$	<i>hierarchy</i>		
ZeroSm	Default time-zero tail smoothing frequency	$\mathcal{R} > 0$	<i>hierarchy</i>	N	
EndSm	Default final time tail smoothing frequency	$\mathcal{R} > 0$	<i>hierarchy</i>	N	
ConV	Default velocity confidence band factor	$(\mathcal{R}) > 0$	<i>hierarchy</i>	N	
ConP	Default position confidence band factor	$(\mathcal{R}) > 0$	<i>hierarchy</i>	N	
ConSS	Default smoothness confidence band factor	$\mathcal{R} > 0$	<i>hierarchy</i>	N	
ConI	Default impulse confidence band factor	$(\mathcal{R}) > 0$	<i>hierarchy</i>	N	
ConJ	Default integrated impulse confidence band factor	$(\mathcal{R}) > 0$	<i>hierarchy</i>	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 and Structures. Specifying the `Weight` is optional in this case but provides a check on the weights. Specifying `Weight=#` causes the computed Substructure weight to appear in the output Event file.
 - 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 and Structures). If the Substructure `Weight` is unspecified or if `Weight=?` or `Weight=#` is specified no total weight constraint is generated and the Substructure weight is an (implicitly) extracted value. Specifying `Weight=?` or `Weight=#` causes the extracted Substructure weight to appear in the output Event file.

- An expression specified for *Weight* can contain references but must resolve to a defined value or an expression that is linear in the extracted Parameters so that the corresponding equality constraint can be generated. Default-based expressions other than *Weight=#* would be self-inconsistent and are not allowed.
- *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 Parameters 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 Parameters 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	$\mathcal{C} \in \mathcal{S}, \mathcal{C}, \mathcal{T}, \mathcal{D}, \mathcal{d}$	<i>see Notes</i>		
Weight	Weight (in mass units)	$\mathcal{R} > 0$	<i>none</i>		?
Inertia	Inertia tensor	(\mathcal{R})	<i>none</i>		
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	$\mathcal{R} \geq 0$	<i>hierarchy</i>		
ZeroSm	Time-zero tail smoothing frequency	$\mathcal{R} > 0$	<i>hierarchy</i>	N	
EndSm	Final time tail smoothing frequency	$\mathcal{R} > 0$	<i>hierarchy</i>	N	
ConA	Acceleration confidence band	$(\mathcal{R}) > 0$	<i>see Notes</i>		
ConV	Velocity confidence band factor	$(\mathcal{R}) > 0$	<i>hierarchy</i>	N	
ConP	Position confidence band factor	$(\mathcal{R}) > 0$	<i>hierarchy</i>	N	
TarSignal	Target motion signal specifiers	(\mathcal{S})	<i>none</i>		
DrvSignal	Driving motion signal specifiers	(\mathcal{S})	<i>none</i>		
CmpSignal	Comparison motion signal specifiers	(\mathcal{S})	<i>none</i>		
PointID	Point ID	$\mathcal{S} \leq 20$			
ShapeID	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

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

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

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

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. 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.

The inertia tensor entries must obey certain rules to represent a realizable rigid body:

$$\begin{array}{lll} i_{xx} + i_{yy} \geq i_{zz} & i_{yy} + i_{zz} \geq i_{xx} & i_{zz} + i_{xx} \geq i_{yy} \\ i_{xx} \geq 2 * i_{yz} & i_{yy} \geq 2 * i_{zx} & i_{zz} \geq 2 * i_{xy} \\ i_{xx} * i_{yy} \geq i_{xy}^2 & i_{yy} * i_{zz} \geq i_{yz}^2 & i_{zz} * i_{xx} \geq i_{zx}^2 \end{array}$$

- *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 [8, 10] 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 $2.5 \times \text{Cutoff}$. 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 $10 \text{ g} \sqrt{\tilde{w}/w_i}$ (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 $100 \text{ radians/s}^2 \sqrt{\tilde{I}/I_i}$, 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 high weight/inertia Masses.

- *ConV* and *ConP* are confidence band factor tuples for the velocities and positions of target and comparison Masses. They are keyed by axis type, **l** for linear and **r** for rotational. The weighting of these target equations is based on *ConA*. *ConV* and *ConP* default to the enclosing Structure's corresponding Parameters and then to **1**. *ConP* defaults to the enclosing Structure's *ConP* and then to **1**. *ConV* and/or *ConP* Parameters 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 separate field with the same **Signal** Tag.
 - 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.)
- The **CG Point Size**, *Color*, and *Transp* (but not *RelP*) can be specified following a **PointID=CG** field to override their defaults.
- Specifiers of the form *ShapeID* 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	$\mathcal{S} \leq 20$			
ShapeID	Shape ID	$\mathcal{S} \leq 20$			

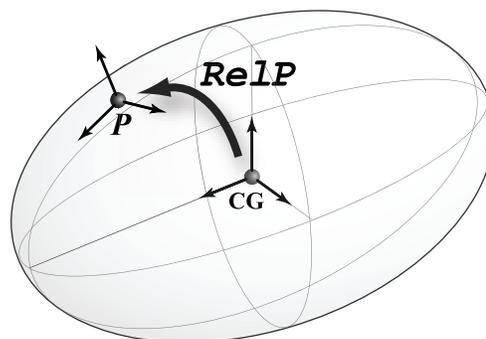
Fixed Mass Specification Notes

- **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 Fixed 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 Fixed 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 Fixed Mass Point. Any number of Points can be specified for each Fixed Mass following the general Fixed Mass specifications. (See Section 3.13.)
- The **CG Point Size**, **Color**, and **Transp** (but not **RelP**) can be specified following a **PointID=CG** field to override their defaults.
- Specifiers of the form **ShapeID** are leading Tags for the specifications of a Fixed 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 Fixed 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. Each Mass and Fixed Mass has an automatically generated Point with ID `CG` with the matching coordinate frame and that, if the Mass has an associated weight, is taken to be its center of gravity. 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	$S \leq 20$			
RelP	CG-relative generalized position	(\mathcal{R})	$(\mathbf{0})$		
Size	Size (radius) of Point sphere	(\mathcal{R})	<i>see Notes</i>		
Color	Color of Point sphere	S	<i>see Notes</i>		
Transp	Transparency of Point sphere	$\mathcal{R} \in [0, 1]$	0		

Point Specification Notes

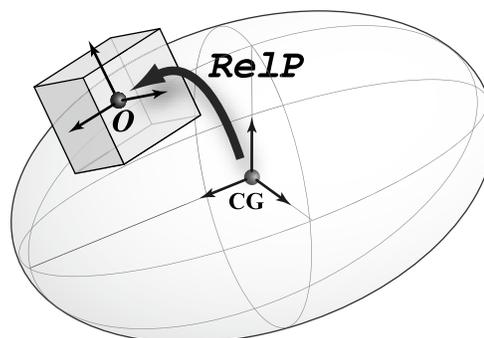
- `PointID` is the leading Tag for the Point specifications.
- Any number of Points can be specified following the general specifications of a Mass or FixedMass (Point and Shape specifications can be interspersed).
- `RelP` is an axis-keyed Parameter tuple that specifies the generalized position of a 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 `SpringGreen`. The *Color* of other Points defaults to `SpringGreen`.
- *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** *Size*, *Color*, and *Transp* (but not *RelP*) can be specified following a `PointID=CG` field to override their defaults.

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	Size of Shape	(\mathcal{R})			
Color	Color of Shape	\mathcal{S}	<i>see Notes</i>		
Transp	Transparency of Shape	$\mathcal{R} \in [0, 1]$	0		

Shape Specification Notes

- BoxID, CylinderID, ConeID, SphereID, EllipsoidID, and PlaneID are 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. 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

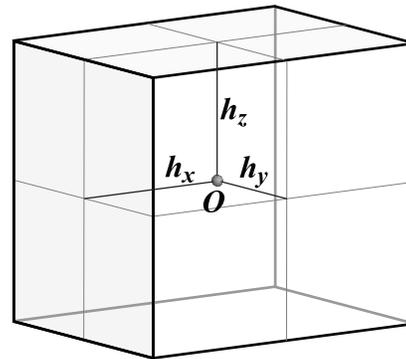
A Box is a hyperrectangle with its coordinate frame origin at its center and axes parallel to its sides.

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

Examples:

Size=(r:200)

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



Size=(r:h x:h_x y:h_y z:h_z)

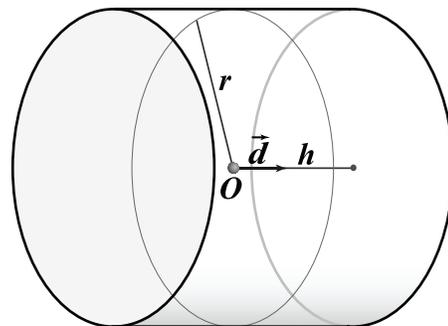
3.14.2 Cylinder

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

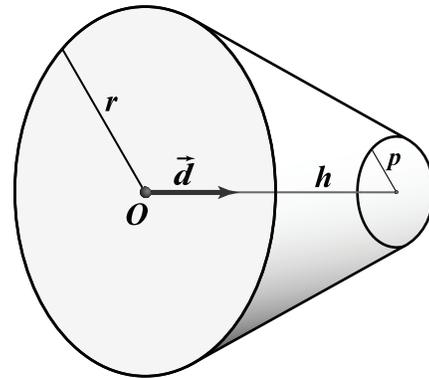
3.14.3 Cone

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, $d:h$, and an optional tip radius, $p:p$, for a capped cone. Any of the three linear axes can be designated as the directional axis. The value of h can be negative to point the Cone along the negative directional axis (equivalent to a *RelP* rotation of π about a non-directional axis).

Example:

Size=(r:100 x:50)



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

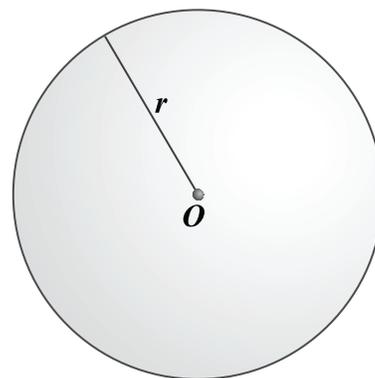
3.14.4 Sphere

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

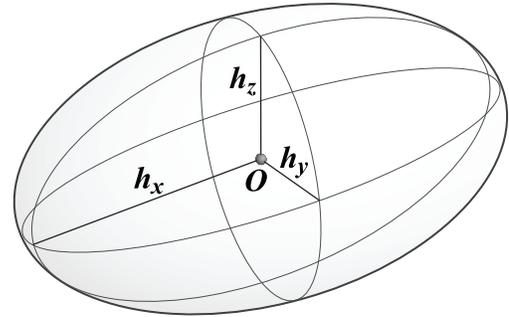
An Ellipsoid has its coordinate frame origin at its center with axes coincident with its principal axes.

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:h x:h_x y:h_y z:h_z)

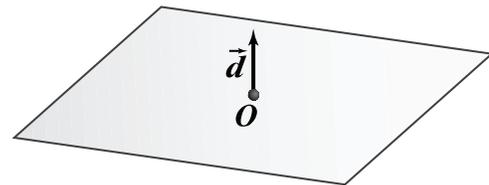
3.14.6 Plane

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	$S \leq 20$			
Descr	Description	S			
Class	Class	$C \in D, E$	<i>see Notes</i>		
Point1	Attachment Point #1	S			
Point2	Attachment Point #2	S			
Size	Size (radius) of Link cylinder	(\mathcal{R})	<i>see Notes</i>		
Color	Color of Link cylinder	S			
Transp	Transparency of Link cylinder	$\mathcal{R} \in [0, 1]$	0		
Ref	Reference Link	S	<i>none</i>		
XSc1	Deflection scale factor (X_{\otimes})	$\mathcal{R} > 0$	1		
FSc1	Static force scale factor (F_{\otimes})	$\mathcal{R} \geq 0$	1		
RSc1	Relative velocity scale factor (R_{\otimes})	$\mathcal{R} > 0$	1		
DSc1	Damping force scale factor (D_{\otimes})	$\mathcal{R} \geq 0$	1		
MSc1	Dynamic Magnifier scale factor (M_{\otimes})	$\mathcal{R} \geq 0$	1		
Mode	Link loading mode	$S \in \text{Axial, Torsion, Bending1, Bending2, Rotation}$	Axial		
Sta	Static Aspect type	$S \in \text{LE, EE, AE, BE, SE, AI, BI, SI}$	<i>none</i>		
Dyn	Dynamic Aspect type	$S \in \text{LD, AD, LM, AM}$	<i>none</i>		

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 *Point1* or *Point2* refer to a Mass or FixedMass, omitting the Point ID, the CG Point 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 in 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 default tail deflections. Inactive Links are counted in the defined Link tallies. Parametric outputs are available for inactive Links.
- **Size** is a Parameter tuple indicating the radius of the cylinder representing the Link in model images, for example **Size=(r:50)**. Only the **r** specifier is allowed in a Link **Size**. The default Link **Size** is half of the smaller of the *Point1* and *Point2* sphere radii.
- **Color** specifies the color of the cylinder representing the Link in model images (see Section 3.14). The **Color** can be any POV-Ray color specifier such as **MediumBlue** or **rgb<1,.5,.5>**. The Link Color defaults to **VioletRed**.
- **Transp** specifies the transparency of the cylinder representing the Link in model images (see Section 3.14). **Transp** defaults to zero (opaque) and can be any value from zero to one.
- **Ref** identifies a Link whose modes and Static and Dynamic Aspects this Link's Aspects, if not specified, will refer to. Specified Link modes override the corresponding modes of the referenced Link even if they contain no Static or Dynamic Aspects. Link color, transparency, and transform Parameters (see below) will refer to the referenced Link if unspecified. Acyclic chains of **Ref** references are supported. Note that the referenced Link's deflection range determines any default tail deflections.
- **Link transforms** provide an easy way to modify Static and Dynamic Aspect behavior. **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. Link transforms alter the internal behavior of defined and extracted Link Parameters but not their external value.
- **Mode** indicates the Link loading mode that the next Static and/or Dynamic specifications apply to:
 - **Axial** is the linear deflection along the Link axis that passes through *Point1* and *Point2*.
 - **Torsion** is the rotation about the Link axis passing through *Point1* and *Point2*.
 - **Bending1** and **Bending2** are the components of the rotations at *Point1* and *Point2*, respectively, that are orthogonal to the Link axis passing through *Point1* and *Point2*.
 - **Rotation** is the relative rotation of the Masses containing *Point1* and *Point2* and is independent of the specific locations of the Points on the Masses.
 - The Static and Dynamic Aspect specifications are in units of linear force and motion for the Axial mode and in units of rotational force (torque) and motion for the other modes.
 - The **Mode** defaults to **Axial**.
- **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 identifier 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. The Static Aspects are described in terms of forces and deflections: for rotational Link Modes these are torques and angular deflections but otherwise there is no difference in behavior from the Axial Link Mode.

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 the behavior of Static Aspects. **Scale factor** transforms X_{\otimes} and F_{\otimes} scale the effective Static characteristic along the deflection and force axes. **Offset** transforms X_{Δ} and F_{Δ} add offset values to the effective Static characteristic deflections and forces after the effects from any scale factor transforms are applied.

Static failure can be specified to occur if the deflection reaches or goes below a given $X_{\text{Min}}^{\text{Fail}}$ value or if it reaches or goes above a given $X_{\text{Max}}^{\text{Fail}}$ 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	$C \in \mathcal{C}, \mathbb{T}$	<i>none</i>	N	
XSc1	Deflection scale factor (X_{\otimes})	$\mathcal{R} > 0$	1		
FSc1	Force scale factor (E_{\otimes})	$\mathcal{R} \geq 0$	1		
XDe1	Deflection offset (X_{Δ})	\mathcal{R}	0		
FDe1	Force offset (E_{Δ})	\mathcal{R}	0		?
XFailMin	Tension failure deflection ($X_{\text{Min}}^{\text{Fail}}$)	\mathcal{R}	<i>none</i>	N	
XFailMax	Compression failure deflection ($X_{\text{Max}}^{\text{Fail}}$)	\mathcal{R}	<i>none</i>	N	

Universal Static Parameter Notes

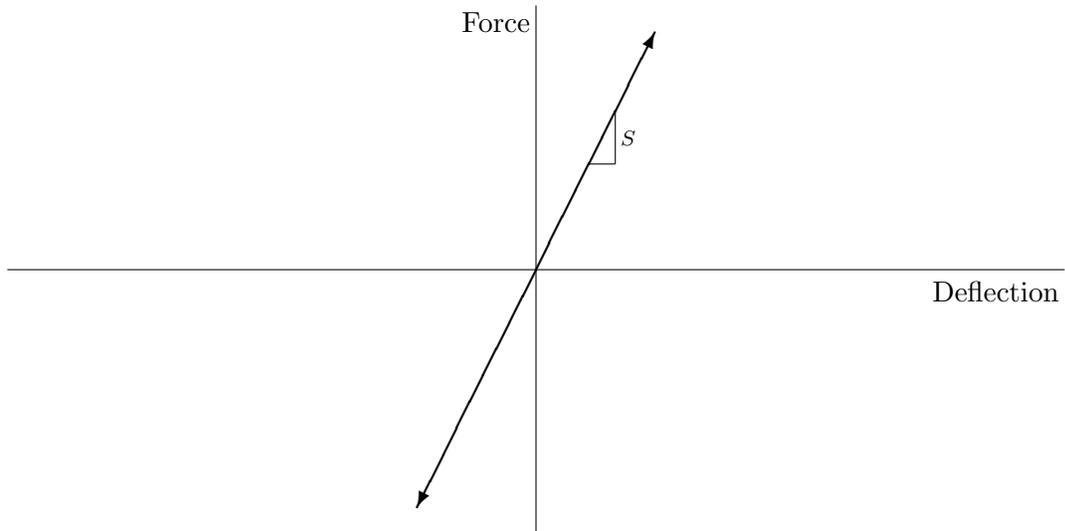
- Extracted one-way Static Aspects can hamper the solution convergence. Reducing the *ConPC*, *ConPD*, *MultPD*, and/or (for segmented Static Aspects) *ConSS* values may improve convergence in such cases.
- Static transforms alter the internal behavior of defined and extracted Static Parameters but not their external value. Parameters must obey their value range constraints both before and after the effects of any Link and Static transforms are applied.
- The effects of X_{Δ} and E_{Δ} are applied after the Static scale factor transforms so their effective values are transformed by Link-level (but not Static) scale factor transforms.
- The effect of X_{Δ} and E_{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 $E_{\Delta} \neq 0$ 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.
- $X_{\text{Min}}^{\text{Fail}}$ and $X_{\text{Max}}^{\text{Fail}}$ 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

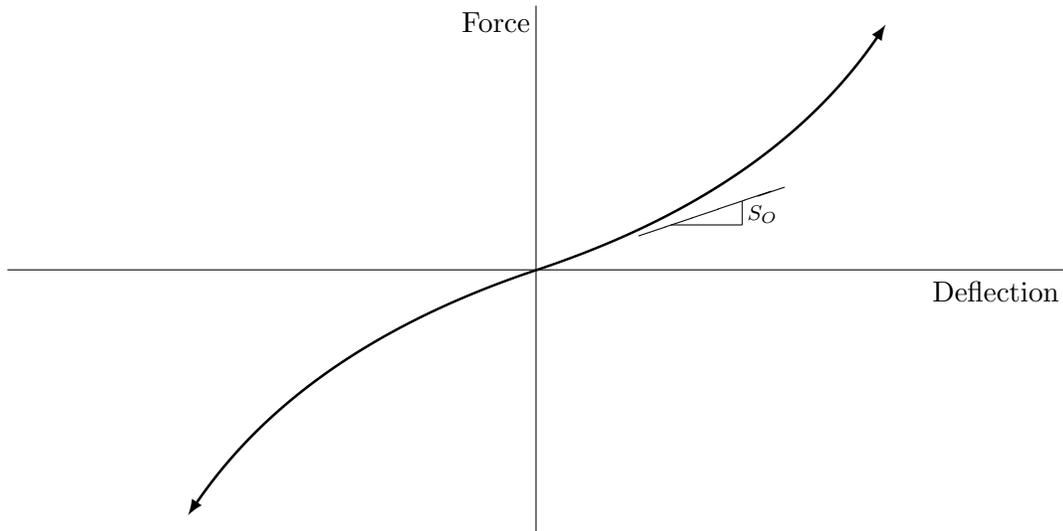


Static type LE is linear elastic with **force** and **stiffness**

$$f^S = Sx \quad s^S = S.$$

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**

$$f^s = S_o x + C \sigma(x) |x|^P \quad s^s = S_o + C P |x|^{P-1}$$

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 $|x|$.

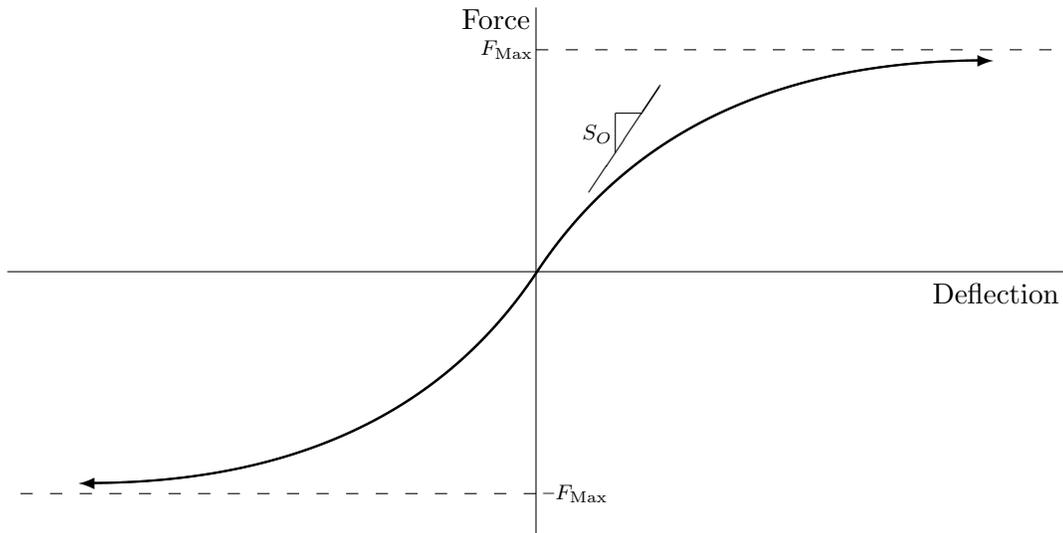
Static Type **EE** Parameters

Tag	Description	Value	Default	N	?
S0	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 . P cannot be extracted with deflection scaling (X_\otimes) (other than the default of 1.0).
- **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.

3.16.2.3 Static Type AE : Arctangent Elastic



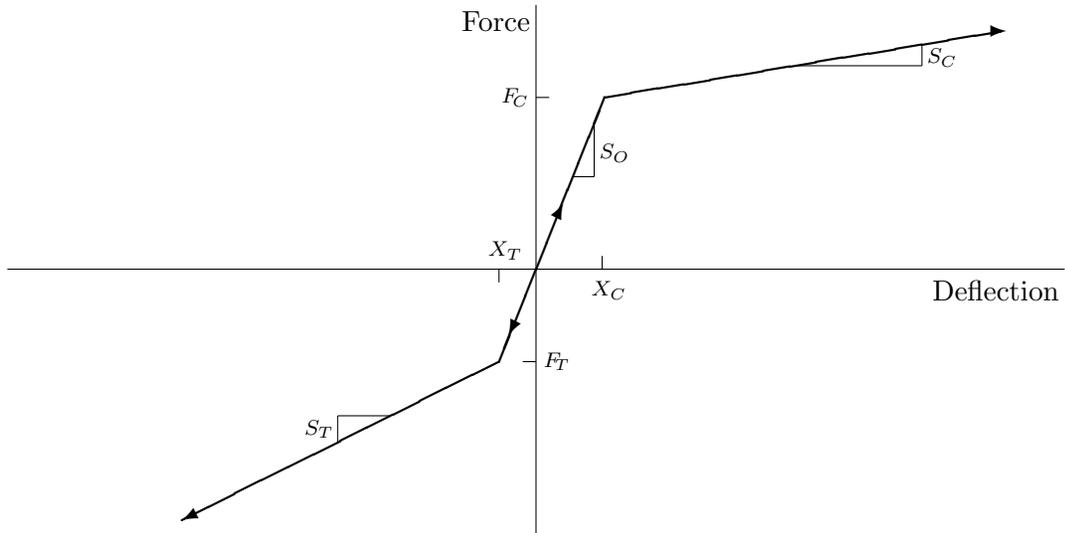
Static type AE is arctangent elastic with **origin stiffness** S_o and asymptotic **maximum force** magnitude F_{Max} .

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

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

Static Type AE Parameters

Tag	Description	Value	Default	N	?
S0	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

$$f_o^s = S_o x \quad s_o^s = S_o.$$

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

$$f_c^s = F_c + [x - X_c] S_c \quad s_c^s = S_c.$$

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

$$f_t^s = F_t + [x - X_t] S_t \quad s_t^s = S_t.$$

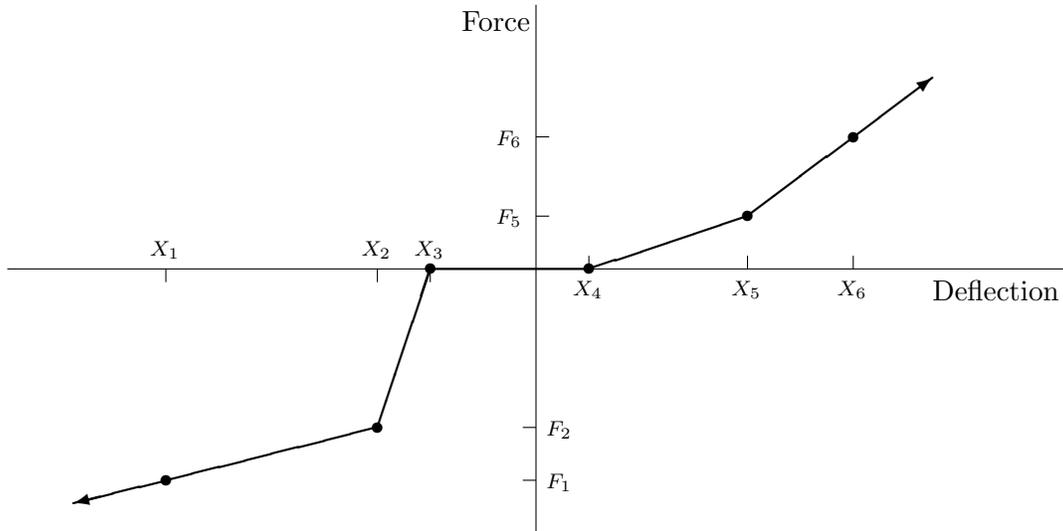
Static Type **BE** Parameters

Tag	Description	Value	Default	N	?
S0	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 $FC=S_O*X_C$ or $FT=S_O*X_T$, or alternatively, $S_O=FC*/X_C$ or $S_O=FT*/X_T$.
- 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 $SC=S_O$ or $ST=S_O$.
- $F_T \neq -F_C$ 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 $i = 1, \dots, N$ 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.

A **symmetric** variant has 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	?
Var	Variant selector	$\mathcal{C} \in \mathcal{S}$	<i>none</i>		
XiSc1	Deflection block scale factor ($X_{i\otimes}$)	$\mathcal{R} > 0$	1		
FiSc1	Force block scale factor ($F_{i\otimes}$)	$\mathcal{R} \geq 0$	1		
ConSS	Segment smoothness confidence band factor	$\mathcal{R} > 0$	<i>hierarchy</i>	N	
AnySlope	Extracted segment slope control	\mathcal{B}	False		
X	Deflection (X_i) block	$[\mathcal{R} \nearrow]$	<i>see Notes</i>		
F	Force (F_i) block	$[\mathcal{R}]$?

Static Type SE Notes

- **Var=S** selects the symmetric SE variant. The default is the non-symmetric variant.
- $X_{i\otimes}$ and $F_{i\otimes}$ scale the effective 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 $\text{ConSS} \cdot 5000(\tilde{w}/\tilde{x})$ 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 $?(>Fi)$ for the entry for F_{i+1} , or $?n(>Fi:)$ 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 3.3.2.1).
- Default **X** block deflections can be selected by entering **#** instead of a value. For Links between instrumented Masses SISAME-3D uses the actual deflection range to set default X_1 or X_N tail deflections. For symmetric SE Aspects the default tail deflection values are $X_1 = 0$ and $X_N = \max(|x|)$. Non-tail default deflections are evenly spaced between the surrounding deflection values. The repeat count form $n\#$ can be used to indicate n successive default 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.

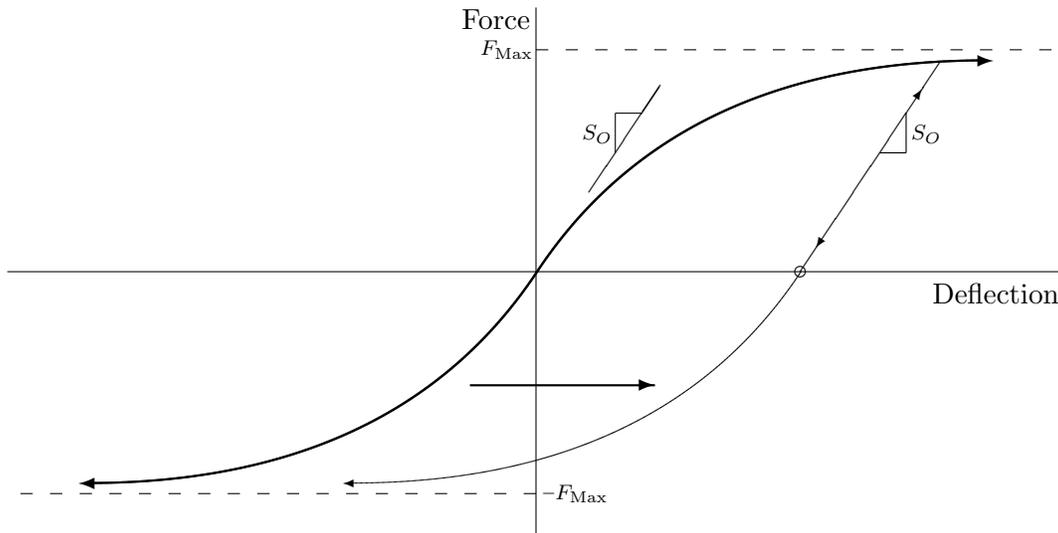
One-way Aspects do not yield (thus the unloading path and opposite boundary do not shift) while the force is clipped at zero.

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

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

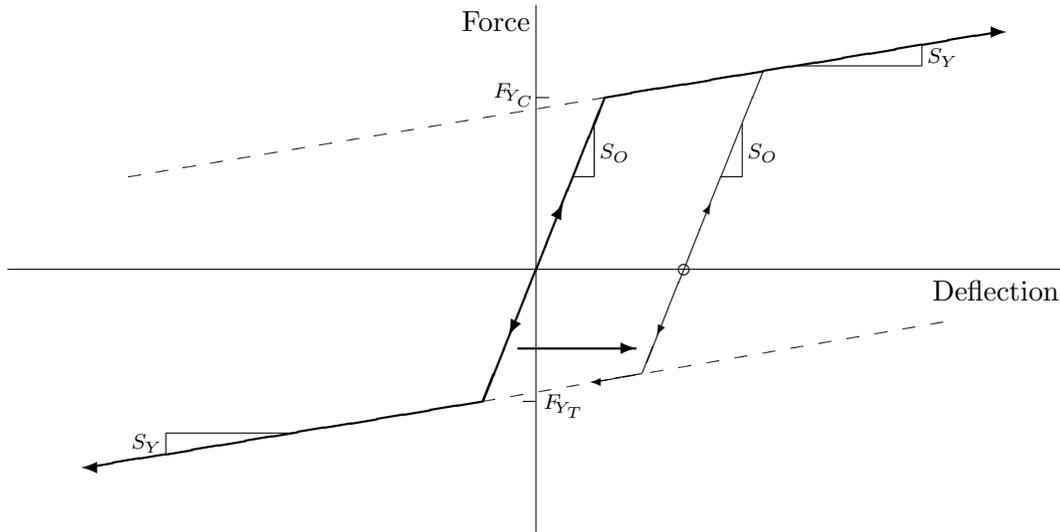
Yielding occurs during traversal of the compression or tension boundaries. The opposite boundary shifts as necessary so that the unloading line intersects it.

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	?
S0	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_{YC} , initial **tension yield force** F_{YT} , and **yielding stiffness** of S_Y .

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

$$f_o^s = S_O x \quad s_o^s = S_O.$$

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

$$f_c^s = F_{YC} + [x - X_{YC}] S_Y \quad s_c^s = S_Y.$$

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

$$f_t^s = F_{YT} + [x - X_{YT}] S_Y \quad s_t^s = S_Y.$$

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	?
S0	Origin and unloading stiffness (S_O)	$\mathcal{R} \geq S_Y$?
FYC	Compression yield force (F_{YC})	$\mathcal{R} \geq 0$	$-F_{YT}$	N	?
FYT	Tension yield force (F_{YT})	$\mathcal{R} \leq 0$	$-F_{YC}$	N	?
SY	Yield stiffness (S_Y)	$S_O \geq \mathcal{R} \geq 0$		N	?

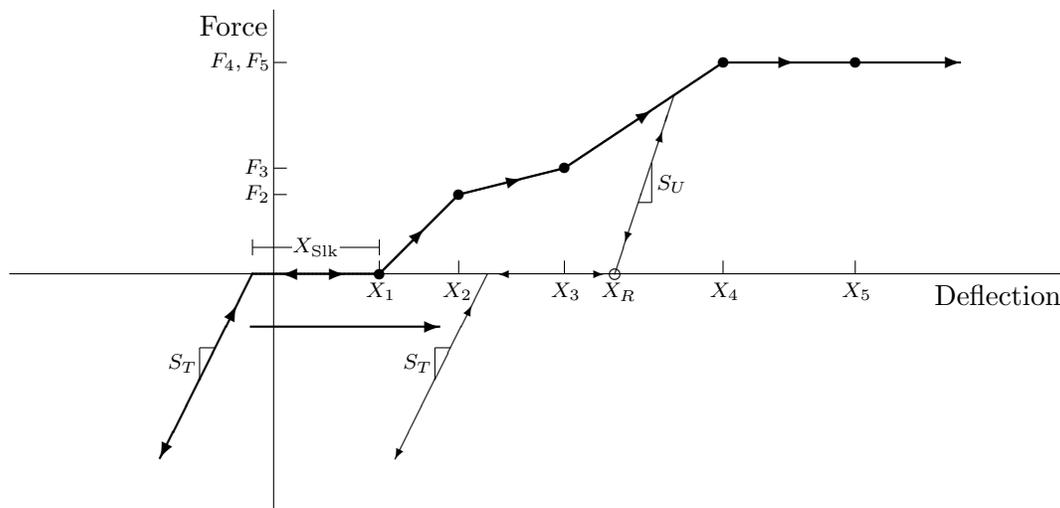
Static Type BI Notes

- If $S0 = 0$ the force is constant for all deflections and the other BI Parameters are inactive and are suppressed by default.
- Yield deflection values, X_{YC} or X_{YT} , can be implicitly defined by specifying $FYC=S0*X_{YC}$ or $FYT=S0*X_{YT}$.
- 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_{YC} and F_{YT} by default.
- In One-way BI Aspects with $S_Y > 0$ both F_{YC} and F_{YT} 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.
- $F_{YT} \neq -F_{YC}$ can be used to model asymmetric compression–tension behavior.

3.16.3.3 Static Type **SI** : Segmented Inelastic

Static type **SI** is segmented inelastic with boundaries defined by linear segments between specified deflection–force points. There are three **SI** variants having compression, tension, and symmetric (bi-directional) boundaries. The **compression** variant yields in compression along the boundary and has elastic unloading and tension behavior. The **tension** variant yields in tension along the boundary and has elastic unloading and compression behavior. The **symmetric** variant yields in both directions along a shifting symmetric boundary and has elastic unloading behavior.

Static Type **SI** : Compression Variant



The Static type **SI** compression variant 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 $i = 1, \dots, N$ 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 $x > X_N + X_\Delta$.

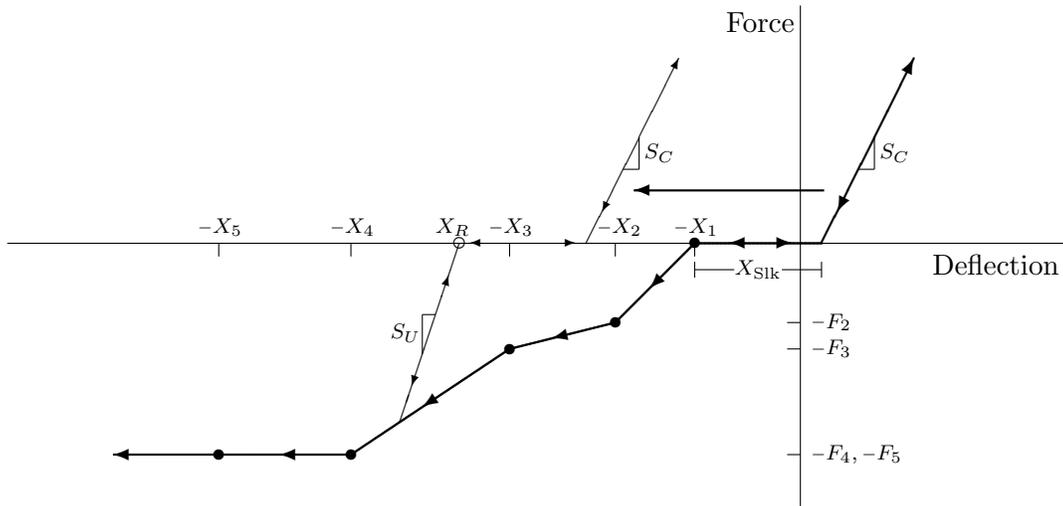
Yielding occurs during increasing deflection traversal of the compression boundary.

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_{sik} 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 $X_R - X_{sik}$.

Static Type SI : Tension Variant



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

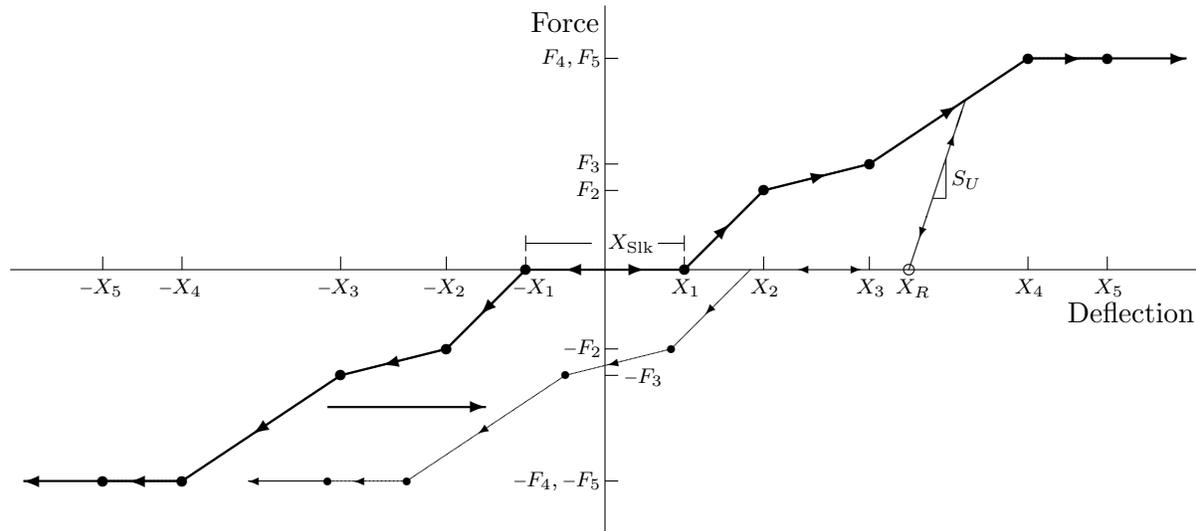
The **tension boundary** is defined by the points $(-X_i, -F_i)$ (the sign-inverted X and F block values) for $i = 1, \dots, N$ where X_i is increasing with i and $F_i + F_\Delta \geq 0$. The last compression segment is extrapolated to define the boundary for $x < -(X_N + X_\Delta)$.

Yielding occurs during decreasing deflection traversal of the tension boundary.

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

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

Compression traverses an elastic line with slope S_C for all deflections above $X_R + X_{Slk}$.

Static Type **SI** : Symmetric Variant

The Static type **SI** symmetric variant 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 $i = 1, \dots, N$ 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	?
Var	Variant selector	$\mathcal{C} \in \mathcal{C}, \mathcal{T}, \mathcal{S}$	\mathcal{C}		
SU	Unloading slope (S_U)	$\mathcal{R} \geq 0$?
ST	Tension slope (S_T) for compression variant	$\mathcal{R} \geq 0$	S_U	N	?
SC	Compression slope (S_C) for tension variant	$\mathcal{R} \geq 0$	S_U	N	?
XSlk	Slack (X_{Slk}) for nonsymmetric SI	$\mathcal{R} \geq 0$	0		?
XiSc1	Deflection block scale factor ($X_{i\otimes}$)	$\mathcal{R} > 0$	1		
FiSc1	Force block scale factor ($F_{i\otimes}$)	$\mathcal{R} \geq 0$	1		
ConSS	Segment smoothness confidence band factor	$\mathcal{R} > 0$	<i>hierarchy</i>	N	
AnySlope	Extracted segment slope control	\mathcal{B}	False		
X	Boundary deflection (X_i) block	$[\mathcal{R} \nearrow]$	<i>see Notes</i>		
F	Boundary force (F_i) block	$[\mathcal{R}]$?

Static Type SI Notes

- **Var** selects the SI variant: \mathcal{C} for the compression variant, \mathcal{T} for the tension variant, and \mathcal{S} for the symmetric variant. **Var** defaults to the compression variant.
- 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.
- For the symmetric SI variant the unloading slope, S_U , is used for unloading/reloading in both compression and tension.
- S_T is active only for the compression SI variant.
- S_C is active only for the tension SI variant.
- X_{Slk} can be specified only for the nonsymmetric SI variants. For the symmetric variant the **implied slack** value is $\max(2(X_1 - (F_1/S_U)), 0)$.
- $X_{i\otimes}$ and $F_{i\otimes}$ 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 $\text{ConSS} \cdot 5000(\tilde{w}/\tilde{x})$ 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 3.3.2.1).
- Default **X** block deflections can be selected by entering **#** instead of a value. For Links between instrumented Masses SISAME-3D uses the actual deflection range to set default X_1 or X_N tail deflections. For the symmetric **SI** variant the default tail deflection values are $X_1 = 0$ and $X_N = \max(|x|)$. Non-tail default deflections are evenly spaced between the surrounding deflection values. The repeat count form $n\#$ can be used to indicate n successive default 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 the tension and symmetric **SI** variants the $(-X_i, -F_i)$ tension boundary points are offset by $(-X_\Delta, -F_\Delta)$.
- 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 the tension and symmetric **SI** variants 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: two Dampers and two Magnifiers. A Link can have a Damper and/or a Magnifier or neither. 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 identifier in references.

The Dampers 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 without a Static Aspect to obtain a pure damping Element.

The Magnifiers 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](#). The Dynamic Aspects are described in terms of forces and velocities: for rotational Link Modes these are torques and angular velocities but otherwise there is no difference in behavior from the Axial Link Mode.

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 the behavior of Dynamic Aspects. **Scale factor** transforms X_{\otimes} , R_{\otimes} , D_{\otimes} , and M_{\otimes} scale the effective Dynamic characteristic along the corresponding axes.

Universal Dynamic Parameters

Tag	Description	Value	Default	N	?
<code>OneWay</code>	One-way Dynamic Aspect flag	$C \in C, T$	<i>none</i>	N	
<code>XSc1</code>	Deflection scale factor (X_{\otimes})	$\mathcal{R} > 0$	1		
<code>RSc1</code>	Relative velocity scale factor (R_{\otimes})	$\mathcal{R} > 0$	1		
<code>DSc1</code>	Damping force scale factor (D_{\otimes})	$\mathcal{R} \geq 0$	1		
<code>MSc1</code>	Dynamic Magnifier scale factor (M_{\otimes})	$\mathcal{R} \geq 0$	1		

Universal Dynamic Parameter Notes

- Dynamic transforms alter the internal behavior of defined and extracted Dynamic Parameters but not their external value. Parameters must obey their value range constraints both before and after the effects of any Link and Dynamic transforms are applied.
- M_{\otimes} scales the parametric contribution to the dynamic magnification that is added to one to get the magnification: M_{SIP} and, for Dynamic type AM, $(M_{Max} - 1)$ are scaled by M_{\otimes} .
- X_{\otimes} affects only the damping slack deflection, D_{Sik} .
- 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_{Sik} , can be specified to prevent damping at deflections less than D_{Sik} .

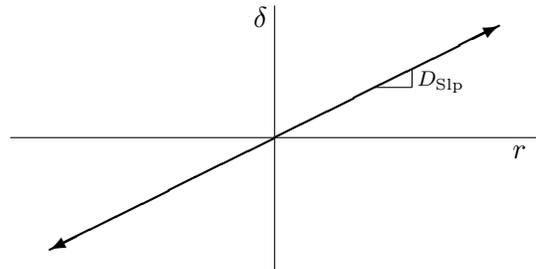
A compression-only (**OneWay=C**) Damper generates zero force at negative relative velocities. A tension-only (**OneWay=T**) Damper generates zero force at positive relative velocities.

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

The damping slope, D_{Sip} , 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

$$f = f^S + D_{Slp} r$$

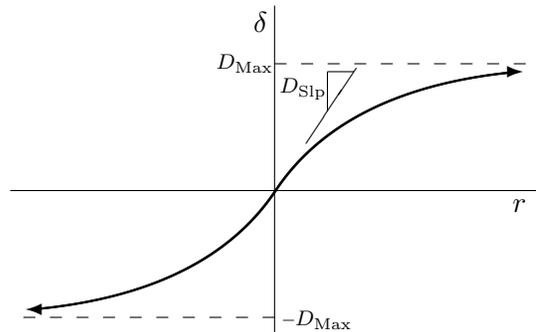


Dynamic Type **LD** Parameters

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

3.17.2.2 Dynamic Type **AD** : Arctangent Damper

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



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}	<i>none</i>	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 [13]. 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.

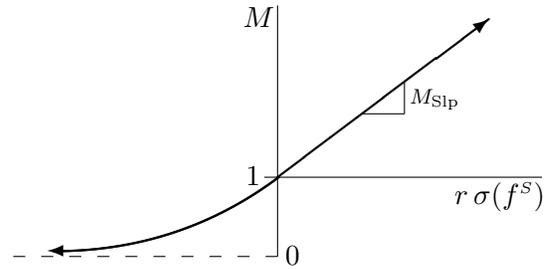
A compression-only (**OneWay=C**) Magnifier has a factor of one at negative relative velocities. A tension-only (**OneWay=T**) Magnifier has a factor of one at positive relative velocities.

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

3.17.3.1 Dynamic Type **LM** : Linear Magnifier

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

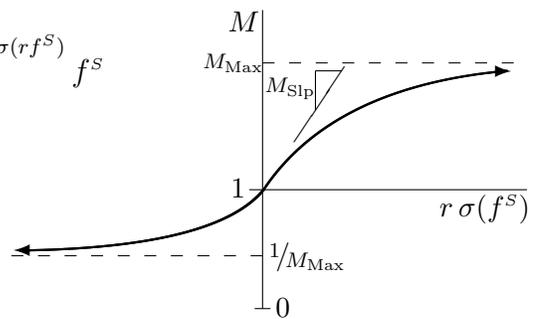


Dynamic Type **LM** Parameters

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

3.17.3.2 Dynamic Type **AM** : Arctangent Magnifier

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



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

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*, *Link*, and *Point* member lists support set reference specifiers (see Section 3.3.3.5). 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 [8, 10] 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 $2.5 \times \textit{Cutoff}$. 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 $10 w * 1 \text{ m}$, 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 factor tuples for the impulses and integrated impulses of target and comparison Loads. They are keyed by axis type, **l** for linear and **r** for rotational. The weighting of these target equations is based on *ConF*. *ConI* and *ConJ* default to the enclosing Structure's corresponding Parameters and then to **1**. *ConI* and/or *ConJ* Parameters 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 separate field with the same **Signal** Tag.
 - 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

The `Output Information` template is repeated here with descriptions of each output class followed by descriptions of the general and POV-Ray output specifications.

Output Information

<code>OutClass=MassTbl</code>	<code>Sub=</code>	<code>Mass=</code>	Mass Tabular
<code>OutClass=LinkTbl</code>	<code>Sub=</code>	<code>Link=</code>	Link Tabular
<code>OutClass=LoadTbl</code>	<code>Sub=</code>	<code>Load=</code>	Load Tabular
<code>OutClass=MassTS</code>	<code>Qty=AVPavp</code>	<code>Mass=</code>	Mass Time Series
<code>OutClass=LinkTS</code>	<code>Qty=XRSFDE</code>	<code>Link=</code>	Link Time Series
<code>OutClass=LoadTS</code>	<code>Qty=FIJfij</code>	<code>Load=</code>	Load Time Series
<code>OutClass=LinkXY</code>	<code>Qty=SFDE</code>	<code>Link=</code>	Link X-Y History
<code>OutClass=LinkPar</code>	<code>Qty=SD</code>	<code>Link=</code>	Link Parametric
<code>OutClass=FitRep</code>	<code>File=</code>		Fit Measure Report
<code>OutClass=Event</code>	<code>File=</code>		Event specifications
<code>OutClass=POV</code>	<code>Motion=</code>	<code>Camera=</code>	<code>LookAt=</code>
	<code>Angle=</code>	<code>Time=</code>	<code>File=</code>
			POV-Ray images

Output Specifications

Tag	Description	Value	Default	N	?
<code>OutClass</code>	Output class	<i>see below</i>			
<code>Sub</code>	Subsampling rate	$\mathcal{I} \geq 1$	1		
<code>Qty</code>	Output quantity flags	<i>see below</i>			
<code>Mass</code>	Mass set list	\mathcal{S}	<i>none</i>		
<code>Link</code>	Link set list	\mathcal{S}	<i>none</i>		
<code>Load</code>	Load set list	\mathcal{S}	<i>none</i>		
<code>File</code>	Output file name	\mathcal{S}	<i>see Notes</i>		

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 time-zero 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 inertia, 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, F, D, E }, which request deflection, relative velocity, static force, magnified force, damping force, and energy output, respectively. Only S, F, D, 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 can contain lists of Element and Element set references (see Sections 3.3.3 and 3.3.3.5). 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 with the effect of any Link and Static transforms applied. 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 with any Link and Dynamic transforms applied. 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.
 - 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.
- Mass and Load output signal files are given names the form *Element.Axis.DataType.FileType*, and Link output signal files are given names of the form *Link.Mode.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
Link					c	comparison
	x	deflection	t	time series	d	defined
	r	relative velocity	x	vs. deflection	e	extracted
	s	static force				
	d	dynamic force				
Parametric Link	f	force	p	parametric	d	defined
	e	energy			e	extracted
Load	m	magnifier	t	time series	g	group
	f	force			i	instrumented
	i	impulse			d	driving
	j	integrated impulse			c	comparison

POV-Ray Output Specifications

Tag	Description	Value	Default	N	?
Motion	Motions (Actual and/or Effective)	$S \in A, E, AE$	A		
Camera	Camera location specification	$S \in \text{Auto} (x, y, z)$ [+Fixed Track ($\Delta_x, \Delta_y, \Delta_z$)]	Auto+Fixed		
LookAt	Look-at position/part	$S \in \text{Auto}, (x, y, z), \text{part}$	Auto		
Angle	Lens viewing angle (degrees)	\mathcal{R}	50°		
Time	Frame time specifications	\mathcal{R} or R	0		
File	Base frame POV-Ray file name	S	<i>see Notes</i>		

POV-Ray Output Specification Notes

- The `POV` request produces one or more POV-Ray image files of the Model that can be used to verify and document the Model configuration and to generate animations. Both actual and effective extracted motions can be displayed and these can be overlaid in the same images to produce a visual assessment of the model's fidelity to the target motions. Methods and tools for generating animations from SISAME-3D POV-Ray images are platform-specific and are covered in a separate document.
- *Motion* can be **A** to produce the actual motions of all Masses, **E** to get the effective extracted motions of target Masses, or **AE** to get both overlaid with the effective motion Mass images made more transparent (and slightly smaller to avoid image artifacts). *Motion* defaults to **A**. In test runs only the actual time-zero position will be shown.

- *Camera* specifies the camera start location and how the camera moves. *Camera* is of the form :

`Camera=start[+move]`

The syntax of *start* is:

`Auto | (x,y,z)`

where `Auto` specifies the default automatic camera start location and (x,y,z) specifies the camera start point. The *start* can also be `#` or omitted to get the default automatic camera start location.

The syntax of the optional *move* specifier is:

`Fixed | Track | ($\Delta_x, \Delta_y, \Delta_z$)`

where `Fixed` specifies the default fixed camera, `Track` specifies that the camera should track at the same relative position to the look-at point, and $(\Delta_x, \Delta_y, \Delta_z)$ specifies the amount the camera should move each frame in the series after the first frame.

Camera defaults to **Auto+Fixed**, giving a fixed position camera in the default location.

- *LookAt* specifies the point or model part that the camera should look at. The default (**Auto**) behavior is to look at the center of the model's bounding box and to follow that point as the model moves. A fixed point to look at can be specified with the syntax (x,y,z) . A specific part

of the model can be looked at by specifying its name: any Mass, FixedMass, Point, or Shape can be looked at.

- *Angle* specifies the viewing angle. Reducing *Angle* zooms the image in and increasing *Angle* zooms out.
- With 3-D models perspective images are generated. Moving the camera closer to the model increases the perspective effect of bringing the vanishing points closer and making parallel model edges less parallel in the image. By compensating the angle for the camera distance you can obtain essentially the same sized model components but with varying levels of perspective effect.
- *Time* specifies the output time or times at which to generate a POV-Ray image. One or a sequence of time specifiers can follow *Time=* and multiple *Time* specifications can be included in one *OutClass=POV* specification. A time specifier can have one of these forms:

$$t \quad t + \Delta_t \quad + \Delta_t \quad R$$

where t is a start time, Δ_t is a time step, and R specifies the rest state position of the Model. The first form requests a single output time, the second form specifies a start time and a time step, and the third form has an implied start time of zero with a specified time step. The list of times for each *OutClass=POV* specification is sorted with duplicates removed and rounded to the nearest output time step to obtain a time sequence for output.

- For normal animation use a single start time and time step should be specified, for example *Time=+.002*. Use separate *OutClass=POV* specifications for distinct animation sequences.
- A sequence of frames with non-uniform time steps will produce a strange effect if combined with a camera that is moving with a fixed $(\Delta_x, \Delta_y, \Delta_z)$ step each frame.
- *Time* defaults to 0, producing a single image at time-zero.
- Only time-zero POV-Ray images are generated from a test run.
- Rest state images contain actual positions only and are treated as the first frame when R is combined with frame time specifiers, which may not produce the desired animation results: normally rest state images are specified on a separate POV specification line.
- *File* specifies the base file name for the POV-Ray image sequence. A suffix of *.frame* is appended to the base portion of the name, where *frame* is the sequential number of the image in this series. Distinct *File* names should be used for each *OutClass=POV* specification to avoid overwriting frame files.
- POV-Ray file names default to the name of the input Event file with the file extension replaced by *.series.frame.pov* where *series* is the sequential number of the *OutClass=POV* specification and *frame* is the sequential number of the image in this series.
- POV-Ray files show the positions of the Masses and Fixed Masses and the Links that connect them. Shapes and Points appear with their default or specified size and color. CG Points are shown even if their Mass or FixedMass has no specified Shapes or Points. Links are shown as cylinders connecting their attachment Points.
- POV-Ray 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-Ray 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 specifications in the POV-Ray files can be altered to create model images from other angles and distances.
- The Python script, `POVpost.py`, can be used to generate new frame sequences with altered camera specifications without re-running SISAME-3D. Python must be installed on the computer to use `POVpost.py` (<http://www.python.org>).
- Shapes within a Mass or Fixed Mass can be combined to eliminate internal surfaces by wrapping their POV-Ray specifications in a `merge` object. Using `merge` will increase the POV-Ray render times.
- The POV-Ray 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-Ray file will differ, in general, from the body-fixed rotations used in SISAME-3D.

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 default tail deflections will 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 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 *EventIDs* 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	$S \leq 20$		
Descr	Description	S		
Class	Run class	$C = S, E, T$	<i>see Notes</i>	
Title	Run title	S		
MaxIter	Max number of extraction iterations	$I \geq 0$	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 [14]. 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

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

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

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

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

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

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

$$\begin{pmatrix} \mathbf{f}_S^{(A)} \\ \mathbf{f}_T^{(A)} \end{pmatrix} = \begin{bmatrix} \Phi_{SD}^{(L)} & \Phi_{SE}^{(L)} \\ \Phi_{TD}^{(L)} & \Phi_{TE}^{(L)} \end{bmatrix} \begin{pmatrix} \mathbf{f}_D^{(L)} \\ \mathbf{f}_E^{(L)} \end{pmatrix} + \begin{pmatrix} \Phi_S^{(D)} \mathbf{f}_S^{(D)} \\ \Phi_T^{(D)} \mathbf{f}_T^{(D)} \end{pmatrix} + \begin{pmatrix} \mathbf{f}_S^{(G)} \\ \mathbf{f}_T^{(G)} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_S^{(I)} \\ \mathbf{f}_T^{(I)} \end{pmatrix} \quad (4.4)$$

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 $\Phi^{(L)}$ 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 $\Phi_{SE}^{(L)} = \mathbf{0}$, 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

$$\Phi_{TE}^{(L)} \mathbf{f}_E^{(L)} = \mathbf{f}_T^{(I)} - \Phi_{TD}^{(L)} \mathbf{f}_D^{(L)} - \Phi_T^{(D)} \mathbf{f}_T^{(D)} - \mathbf{f}_T^{(G)} \quad (4.5)$$

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 $\Phi_{TE}^{(L)}$ 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, 7] but there are serious drawbacks with such an approach. To get a square, non-singular $\Phi_{TE}^{(L)}$ 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

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

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

$$\mathbf{f}^{(L)} = \mathbf{\Pi}^{(L)} \mathbf{p}^{(L)} \quad (4.7)$$

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

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

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 $\Phi_{SE}^{(L)} = \mathbf{0}$) 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 applied force from both defined and extracted parameters, so

$$\mathbf{f}_T^{(A)} = \mathbf{\Psi}_{TD}^{(L)} \mathbf{p}_D^{(L)} + \mathbf{\Psi}_{TE}^{(L)} \mathbf{p}_E^{(L)} + \mathbf{\Phi}_T^{(D)} \mathbf{f}_T^{(D)} + \mathbf{f}_T^{(G)} \quad (4.9)$$

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

$$\mathbf{f}_T^{(I)} = \Phi_{TD}^{(I)} \mathbf{p}_D^{(I)} + \Phi_{TE}^{(I)} \mathbf{p}_E^{(I)} \quad (4.10)$$

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

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

$$\mathbf{f}_T^{(A)} = \Psi_{TD}^{(L)} \mathbf{p}_D^{(L)} + \Psi_{TE}^{(L)} \mathbf{p}_E^{(L)} + \Phi_T^{(D)} \mathbf{f}_T^{(D)} + \mathbf{f}_T^{(G)} = \Phi_{TD}^{(I)} \mathbf{p}_D^{(I)} + \Phi_{TE}^{(I)} \mathbf{p}_E^{(I)} = \mathbf{f}_T^{(I)} \quad (4.11)$$

or

$$\Psi_{TE}^{(L)} \mathbf{p}_E^{(L)} - \Phi_{TE}^{(I)} \mathbf{p}_E^{(I)} = \Phi_{TD}^{(I)} \mathbf{p}_D^{(I)} - \Psi_{TD}^{(L)} \mathbf{p}_D^{(L)} - \Phi_T^{(D)} \mathbf{f}_T^{(D)} - \mathbf{f}_T^{(G)} \quad (4.12)$$

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

Letting

$$\Gamma_T^{(M)} = \begin{bmatrix} \Psi_{TE}^{(L)} & -\Phi_{TE}^{(I)} \end{bmatrix} \quad \text{and} \quad \gamma_T^{(M)} = \Phi_{TD}^{(I)} \mathbf{p}_D^{(I)} - \Psi_{TD}^{(L)} \mathbf{p}_D^{(L)} - \Phi_T^{(D)} \mathbf{f}_T^{(D)} - \mathbf{f}_T^{(G)}$$

the instantaneous target Mass equations can be expressed

$$\Gamma_T^{(M)} \mathbf{p}_E = \gamma_T^{(M)} \quad (4.13)$$

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

$$\Gamma_T^{(M)} \mathbf{p}_E \approx \gamma_T^{(M)}. \quad (4.14)$$

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)

$$\left[\int_t \Gamma_T^{(M)} \right] \mathbf{p}_E \approx \int_t \gamma_T^{(M)} \quad \text{and} \quad \left[\int_t \int_t \Gamma_T^{(M)} \right] \mathbf{p}_E \approx \int_t \int_t \gamma_T^{(M)}. \quad (4.15)$$

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

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

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

$$\mathbf{f}^{(\mathcal{L})} = \mathbf{\Phi}^{(\mathcal{L})} \mathbf{p} \approx \mathbf{f}^{(\mathcal{L})} \quad (4.17)$$

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 $\mathbf{f}^{(\mathcal{L})}$ is the target Load signal force vector at that instant.

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

$$\mathbf{\Phi}_T^{(\mathcal{L})} \mathbf{p}_E \approx \mathbf{f}^{(\mathcal{L})} - \mathbf{\Phi}_D^{(\mathcal{L})} \mathbf{p}_D \quad (4.18)$$

and letting

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

the instantaneous target Load equations can be expressed

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

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)

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

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

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

4.5 Weighted Target System

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

$$\mathbf{\Gamma} \mathbf{p}_E \approx \boldsymbol{\gamma}. \quad (4.22)$$

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 $0, \dots, N$ are used to generate the instantaneous target equations, the complete target system is

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

which is written more concisely as

$$\mathbf{\Upsilon} \mathbf{p}_E \approx \mathbf{v}. \quad (4.24)$$

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, $\mathbf{\Omega}$.

$$\mathbf{\Omega} \mathbf{\Upsilon} \mathbf{p}_E \approx \mathbf{\Omega} \mathbf{v} \quad (4.25)$$

or

$$\mathbf{T} \mathbf{p}_E \approx \boldsymbol{\tau} \quad (4.26)$$

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

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

Letting

$$\mathbf{G} = \mathbf{T}^T \mathbf{T} \quad \text{and} \quad \mathbf{h} = \mathbf{T}^T \boldsymbol{\tau} \quad \text{and} \quad b = \boldsymbol{\tau}^T \boldsymbol{\tau}$$

this becomes

$$e = \mathbf{p}_E^T \mathbf{G} \mathbf{p}_E - 2 \mathbf{p}_E^T \mathbf{h} + b \quad (4.28)$$

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

The gradient of the error is

$$\nabla e = \mathbf{2} \mathbf{G} \mathbf{p}_E - \mathbf{2} \mathbf{h} \quad (4.29)$$

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 $\mathbf{C} \mathbf{p}_E \leq \mathbf{c}$. To incorporate the constraints in the solution process the Lagrangian is formed

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

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 [6] for the globally optimal Parameters are

$$\begin{aligned} \nabla_{\mathbf{p}_E} \ell &= \nabla e + \mathbf{C}_=^T \boldsymbol{\lambda} + \mathbf{C}^T \boldsymbol{\mu} = \mathbf{0} \\ \mathbf{C}_= \mathbf{p}_E &= \mathbf{c}_= \\ \mathbf{C} \mathbf{p}_E &\leq \mathbf{c} \\ \boldsymbol{\mu} &\geq \mathbf{0} \\ \boldsymbol{\mu}^T (\mathbf{C} \mathbf{p}_E - \mathbf{c}) &= 0. \end{aligned} \quad (4.31)$$

This is a Linear Complementary Programming (LCP) [15] 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 $\mu_I = 0$ where the inactivity set is I . The optimality conditions for the tentative activity set A reduce to

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

with the additional conditions

$$\begin{aligned} \mathbf{C}_I \mathbf{p}_E &\leq \mathbf{c}_I \\ \mu_A &\geq \mathbf{0}. \end{aligned} \quad (5.2)$$

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 [15] 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

$$\begin{bmatrix} \mathbf{P}\mathbf{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{G} & \mathbf{C}_=^T & \mathbf{C}_A^T & \mathbf{h} \\ \mathbf{C}_= & \mathbf{0} & \mathbf{0} & \mathbf{c}_= \\ \mathbf{C}_A & \mathbf{0} & \mathbf{0} & \mathbf{c}_A \end{bmatrix} = \begin{bmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{Y}_A & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{U} & \hat{\mathbf{C}}_A^T & \hat{\mathbf{h}} \\ \mathbf{0} & \mathbf{Z}_A & \hat{\mathbf{c}}_A \end{bmatrix} \quad (5.3)$$

where $\mathbf{P}\mathbf{S}$ is a permutation and scaling matrix to allow for row scaling and partial pivoting during the reduction of \mathbf{G} and the equality constraints ($\mathbf{P}\mathbf{S}$, \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 $2\mathbf{Z}_A\boldsymbol{\mu}_A = \hat{\mathbf{c}}_A$ then back-substitution on

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

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} :

$$f = Sx + D_{slp} r.$$

A segmented elastic Static SE Aspect has the static force representation

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

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

$$\bar{f}(\mathbf{p}_E) = f(\widetilde{\mathbf{p}}_E) + \nabla f(\mathbf{p}_E)(\mathbf{p}_E - \widetilde{\mathbf{p}}_E)$$

where \bar{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

$$\bar{f}^S = [x] S_O + [\sigma(x) |x|^{\bar{P}}] C + [\tilde{C} \sigma(x) |x|^{\tilde{P}} \ln |x|] P - [\tilde{C} \sigma(x) |x|^{\tilde{P}} \ln |x| \tilde{P}].$$

The Static AE type has static force

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

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

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

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

$$f_{\square}^S = F_{\square} + [x - X_{\square}] S_{\square}$$

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

$$\overline{f_{\square}^S} = [\tilde{X}_{\square}] S_O + [\tilde{S}_O - \tilde{S}_{\square}] X_{\square} + [x - \tilde{X}_{\square}] S_{\square} - [(\tilde{S}_O - \tilde{S}_{\square}) \tilde{X}_{\square}].$$

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

$$\overline{f_{\square}^S} = \left[\frac{\tilde{F}_{\square} \tilde{S}_{\square}}{\tilde{S}_O^2} \right] S_O + \left[1 - \frac{\tilde{S}_{\square}}{\tilde{S}_O} \right] F_{\square} + [x - \tilde{X}_{\square}] S_{\square}$$

(when $\tilde{S}_O = 0$ 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

$$f_T^S = S_T \left(x - \left(X_B - \frac{F_B}{S_U} - X_{\text{slk}} \right) \right)$$

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

$$\overline{f_T^S} = \left[\frac{\tilde{S}_T}{\tilde{S}_U} \right] F_B - \left[\frac{\tilde{F}_B \tilde{S}_T}{\tilde{S}_U^2} \right] S_U + \left[x - \left(X_B - \frac{\tilde{F}_B}{\tilde{S}_U} - \tilde{X}_{\text{slk}} \right) \right] S_T + [\tilde{S}_T] X_{\text{slk}} - [\tilde{X}_{\text{slk}} \tilde{S}_T].$$

The Dynamic AD Damper type has force

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

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

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

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

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

The Dynamic AM Magnifier type has force

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

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

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

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 $(\mathbf{p}_E - \widetilde{\mathbf{p}}_E) \rightarrow 0$ 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

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

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

$$\mathbf{I} \mathbf{p}_E \approx \mathbf{p}_f \quad (5.7)$$

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

$$\frac{\Delta_i}{\Delta_i^{(2)}} F_{i-1} - F_i + \frac{\Delta_i}{\Delta_i^{(2)}} F_{i+1} \approx 0 \quad (5.8)$$

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 $\Psi \mathbf{p}_E \approx \psi$ system of (4.23).

5.6 Target Equation Weighting

The target equations are weighted by the matrix Ω . 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 Ω , 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, $0, \dots, N$. 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

$$\frac{1}{\mathit{ConA}\sqrt{N+1}} \frac{\eta_M}{\mathit{ConV}(N+1)} \quad \text{and} \quad \frac{1}{\mathit{ConA}\sqrt{N+1}} \frac{\eta_M}{\mathit{ConP}(N+1)^2}$$

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 ConV and ConP defaults of **1** give a reasonable balance between the acceleration, velocity, and position target weighting.

Similarly, Load impulse and integrated impulse targets are weighted by

$$\frac{1}{\mathit{ConF}\sqrt{N+1}} \frac{\eta_L}{\mathit{ConI}(N+1)} \quad \text{and} \quad \frac{1}{\mathit{ConF}\sqrt{N+1}} \frac{\eta_L}{\mathit{ConJ}(N+1)^2}$$

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 ConI and ConJ defaults 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 $i = 1, \dots, N$
- $F_i \geq F_{i-1}$ for $i = 2, \dots, N$ ensuring nonnegative segment slopes (unless **AnySlope=True**)
- $F_i - F_{i-1} \leq (X_i - X_{i-1}) S_U$ for $i = 2, \dots, N$ 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

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

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

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

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 [8, 10].

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 revolutionary frequencies, damping characteristics, and the output and DFT timing. The default time steps should provide sufficient accuracy for most applications.

The highest revolutionary frequency of the simulated Masses is not known before the simulation is performed so SISAME-3D considers only the initial angular velocities. If the revolutionary 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 i^{th} time step for each simulated Mass is

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

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

$$\mathbf{a}_S^* = \mathbf{W}_S^{-1} \Phi_{SD} \mathbf{f}_D^* \quad (5.11)$$

The first corrector step is

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

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

$$\mathbf{a}_S^{**} = \mathbf{W}_S^{-1} \Phi_{SD} \mathbf{f}_D^{**} \quad (5.13)$$

The second corrector step has the same form as the first

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

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

$$\mathbf{a}_S = \mathbf{W}_S^{-1} \Phi_{SD} \mathbf{f}_D \quad (5.15)$$

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.
- When a position domain input signal has an initial velocity (as computed by numerical differentiation in SISAME-3D) that is not consistent to sufficient precision with the *IniV* specifications some distortion of the filtered signal representation near time-zero can occur.
- 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 `XSc1` 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 Parameter tuples 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 *DeITSim* can be used. Smaller than default *MaxIter*, *ConPD*, and *MultPD* values and a larger than default *ConvTo1* value can reduce the number of iterations. Decreasing the *Cutoff* value(s) will speed each iteration. Suppressing *ConV* and *ConP* 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 *DeITSim* should be used. Larger than default *MaxIter*, *ConPD*, and *MultPD* may be helpful by allowing more iterations. A smaller than default *ConvTo1* may slightly improve the solution. The *ConA*, *ConV*, *ConP*, *ConF*, *ConI*, and *ConJ* confidence band Parameters 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 *MultiPD* 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:
 - Reversed 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 `DeITSim` than default can be used (up to a factor of 100 can generally be used without serious loss of precision).
- Very small `DeITSim` 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 `DeITSim` 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 Non-linear 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] Ginsberg, J.H. (1998). *Advanced Engineering Dynamics, 2nd ed.* Cambridge University Press, Cambridge, UK.
- [5] Lawson, C.L. and Hanson, R.J. (1974). *Solving Least Squares Problems*. Prentice-Hall, Inc., Englewood Cliffs, NJ.
- [6] Luenberger, D.G. (1973). *Introduction to Linear and Nonlinear Programming*. Addison-Wesley, Reading, MA.
- [7] 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.
- [8] Mentzer, S.G. (1991). *The SIMFIL Program: Simultaneous Filtering*. U.S. Department of Transportation Report No. DOT HS 807 731.
- [9] Mentzer, S.G. et al (1992). *The SISAME Methodology for Extraction of Optimal Structural Crash Models*. SAE Report No. 920358.
- [10] Mentzer, S.G. (1995). *The SimFil Program v.2: Simultaneous Filtering*. U.S. Department of Transportation Draft Report.
- [11] Mentzer, S.G. (1995). *The VeCor Program*. U.S. Department of Transportation Draft Report.
- [12] Mentzer, S.G. (2002). *The SISAME Program: Structural Crash Model Extraction and Simulation*. U.S. Department of Transportation Draft Report.
- [13] Prasad, P. and Padgaonkar, A.J. (1981). *Static-to-Dynamic Amplification Factors for Use in Lumped-Mass Vehicle Crash Models*. SAE Report No. 810475.
- [14] Shabana, A.A. (2001). *Computational Dynamics*. Wiley, New York.
- [15] Shapiro, J.F. (1979). *Mathematical Programming: Structures and Algorithms*. Wiley, New York.

Index

Symbols

π , 32

A

Acceleration, 8
 Actuator, 6
 Angle, 97
 Cardan, 13, 20, 21, 126
 Animation, 96
 Applications, 4, 125–132
 Approximant, 114
 Aspect, 10
 Approximant, 114
 Dynamic, *see* Dynamic Aspect
 Identifier, 24
 Name, 25
 Point, 8, 9
 Reference, 26
 Shape, 8, 9, 56, 58
 Static, *see* Static Aspect
 Tag, 24, 27, 84
 Axes, 13, 19

B

Box, 59
 ID, 58

C

Camera, 96
 Capabilities, 4
 Design, 4
 FEM distillation, 4
 Multiple-Event modeling, 4, 99
 Weight extraction, 4
 Cardan angles, 13, 20, 21, 126
 Center of gravity, 9, 13, 19
 CG, 9, 13, 19, 54–57
 Color, 53, 55, 57, 59, 63
 Component, 6
 ID, 24

Name, 25

Reference, 26

 Shorthand, 27

Cone, 60

 ID, 58

Constrained approximation problem, 111

Constraints, *see* Solution, Constraints

Coordinate frame, 13, 19

 Ground, 13, 19

 Inertial, 13, 19

 Origin, 13, 19

Coordinate system, 94

Cylinder, 59

 ID, 58

D

Damper, 11, 86–87

 AD, 87

 LD, 87

Damping force, 11

Data file, *see* Signal, File

Deflection, 11

Degrees of freedom, 13, 19, 46

Dimensional systems, 23

DOF, 13, 19, 46

Dynamic Aspect, 11, 84–89

 Approximant, 114

 Dampers, 11, 86–87

 AD, 87

 LD, 87

 Domain transitions, 116

 Extraction, 128

 Magnifiers, 11, 88–89, 128

 AM, 89

 LM, 89

 One-way, 85

 Reference, 27

 Tag, 84

 Transform, 85

Universal Parameters, 85
 Dynamic magnification, 11

E

Element, 7
 Fixed Mass, 55
 ID, 24
 Link, 6, 10, 11, 62
 Load, 12, 90
 Mass, 6, 7, 51
 Fixed, 9
 Model, 39
 Reference, 26
 Shorthand, 27
 Ellipsoid, 61
 ID, 58
 Enclosing Structure, 6
 Energy, 11
 Dissipation, 75
 Equations of motion, 105–107
 Errors, 131
 Event
 ID, 24, 27, 104
 Matching, 99
 Specifications, 104
 Weighting, 104
 Event file, 17, 35
 Dynamic Parameter template, 38
 Example, 16
 Field, 39
 Boolean, 41
 Character, 41
 Extractable, 41
 Extracted, 41
 Integer, 41
 Real, 40
 Repeat, 40
 Suppressed, 41
 Suppressible, 41
 Tag, 16, 40
 Value, 41
 Fixed Mass specifications, 55
 Leading Tag, 39
 Link specifications, 62

Link template, 38
 Load specifications, 90
 Mass specifications, 51
 Model Information, 45
 Model specifications, 45
 Organization, 39
 Output Information, 93
 Parameter, 18, 39
 Point specifications, 56
 Run Information, 42
 Shape specifications, 58
 Specifications, 40
 Legend, 40
 Static Parameter template, 38
 Substructure specifications, 49
 Template, 35
 Vehicle specifications, 47
 Executing SISAME-3D, 17
 Exerted force, 8
 Expression, 32
 Parameter, 41
 Extracted Parameter, *see* Parameter, Ex-
 tracted
 Extraction, 2, 6, 125
 Assessment, 130
 Controlling, 129
 Design, 4
 Implementation, 112–121
 No feasible solution, 31
 Parameter specifiers, 31
 Process, 15
 Refining, 129
 Resimulation, 14, 15
 Specifiers, 33
 Time step, 43, 122
 Weight and load-path, 128

F

Failure
 Static, 65
 Fiat methodology, 3, 107
 Field, 17, 39
 Parameter, 18, 39
 Tag, 16, 17, 39–41

- Leading, 39
- Unspecified, 40
- Value, 17, 41
- Filtering, 5, 122
- Finite element modeling, 4, 127
- Fit measure report, 14, 94
- Fixed Mass, 9, 55
 - CG, 55–57
 - Color, 55
 - ID, 55
 - Orientation, 55
 - Point, 55
 - Position, 55
 - Shape, 55
 - Specifications, 55
 - Transparency, 55
- Force, 11, 12, 105
 - Damping, 11
 - Dynamic, 11
 - Exerted, 8
 - Inertia, 8
 - Magnifier, 11
 - Static, 11
- Formulation, 105–111

G

- Generalized force, 105
- Generalized momentum, 106
- Generalized vector, 105
- Ground frame, 13
- Guidelines, 125–132

I

- ID, 18, 24
 - Component, 24
 - Element, 24
 - Event, 24
 - Link, 24
 - Load, 24
 - Mass, 24
 - Substructure, 24
 - Vehicle, 24
 - Wildcard, 29
- Identifier, 24

- Aspect, 24
 - Parameter, 24
- Ill-conditioned least squares system, 131
- Impulse, 12
- Inactive Link, 63
- Inertia
 - Moments, 52
 - Products, 52
 - Tensor, 52
- Inertia force, 8
- inertial frame, 13
- Influence matrix, 105
- Instrumentation, 126
 - Accuracy, 125
 - Additional, 125
 - Artificial, 127
 - File, 5
 - Output, 14
 - Filtering, 126
 - Location, 125
 - Preparation, 15, 126, 130
 - Quality, 126
 - Synthesized, 127
- Integrated impulse, 12, 34

L

- LCP, 111, 112
- Limitations, 12, 18
- Linear approximant, 114
- Link, 6, 10, 62
 - Class, 11
 - Color, 63
 - Connection, 10
 - Damping force, 11
 - Defined, 11
 - Deflection, 11
 - Dynamic Aspect, *see* Dynamic Aspect
 - Dynamic magnification, 11
 - Energy, 11
 - Extracted, 11, 12, 18
 - Force, 11
 - Approximant, 114
 - Damping, 11
 - Dynamic, 11

- Magnifier, 11
 - Representation, 114
 - Static, 11
- ID, 24, 62
- Inactive, 63
- Mode
 - Reference, 27
- Name, 25
- Parameter
 - Extraction, 114
 - Template, 38
- Quantities, 11
- Reference, 26
- Relative velocity, 11
- Sign convention, 11, 65, 84
- Size, 63
- Specifications, 62
- Static Aspect, *see* Static Aspect
- Static force, 11
- Transform, 63
- Transparency, 63
- Load, 6, 12, 90
 - Class, 12
 - Driving, 12, 90
 - Force, 12
 - Group, 12, 90
 - ID, 24, 90
 - Impulse, 12
 - Integrated impulse, 12
 - Name, 25
 - Quantities, 12
 - Reference, 26
 - Specifications, 90
 - Target, 12, 90, 127
 - Equations, 109
- Locational reference, 26
- Log file, 17, 131
- LookAt, 96

M

- Magnifier, 11, 88–89, 128
 - AM, 89
 - LM, 89
- Mass, 6, 7, 51

- Acceleration, 8
 - C-V, 8
 - CG, 54, 56
 - Class, 7
 - Color, 53
 - Constant velocity, 52
 - Constant-velocity, 8
 - Driven, 8, 52
 - Exerted force, 8
 - Fixed, 9
 - ID, 24, 51
 - Inertia force, 8
 - Instrumented, 8
 - Name, 25
 - Orientation, 53
 - Point, 54
 - Position, 8, 53
 - Quantities, 8
 - Reference, 26
 - Shape, 54
 - Simulated, 8, 12, 18, 51
 - Specifications, 51
 - Target, 8, 52
 - Equations, 107–109
 - Transparency, 53
 - Velocity, 8, 53
 - Weight, 8, 127
 - Extraction, 4, 12, 18, 127
 - Magnitude, 46
- Master file, 17, 100
 - Event Information, 104
 - Event specifications, 104
 - Leading Tag, 101
 - Organization, 100
 - Run Information, 102
 - Template, 101
- Model, 1, 6, 7, 45
 - Axes, 13, 19
 - Component, *see* Component
 - Configuration, 12, 18, 127
 - Construction, 17
 - Degrees of freedom, 13, 19
 - Dimensional systems, 23
 - Element, *see* Element

Extraction, *see* Extraction
 ID, 46
 Orientation, 46
 Output, 14
 Parameter, *see* Parameter
 Position, 46
 Simulation, *see* Simulation
 Size, 12, 18
 Specifications, 45
 Structure, *see* Structure
 Substructure, *see* Substructure
 Vehicle, *see* Vehicle
 Velocity, 46
 Weight, 46, 111, 128
 Modeling
 Concepts, 6–15
 Momentum, 106
 Multiple-Event, 4, 99
 Usage, 130

N

Name, 24–26
 Aspect, 25
 Component, 25
 Link, 25
 Load, 25
 Mass, 25
 Parameter, 25
 Block, 25
 Tuple, 26
 Numerical integration, 123
 Numerical simulation, 122–123

O

One-way
 Dynamic Aspect, 85
 Static Aspect, 65
 Orientation, 13, 20, 21
 Fixed Mass, 55
 Mass, 53
 Model, 46
 Output signal, 94
 Point, 56
 Shape, 58

Substructure, 50
 Vehicle, 48
 Output, 14, 93
 Effective, 14
 Extracted, 14
 Fit measure report, 14, 94
 Instrument, 14
 Specifications, 93
 Output signal
 Files, 24, 25, 95
 Orientation, 94

P

Parameter, 18, 39
 Block, 18, 25, 39, 40
 Extracted, 33
 Monotonic, 41
 Name, 25
 Reference, 29, 33
 Specifier, 29
 Tag, 29
 Boolean, 41
 Character, 41
 Conditioning, 15, 43, 95, 102, 118, 131
 Equivalence, 118
 Damping, 43, 95, 103, 119, 129
 Default, 31, 41
 Expression, 32
 Extractable, 41
 Extracted, 41
 Bound, 33
 Constraints, 120
 Estimate, 33
 Reference, 31
 Specifier, 33
 Identifier, 24
 Integer, 32, 41
 Name, 25
 Real, 40
 Block, 40
 Tuple, 40
 Reference, 26, 28, 31
 Default, 31
 Extracted, 31

- Specifier, 29
 - Default, 31
 - Defined value, 30
 - Expression, 32
 - Extraction, 33
 - Reference, 31
 - Suppressed, 41
 - Suppressible, 41
 - Tag, 16, 17, 24, 28, 29, 40
 - Leading, 101
 - Tuple, 18, 26, 39, 40
 - Name, 26
 - Reference, 30
 - Spatial, 19
 - Specifier, 30
 - Tag, 30
 - Value, 17, 40, 41
 - Phase I procedure, 31, 121
 - Pi, 32
 - Plane, 61
 - ID, 58
 - Point, 8, 9, 54–56
 - CG, 56
 - Color, 57
 - ID, 56
 - Orientation, 56
 - Position, 56
 - Size, 56
 - Specifications, 56
 - Transparency, 57
 - Position, 8
 - Generalized, 13, 19
 - Initial, 13, 19, 21
 - Resting, 13, 19, 20
 - Rotational, 21
 - POV-Ray, 14
 - Angle, 97
 - Animation, 96
 - Camera, 96
 - LookAt, 96
 - Output
 - Specifications, 96
 - post-processor, 98
 - POVpost, 98
 - Problems
 - Common, 131
- ## R
- Reference, 24–29, 31
 - Ambiguous, 27
 - Aspect, 26
 - Resolution, 26
 - Component, 26
 - Resolution, 26
 - Shorthand, 27
 - Element, 26
 - Resolution, 26
 - Shorthand, 27
 - Link, 26
 - Load, 26
 - Locational, 26
 - Mass, 26
 - Parameter, 26
 - Resolution, 26
 - Set, 28–29
 - Shorthand, 26, 27
 - Structure, 26
 - Substructure, 26
 - Vehicle, 26
 - Relative velocity, 11
 - Run
 - ID, 42, 102
 - Running SISAME-3D, 17
- ## S
- Shape, 8, 9, 54, 55, 58
 - Box, 59
 - Color, 59
 - Cone, 60
 - Cylinder, 59
 - Ellipsoid, 61
 - Orientation, 58
 - Plane, 61
 - Position, 58
 - Size, 59
 - Specifications, 58
 - Sphere, 60
 - Transparency, 59

- Shorthand references, 26, 27
- Signal
 - Comparison, 8, 12
 - File, 5, 34
 - Input, 34, 40
 - Output, 14, 34
 - UDS, 5, 34, 35, 95
 - Input, 34
 - Motion, 8
 - Output, 14, 34
 - Specifier, 54, 92
 - Target, 12
- SimFil, 5, 34, 53, 91, 122
- Simulation, 6, 122–123, 131
 - Numerical integration, 123
 - Time step, 43, 122
- SISAME, 4
- SISAME-3D
 - Applications, 125–132
 - Dimensional systems, 23
 - Event file, *see* Event file
 - Executing, 17
 - Filtering, 122
 - Formulation, 105–111
 - Guidelines, 125–132
 - History, 3
 - Implementation, 112–123
 - Model, *see* Model
 - Modeling, *see* Modeling
 - Overview, 1
 - Running, 17
 - Users Guide, 16–104
- Solution
 - Computations, 110
 - Constraints, 33, 111–113, 120
 - Iteration, 43, 103, 116
 - Weight, 46, 47, 49, 128
 - Convergence, 43, 66, 88, 102, 103, 117, 119, 129
 - Direct, 112
 - Ill-conditioned, 131
 - Implementation, 112–121
 - Iteration, 43, 102, 103, 117, 129
 - Constraints, 43, 103, 116
 - No feasible \sim , 31
 - Optimality conditions, 111
 - Parameter, *see* Parameter
 - Phase I procedure, 31, 121
 - Pivoting, 113
 - Quality, 129
 - Relaxation, 43, 103, 117
 - Speed, 129
 - Sphere, 60
 - ID, 58
 - Static Aspect, 11, 65–83
 - Approximant, 114
 - Domain transitions, 116
 - Elastic, 67–74
 - AE, 70
 - BE, 71
 - EE, 69
 - LE, 68
 - SE, 73
 - Extraction, 114
 - Failure, 65
 - Inelastic, 75–83
 - AI, 76
 - BI, 77
 - Bidirectional yielding, 75
 - SI, 79
 - Unloading, 75
 - Yielding, 75
 - One-way, 65
 - Reference, 27
 - Segment smoothness, 95, 119
 - Transform, 65
 - Universal Parameters, 65
 - Static force, 11
 - Structure, 6, 18
 - Contained, 6
 - Enclosing, 6
 - Member, 6
 - Model, 6, 7, 45
 - Reference, 26
 - Substructure, 6, 7, 49
 - Vehicle, 6, 7, 47
 - Weight, 111, 128
 - Substructure, 6, 7, 39, 49

ID, [24](#), [49](#)
 Orientation, [50](#)
 Position, [50](#)
 Reference, [26](#)
 Specifications, [49](#)
 Velocity, [50](#)
 Weight, [49](#), [111](#), [128](#)
 Suppressed Parameter, [41](#)
 Suppressible Parameter, [41](#)

T

Target
 Equations
 Acceleration, [108](#)
 Confidence bands, [3](#), [33](#), [43](#), [53](#), [54](#), [74](#),
[82](#), [91](#), [92](#), [102](#), [103](#), [119](#), [129](#)
 Force, [109](#)
 Impulse, [109](#), [117](#)
 Integrated Impulse, [109](#), [117](#)
 Load, [109](#)
 Mass, [107–109](#)
 Position, [108](#), [117](#)
 Supplementary, [118](#)
 Velocity, [108](#), [117](#)
 Weighted, [110](#)
 Weighting, [119](#)
 Integral, [117](#)
 Load, [12](#)
 Mass, [8](#)
 Signal, [12](#)
 Test run, [42](#), [102](#)
 Time step
 Extraction, [43](#), [122](#)
 Simulation, [43](#), [122](#)
 Time-based weighting, [43](#)
 Transform
 Dynamic, [85](#)
 Link, [63](#)
 Static, [65](#)
 Transparency, [53](#), [55](#), [57](#), [59](#), [63](#)

U

UDS file, [5](#), [34](#), [35](#), [95](#)
 Users Guide, [16–104](#)

V

VeCor, [5](#), [15](#), [126](#)
 Vector
 Generalized, [19](#)
 Vehicle, [6](#), [7](#), [39](#), [47](#)
 Design, [4](#)
 ID, [24](#), [47](#)
 Impacts, [4](#)
 Orientation, [48](#)
 Position, [48](#)
 Reference, [26](#)
 Specifications, [47](#)
 Velocity, [48](#)
 Weight, [47](#), [111](#), [128](#)
 Velocity, [8](#)
 Generalized, [13](#), [19](#)
 Initial, [19](#), [22](#)

W

Warnings, [131](#)
 Weight, [8](#)
 Extraction, [4](#), [12](#), [18](#), [127](#)
 Magnitude, [46](#)
 Weighted target system, [110](#)
 Weighting, [104](#)