Things that need to be fixed after initial M2 testing in March 2020:
[This is compiled out of my notes during testing and data analysis, after taking into account items on the confluence page which were contributed by various people]

1. Implement detailedStates: open loop; closed loop; diagnostic? Manual standby?
2. Find hard-coded inPosition thresholds, refactor them into configuration files. Make sure this is defined using hardpoint encoders.
3. Find hiTemperatureWarning thresholds, refactor them into configuration files. Formula:
    a. D1 = (Exhaust1 - intake1)
    b. D2 = (exhaust2 - intake2)
    c. D= (D1+D2)/2
4. Add telemetry topics relevant to the internal working mechanisms of the M2 cell, for example, total motor voltage and currents - since they are in Harris binary files already.
    a. Need to be added into the current LTS-162 revision.
5. Add a GUI command to turn on and off the force balance system.
6. Add a GUI command that does the same thing as SAL command resetForceOffsets{}.
7. Add a section to the M2 EUI on the status of the interlock.
8. Display the Fx, Fy, Fz, Mx, My, and Mz that are imposed by the force balance system.
    a. This requires the FB on/off switch to be implemented first - not [applied forces - measured forces], but appliedBalanceForces from the outer control loop.
    b. Need to be added to the GUI and SAL
    c. Not in the current LTS-162, because FB on/off switch requires significant work and was requested by Doug after TCB.
    d. we should still keep the netForcesTotal and netMomentsTotal topics in the currently proposed LTS-162)
9. Clean up GUI display of the forces, to match the categorization of the forces in the SAL telemetry: total forces, measured forces, LUT_gravity forces, LUT_thermal forces, user_applied forces (including AOS forces), balance forces.
10. Clean up the zenith angle inconsistency.
    a. GUI should only display two things: (1) zenith angle from MTMount telemetry (2) zenith angle from M2 inclinometer (processed value); Bo will provide formula for transformation.
    b. Unit should be in degrees everywhere.
    c. Telemetry zenith angle should be consistent with this too.
    d. We will leave the LUT angle alone, and provide a conversion between the two. Done.
    e. Is LUT angle calculated using M2 inclinometer reading or MTMount telemetry?
    f. The system should use its own inclinometer reading whenever available, and fall to MTMount telemetry if its own inclinometer reading is not present.
    g. Users should be able to configure through the EUI which inclinometer to use for the LUT.
11. Need to look at the code and understand M2 actuator control during slewing? Does M2 reply on the FB system for slewing?

12. Under utility view, it provides the values for the individual displacement sensors. It needs to also provide the net values for displacement and rotations. These all need to go into the telemetry.
    a. Bo to clarify how to calculate these values. Done.
    b. Need to be able to reconfigure the offset of individual displacement sensors.
    c. Not in the current LTS-162 revision (overlooked, need to add it)
    d. Need to display the 12 displacement sensor readings together with x/y/z/Rx/Ry/Rz calculated from these sensors.
13. Check whether the currently displayed mirror position is determined from the hard points. If not, then we need to make it right. Display the position determined from hard points and the results from the displacement sensors side by side and make sure they are consistent.
    a. Do we need to define a threshold for the consistency? No.
14. On GUI where we display the hardpoint ID numbers, we need to display B6, B16, B26, A2, A4, A6, instead of (6,16,26,72,74,76)
15. The limit switch panel seemed to never light up even when a limit switch error was triggered. Check this. And make sure this works as expected. And indicate on GUI whether these are software limits (orange) or hardware limits (red).
16. Enable users to reconfigure the limit switches. No.
    a. Do we need the software limits as well? No. only locate them, no need to reconfigure these.
17. Clarify the physical locations of the temperature sensors (with testing). Move the temperature inversion matrix to configuration file, and switch columns as needed to match the ordering the temperature sensors in the GUI and telemetry.
    a. Need to be able to set the zero points (offset) for these temperature sensors.
18. M2 XML should use TAI instead of UTC.
19. Enable disabling use of thermal data in LUT (GUI only)
20. Enable re-configuring the mirror reference position.
21. It would be nice to have VMS in place for the next round of testing. - Andy: most likely this can be ready in time.
22. EUI graphical force distribution has 7 rings? Need to investigate.
23. The following telemetry topics seemed to have fields with corrupt parameter values. Sometimes they did not match GUI and binary files, other times they did. These need to be investigated (Bo+Te-Wei).
    a. closed loop mode, when we change force on an actuator, encoder reading changes on GUI, but not EFD
    b. Open loop mode, when we change an actuator position, steps change on the GUI, but do not change in the EFD. Note that the binary file only has “step_cmd” at 20Hz, but we can accumulate it and get steps.
24. When there are NaNs in a topic, the entire topic gets dropped by influxDB. - Angelo. https://jira.lsstcorp.org/browse/DM-21334
25. EFD time index needs to be private_sndStamp, instead of insertion time. Angelo. Done

Following items are from confluence page

https://confluence.lsstcorp.org/pages/viewpage.action?spaceKey=LTS&title=M2+Wish+List
Followed by my current thinking on each of them.
Blue responses mean it has been adopted in the master list above.
Red responses mean I am going to argue against implementing it.

The changes below were taken from Chris Contaxis's notes on the CR Harris requested on LTS-162

1. 2.2 - M2 Assembly telemetry time-stamp - TCS-M2-CMD-ICD-0018
    a. Harris will not comply, need a waiver, this was in version 2.
    My recommendation is for LSST to add a timestamp field to each telemetry and event topic and then modify the software to populate those fields after receipt of M2 code. I would estimate 80 hours of development effort and 60 hours of testing effort.

    Already done (?) under uniform TSSW framework?

2. 4.1.1 - ApplyBendingModes {} command - TCS-M2-CMD-ICD-0006
    a. Harris will not comply, need a waiver, this was in version 2.
    My recommendation is for LSST to add support for bending modes to the software after receipt of M2 code. I would estimate 80 hours of development effort and 60 hours of testing effort.

    removed

3. 4.1.3 - ResetOffsets {} command - TCS-M2-CMD-ICD-0008
    a. Need exception, Harris will allow reset offsets to handle force but since they won't implement bending modes they will not meet this completely.
    My recommendation is for LSST to add support for removing bending mode offsets after receipt of M2 code. I would estimate 20 hours of development effort and 15 hours of testing effort.

    Not needed. We will not apply bending mode via commands. We only apply forces

4. 5.2 - M2Assembly detailed state - TCS-M2-CMD-ICD-0028
    a. Need waiver, Harris will not implement.
    My recommendation is to determine what detailed states Harris implements and add the reporting of those detailed states after receipt of M2 code. I would estimate 40 hours of development effort and 30 hours of testing effort.

    Yes, this is in our list as well.

5. 5.3 - M2Assembly inPosition - TCS-M2-CMD-ICD-0029
    a. Need exception for configurable thresholds. Harris will implement with hard coded thresholds.

My recommendation is to spend time finding non-configurable thresholds and changing to configurable thresholds after receipt of M2 code. Hard coded thresholds will cause problems during integration and commissioning. I would estimate 160 hours to development effort to fix ALL hard coded thresholds and 120 hours of testing effort.

Added to our list. Subject to Te-Wei's agreement.

6. 5.4 - CellTemperatureHiWarning - TCS-M2-CMD-ICD-0029
    a. Need exception for configurable threshold. Harris will implement with hard coded value. 5.3 recommendation covers this.

    Added to our list. Subject to Te-Wei's agreement.

7. 8 - Telemetry Publication - TCS-M2-CMD-ICD-0034
    a. Need waiver, Harris will only publish telemetry listed here. Harris will not add any telemetry that provides insight into their design.
    My recommendation would be to create the topics to report all telemetry the M2 code has available internal to itself so it can be published on SAL. This would occur after receipt of M2 code. It is difficult to estimate amount of effort due to the lack of documentation on Harris' part, to be safe I will estimate 160 hours of development effort and 120 hours of testing effort.

    Added to our list. Subject to Te-Wei's agreement.

Comments from Doug Neill and Te-Wei 2020 01 31

The Gui for the M1M3 should be used as a general reference for the format for any modifications to the M2 GUI. This will reduce confusion. For example, the Force balance system should use the same annotation as the M1M3. In the M2 GUI it is "HC".

That would be nice. Actually, it is on Bo's wish list that the M2 XML should be like M1M3 as much as possible. Bo understands that this means quite a bit of work, especially if we want to make the GUI alike. It should be relatively easier to do this for the XML.

Te-Wei argues that this is not practical, as it requires a lot of coordination between developers(?).

From the GUI, you should be able to input command forces (force offsets) to individual (axial) actuators. The control system does not allow for sending individual commands to the individual actuators. However, this function can be added by combining the LUT values with the force offset values before applying the force displacement matrix conversion.

Actually the GUI allows the user to input force offsets.

If we only need to apply offset to one actuator, this can be done by keying in the number then click the button to apply; If we need to apply offsets to multiple actuators at a time, then we will need to create csv files, then convert into scp files, then load them using the GUI.

In the GUI you should be able to choose which three tangent links are active (force control) and which three and passive (position defining). This way, if a tangent link fails, it can become a passive link and we can continue operating. This should be in the tab of the rigid body position.

In the GUI you should be able to choose which three axial actuator are passive (position defining). This way, if a actuator fails, it can become a passive actutor and we can continue operating. This should be in the tab of the rigid body position.

This probably won't be done often because we imagine this needs to be done only when an actuator or tangent link fails. If it needs to be done on rare occasions, then we don't have to put this in the GUI.

If we do put this in the GUI, it is a bit complicated, because when the hardpoints changes, almost all the configuration files need to be changed, including the influence matrix, decoupling matrix, and the axial and tangent hardpoint matrices and their feed forward matrices. It will be tricky to design the GUI in a way that when a button is clicked, it automatically re-calculates all the matrices and regenrated all the configration files then start using those (in the minimum, a system restart will be needed to reload the configuration files).

Also, note that Harris provided code that helps pick alternative actuators as hardpoints when the current hardpoints fail. So we shouldn't allow the user to simply change it on the GUI.

In the GUI it should be clear which three tangent links and which tree axial actuator are passive and used to define the location. This should also show the loads on three items, and the forces and moments compensated by the force balance system. See M1M3 Gui

The way the force balance system works for the axial actuators, is that the forces on the passive actuator are measured and compared to the command values. The difference between these values is removed by the force balance system. The resulting net forces and moments, from the difference ,are determine and the opposite forces and moments are applied as distributed value across the rest of the actuator.

The current GUI does show the ID numbers of the 6 hardpoints - 3 axial and 3 tangent.

However it doesn't show Fx, Fy, Fz, Mx, My, and Mz that are commanded by the force balance system. As a compromise, we've decided to add the current total Fx, Fy, Fz, Mx, My, and Mz as commanded by all the actuators. These would be useful in determining whether the FB system is working properly, but the precision will be 72/3 times worse.

For Force in the "detailed force page", it needs to be clear what is the input force (LUT), total command force, measured force, force balance force and offset force. F_CUR is defined as the

measured force at the load cell. This does not appear to be correct. The load cell value should be very close to the command value. The force from the LUT appears to be F_cur. However this is unclear. This value has the force balance force (F_Hc) added to it to produce what is called the F_CMD. The actual force read from the load cell does not appear to be provided. This gui is very unclear. The forces shold be arranged as

1 Total Command Value which is the sum of the rest of the value

2) measured value from load cell after applying calibration value

3) Look up table value which are mostly based on gravity correction

4) force balance value used in the hard point force correction

5) offset values. Note: There are actually two sets of offset value. There are the external value provided by the AOS and GUI applied values to be added with a software update. The best way to deal with this is not clear at this time. Perhaps the GUI should just state which set is being used.

See list at the top. Yes, these are very confusing, and we need to clean it up. The understanding described above doesn't appear to match exactly what we observed during the tests on the summit:

F_cmd = total commanded forces

F_cur = currently measured forces

F_delta = forces applied by the user (on top of the LUT)

F_HP = seems to be the balance forces, but values do not really make sense.

F_error = not sure what this is exactly.

We will make this right.

For the Actuator Control, actuator selection, the tangent links (A-Ring in GUI) should be in a separate window. We never refer to tangent links as actuators to reduce confusion.

Bo thinks this is a relatively cosmetic thing. We should probably leave it the way it is, considering all the other work Te-Wei has to do. And, tangent links are referred to as tangent actuators at a lot of places (in XML for example), so I assume it is the same in the code.

Under cell status, it provides the elevation angle both from the inclinometer and from the TMA. It should be clear which one the system is using. You need to be able to tell it which one to use. It is not clear how you do that.

See list at the top.

Under utility view, it provides the values for the individual sensors. It needs to also provide the net values for displacement and rotations.

This is in our list too.

For the rigid body motion, it provides the "current position". It is unclear if this is the position as determined by the 6 passive support (3 axial and 3 tangential) or by the 12 displacement sensors. The position should be provide by the 6 passive support. The position from the displacement sensors should be provided for verification.

This is in our list too.

Detailed forces: The "HardPoint" provides number for the passive supports. Everywhere else the nomenclature uses a combination number and letter (B6. C14, D9 etc). These values should use the same nomenclature and they need to be changeable.

This is doable. See list at the top.

As discussed previously we need to be able to choose which axial actuators and tangent links are used as the passive supports to define the mirrors location relative to its cell.

See discussions above.

The Alarms/warnings: The Limit switch status, does not specify if these are the software limits or the hardware (electrical) limits. Both should be shown and it should be clear which is which.

See list at the top.

Of interest on LTS-162:

4.2.1. - MoveAxialActuator {} command - TCS-M2-CMD-ICD-0026

Specification: The MoveAxialActuator command shall cause the M2 Assembly to move a particular axial actuator the specified number of steps.

Parameters:

- axialActuatorID (type=U16)
- relativeStepsToMove: (type = I64). Increasing steps correspond to counterclockwise rotation of the motor shaft, looking along the motor shaft in the direction of the motor.

Discussion: The MoveAxialActuator command shall move the particular axial actuator the specified number of steps. This command will only be processed when the M2 assembly is in the enabled state and under open loop control

According to current Harris software design, SAL commands only apply to closed loop mode. In closed loop mode, only force commands are allowed. We feel that it may be too much work to change this basic design.
Therefore we are proposing to leave this out in the current change request.