Protection

Functions in "Protection.h" are to protect tip under 2 conditions: Ramp and Scan.

*protectRamp*

This function protect tip while ramping. Return if protection is triggered. The core function is *zAutoT*. It automatically adjusts Zoffset to make sure Zfeedback reached a target value. In *protectRamp*, it’s used to adjust Zoffset when Zfeedback is out of limit. The flag of this protection : Ramp protection is enabled && Feedback is on && Retract is off. With this protection, if Zfeedback is out of limit and the feedback loop is fully functional, it’ll be re-target to zero by adjusting Zoffset.

*protectScan*

This function protect tip while scanning. Return if protection is triggered. The goal of is *protectScan* is same with *protectRamp*, which is re-target Zfeedback to make sure it’s in range by adjusting Zoffset. The difference is that 3 specific protection mode is offered.

|  |  |
| --- | --- |
| Option 1 | Stop Scan |
| Option 2 | Auto 0 and continue |
| Option 3 | Auto to previous midpoint and continue |

Option 1 stops scan directly; option 2 re-targets Zfeedback to zero and continue scan; option 3 re-targets Zfeedback to the median point of the last scan line. *protectScan* has the same flag with *protectRamp*.

*protectTip*

This function protect sharp tip during scan, requires knowing the scan area. When we start scan, we need to move the tip to one of the corner of the scan area. In order to protect tip through this process, we will retract Zoffset fine a little bit, and then ramp it back to original value. Before we send the tip, we call *tipProtect(tip\_protect\_data, False)* to trigger tip protection; after returning the tip to origin position, we call *tipProtect(tip\_protect\_data, True)* to unprotect tip.