**LbcbPlugin Data Organization**

1. Introduction. This document is a guide to the organization of the LbcbPlugin step data. The guide shows how this data can be accessed by the customizable plugins which are written for specific tests. The step data is organized using MATLAB classes.
2. Anatomy of a Plugin Function. The LbcbPlugin provides a customization feature called plugin functions that allow the researcher to specify the control protocol of a test. The application supports two types of plugin functions; control point transformation functions and step correction functions. Examples of these functions can be found in the UserFunctions folder.
   1. Control Point Transformation Functions. These functions are used to convert step data between LBCB coordinate space and UI-SimCor model coordinate space. The functions have two input arguments and one output argument. The two input arguments are the ‘me’ parameter and the input displacement and force data expressed in the input coordinate space. The output argument is the same data expressed in the output coordinate space.
   2. Step Correction Functions. These functions are used during the execution of a step or sub-step as indicated in Figure 1 Step Correction Flowchart . The functions are executed at key points of the flow as shown in the following table:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

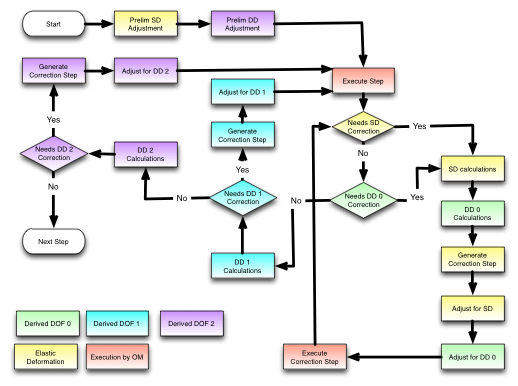


Figure Step Correction Flowchart

* 1. The ‘me’ Parameter. A typical plugin function usually has two input parameters

1. Accessing Step Data.
   1. Accessing the LBCB Response.
   2. Accessing the LBCB Command.
2. Class Hierarchy. All of these classes are contained in the SimData folder.
   1. StepData Class. The step data contains a complete set of everything that occurred during the step. The following data is included:

Table StepData Properties

|  |  |  |
| --- | --- | --- |
| Name | Description | Property Labels |
| LBCB Data | Cell array if one or two LbcbControlPoint (4.2) objects which contain commands and responses from the OM. Also contains the setup deformation calculations | lbcbCps |
| Model Data | Cell array of ModelControlPoint (3.6) objects which contain commands and response in the UI-SimCor coordinate space | modelCps |
| Step Number | Handle to the StepNumber (3.7) object contain all of the numbers for the step. | stepNum |
| Calculated Data | Names and values of all of the archive variables set by the plugins stored in a CorrectionData (3.8) object | cData |
| Configuration Data | Handle to a ConfigDaoProvider which contains the current configuration of the application as well as a set of convenience functions | cdp |
| Flags | Miscellaneous flags which characterize the step | containsModelCps isFirstStep  isInitialPosition |

* 1. LbcbControlPoint Class. This class contains the response and command for one LBCB as well as data for the control sensor associated with the LBCB.

Table LbcbControlPoint Properties

|  |  |  |
| --- | --- | --- |
| Name | Description | Property Labels |
| Response Data | Handle to an LbcbReading (3.3) object | response |
| Command Data | Handle to a Target (3.5) object | command |
| Control Sensor Data | Numeric array of control sensor readings | externalSensors |

* 1. LbcbReading Class. This class contains the responses of an LBCB. It has two virtual properties which return the appropriate response regardless of whether the application is configured to calculate setup deformation or not.

|  |  |  |
| --- | --- | --- |
| Name | Description | Property Labels |
| Displacements | Numeric array containing the position of the LBCB in LBCB Cartesian space. | disp |
| Forces | Numeric array containing the force readings of the LBCB in LBCB Cartesian space. | force |
| LBCB Data | Handle of DofData (3.4) containing the LVDT & load cell readings of the LBCB in LBCB Cartesian space | lbcb |
| Setup Deformation Data | Handle of DofData (3.4) containing the position of the LBCB in LBCB Cartesian space calculated from the control sensors as well as the Cartesian forces | ed |

* 1. DofData Class. This class contains the 12 degrees of freedom in Cartesian space of displacements and forces.

Table DofData Properties

|  |  |  |
| --- | --- | --- |
| Name | Description | Property Labels |
| Displacements | Numeric array of displacements; 3 translations, 3 rotations. | disp |
| Forces | Numeric array of 3 forces and 3 moments. | force |
| Labels | Cell array of DOF labels | labels |

* 1. Target Class. Class containing the displacement and force commands.

Table Target Properties

|  |  |  |
| --- | --- | --- |
| Name | Description | Property Labels |
| Displacements | Numeric array of displacements; 3 translations, 3 rotations. | disp |
| Forces | Numeric array of 3 forces and 3 moments. | force |
| Control DOFs for Displacements | Binary array indicating which displacement DOFs are being commanded | dispDofs |
| Control DOFs for Forces | Binary array indicating which force DOFs are being commanded | forceDofs |
| Labels | Cell array of DOF labels | labels |

* 1. ModelControlPoint Class. Contains displacement and force commands and responses in the coordinate space of UI-SimCor.

Table Model Control Point

|  |  |  |
| --- | --- | --- |
| Name | Description | Property Labels |
| Response Data | Handle to an Target (3.5) object | response |
| Command Data | Handle to a Target (3.5) object | command |
| UI-SimCor Address | The model control point address referenced by UI-SimCor | address |

* 1. StepNumber Class. Class containing the Step, sub-step, and correction step number of the step.

Table StepNumber Properties

|  |  |  |
| --- | --- | --- |
| Name | Description | Property Labels |
| Step number | Step number of the step which comes from either the input file or UI-SimCor | step |
| Sub-Step Number | Number of the sub-step. This is created when a step is split up by DOF increments | subStep |
| Correction Step Number | Number created by the correction plugins. Setup deformations increase this number by single digits. Derived DOF corrections increase this number by increments of 100 times what every control level the correction happens in. | correctionStep |

* 1. CorrectionData Class. Class containing the labels and values of all of the archive variables.

|  |  |  |
| --- | --- | --- |
| Name | Description | Property Labels |
| Variable Names | Cell array of variable names | labels |
| Variable Values | Numeric array of variable values | values |

* 1. ConfigDaoProvider Class. Class containing the application configuration data (in a property called ‘cfg’). It also contains a number of convenience methods:

Table ConfigDaoProvider Methods

|  |  |
| --- | --- |
| Method Name | Description |
| getAddress() | Get the primary UI-SimCor address of the test |
| numLbcbs() | Returns the number of LBCBs configured |
| numModelCps() | Returns the number of control points passed between UI-SimCor and the application |
| getAddresses() | Returns an array of control point addresses. |
| getExtSensors() | Returns arrays of names, sensitivities, and LBCB labels for all of the control sensors. The LBCB labels array can be one of ‘LBCB1’, LBCB2’, or ‘BOTH’ |
| getNumExtSensors() | Returns the number of control sensors |
| getFilteredExtSensors(isLbcb1) | Returns the same arrays as getExtSensors() for a particular LBCB |
| useEd() | Returns true if setup deformation is being corrected. |
| useDd(level) | Returns true if the specified level of Derived DOF is being corrected. |
| doStepSplitting() | Returns true if steps are being subdivided into sub-steps |
| forceAccept() | Returns true if only sub-steps and correction steps are auto-accepted. |
| getSubstepInc(isLbcb1) | Returns a numeric array of the sub-step increment for each displacement DOF. |