### Finite State Machine

#### ACC Single Mode Booster

* 1. There shall be a function provided to control the Booster’s Single Mode (SM) laser’s forward current when in ACC mode, based on a pre-determined setpoint value.
  2. Null values as parameters shall not be allowed.
  3. If the current is below the setpoint, then it needs to be incremented to get to this value. This should happen in stages of a pre-determined size. The number of smaller incrementations (steps) needed should also be pre-determined.
  4. If the current is above the setpoint, then it needs to be decremented to get to this value. This should happen in stages of a pre-determined size. The number of smaller decrements needed should also be pre-determined.
  5. If the number of steps needed (detailed in Requirement 1.3) is zero, then the value of the current shall not be altered.
  6. If the number of steps is greater than zero, and the current needs to be increased, then the setpoint value shall be compared to the current value to determine how to proceed. If the difference between the setpoint and the current value is less than the amount by which the current is increased at each step, then the new value for the current shall be set to the setpoint.
  7. If the number of steps is greater than zero, and the current needs to be increased, then the setpoint value shall be compared to the current value to determine how to proceed. If the difference between the setpoint and the current value is greater than or equal to the amount by which the current is increased at each step, then the current shall be incremented by this value.
  8. If the number of steps is greater than zero, and the current needs to be decreased, then the setpoint value shall be compared to the current value to determine how to proceed. If the difference between the current value and the setpoint is less than the amount by which the current is decreased at each step, then the current shall be set to the setpoint.
  9. If the number of steps is greater than zero, and the current needs to be decreased, then the setpoint value shall be compared to the current value to determine how to proceed. If the difference between the current value and the setpoint is greater than the amount by which the current is decreased at each step, then the current shall be decreased by this value.
  10. At the end of the comparisons and assignments detailed above, the number of steps is decremented, indicating that another increment/decrement stage has ended. If the number of steps is now zero, the current shall be set to the setpoint value, regardless of what value it currently holds.
  11. The Booster SM1 shall be updated with the newly determined value for the current.

#### Trajectory Calculator

* 1. When a laser’s current needs to be increased or decreased to reach a setpoint, this needs to happen in stages. A pre-determined measure of current marks an upper bound on the amount that can be added or deducted per stage. There shall be a function to determine the trajectory (whether the current needs to increase or decrease, and how many stages are necessary) for this current to reach the setpoint value, based on the current’s value, the setpoint, and the amount by which the current can increase or decrease in a single step.
  2. Null pointers as parameters shall not be allowed.
  3. If the current is below the setpoint, then a flag to indicate that the current needs to increase shall be set to true. The number of steps needed shall be calculated from the difference between the current and the setpoint, and the maximum amount of current that can be added per step. It is presumed that, at each step, the maximum amount of current will be added (if necessary). If, at any point, only a fraction of this maximum amount of current is needed to reach the setpoint, then this fraction will be added in a single (final) step.
  4. If the current is greater than or equal to the setpoint, then a flag to indicate that the current needs to increase shall be set to false. The number of steps needed shall be calculated from the difference between the current and the setpoint, and the maximum amount of current that can be subtracted per step. It is presumed that, at each step, the maximum amount of current will be subtracted (if necessary). If, at any point, only a fraction of this maximum amount of current is needed to reach the setpoint, then this fraction will be subtracted in a single (final) step.

#### Shifting the Current Towards the Setpoint

* 1. There shall be a function provided to shift the current towards the setpoint, one step at a time; a single call to this function should increase or decrease the current by a single step. Refer to *Section 2: Trajectory Calculator* (specifically *Requirement 2.1*) for an explanation of how the current is moved towards the setpoint in steps.
  2. This function shall be only ever be called after the function detailed in *Section 2: Trajectory Calculator* has been executed, and information regarding the current trajectory has been retrieved: whether the current must increase or decrease, and the number of steps needed.
  3. Null pointers as parameters shall not be allowed.
  4. If the current must increase, then the current and the setpoint must be compared. At each stage, as outlined in *Section 2*, there is an upper limit on the amount that the current can be increased per stage. If the difference between the current and the setpoint is greater than or equal to this upper limit, then the current shall be increased by the upper limit current value.
  5. If the current must increase, then the current and the setpoint must be compared. At each stage, as outlined in *Section 2*, there is an upper limit on the amount that the current can be increased per stage. If the difference between the current and the setpoint is less than this upper limit, then the current shall be set to the setpoint.
  6. If the current must decrease, then the current and the setpoint must be compared. At each stage, as outlined in *Section 2*, there is an upper limit on the amount by which the current can be decreased per stage. If the difference between the current and the setpoint is greater than or equal to this upper limit, then the current shall be decreased by the upper limit current value.
  7. If the current must decrease, then the current and the setpoint must be compared. At each stage, as outlined in *Section 2*, there is an upper limit on the amount by which the current can be decreased per stage. If the difference between the current and the setpoint is less than this upper limit, then the current shall be set to the setpoint.

#### Booster APC State Machine

* Booster APC Finite State Machine (FSM) States:

1. HpaApcInit: Initialization stage for the SM pump
2. HpaApcSm: The SM current is building towards a minimum threshold current
3. HpaApcMM1\_Init: Initialization stage for the MM1 pump
4. HpaApcMM1\_Control: The PID loop for the MM1 pump is run to obtain a setpoint current
5. HpaApcMM2\_Init: Initialization stage for the MM2 pump
6. HpaApcMM2\_Control: The PID loop for the MM2 pump is run to obtain a setpoint current
7. HpaApcMM1andMM2\_Init: Initialization stage for the MM1 and 2 pumps
8. HpaApcMM1andMM2\_Control: The PID loop for the MM1 and 2 pump is run to obtain a setpoint current

#### Initializing the Booster APC Finite State Machine

* 1. There shall be a function provided to initialize the Booster APC State Machine
  2. This function shall set the initial Finite State Machine (FSM) state to ‘HpaApcInit’.

#### Updating the Booster APC Finite State Machine

* 1. There shall be a function provided to update the state of the Booster APC FSM, depending on the current state, and whether or not the SM, MM1 and MM2 pumps are enabled or disabled.
  2. Externally implemented functions shall be called to collect information on the enable/disable status of each pump.
  3. If the enable/disable status of any pump has changed, the state should be reset to ‘HpaApcInit’; a flag to indicate the most recent enable/disable state should be compared to the value of each at the onset of this function’s execution. In other words, if any pump was most recently disabled, but has since been enabled, or vice versa, the FSM shall be reset.
  4. An externally implemented function shall be called to retrieve the value of the setpoint. There shall also be a flag to indicate the most recent value of this setpoint. If the newly retrieved value does not equal the most recent value, then each of the three PID loops shall be reset (MM1, MM2, MM1 and MM2 together).
  5. The PID loop resetting, mentioned in *Requirement 5.3* shall be executed by an externally implemented function.
  6. This function shall call a state-specific function for the FSM, depending on what the current state is (which may or may not have been reset, as detailed in requirements 5.1 and 5.2).
  7. Flags to store the most recent enable/disable status of the pumps shall be set to their new values, and the new setpoint shall also be stored in an appropriate variable. A flag to indicate whether the current needs to increase or decrease to achieve the required value shall also be set.

#### Booster APC Initialization State

* 1. There shall be a function provided to enact the necessary sequence of events for the Booster APC Initialization state.
  2. Externally implemented functions shall be called to collect information on the enable/disable status of each pump.
  3. If the value of the setpoint (retrieved in the function detailed in *Section 5: Updating the Booster APC Finite State Machine*) is greater than or equal to a pre-determined minimum threshold for the setpoint, and the SM pump is enabled, then the FSM shall progress to state ‘HpaApcSm’.
  4. If the value of the setpoint is less than the pre-determined minimum threshold mentioned in *Requirement 6.1*, and/or the SM pump is disabled, then the FSM shall remain in stage ‘HpaApcInit’.

#### Booster APC SM State

* 1. There shall be a function provided to enact the necessary sequence of events for the Booster APC SM state.
  2. Externally implemented functions shall be called to collect information on the enable/disable status of each pump.
  3. If the SM pump is disabled, then the current shall be lowered, in portions of a pre-determined size, to a threshold. This threshold represents the SM current low enough to shutdown the diode.
  4. If the SM pump is disabled and the SM current is below the threshold detailed in *Requirement 7.2*, then the SM current shall be set to zero, and the FSM state shall be reset to ‘HpaApcInit’.
  5. If the SM pump is disabled and the SM current is not below the threshold detailed in *Requirement 7.2*, then the appropriate portion, also mentioned in *Requirement 7.2* shall be subtracted from the current, and the FSM shall remain in state ‘HpaApcSm’.
  6. If the SM pump is enabled, then the SM current shall be increased in portions of a pre-determined size, up to a nominal threshold value.
  7. If the SM pump is enabled, and the SM current is above the nominal threshold value mentioned in *Requirement 7.5*, then if the MM1 pump is enabled, the FSM shall transition to state ‘HpaApcMM1\_Init’. The SM current shall be set to the nominal threshold current.
  8. If the SM pump is enabled, and the SM current is above the nominal threshold value mentioned in *Requirement 7.5*, then if the MM1 pump is disabled and the MM2 pump is enabled, the FSM shall transition to state ‘HpaApcMM2\_Init’. The SM current shall be set to the nominal threshold current.
  9. If the SM pump is enabled, and the SM current is above the nominal threshold value mentioned in *Requirement 7.5*, then if both the MM1 and MM2 pumps are disabled, the FSM shall transition to state ‘eHpaApcInit’. The SM current shall be set to the nominal threshold current.
  10. An externally implemented function shall be called to update the new SM current value.

#### Booster APC MM1 Initialization State

* 1. There shall be a function provided to enact the necessary sequence of events for the Booster APC MM1 Initialization state.
  2. Externally implemented functions shall be called to collect information on the enable/disable status of each pump.
  3. If the SM and/or the MM1 pump(s) is/are disabled, then the MM1 current shall be incrementally decreased to a minimum value, representing a low enough value to shutdown the diode. In a single call to this function, in this situation, the current shall be decreased by a pre-determined step size.
  4. If the SM and/or the MM1 pump(s) is/are disabled, and, after the current has been decreased, it is below the necessary threshold, then the FSM shall transition to state ‘eHpaApcSm’. An externally implemented function shall be called to update the new MM1 current value.
  5. If the SM and/or the MM1 pump(s) is/are disabled, and, after the current has been decreased, it is not below the necessary threshold, then the FSM shall remain in state ‘HpaApcMM1\_Init’. An externally implemented function shall be called to update the new MM1 current value.
  6. If both the SM and MM1 pumps are enabled, then the FSM shall transition to state HpaApcMM1\_Control, and the MM1 PID loop shall be reset, through an externally implemented function.

#### Booster APC MM1 Control State

* 1. There shall be a function provided to enact the necessary sequence of events for the Booster APC MM1 Control state.
  2. Externally implemented functions shall be called to collect information on the enable/disable status of each pump.
  3. If either the MM1 or the SM pumps are disabled, the FSM shall transition to state ‘HpaApcMM1\_Init’.
  4. If the MM1, SM and MM2 pumps are enabled, and the MM1 current is greater than a pre-determined upper threshold, then the MM1 current shall be updated to this threshold value (by an externally implemented function), and the FSM shall transition to state ‘HpaApcMM2\_Init’.
  5. If the MM2 pump is disabled, and/or the MM1’s current is below the threshold mentioned in *Requirement 9.4*, then the MM1 PID control loop shall be run, with an externally implemented function.

#### Booster APC MM2 Initialization State

* 1. There shall be a function provided to enact the necessary sequence of events for the Booster APC MM2 Initialization stage.
  2. Externally implemented functions shall be called to collect information on the enable/disable status of each pump.
  3. If the SM or the MM2 pumps are disabled, then the MM2 current should be decreased in steps of a pre-determined size, lowered to a value that is low enough for the diode to be shutdown
  4. In the case that the SM or MM2 pumps are disabled, and the value of the MM2 current (after being decreased by the pre-determined amount mentioned in *Requirement 10.*3) is low enough to shut down the diode, then this current shall be set to zero (by an externally implemented function), and the FSM shall transition to state ‘HpaApcMM1\_Control’.
  5. In the case that the SM or MM2 pumps are disabled, and the value of the MM2 current (after being decreased by the pre-determined amount mentioned in *Requirement 10.*3) is not low enough to shut down the diode, then the FSM shall remain in state ‘HpaApcMM1\_Init’.

#### Booster APC MM2 Control State

* 1. There shall be a function provided to enact the necessary sequence of events for the Booster APC MM2 Control stage.
  2. Externally implemented functions shall be called to collect information on the enable/disable status of each pump.
  3. If either the MM2 or SM1 pumps are disabled, the FSM shall transition to stage ‘HpaApcMM2\_Init’.
  4. If the flag, mentioned in *Requirement 5.7*, that indicates that the current needs to increase to achieved the setpoint value is set to true, and the MM2 current is greater than a pre-determined value (a value that indicates that the FSM can progress to the PID loop for MM1 and MM2 together), and if MM1 is enabled, then the FSM shall transition to state ‘HpaApcMM1MM2\_Init’. The MM2 current shall be updated to the pre-determined value just mentioned, by an externally implemented function. The PID loop for MM1 and MM2 shall be reset, again, by an externally implemented function.
  5. If the flag, mentioned in *Requirement 5.7*, that indicates that the current needs to increase to achieved the setpoint value is set to false, MM1 is enabled and the MM2 current is less than 100, then the FSM shall transition to stage ‘HpaApcMM1\_Control’. The MM2 current shall be set to zero by an externally implemented function, and the MM1 PID loop shall be reset by an externally implemented function.
  6. In all other scenarios (not covered by *Requirements 11.3 – 11.5*), the MM2 PID control loop shall be run.

#### Booster APC MM1 and MM2 Initialization State

* 1. There shall be a function provided to enact the necessary sequence of events for the Booster APC MM1 and MM2 Initialization stage.
  2. Externally implemented functions shall be called to collect information on the enable/disable status of each pump, and, in case a pump has been disabled, the minimum value of current required to pass to a previous stage.
  3. If either of the SM or the MM2 pumps have been disabled, then the MM1 and MM2 currents need to be decreased in stages of a pre-determined size, to a minimum value at which the FSM can progress to the ‘HpaApcMM2\_Control’ stage.
  4. If either of the SM or the MM2 pumps have been disabled, and the MM1 and MM2 currents are less than the pre-determined minimum value (mentioned in *Requirement 12.3)*, then the FSM shall transition to stage ‘HpaApcMM2\_Control’, and, by externally implemented functions, the MM1 and MM2 currents shall be separately set to the pre-determined minimum value.
  5. If either of the SM or the MM2 pumps have been disabled, and the MM1 and MM2 currents are not less than the pre-determined minimum value (mentioned in *Requirement 12.3)*, then the FSM shall remain in stage ‘HpaApcMM1MM2\_Init’.
  6. If only the MM1 pump has been disabled, then the MM1 and MM2 currents need to be decreased in stages of a pre-determined size, to a minimum value at which the FSM can progress to the ‘HpaApcMM1\_Control’ stage.
  7. If only the MM1 pump has been disabled, and the MM1 and MM2 currents are less than the minimum value (mentioned in *Requirement 12.6*), then the FSM shall transition to stage ‘HpaApcMM1\_Control’, and by externally implemented functions, the MM1 and MM2 currents shall be separately set to the minimum value.
  8. If only the MM1 pump has been disabled, and the MM1 and MM2 currents are not less than the minimum value (mentioned in *Requirement 12.6)*, then the FSM shall remain in stage ‘HpaApcMM1MM2\_Init’.
  9. In all other cases (not covered by *Requirements 12.3-12.8*, i.e., no pumps are disabled), the MM1 MM2 PID loop shall be reset, and the FSM shall transition to stage ‘HpaApcMM1MM2\_Control’.

#### Booster APC MM1 and MM2 Control State

* 1. There shall be a function provided to enact the necessary sequence of events for the Booster APC MM1 and MM2 Control stage.
  2. Externally implemented functions shall be called to collect information on the enable/disable status of each pump.
  3. If any of the SM, MM1 and MM2 pumps are disabled, the FSM shall transition to stage ‘HpaApcMM1\_Init’.
  4. If the flag, mentioned in *Requirement 5.7*, that indicates that the current needs to increase to achieved the setpoint value is set to false, then externally implemented functions shall be called to retrieve the setpoint, and the minimum current values that need to be achieved for the MM1 and MM2, before they can transition to previous stages, where further reductions in current will occur.
  5. If the flag, mentioned in *Requirement 5.7*, that indicates that the current needs to increase to achieved the setpoint value is set to false, and the MM2 current (should be equal to the MM1 current) is less than the minimum current value, mentioned in *Requirement 13.4*, then the MM1 and MM2 PID loop shall be reset. Externally implemented functions should be called to set the values of the MM1 and MM2 currents to this minimum current value.
  6. If the flag, mentioned in *Requirement 5.7*, that indicates that the current needs to increase to achieved the setpoint value is set to true, or it is set to false, but the value of the MM1 and MM2 currents are greater than the minimum value from *Requirement 13.4*, the MM1 and MM2 PID loops shall be run.