WHEN YOU NEED TO BE SURE

CyFlex® Multi-Load Bank Control
User Guide

Version 6

June 8, 2022

Developed by SGS North America, Inc.

Version History

Version	Date	Revision Description
1	1/15/2014	Initial publication
2	8/23/2018	Format with SGS brand
3	4/7/2020	Retrofit to new template
4	2/16/2021	Add examples of computed expressions and event handling when using an alternator as a dynamometer. Revised *Section 3.4* title on page 6 to *Configuring Real Power Control* Added the following: • *Section 3.6 Configuring the Generator to Behave as a Dynamometer* on page 8 • *Section 3.7 Converting the Torque Output Command from the Dyno Controller to an Electrical Power Request* on page 9 • *Section 3.8 Configuring the Voltage Regulator (VR) Control Loop* on page 9 • *Section 3.9 Getting Started* on page 11 • *Appendix D. Sample Efficiency Table* on page 19 • *Appendix E. Sample Speed Versus Voltage Table* on page 20 • *Appendix F. Sample Gen Labels File* on page 22 • *Appendix G. Event Response File* on page 37 • *Appendix H. Dyno Control Configuration* on page 48 • *Appendix I. Voltage Control Configuration* on page 52
5	12/6/2021	Removed inline usage content for `multi_lb_ctrl` and added hypertext linked cross-reference to its cyflex.com usage help in *Section 2 Starting the Program* on page 2
6	6/8/2022	Updated all hypertext linked cross-references to cyflex.com usage help descriptions

Document Conventions

This document uses the following typographic and syntax conventions.

- Commands, command options, file names or any user-entered input appear in Courier type. Variables appear in Courier italic type.

  Example: Select the `cmdapp-`*`relVersion-buildVersion`*`.zip` file….

- User interface elements, such as field names, button names, menus, menu commands, and items in clickable dropdown lists, appear in Arial bold type.

  Example: **Type**: Click **Select Type** to display drop-down menu options.

- Cross-references are designated in Arial italics.

  Example: Refer to *Figure 1*…

- Click intra-document cross-references and page references to display the stated destination.

  Example: Refer to *Section 1 Overview* on page 1.

  The clickable cross-references in the preceding example are *1*, *Overview*, and on page 1.

CyFlex Documentation

CyFlex documentation is available at https://cyflex.com/. View **Help & Docs** topics or use the **Search** facility to find topics of interest.

Table of Contents

1 Overview

The load bank control program selects digital outputs for one or more electrical load banks that are used to absorb load from a generator set. The load banks can be of the resistive or reactive type and can be located on the same or different test stands. If they are located on separate stands, then certain variables must be shared between the stands using the CyFlex node linking feature.

Refer to the following for information on node-linked variables and computed expressions:

- *CyFlex Variables, Units and Computed Expressions*
- cyflex.com usage for `snode_link`

The program can also be combined with computed expressions, events, and control loops that will make the generator respond in nearly the same way as conventional dynamometer.

2 Starting the Program

Start the `multi_lb_ctrl` program in the CyFlex `go` script.

Refer to cyflex.com usage help for `multi_lb_ctrl` for command options.

3 Configuration File

The multi-load bank control program is configured with a keyword/value style specification file. Each keyword is preceded with the at sign (@) and must be spelled exactly as shown. The value is placed on the line following the keyword. Refer to *Appendix A. Sample Configuration File* on page 12 for a sample file.

The keywords can be entered in any order. They are described below.

3.1 Program Instances

Each instance of the program must have a registered name. It can be any valid task name but must be unique in the system. It is anticipated that a separate instance will be used for reactive and resistive load banks.

```
@LOADBANK_CONTROLLER_REGISTERED_NAME
            ResistiveControl
```

3.2 CyFlex Events

The program uses five CyFlex events. All are inputs and will be created by the task when it initializes if they have not been created elsewhere. The events can be shared with other instances of the load bank control task if desired.

1. The first CyFlex event, when set, will enable changes in the load that is applied to the genset. Until it is set, the program will do nothing. It must be reset after the freeze load event, explained next, is set.

   ```
   @LOAD_CHANGE_ENABLE_EVENT_NAME
     ChangeLoadEv
   ```

2. The name of a CyFlex event that, when set, will cause the output values to be frozen at their present settings. The frozen load values are saved and can be restored later if desired.

   ```
   @LOAD_FREEZE_EVENT_NAME
     FreezeLoadEv
   ```

3. The name of a CyFlex event that, when set, will cause all of the digital output relays to open. This is the only way to assure this state and should be done at the end of each test procedure and prior to engine shutdown.

   ```
   @ZERO_LOAD_EVENT_NAME
     ClearLoadEv
   ```

4. The name of a CyFlex event that will cause the load values to be restored to the value they had when they were last frozen. Setting this event will not cause the load to change beyond restoring the previously frozen values.

   ```
   @RESTORE_LOAD_EVENT_NAME
     RestoreLoadEv
   ```

5. The name of a CyFlex event that will cause the multi-load bank control task to exit. This is generally not used except by developers or during testing.

   ```
   @RELEASE_TASK_EVENT_NAME
     StopKwCtrlEv
   ```

3.3 CyFlex Variables

The next series of keywords define CyFlex variables that are used to configure the program. Each label must be a valid CyFlex name. The variables will be created if they do not already exist when the program is started.

- Up to four labels for LOGICAL variables can be defined that indicate whether or not a load bank is in a condition to apply load to a generator set. The number of labels given with this keyword will define the total number of load banks available to the system. Separate logic, normally implemented with computed expressions, is used to set the value, which will normally be a logical summation of digital inputs received from the load bank. If the value of the variable is TRUE, then the load bank will not be used.

  ```
  @LOAD_BANK_TROUBLE_LABELS
    LB1_Trouble    LB2_Trouble    LB3_Trouble    LB4_Trouble
  ```

- Up to four labels for LOGICAL variables can be defined that indicate whether or not a load bank is enabled for use with a generator set. The number of labels given with this keyword must equal the number of trouble labels (above) and must be in the same order as the trouble labels. Separate logic, normally implemented with computed expressions or test procedures, is used to set the values. The values must be TRUE before load is applied by the load bank.

  ```
  @LOAD_ENABLE_LABELS
  LB1_LoadEnable LB2_LoadEnable LB3_LoadEnable
  LB4_LoadEnable
  ```

- Up to four labels for REAL variables can be defined that indicate the maximum load capability of the various load banks in the system. The number of labels given with this keyword must equal the number of trouble labels (above) and must be in the same order as the trouble labels. The values of these variables are set via computed expressions or test procedures and are normally static, though this is not required. Values can be de-rated dynamically if ambient conditions or other factors apply.

  ```
  @LOAD_BANK_MAX_LOAD_LABELS
    LB1_MaxLoad    LB2_MaxLoad    LB3_MaxLoad    LB4_MaxLoad
  ```

- Up to three labels for REAL variables can be defined that indicate the requested load for each electrical phase of the genset. The number of labels given with this keyword must equal the number of columns in the digital output section of the configuration file described below. Values are set dynamically by test procedures or computed expressions. The difference between the minimum and maximum value should be limited to the value of the maximum load imbalance variable described below.

  ```
  @DESIRED_LOAD_INPUT_LABELS
    PhaseA_RqstLoad   PhaseB_RqstLoad   PhaseC_RqstLoad
  ```

- The label of a REAL variable must be defined that indicates the maximum difference between the requested loads for all electrical phases of the genset. Values can be set dynamically by test procedures or computed expressions.

  ```
  @LOAD_IMBALANCE_LIMIT_LABEL
    MaxLdImbalance
  ```

- The labels for a set of REAL variables must be defined that indicate the actual load requested from each load bank that has been enabled. Values are set by the load bank control program.

  If the stand is a master controlling one or more slaves, then the values for the master stand represent the sum of the nominal ratings of the digital outputs requested. The values for the slave units should be fed into the inputs for the load bank control program on the slave stands.

```
@LOAD_OUTPUT_LABELS
   LB_1_PH_A_Load    LB_1_PH_B_Load    LB_1_PH_C_Load
   LB_2_PH_A_Load    LB_2_PH_B_Load    LB_2_PH_C_Load
   LB_3_PH_A_Load    LB_3_PH_B_Load    LB_3_PH_C_Load
   LB_4_PH_A_Load    LB_4_PH_B_Load    LB_4_PH_C_Load
```

- The label for an INTEGER variable must be defined that indicates the actual number of phases to which load should be applied. The value is set by a computed expression or test procedure. If multiple load banks are slaved together, the values on all stands must be the same. If the system has three phases but all phases are to always be loaded equally, then set the number to one.

```
@LOAD_BANK_NUM_PHASES_LABEL
   NumPhases
```

- The load values and labels for a set of DIGITAL OUTPUT (DO) variables must be defined that contain the state of the outputs. The values are set by the load bank control program. Note that loads for phases A, B, and C do not necessarily need to be the same. Each line consists of:
  - The nominal load value that is applied to a phase if the DO is set TRUE – units must be included.
  - The labels for the DO variables that represents phases A, B, and C. The variables will be created if they do not already exist so it is important that Digital Output labels are created first unless the program is running on a development system.
  - A logical variable that indicates whether or not a load value should be used. The value of the logical variable can be set TRUE or FALSE with computed expressions or test procedures.
  - The list is terminated with a line that contains only a dollar sign ($)

```
@LOAD_VALUES_AND_DO_LABELS
   25[watt]     A1_25W    B1_25W     C1_25W    Use_25W
   50[watt]     A1_50W    B1_50W     C1_50W    Use_50W
   100[watt]    A1_100W   B1_100W    C1_100W   Use_100W_1
   100[watt]    A2_100W   B2_100W    C2_100W   Use_100W_2
   200[watt]    A1_200W   B1_200W    C1_200W   Use_200W_1
   200[watt]    A2_200W   B2_200W    C2_200W   Use_200W_2
   500[watt]    A1_500W   B1_500W    C1_500W   Use_500W_1
   500[watt]    A2_500W   B2_500W    C2_500W   Use_500W_2
   1.0[kw]      A1_1000W  B1_1000W   C1_1000W  Use_1000W_1
   1.0[kw]      A2_1000W  B2_1000W   C2_1000W  Use_1000W_2
   2.0[kw]      A1_2000W  B1_2000W   C1_2000W  Use_2000W_1
```

```
2.0[kw]      A2_2000W B2_2000W   C2_2000W  Use_2000W_2
4.0[kw]      A1_4000W B1_4000W   C1_4000W  Use_4000W_1
4.0[kw]      A2_4000W B2_4000W   C2_4000W  Use_4000W_2
6.0[kw]      A1_6000W B1_6000W   C1_6000W  Use_6000W_1
6.0[kw]      A2_6000W B2_6000W   C2_6000W  Use_6000W_2
8.0[kw]      A1_8000W B1_8000W   C1_8000W  Use_8000W_1
8.0[kw]      A2_8000W B2_8000W   C2_8000W  Use_8000W_2
```

- The label of the `DIGITAL OUTPUT (DO)` variable for the master load switch variable value must be `TRUE` before the load bank control program will select an output load.

```
@MLS_LABEL
   MasterLoad
```

3.4 Configuring Real Power Control

The multi load bank control program is only one part of the CyFlex system that can be used for controlling the load being applied to a generator set. If open loop switching to nominal settings is sufficient, then the program can work on its own. However, closed loop control is often desired to compensate for changes in load values from their nominal settings.

When closed loop control is desired, the standard CyFlex PID control task is applied in a relatively conventional manner. In systems where a single switch controls the load for all electrical phases, a single instance of the control task is used. Where both real and reactive control is desired, then two instances are required.

Six instances are used, three for real and three for reactive, for load banks that can manipulate each phase independently,

All instances of control for real power are configured in the same manner:

1. Set the feedback variable to a measured value of electrical power:
    - The measured value can be for a single phase or for the entire generator set output.
    - The measurement should be sampled at a rate that is at least as fast as the controller to accomplish reliable control.
    - The measurement should be sufficiently filtered so that random noise on the signal is removed.
2. Set the output of the control loop to a real variable that is mapped to the input of the multi load bank control program.
3. Set the target value for the loop with any one of the following:
    - Standard CyFlex commands
    - User command scripts
    - Test procedures
    - Computed expressions

4. Define a feed forward variable and set the value to the desired load value at the same time as the loop target is set:
    - This can be done by a command script or by a test procedure.
    - Computed expressions can be used to arbitrate requests if desired.
5. Set `P`, `I`, and `D` gains as desired. However, since the output will be very close using the feed forward term, only modest integral correction is required.
6. Set output and/or integral bounds via CyFlex variables.
    - Output bounds can be used to prevent overloading a genset if the load bank capacity is larger than the generator set output.
    - Integral bounds can prevent wind-up in circumstances where the desired load cannot be reached.

Refer to *Appendix B. Sample Real Power Control Specification* on page 15 for an example configuration file.

3.5 Configuring Reactive Power Control

Multiple approaches to controlling the reactive power on a generator set can be used depending on the desired user target. If the object of the controller is to set a reactive value directly, then the measured value is adjusted until the target is achieved.

If power factor is to be the controlled parameter, then some intermediate calculations may be necessary:

- The output of the control loop should be a real variable that represents the reactive power that is to be applied to the generator set.
- The loop output is mapped to the load bank controller input.
- The desired power factor must be converted to desired reactive power using a computed expression that is a function of desired PF and measured or desired real power.
- The loop target and feed forward variables are set to the desired reactive power.
- Similar integral gains to those used for real power should be used on reactive power.

Refer to Appendix *C. Sample Reactive Power Control Specification* on page 17 for an example configuration file.

3.6 Configuring the Generator to Behave as a Dynamometer

The various capabilities of CyFlex can be linked together so that a generator/load bank combination can act the same as a dynamometer and respond to speed and torque commands.

Execute the following steps to configure the generator to behave as a dynamometer:

1. Modify a voltage regulator that accepts an analog input to adjust the alternator output voltage:
    a. Using the manual adjustment potentiometer on the voltage regulator, set the nominal output voltage at the highest allowable operating speed to be 4-5% below the maximum voltage rating of the load bank.
    b. Adjust or disable the Under Frequency Roll Off (UFRO) settings so that the generator can produce sufficient electrical power to load the engine at lower speeds.
2. Calculate filtered engine speed and torque values based on measured frequency, electrical power, and alternator efficiency. Note the following:
    - Speed may be measured directly or may be calculated as (X * electrical frequency) where the value of X depends on the number of alternator poles. For larger engines where the 60 Hz speed is 1800 rpm, X equals 30.
    - Engine torque is calculated as (electrical power / generator efficiency) or may be measured directly with an in-line torque meter.
    - The alternator efficiency is normally published by the manufacturer for 50 and 60 Hz (1500 and 1800 rpm for larger alternators) but the curves often do not go down to very low loads.
    - Filling in the table for speeds and loads other than those published requires good engineering judgement. Errors in efficiency will result in calculating the wrong engine torque.
    - Use a 3D table to store the efficiency values as shown in *Appendix D. Sample Efficiency Table* on page 19.. For this example, the output from the table is called `AltTableEff`.
    - The 3D table mentioned above was generated by using an inline meter to measure torque directly and then calculating efficiency from (measured electrical power / measured engine power).
    - Once the efficiency table is completed, the in-line torque meter is no longer required except as a calibration device.
    - The efficiency table can be used for other similarly sized alternators.
    - The speed and torque variables for control in the special channels file must be set to the above calculated or measured variables.
3. Configure the dynamometer control loop:
    a. Send the output to a real variable that has units of percent (pct.), Use `dyno_cmd` for this example.
    b. Multiply the `dyno_cmd` times the maximum engine torque in a computed expression to arrive at a commanded torque: `dyno_cmd_torq`.

c. For engine control modes one and two, set the PID gains to zero and specify a feed forward variable which is the torque reference value times the maximum engine torque: Torque_RF * max_eng_torque.
d. In control modes four and five, set the PID gains as required to control speed. The feed forward value is generally set to zero but may be used if desired and an appropriate control model is available.
e. Set reasonable output bounds according to the engine being tested.

4. Configure the throttle control loop as is normally done.

3.7 Converting the Torque Output Command from the Dyno Controller to an Electrical Power Request

Execute the following steps to convert the torque output command from the dyno controller to an electrical power request to the multi load bank control program:

1. Multiply the speed reference value (Speed_RF) by the desired torque command to obtain the desired engine power:

   DesiredEnginePwr = Speed_RF * dyno_cmd_torq

2. Multiple the desired engine power is by the alternator efficiency at the desired generator operating condition to obtain the desired electrical load:

   DesiredAltPower = AltTableEff * DesiredEnginePwr

   The alternator efficiency above is taken from the 3D table mentioned earlier.

3. Correct the desired power. Because the published electrical increments for the load bank are only valid at the nameplate voltage, and because the output voltage of the alternator depends on the engine speed, a correction to the desired power must be made. This correction requires two steps:
   a. A table of speed versus nominal alternator voltage must be created. Refer to Appendix *E. Sample Speed Versus Voltage Table* on page 20 for an example.
   b. The desired electrical load found above must be multiplied by:

      (Nameplate voltage / table voltage) ^ 2

4. The final formula for desired electrical load in CyFlex looks like:

   Speed_RF * dyno_cmd_torq * AltTableEff * @pow(AltNamePlateVolts / AltTableVolts) , 2[none])

3.8 Configuring the Voltage Regulator (VR) Control Loop

When configuring the alternator to behave as a dynamometer, a CyFlex User Loop is set up with an analog output that is directed to the voltage regulator. This loop is used to bias the alternator output voltage so that load settings between the discrete load bank steps can be achieved. Logic implemented by the event response program decides when to set the dyno and voltage controllers in open and closed loop.

Note the following:

- When the engine is in control modes one or two where the dyno (alternator) is controlling torque:

- The dyno loop uses feed forward only and that value is based on the desired engine power. It will not change until a new setpoint is requested. Therefore, in these engine control modes, the dyno controller should always stay in closed loop.
  - Switching from open to closed loop and back again may cause the feed forward term to be carried to the integral term, doubling the output. This behavior is inherent with the old CyFlex control task (`ctrl_task`) but can be turned off with the new one (`eng_ctrl_task`).
- The voltage regulator loop uses a calculated value for load bank requested power *error* as the feedback value.
  - The target for this loop is always set to zero.
- While the desired torque is ramping (Torque_RF != Torque_TR), the voltage regulator control is set to open loop.
- When the ramping is complete, and a user selectable time delay has been met, the voltage regulator:
  - Integral term is reset
  - Controller is set to closed loop
  - Gains are set for engine control modes one and two
  - A freeze load event is sent to the multi load bank control program which causes the switch settings to be fixed
  - Control loop will adjust the alternator output voltage slightly to bring the desired power error to zero
  - Output limits on the VR loop should be set such that the maximum safe voltage of the load bank is not exceeded. This can easily be done with a computed expression.

- When the engine control mode is four or five:
  - The dyno uses conventional PID gains to control speed
  - While the desired speed is ramping (Speed_RF != Speed_TR), the dyno is in closed loop and the voltage regulator is set to open loop
  - When ramping is complete, the speed is within tolerance, and a user selectable delay has been met:
    - The dyno controller is set to open loop
    - The voltage controller:
      - Integral term is reset
      - Is set to closed loop
      - Gains are set for speed control

3.9 Getting Started

Execute the following steps before activating the system:

1. Take the engine to the speed where the alternator produces the nameplate voltage of the load bank
2. Manually set each digital output or switch ON/OFF and ensure that the load bank is producing very close to the advertised load
    a. Fix any digital output, switch, relay, or load resistor that is found to be defective.
3. Double check to see that the voltage out of the generator matches the table that was created earlier. Experience shows that periodic VR adjustments might be required
4. Make sure that the `gen_labels` file contains the correct maximum alternator and load bank capacity.
5. Make sure that the rated and torque peak torque values in the engine specifications file are correct
6. Make sure that the special channels file contains the correct variable names for speed and torque control
7. Set all logical variables true that represent whether a load step is working or not. These variables are normally defined in a `gen_labels` file and must be manually set TRUE or with a command script since logical variables are set FALSE at system startup
8. Check the Dyno and Load Bank Error controller states:
    a. Only one controller should be in closed loop at a time when running in engine modes four or five.
    b. The Dyno controller will always stay in closed loop when in engine modes one or two.

Note:

This document does not cover using an alternator for control modes three or six

Appendices

Appendix A. Sample Configuration File

```
                                # the name of this instance of
                                # the program - variables that are
                                # created will be owned by this name

@LOADBANK_CONTROLLER_REGISTERED_NAME
   ResistiveControl
                                # NOTE: all events listed are inputs to
                                #       this program - none are outputs

                                # the name of the event that will enable
                                # changes in load

@LOAD_CHANGE_ENABLE_EVENT_NAME
   ChangeLoadEv
                                # the name of the event that will
                                # cause the load to be frozen at the
                                # current requested value
@LOAD_FREEZE_EVENT_NAME
   FreezeLoadEv
                                # the name of the event that will set
                                # the load to zero - it is the only
                                # safe way to assure that all relays
                                # are open
@ZERO_LOAD_EVENT_NAME
   ClearLoadEv
                                # the name of the event that will
                                # set load back to the value when
                                # it was frozen
@RESTORE_LOAD_EVENT_NAME
   RestoreLoadEv
                                # the name of the event that will
                                # cause this instance of the control
                                # task to die
@RELEASE_TASK_EVENT_NAME
   StopKwCtrlEv
                                # NOTE: unless stated explicitly,
                                #       all variables are inputs
                                #       to this program -  if they
                                #       do not exist at the start
                                #       of execution of this program
                                #       they will be created

                                # labels for up to four logical
                                # variables that will indicate
                                # whether a load bank has a problem

                                # NOTE - the number of labels for this
                                #        keyword will establish the
                                #        number of load banks being
                                #        controlled - similar
```

```
                                #        keywords must have the same
                                #        number of entries
@LOAD_BANK_TROUBLE_LABELS
   LB1_Trouble    LB2_Trouble    LB3_Trouble    LB4_Trouble

                                # labels for up to four logical
                                # variables that will indicate
                                # whether load can be applied
                                # to a particular load bank - the
                                # labels MUST be in the same order
                                # as the trouble labels
@LOAD_ENABLE_LABELS
   LB1_LoadEnable LB2_LoadEnable LB3_LoadEnable LB4_LoadEnable

                                # labels for up to four real variables
                                # that will contain the maximum load
                                # that should be requested from a
                                # particular load bank - useful when
                                # load banks are of different sizes
@LOAD_BANK_MAX_LOAD_LABELS
   LB1_MaxLoad    LB2_MaxLoad    LB3_MaxLoad    LB4_MaxLoad

                                # labels for up to three real variables
                                # that will indicate the desired load
                                # for each phase - values are taken
                                # as phases A, B, and C
@DESIRED_LOAD_INPUT_LABELS
   PhaseA_RqstLoad   PhaseB_RqstLoad   PhaseC_RqstLoad

                                # the label of a real variable that
                                # will contain the maximum load
                                # imbalance that can be applied -
                                # the maximum load for a phase will
                                # be limited to this value above
                                # the minimum load requested
@LOAD_IMBALANCE_LIMIT_LABEL
   MaxLdImbalance
                                # labels for up to four groups of up to
                                # three variables in each group that
                                # will show how much load is being
                                # absorbed by each phase of the
                                # various load banks - these labels
                                # represent program OUTPUTS

                                # NOTE: outputs for load banks other
                                #       than the local one must be
                                #       node-linked to the slave
                                #       computers, evaluated with
                                #       computed expressions, and used
                                #       as desired load inputs
@LOAD_OUTPUT_LABELS
   LB_1_PH_A_Load    LB_1_PH_B_Load    LB_1_PH_C_Load
   LB_2_PH_A_Load    LB_2_PH_B_Load    LB_2_PH_C_Load
   LB_3_PH_A_Load    LB_3_PH_B_Load    LB_3_PH_C_Load
   LB_4_PH_A_Load    LB_4_PH_B_Load    LB_4_PH_C_Load
```

```
                                # the label for the integer variable
                                # that will contain the number of
                                # phases that are connected for the
                                # genset under test
@LOAD_BANK_NUM_PHASES_LABEL
   NumPhases
                                # up to 48 groups of variable labels
                                # that represent the load values
                                # for each digital output - labels are
                                # to be entered in the order of
                                # phase A, B, C, and validity logical -
                                # valid CyFlex units for real power
                                # must be used

                                # NOTE: the load value represents the
                                #       load per phase
@LOAD_VALUES_AND_DO_LABELS
   25[watt]       A1_25W     B1_25W       C1_25W      Use_25W
   50[watt]       A1_50W     B1_50W       C1_50W      Use_50W
   100[watt]     A1_100W    B1_100W      C1_100W     Use_100W_1
   100[watt]     A2_100W    B2_100W      C2_100W     Use_100W_2
   200[watt]     A1_200W    B1_200W      C1_200W     Use_200W_1
   200[watt]     A2_200W    B2_200W      C2_200W     Use_200W_2
   500[watt]     A1_500W    B1_500W      C1_500W     Use_500W_1
   500[watt]     A2_500W    B2_500W      C2_500W     Use_500W_2
   1.0[kw]      A1_1000W   B1_1000W     C1_1000W    Use_1000W_1
   1.0[kw]      A2_1000W   B2_1000W     C2_1000W    Use_1000W_2
   2.0[kw]      A1_2000W   B1_2000W     C1_2000W    Use_2000W_1
   2.0[kw]      A2_2000W   B2_2000W     C2_2000W    Use_2000W_2
   4.0[kw]      A1_4000W   B1_4000W     C1_4000W    Use_4000W_1
   4.0[kw]      A2_4000W   B2_4000W     C2_4000W    Use_4000W_2
   6.0[kw]      A1_6000W   B1_6000W     C1_6000W    Use_6000W_1
   6.0[kw]      A2_6000W   B2_6000W     C2_6000W    Use_6000W_2
   8.0[kw]      A1_8000W   B1_8000W     C1_8000W    Use_8000W_1
   8.0[kw]      A2_8000W   B2_8000W     C2_8000W    Use_8000W_2
$
                                # terminate the list with a "$"

                                # the label of the digital output that
                                # controls the master load control -
                                # outputs will not be changed if the
                                # master load control is open
@MLS_LABEL
   MasterLoad
```

Appendix B. Sample Real Power Control Specification

```
# Name: /specs/ctrl_specs.<PID_LOOP>

@USER_LOOP_CTRLER
#       label        command            interval     open-loop
#       root         control units                   ramp rate
        RealPwrPhA   kw                 MED          0

@CTRLER_FEEDBACK_VRBL
#       label        closed-loop        error-term
#                    ramp rate          tolerance
        RealPwrPhAMsd   0               0.025

@USER_LOOP_VRBL_OPTS_TR
#       display      history            history
#       precision    active             tolerance
        3            yes                1

@USER_LOOP_VRBL_OPTS_ER
#       display      history            history
#       precision    active             tolerance
        3            yes                1

@USER_LOOP_VRBL_OPTS_IT
#       display      history            history
#       precision    active             tolerance
        3            yes                1

@USER_LOOP_VRBL_OPTS_PT
#       display      history            history
#       precision    active             tolerance
        3            yes                1

@USER_LOOP_VRBL_OPTS_CM
#       display      history            history
#       precision    active             tolerance
        3            yes                1

@CTRLER_GAINS
#   proportional           integral          derivative
    0.0                    0.1               0

@CTRLER_INTEGRAL_BOUND_LABELS
# Lower          Upper
RealPwrIntLB     RealPwrIntUB

@CTRLER_OUTPUT_BOUNDS
# lower   upper
  0       30

@CTRLER_FEED_FORWARD
#   label             active         gain
    RealPwrPhASetpt     YES            1

@CTRLER_OUTPUT_CHAN
```

```
#  RV Target           Bias Gain Filter
   RV RealPwrPhARqstPID  0    1    0

$END
```

Appendix C. Sample Reactive Power Control Specification

```
# Name: /specs/ctrl_specs.<PID_LOOP>

@USER_LOOP_CTRLER
#       label        command            interval     open-loop
#       root         control units                   ramp rate
        ReacPwrPhA   kw                 MED          0

@CTRLER_FEEDBACK_VRBL
#       label        closed-loop        error-term
#                    ramp rate          tolerance
        ReacPwrPhAMsd      0            0.01875

@USER_LOOP_VRBL_OPTS_TR
#       display      history            history
#       precision    active             tolerance
        3            yes                1

@USER_LOOP_VRBL_OPTS_ER
#       display      history            history
#       precision    active             tolerance
        3            yes                1

@USER_LOOP_VRBL_OPTS_IT
#       display      history            history
#       precision    active             tolerance
        3            yes                1

@USER_LOOP_VRBL_OPTS_PT
#       display      history            history
#       precision    active             tolerance
        3            yes                1

@USER_LOOP_VRBL_OPTS_CM
#       display      history            history
#       precision    active             tolerance
        3            yes                1

@CTRLER_GAINS
#   proportional           integral          derivative
    0.0                    0.1               0

@CTRLER_INTEGRAL_BOUND_LABELS
# Lower        Upper
  ReacPwrIntLB ReacPwrIntUB

@CTRLER_OUTPUT_BOUNDS
# Lower Upper
0       30

@CTRLER_FEED_FORWARD
#   label            active         gain
    ReacPwrPhASetpt  YES            1

@CTRLER_OUTPUT_CHAN
```

```
#  RV Target             Bias Gain Filter
   RV ReacPwrPhARqstPID  0    1    0

$END
```

Appendix D. Sample Efficiency Table

```
# Description

Efficiency table as a f() of speed and power
# Table Type     Table Name
  5              loc_two_d_4

# x units     y units    z units
  rpm         kw          pct

# x label
  ctl_spd

# x coordinate values
   -1          650          900          1200          1500          1800          2100          3000
# y/z coordinate pairs
# kW  Eff    kW    Eff    kW   Eff     kW      Eff      kW      Eff      kW      Eff      kW      Eff      kW    Eff
 -1    9    0.6     9    10    48     12       48      -1       10       8       20      -1       10      -1    10
 11   28    2.8    28    38    80     50       80      20       46      49       66     116       66     116    66
 20   45    6      45    56    85    111       89     135       86     158       87     231       88     231    88
 40   55    9.2    55    74    88    112       89     214       91     253       91     463       94.5   463    94.5
 61   63    12     63    92    89    173       91     388       95.4   348       93     694       95.85  694    95.85
 81   70    15.5   70   110    89    223       93     465       96.0   443       94     925       96.55  925    96.55
 101  70    19     70   128    91    295       94     620       96.6   719       96.0  1157       96.95 1157    96.95
 121  72    22     72   137    91    356       95     681       96.9   863       96.79 1388       97.15 1388    97.15
 142  72    24.5   72   173    92    393       96.5   795       96.9  1006       96.85 1619       97.25 1619    97.25
 162  78    31     78   259    92    449       96.45  908       96.8  1150       96.80 1851       97.3  1851    97.3
 182  81    38     81   291    91    505       96.4  1022       97.0  1294       96.68 2082       97.25 2082    97.25
 202  81    43.5   81   324    91    561       96.35 1135       96.8  1438       96.53 2313       97.2  2313    97.2
 223  84    57     84   356    90    617       96.2  1249       96.7  1582       96.30 2544       97.15 2544    97.15
```

Appendix E. Sample Speed Versus Voltage Table

```
#*************************************************************#
#               USE THIS FOR 1.5 MW ALTERNATOR                #
#*************************************************************#

#revision code
@AUG07

# user comment
no comment

#     last calibrated on: Thu May 08 22:51:00 2014
#next calibration due on: 05/07/2015

#last_cal(time_t)        cal_interval
1399603860    52[week]

#Calibration Method
NA

#Technician
NA

#Description
NA

#calibration units
#eng_units   raw_units
   V          RPM

# calibration limits
#  min_cal_range        max_cal_range      tolerance   tolerance_units
   0[V]                 500[V]             1           V

#MUA Instance Name
none

#Sensor Model
mx342
```

```
#Sensor Serial Number
na

#Sensor Range Units
V

#Sensor range limits
#  lower_limit         upper_limit
   0[V]                600[V]

#type_code
LONG_INTERPOLATION

#number of points
17

#    RPM    V
    0      0
    750    208
    800    220
    900    244
    1000   267
    1100   291
    1200   315
    1300   340
    1400   368
    1500   395
    1600   421
    1700   450
    1800   480
    1900   486
    2000   491
    2100   498
    3000   500
```

Appendix F. Sample Gen Labels File

```
VERSION_2  (do not remove this line)
@REG_NAME
GL_Cntrl

# ****************************
# Real variables section
# ****************************


#*****************************************************************************
#
#  Calculate torque flange variables
#
#*****************************************************************************

#  label      units     format   initial_value   interval/event   hst_flag    tolerance
TargetPwr     kw        1        -               FAS             NO          1 -
SPEED_RF * dyno_cmd_trq

#  label      units     format   initial_value   interval/event   hst_flag    tolerance
TargetEff     pct       3        -               FAS             NO          1 -

@long_3d_comp( TargetPwr, 4[none])

#  Torque flange filter factor
#  label      units     format   initial_value   interval/event   hst_flag    tolerance
HBM_FF        none      3        .995             -               NO          1 -

-

#  Calculated power from a torque flange
#  label      units     format   initial_value   interval/event   hst_flag    tolerance
EngPwrTF      kw        2        -               FAS             NO          1 -
SPEED * HBM_TORQUE

#  Calculated efficiency when using a torque flange
#  label      units     format   initial_value   interval/event   hst_flag    tolerance
Eff_calc      pct       3        0               FAS             NO          1  -
```

```
LdbnkPwr / EngPwrTF

# The filtered alternator efficiency
# The new value = [(1 - filt) * input value] + [filt * last value]
#  label      units     format   initial_value   interval/event   hst_flag    tolerance
FltrdCalcEff  pct       3            -           MED              OFF              1 -
((1[none] - AltrnFltrCoeff) * Eff_calc) + (AltrnFltrCoeff * Eff_calc)

#  Filtered torque flange torque
#  label      units     format   initial_value   interval/event   hst_flag    tolerance
HBM_TORQUE    N_M       1            -           MED              OFF              1 -
((1[none] - HBM_FF ) * HBM_TORQUE_Flt1) + ( HBM_TORQUE_Flt1 * HBM_FF  )

#*************************************************************************
#
#  End calculate torque flange variables
#
#*************************************************************************
```

```
#*********************************************************************************
#
#  Variables for speed/power calculation and control
#
#*********************************************************************************

#  Calculate the average voltage from the 3 individual phase measurements
#  label      units     format   initial_value   interval/event   hst_flag   tolerance
AvgVolt       V         1              -         FAS              OFF          1 -
(LB_a_V + LB_b_V + LB_c_V) / 3[none]

#  Calculate the average current from the 3 individual phase measurements
#  label      units     format   initial_value   interval/event   hst_flag   tolerance
AvgCrnt       A         1              -         FAS              OFF           1 -
(LB_a_A + LB_b_A + LB_c_A) / 3[none]

#  If the current is > 0 then use it, else set it to 0.  This
#  is to get away from having a negative (motoring) power/torque.
#  label      units     format   initial_value   interval/event   hst_flag   tolerance
AvgCrntGt0    A         1            -          FAS              OFF           1 -
if (AvgCrnt >= 0[A]) then AvgCrnt else 0[A]


   # The Electrical power consumed by load bank
   # P(kw)= sqrt(3) x PF x I(a) x V(L-L)/1000
   # Power factor (PF) = 1 for resistive load banks.
   # 1.7205080757 = sqrt(3)
   # CyFlex takes care of dividing by 1000

#  label      units     format   initial_value   interval/event   hst_flag   tolerance
LdbnkPwr      kw        2              -         FAS              OFF           1 -
AvgVolt * AvgCrntGt0 * 1.73205080757[none]

###### Black 1.15MW Alt. loc_two_d_4.tbl

#  The alternator efficiency is taken from a 3D table
#  label      units     format   initial_value   interval/event   hst_flag   tolerance
AltrnEff      pct       4           96          FAS              OFF           1 -
@long_3d_comp( LdbnkPwr, 4[none])
```

```
#  The alternator efficiency will be recursively filtered
#  label      units     format   initial_value   interval/event   hst_flag   tolerance
AltrnFltrCoeff none     2           0.8          -                OFF           1 -
-

#  The filtered alternator efficiency
#  The new value = [(1 - filt) * input value] + [filt * last value]
#  label      units     format   initial_value   interval/event   hst_flag   tolerance
FltrdAltrnEff  pct      4            -           FAS              OFF           1 -
((1[none] - AltrnFltrCoeff) * AltrnEff) + (AltrnFltrCoeff * FltrdAltrnEff)

#  The engine power including LdbnkPwr and alternator efficiency.
#  The value will be set to zero when the speed is very low.
#  label      units     format   initial_value   interval/event   hst_flag   tolerance
EngPower      HP        1            -           FAS              OFF           1 -
SpeedGT400 * (LdbnkPwr / FltrdAltrnEff)


#  Map the actual speed measurement to the variable used
#  in this generic file.
#  label       units     format   initial_value   interval/event   hst_flag  tolerance
Speed          rpm       1              -         FAS              OFF           1 -
SPEED


#  Calculated Torque from calculated power and measured Speed
#  label       units     format   initial_value   interval/event   hst_flag  tolerance
raw_trq        n_m       1              -         FAS              OFF           1 -
EngPower / SPEED

#  Torque will be filtered and this variable allows the user
#  to change the filter coefficient on the fly.
#label         units     format   initial_value   interval/event   hst_flag  tolerance
TrqFltrCoeff   none      1           0.9          -                OFF           1 -
-

#  The new value = [(1 - filt) * input value] + [filt * last value]
#label         units     format   initial_value   interval/event   hst_flag  tolerance
Torque         n_m       1            -           MED              OFF           1 -
((1[none] - TrqFltrCoeff) * raw_trq) + (TrqFltrCoeff * Torque)
```

```
#  Just to make PAM happy with units
#  label       units     format   initial_value   interval/event   hst_flag  tolerance
dyn_cm_pam    %         1               -         FAS              OFF           1 -
Dyno_CM * 100[%]


############ UPDATE THESE VALUEs PER COMPONENTS BEING USED ######################

#  Rating of the loadbank
#  label       units     format   initial_value   interval/event   hst_flag  tolerance
RatedLdbnkPwr  KW        0        1950             -               OFF           1 -
-

#  Continuouse rating of the alternator
#  label       units     format   initial_value   interval/event   hst_flag  tolerance
RatedAltOutpt  KW        0        1500             -               OFF           1 -
-

#  Some alternators will allow higher ratings for short periods
#  of time so the multiplier can be changed from something other
#  than 1.0
#  label       units     format   initial_value   interval/event   hst_flag  tolerance
OvrRatedAltOutpt  KW       0          -              -               OFF         1 -
1.0[none] * RatedAltOutpt

#  Maximum KW that we will let multi_lb_ctrl apply.
#  We'll allow the overrated alternator output for up to 1hr
#  cumulative in 6hrs, else we will be at the rated output of the alternator
#  If the alternator is more than the loadbank is capable of,
#  max is max of loadbank
#  label       units     format   initial_value   interval/event   hst_flag  tolerance
MaxLdbnkLd     KW        0          -               -               OFF          1 -
if (RatedAltOutpt <= RatedLdbnkPwr) then RatedAltOutpt else RatedLdbnkPwr

#  Variable used later. Just renaming of the rated torque.
#  label       units     format   initial_value   interval/event   hst_flag  tolerance
MaxEngTorq     n_m       0          -             -               OFF           1 -
if (ra_torq > tp_torq) then ra_torq else tp_torq
```

```
############ END UPDATE THESE VALUEs PER COMPONENTS BEING USED ####################


#  Real variable set from output of dyno control loop in percent
#  label       units     format   initial_value   interval/event   hst_flag  tolerance
dyno_cmd       pct       2               -         -               OFF           1 -
-

#  Dyno commanded value converted to n_m
#  label       units     format   initial_value   interval/event   hst_flag  tolerance
dyno_cmd_trq   n_m       2               -         FAS             OFF           1 -
dyno_cmd * MaxEngTorq/100[pct]


#   Feedforward calculation for the alternator controller which is
#  only used when the alternator is controlling torque
#  label       units     format   initial_value   interval/event   hst_flag  tolerance
DynoFdFrwrd    pct       2               -         FAS             OFF           1 -
if (EngCtrlMode == 1[none] || EngCtrlMode == 2[none]) \
   then (TORQUE_RF / MaxEngTorq) else 0[pct]
###   then (HBM_TORQUE_RF / MaxEngTorq) else 0[pct]


#  The nameplate voltage for the loadbank
#  label       units     format   initial_value   interval/event   hst_flag  tolerance
LdbnkRaVolt    volt      0        480             -               OFF           1 -
-

#  The speed used to get the nominal alternator voltage is filtered
#  using this filter factor
#  label       units     format   initial_value   interval/e vent   hst_flag tolerance
TblSpeedFltrCoeff  none  3        0.95            -               OFF           1 -
-

#  The filtered speed used to find the maximum voltage produced by
#  the AVR
#  label       units     format   initial_value   interval/event   hst_flag  tolerance
TblSpeed       RPM       3               -         FAS             OFF           1 -
### ((1[none] - TblSpeedFltrCoeff) * SPEED) + (TblSpeedFltrCoeff * TblSpeed)
```

```
SPEED_RF

#  The filtered speed error
#  label       units     format   initial_value   interval/event   hst_flag  tolerance
TblSpeedER     RPM       3              -         FAS              OFF           1 -
((1[none] - TblSpeedFltrCoeff) * SPEED_ER) + (TblSpeedFltrCoeff * TblSpeedER)


#  The nominal voltage taken from a table of voltage vs speed
#  label      units     format    initial_value   interval/event   hst_flag  tolerance
TblVolt       V         3               -         FAS              OFF           1 -
@cal_table( TblSpeed , 'volt_vs_speed')


#  The voltage correction to apply to the requested load for
#  load bank control
#  label       units     format   initial_value   interval/event   hst_flag  tolerance
VoltCrct       none      3              -         FAS              OFF           1 -
if (@pow( LdbnkRaVolt/TblVolt , 2[none]) < 8[none])   \
   then @pow( LdbnkRaVolt/TblVolt , 2[none]) else 8[none]

#  The restored value. Also used to do "open loop" ldbank settings.
#  (for starts, etc.)
#label         units     format   initial_value   interval/event   hst_flag  tolerance
LdbnkRestLd    kw        1              -         FAS              OFF           1 -
-

#  The value sent to multi_lb_ctrl
#  label       units     format   initial_value   interval/event   hst_flag  tolerance
LdbnkReqLd     kw        1              -         FAS              OFF           1 -
if UseRestoredVals then LdbnkRestLd else ((dyno_cmd_trq * TblSpeed) * \
   VoltCrct * TargetEff )

#  This variable is used by the load bank control process as an output
#  of the load that is being requested
# label         units   format    initial_value  interval/event   hst_flag  tolerance
LdbnkOutputLd   kw      1         0.0                -            OFF      1.00
-

#  The error from multi_lb_ctrl output vs what is requested.
```

```
#  label      units     format   initial_value   interval/event   hst_flag   tolerance
LdbnkErrLd    kw         4              -          FAS             OFF           1 -
LdbnkReqLd - LdbnkOutputLd

#  The error from multi_lb_ctrl output vs what is requested.
#  Changed the Speed error to TblSpeedER/SPEED_RF instead of TblSpeedER/TblSpeed.
#  The percentage error should be based on absolute error vs reference
#  label      units     format   initial_value   interval/event   hst_flag   tolerance
LdbnkErr      pct       2              -          FAS             OFF           1 -
if ((EngCtrlMode == 1[none] || EngCtrlMode == 2[none])) \
### then ( ( HBM_TORQUE_RF - HBM_TORQUE ) / HBM_TORQUE_RF ) \
### else if( ( EngCtrlMode == 4[none] || EngCtrlMode == 5[none] ) ) \
### then (-TblSpeedER/SPEED_RF) else 0[pct]
 then ( ( TORQUE_RF - TORQUE ) / TORQUE_RF ) \
 else if( ( EngCtrlMode == 4[none] || EngCtrlMode == 5[none] ) ) \
 then (-TblSpeedER/SPEED_RF) else 0[pct]

#  Dyno controller integral lower bound
#  label         units     format   initial_value   interval/event  hst_flag tolerance
DynoIntLB        pct       2           -15           -              OFF          1 -
-

#  Dyno controller integral upper bound
#  label         units     format   initial_value   interval/event  hst_flag tolerance
DynoIntUB        pct       2           15            -              OFF          1 -
-

#  AVR bias controller integral lower bound
#  label         units     format   initial_value   interval/event  hst_flag tolerance
LdbnkErrIntLB    pct      2             -15          -              OFF          1 -
-

#  AVR bias controller integral upper bound
#  label         units     format   initial_value   interval/event hst_flag  tolerance
LdbnkErrIntUB    pct      2            5            FAS            OFF          1 -
if AvgVolt < 450[volt] then 10[pct] else 3[pct]

#  AVR bias controller lower bound
#  label         units     format   initial_value   interval/event  hst_flag tolerance
LdbnkErrOutLB    pct      2           -15           -              OFF          1 -
```

```
-

#  AVR bias controller upper bound
#  label         units     format   initial_value   interval/event hst_flag  tolerance
LdbnkErrOutUB    pct       2            5          FAS            OFF           1 -
if AvgVolt < 450[volt] then 10[pct] else 3[pct]


#  The lower bound for the error ratio calculation
#  label      units     format   initial_value   interval/event   hst_flag  tolerance
LdbnkErrRatLB  pct        2           -5         -                OFF           1 -
-

#  The upper bound for the error ratio calculation
#  label      units     format   initial_value   interval/event   hst_flag  tolerance
LdbnkErrRatUB  pct        2            5         -                OFF           1 -
-

#  The error ratio calculation stated in terms of load bank applied error
#  vs actual loadbank power
#  label      units     format   initial_value   interval/event   hst_flag  tolerance
LdbnkErrRat   pct        2              -         FAS              OFF           1 -
if (LdbnkReqLd > 1[kw]) \
   then (if (LdbnkErrLd/LdbnkPwr) < LdbnkErrRatLB \
      then LdbnkErrRatLB \
      else if (LdbnkErrLd/LdbnkPwr) > LdbnkErrRatUB \
         then LdbnkErrRatUB \
         else (LdbnkErrLd/LdbnkPwr)) \
   else 0[pct]

#  Gain multiplier for the bias voltage calculation
#  label          units     format   initial_value   interval/event hst_flag tolerance
AltrnBiasVltGain  none      2              1         -                OFF       1 -
-

#  The new voltage that should be output from the alternator.  Will be higher or lower
#  than average voltage based on power being higher or lower than requested.
#  label          units     format   initial_value   interval/event hst_flag tolerance
AltrnBiasVlt      volt      2              -         FAS              OFF       1 -
if ( AvgVolt + ( AvgVolt * ( AltrnBiasVltGain * LdbnkErrRat ))) <= LdbnkRaVolt \
```

```
   then ( AvgVolt + ( AvgVolt * ( AltrnBiasVltGain * LdbnkErrRat ))) \
   else LdbnkRaVolt

#  This variable is used by the load bank control process as a limit on
#  the load imbalance between phases if controlled independently
#  label         units   format initial_value interval/event hst_flag  tolerance
MaxLoadImbalance kw     1      0.0           -              off       1.00
-

#   Calculate the time that the speed is in tolerance
#  label         units   format initial_value interval/event hst_flag  tolerance
TmrSpeedInTL    sec     1      0.0           MED              off       1.00
if SPEED_TL then (TmrSpeedInTL + 0.100[sec]) else 0[sec]

#  Speed in tolerance delay timer
#  label         units   format initial_value interval/event hst_flag  tolerance
SpeedInTLDlyTm  sec       1      5.0           -              off       1.00
-

#  Calculate the time that the torque is in tolerance
#  label         units   format initial_value interval/event hst_flag  tolerance
TmrTorqueInTL   sec       1      0.0           MED              off       1.00
if Dyno_MD then (TmrTorqueInTL + 0.100[sec]) else 0[sec]

#  Torque in tolerance delay timer
#  label         units   format initial_value interval/event hst_flag  tolerance
TorqueInTLDlyTm  sec      1      0.0           -              off       1.00
-

#  Load bank requested load
#  label         units   format initial_value interval/event hst_flag  tolerance
LdbnkReqLd       kw       1       -             -              off       1.00
-

$ end of REAL_VARIABLE section - start of INTEGERS
```

```
#############      INTEGER VARIABLES SECTION ########################################

   #  Number of separate phases to be controlled by the load bank
#label          units    initial_value interval hst_flag transition_event
#expression
phase           none     1             -        OFF      -
1[none]


$  end of INTEGER_VARIABLE section - start of LOGICALS

# ****************************
# Logical variables section
# ****************************
#

#  Indicator of trouble with the load bank - use if feedback bits are wired
#label         true_event false_event true_desc     false_desc  int/event  hst_flag
LBTrouble      -          -           TROUBLE       OK          FAS       OFF
#!ldbnk1_air_ok || !ldbnk1_blwr1_sts || !ldbnk1_blwr2_sts || !ldbnk1_blwr3_sts || !ldbnk1_ovrtmp_ok
-

#  Indicator of whether to use the load bank
#label         true_event false_event true_desc     false_desc  int/event  hst_flag
UseLB          -          -           USE           NOTUSE      SLO        OFF
ON

#  Indicates whether the master load switch is on -
#  Should be set to false in an emergency is detected
#label     true_event  false_event  true_desc   false_desc   interval
#Mstr_load -           -            MLS_ON      MLS_OFF       -
#-
#ldbnk_enbl && ldbnk_ctrl && !LBTrouble

# Indicates whether the load increment should be used
#label     true_event  false_event  true_desc   false_desc   interval
Ldbnk_5kw_Vld    -       -            OK         OFF           -
-

#label     true_event  false_event  true_desc   false_desc   interval
```

```
Ldbnk_10kw_1_Vld    -       -           OK      OFF          -
-

#label     true_event  false_event  true_desc   false_desc   interval
Ldbnk_10kw_2_Vld    -       -           OK      OFF          -
-

#label     true_event  false_event  true_desc   false_desc   interval
Ldbnk_25kw_Vld    -       -           OK       OFF          -
-

#label     true_event  false_event  true_desc   false_desc   interval
Ldbnk_50kw_Vld    -       -           OK       OFF          -
-

#label     true_event  false_event  true_desc   false_desc   interval
Ldbnk_100kw_1_Vld    -       -         OK       OFF          -
-

#label     true_event  false_event  true_desc   false_desc   interval
Ldbnk_100kw_2_Vld    -       -         OK        OFF         -
-

#label     true_event  false_event  true_desc   false_desc   interval
Ldbnk_100kw_3_Vld    -       -           OK      OFF      -
-

#label     true_event  false_event  true_desc   false_desc   interval
Ldbnk_200kw_1_Vld    -       -           OK      OFF      -
-

#label     true_event  false_event  true_desc   false_desc   interval
Ldbnk_200kw_2_Vld    -       -           OK      OFF      -
-

#label     true_event  false_event  true_desc   false_desc   interval
Ldbnk_200kw_3_Vld    -       -           OK      OFF      -
-

#label     true_event  false_event  true_desc   false_desc   interval
```

```
Ldbnk_500kw_1_Vld    -       -            OK      OFF      -
-

#label     true_event  false_event  true_desc   false_desc   interval
Ldbnk_500kw_2_Vld    -       -            OK      OFF      -
-

#  Calculate if the speed has been in tolerance long enough
#  label     true_event  false_event  true_desc   false_desc   interval
SpeedInTL    e_SpdInTL    e_SpdOutTL    IN       OUT          FAS
if (SPEED_RF == SPEED_TR) && (TmrSpeedInTL >= SpeedInTLDlyTm) &&  \
  ((EngCtrlMode == 4[none]) || (EngCtrlMode == 5[none] ))   \
  then ON else OFF

#  Calculate if the torque has been in tolerance long enough
#  label     true_event  false_event  true_desc   false_desc   interval
TorqueInTL   e_TrqInTL   e_TrqOutTL   IN          OUT          FAS
###if(( HBM_TORQUE_RF == HBM_TORQUE_TR ) && (TmrTorqueInTL >= TorqueInTLDlyTm) &&  \
if(( TORQUE_RF == TORQUE_TR ) && (TmrTorqueInTL >= TorqueInTLDlyTm) &&  \
  ((EngCtrlMode == 1[none]) || (EngCtrlMode == 2[none] )))   \
  then ON else OFF

#label     true_event  false_event  true_desc   false_((desc   interval
UseRestoredVals  -     -            TRUE        FALSE        -
-

$   end of LOGICAL_VARIABLE section - start of PRE-EXISTING VARIABLES



###############      PRE-EXISTING VARIABLES SECTION   ############################
#Sample: #label       process_interval or event_name
#Sample: computed expression



Dyno_OL        FAS
if( (EngCtrlMode == 1[none]) || (EngCtrlMode == 2[none] ) )  \
   then DynoFdFrwrd else Dyno_OL
```

```
$   end of PRE-EXISTING VARIABLE section - start of STRINGS


##############      STRING VARIABLES SECTION   ###################################
#Sample: #label          initial_value          event/interval   hst_flag
#Sample: ar_string       '-'              -          OFF
#Sample: -

LdbnkCtrlDesc  '' - OFF
-

$   end of STRING_VARIABLE section - start of COMPUTED STATUS


##############      COMPUTED STATUS SECTION   ###################################

LBTrouble   SLO
if LBTrouble then RED else NORMAL

Mstr_load   SLO
if !Mstr_load then RED else NORMAL

$   end of COMPUTED-STATUS section - start of EVENTS

# ****************************
# Events section
# ****************************


##############      EVENTS SECTION   ############################################

#  Event that enables changes in load
#  Set By:test procedure
  ChangeLoadEv

#  Event that sets load to zero (opens all relays)
#  Set By:test procedure
```

```
  ClearLoadEv

#  Event that freezes load relays at present load setting
#  Set By:test procedure
  FreezeLoadEv

#  Restores load value to frozen value
#  Set By:test procedure
  RestoreLoadEv

#  Resleases load control program
  LBC_RealRels

$   end of EVENTS section
```

Appendix G. Event Response File

```
#*****************************************************************************
#
#             er_specs.multi
#
#*****************************************************************************

@REG_NAME
   ER_MULTI

#*****************************************************************************
#
#  If the speed is in tolerance, then stop changing the load
#  switch settings and start using the AVR bias control
#
#*****************************************************************************

@INPUT_EVENT
   e_SpdInTL

@IF_TRUE
   Dyno_MD
  "EngCtrlMode == 5[none] || EngCtrlMode == 4[none]"
  "SPEED_TR >= 725[rpm]"

  # Set Dyno to open loop at present closed loop command so that it does not
  # hunt while it has no authority to change the load bank settings.
  # The dy command does not allow a variable to be used as a target value
  # so use svar to set the _OL value and the _MD

@PASS_SCRIPT
  svar Dyno_OL Dyno_CM
  reset_int_term LdbnkErr
  u_mode c LdbnkErr
  u_gains LdbnkErr -2 -.1 -.05

@PASS_PARAMETERS
  Dyno_MD   OFF
  LdbnkCtrlDesc 'Freeze'
```

```
@PASS_OUTPUT_EVENT
  FreezeLoadEv

@END


#*****************************************************************************
#
#  If the torque is in tolerance, then start using the AVR bias control
#
#  The dyno is using feedfowared only control so it should stay in closed loop
#
#  Adjust the speed check if the alternator will be used at idle speed
#
#*****************************************************************************

@INPUT_EVENT
  e_TrqInTL

@IF_TRUE
  "EngCtrlMode == 1[none] || EngCtrlMode == 2[none]"
  "SPEED_TR >= 725[rpm]"

@PASS_SCRIPT
  reset_dyno_int
  reset_int_term LdbnkErr
  u_mode C LdbnkErr
  u_gains LdbnkErr -.1 -.05 -0

@PASS_PARAMETERS
  LdbnkCtrlDesc 'Freeze'

@PASS_OUTPUT_EVENT
  FreezeLoadEv

@END


#*****************************************************************************
```

```
#
#  If the speed is out of tolerance, then start changing the load
#  switch settings and stop using the AVR bias control
#
#*******************************************************************************

@INPUT_EVENT
  e_SpdOutTL

@IF_TRUE
  "EngCtrlMode == 5[none] || EngCtrlMode == 4[none]"

@PASS_SCRIPT
  u_target o LdbnkErr 0
  u_mode o LdbnkErr

@PASS_PARAMETERS
  Dyno_MD   ON
  LdbnkCtrlDesc 'Loading'

@PASS_OUTPUT_EVENT
   ChangeLoadEv

@END


#*******************************************************************************
#
#  If the torque is out of tolerance, then stop using the AVR bias control
#
#*******************************************************************************

@INPUT_EVENT
  e_TrqOutTL

@IF_TRUE
  "EngCtrlMode == 1[none] || EngCtrlMode == 2[none]"

@PASS_SCRIPT
  u_target o LdbnkErr 0
```

```
  u_mode o LdbnkErr

@PASS_PARAMETERS
   Dyno_MD    ON
   LdbnkCtrlDesc 'Loading'

@PASS_OUTPUT_EVENT
   ChangeLoadEv

@END


#*****************************************************************************
#
#  Speed safety - if the engine has been running at high speed too long
#  then set the max speed back to 1800
#
#  Implement this logic according to the alternator manufacturer
#
#*****************************************************************************

@INPUT_EVENT
   e_SpeedOvr

@PARAMETERS
   max_spd 1800[rpm]
   max_dyno_spd 1800[rpm]

NOTIFY 'Above 1800rpm for 5 mins.  Now MUST stay below 1800 for mininum 1 min'

@OUTPUT_EVENT
   idle_mode

@END
```

```
#*****************************************************************************
#
#  Speed safety - if the engine has not been running at high speed for a
#  while then set the max speed 2150
#
#  Implement this logic according to the alternator manufacturer
#
#*****************************************************************************

@INPUT_EVENT
   e_SpeedUndr

@PARAMETERS
   max_spd 2150[rpm]
   max_dyno_spd 2150[rpm]

   NOTIFY 'OK to go back above 1800rpm'

@END


#*****************************************************************************
#
#  When Dyno_CM gets above 0, we need the load bank – implement in limit specs
#  Make sure status is ok and turn on the enables and set the
#  change load event.  If status not ok, set the idle mode event
#
#*****************************************************************************

@INPUT_EVENT
   e_DynoNeed

@IF_TRUE
   !LBTrouble

@PASS_SCRIPT
   reset_dyno_int 0
   reset_int_term LdbnkErr

@PASS_PARAMETERS
```

```
   UseLB ON

@PASS_PARAMETERS
  LdbnkCtrlDesc 'Loading'

@PASS_OUTPUT_EVENT
   ChangeLoadEv

@FAIL_OUTPUT_EVENT
   idle_mode

@END

#*****************************************************************************
#
#  If we're not asking for load then turn off the AVR control – implement in
#  limit specs
#
#*****************************************************************************

@INPUT_EVENT
   e_NoDyno

@SCRIPT
   u_target o LdbnkErr 0
   u_mode o LdbnkErr
   reset_int_term LdbnkErr

@END


#*****************************************************************************
#
#  Note string descriptions for load bank control
#
#*****************************************************************************

@INPUT_EVENT
   ChangeLoadEv
```

```
@IF_TRUE
  "EngCtrlMode == 5[none] || EngCtrlMode == 4[none]"

@PASS_SCRIPT
   svar Dyno_MD ON

@PARAMETERS
   LdbnkCtrlDesc 'Loading'

@SCRIPT
   u_target o LdbnkErr 0
   u_mode o LdbnkErr

@END


#*******************************************************************************

@INPUT_EVENT
   ClearLoadEv

@PARAMETERS
   LdbnkCtrlDesc 'Cleared/dumped'

@END


#*******************************************************************************

@INPUT_EVENT
   FreezeLoadEv

@PARAMETERS
   LdbnkCtrlDesc 'Freeze'

@END


#*******************************************************************************
```

```
@INPUT_EVENT
   RestoreLoadEv

@PARAMETERS
   LdbnkCtrlDesc 'Restore'

@END


#*************************************************************************
#
#  When the control mode changes, reset the integrators.
#
#*************************************************************************

@INPUT_EVENT
   EngCtrlModeT

@SCRIPT
   u_target o LdbnkErr 0
   u_mode o LdbnkErr
   reset_int_term LdbnkErr

@PASS_OUTPUT_EVENT
   ChangeLoadEv

@END


#*************************************************************************
#
#  When the emergency event is received, disable the load bank
#
#*************************************************************************

@INPUT_EVENT
   emergency

@PASS_PARAMETERS
  #label       value       delay
```

```
   UseLB       OFF         0[sec]

@END


#*****************************************************************************
#
#  When the abort event is received, disable the load bank
#
#*****************************************************************************

@INPUT_EVENT
   abort_limit

@PASS_PARAMETERS
  #label       value       delay
   UseLB       OFF         0[sec]
@END


#*****************************************************************************
#
#  When the off_push event is received, disable the load bank
#
#*****************************************************************************

@INPUT_EVENT
   off_push

@IF_FALSE
   key_switch

@PASS_PARAMETERS
  #label       value       delay
   UseLB       OFF         0[sec]

@END


#*****************************************************************************
```

```
#
#  When the cell_push event is received, disable the load bank
#
#  Implement this as desired
#
#*******************************************************************************

@INPUT_EVENT
   cell_sys_push

@IF_TRUE
   run_sns

@PASS_PARAMETERS
  #label       value       delay
   UseLB       OFF         0[sec]

@END


#*******************************************************************************
#
#  When the run_push event is received, enable the load bank
#
#*******************************************************************************

@INPUT_EVENT
   run_on

@PASS_PARAMETERS
  #label       value       delay
   UseLB       ON          0[sec]

@END
```

```
#*******************************************************************************
#
#  Initialize settings when the engine starts
#
#*******************************************************************************

@INPUT_EVENT
   spd_gt_400

@PASS_SCRIPT
  u_target o LdbnkErr 0
  u_mode o LdbnkErr
  /cyflex/bin/dy 0

@PASS_OUTPUT_EVENT
   ChangeLoadEv

@PASS_PARAMETERS
   UseLB    ON

@END


#*******************************************************************************
#
#     END OF FILE
#
#*******************************************************************************
```

Appendix H. Dyno Control Configuration

```
SPECS_VERSION
   Version_1

#                          Control Specifications
#*****************************************************************************
#        Default Engine Control Mode
#
#        Codes as defined in controls.h:
#
# *      1     Dyno control dyno torque; throttle control speed
#        2     Dyno control net torque; throttle control speed
#        3     Dyno control other channel; throttle control speed
# *      4     Dyno control speed; throttle control gross torque
#        5     Dyno control speed; throttle control net torque
#        6     Dyno control speed; throttle control other channel
#*****************************************************************************

   5    #     Dyno control speed; throttle control net torque
#   1    #     Dyno control dyno torque; throttle control speed

#*****************************************************************************
#Ctrl_var  label(12)       units (12)   fmt   hst    dsply   ramp_rate tolerance

   0      SPEED           rpm          1      1      1        10.0        10.0
   1      TORQUE          n_m          1      1      1        0.0         25.0

$END
#
#*****************************************************************************
#
#     User Control Variable Gains
#
#*****************************************************************************
#  End the list of control variable gains with $END
#
#  Ctrl_var       is the index into the CONTROL_GLOBAL array; user control
#                 loops start with ctrl_var = 3.
#
```

```
#  Proportional   the proportional gain
#
#  Integral       the integral gain
#
#  Derivative     the derivative gain
#*************************************************************************
# Ctrl_var          proportional    integral     derivative
#
$END

#*************************************************************************
#
#     Dyno Dual Loop Control - For water brake dynos with inlet and outlet
#     restrictions that need to be switched between as a function of an
#     independent variable.
#
#*************************************************************************

# Active    Threshold_value    Threshold_hysteresis    Switch_variable
   0         135[n_m]        27[n_m]               TORQUE

#*************************************************************************
#                                                                        #
#  ALTERNATOR CONTROLLER USE THIS SECTION FOR MULTI LOAD CONTROL
#                                                                        #
#*************************************************************************

#  Dyno controller is controller 0
#  ctrlr no.  label (12)  units (12)  intvl  fmt   hst   dsply ramp_rate
   0          Dyno        pct         FAS     2     1     1     0

#  Output channel
#  Output type  Channel    Bias(Zero)   Span   Filter Constant
   RV           dyno_cmd       0         1     0.0

#  Output Term Bounds
#  Upper bound is very high so we can get higher output at
#  low AVR voltage settings
#  Lower Bound  Upper Bound
   0            100
```

```
#  Label for integral lower bound, $null means none
   $null

#  Label for integral upper bound, $null means none
   $null

#  Hysteresis
   0

#  Command Map File
   $null

# Label for gain scaling, $null means none
   $null

#  Feed Forward
#  Label         Active       Gain
   DynoFdFrwrd   1            1.0

# Local Command Input channel label
   $null

# Remote Sense channel label
   $null

#************ Dyno PID Gains Section ****************************************
#
#  Dyno controlling Speed - adjust as necessary
#  Prop       Integ     Deriv
   -0.25      -0.25     0.0

#  When dyno controlling torque, all done with feedforward and
#  voltage biasing control loop

#  Dyno controlling Dyno torque
#  Prop       Integ     Deriv
   0.0        0.0       0

#  Dyno controlling Net torque
```

```
#  Prop       Integ     Deriv
   0.0        0.0       0

#  Dyno controlling Other channel
#  Prop       Integ     Deriv
   0          0         0
#
#************ end of Dyno Controller ****************************************
```

Appendix I. Voltage Control Configuration

```
#*************************************************************************
#
#  This user loop is for AVR bias control
#
#*************************************************************************

@USER_LOOP_CTRLER
#  label        command       interval        open-loop
#  root         control units                 ramp rate
   LdbnkErr     pct            FAS            10

@CTRLER_FEEDBACK_VRBL
#  label      closed-loop      error-term
#             ramp rate        tolerance
   LdbnkErr     0.1              .1

@CTRLER_INTEGRAL_BOUND_LABELS
#  <lower_bound_label>     <upper_bound_label>
   LdbnkErrIntLB           LdbnkErrIntUB

@CTRLER_OUTPUT_BOUNDS
#  <lower_bound>     <upper_bound>
   LdbnkErrOutLB     LdbnkErrOutUB

@CTRLER_GAINS
#  proportional   integral   derivative
    -2.0          -0.1      -0.05

@CTRLER_OUTPUT_CHAN
#  AO  <channel>   <zero_value>   <span_value>    <filter_constant>
   AO      22         -50            50             0.0

@INITIAL_CTRLER_STATE
#  initial mode    initial state
   OPEN_LOOP       0[none]

$END
```