TCS software for SONG telescope

Hai Wang , Changzhi Ren
Nanjing Institute of Astronomical Optics & Technology
188 Bancang Street，210042, Nanjing，China

Abstract

Stellar Observations Network Group, SONG, is a Danish led international collaboration project to construct a global network of small 1m telescope around the globe. The second 1 meter SONG node telescope designed by NIAOT is installed at Delingha site in west China.
TCS hardware is based on PC, UMAC, tape encoder, motor and driver. TCS software is developed in powerful Qt Creator environment under stable Debian 6.0 operation system. The design rules are modularity and simplification. Several software modules work together to realize telescope usual function . Tracking algorithm is comprised of two parts. One is UMAC motion program, another is tracking thread in PC program. Communication between TCS and OCS is complicated. The method to process remote command is described.

Keyword: TCS, Software, SONG Telescope, Control, Tracking, UMAC

1 INTRODUCTION

Stellar Observations Network Group, SONG, is a Danish led international collaboration project to construct a global network of small 1m telescope around the globe. SONG aims to stellar physics, asteroseismology, extra-solar planets, transient phenomena and much more. Currently, the Danish prototype SONG node is now just about to enter the operation mode at Observatorio del Teide. The second 1 meter SONG node telescope designed by NIAOT is installed at Delingha site in west China and now just in field tuning stage.
The SONG telescope control system also includes Observatory Control System (OCS), Telescope Control System (TCS), Active Optic Control System (AOCS), Instrument Control System (ICS), and Data Handling System (DHS), connected together via Local Area Networks. The TCS is responsible for target trajectories calculation, pointing, tracking, coordination of observations, status monitoring and safety protection.

2 HARDWARE SYSTEM

The main physical devices controlled by TCS are azimuth, altitude, M3 mirror, view field rotator and atmospheric dispersion corrector(ADC). About 8 motors are controlled by same motion controller and one PC. Figure 1 displays several key parts of hardware system for only one axis. The motion controller is Universal Motion and Automation Controller (UMAC) from Delta Tau Data Systems, Inc. The UMAC can be think as a PC, because it has real time operation system, can handle all of the tasks required for machine control, constantly switch back and forth between the different tasks thousands of times per second, also can execute its own motion program and PLC program. The position and velocity loops are closed in UMAC at repeat frequency 2 KHz. The azimuth and altitude axes of mount both are driven directly by the ETEL dc brushless torque motor. Delta Tau's MACRO ring is chosen as interface between UMAC and motor driver. The Haidenhain tape encoder for azimuth and altitude axes are ERA 4282C, 52000 lines, 4 reading heads. Delta Tau's ACC-51E UMAC Interpolator card can creates 4,096 steps per encoder sine wave cycle.

Ground-based and Airborne Telescopes V, edited by Larry M. Stepp, Roberto Gilmozzi,
Helen J. Hall, Proc. of SPIE Vol. 9145, 914552 • © 2014 SPIE
CCC code: 0277-786X/14/$18 • doi: 10.1117/12.2054589

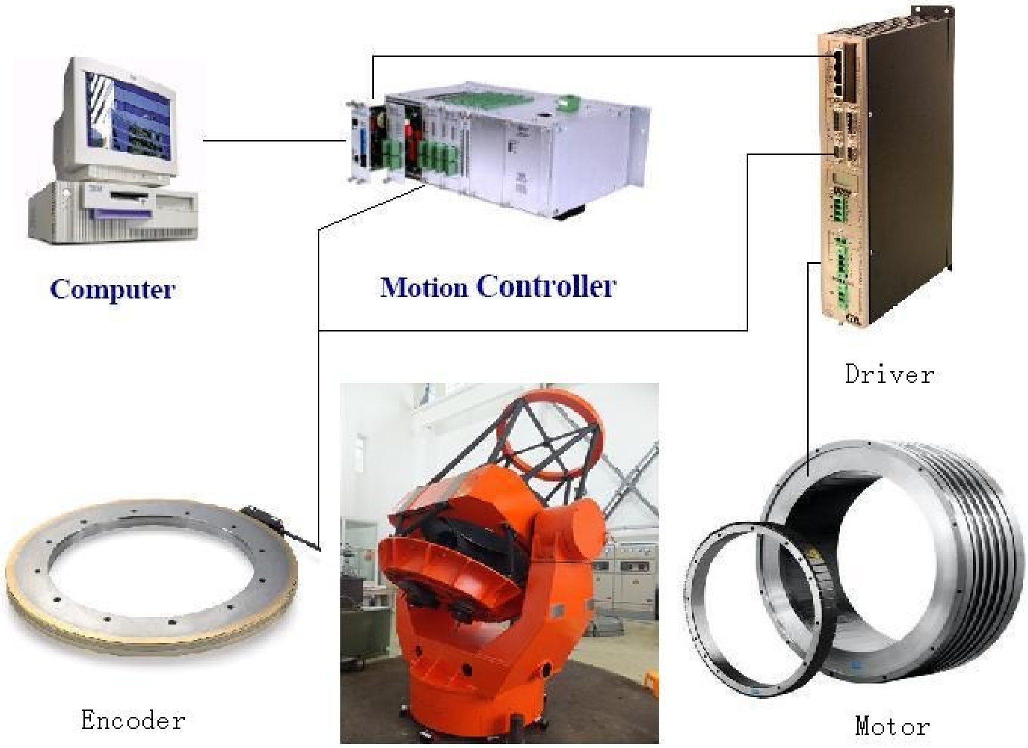

Figure 1. Hardware system layout

3 TCS SOFTWARE DESIGN

The TCS software is developed in Qt Creator environment under stable Debian 6.0 operation system. Qt is a leading cross-platform application and UI development framework. Qt uses standard C++ and is widely used for developing software applications. The basic ideas in the TCS software architecture are modularity and simplification.

The TCS software is divided into eight modules, namely, azimuth altitude, M3 mirror, image field rotator, atmospheric dispersion corrector, set parameter, OCS simulation, dialogue with UMAC and tracking. Each module has a independent child window and one or more independent processes. Exclude tracking module other modules are for test and maintenance.

The set parameter module is for setting up different parameters, such as geographic coordinates, IP address, different file directory and name, axis rotation range and different velocity magnitude, etc. Because receiving commands from LAN and sending back the corresponding replies are becoming more and more complex, so the OCS simulation module is designed for testing communication with OCS computer.

Delta Tau supply a windows program Pewin32PRO this software provides a terminal, a text editor for editing Motion/PLC programs and a workspace environment to configure, control and trouble-shoot UMAC. Under TCS developing process we often need switching to and fro between Windows and Linux OS. So for increasing work efficiency the dialogue with UMAC module is designed for sending command to UMAC and getting reply directly.

The azimuth altitude module is for rotating azimuth and altitude axes at different velocity. A simple command like this "I122=velocity #1J=position" will be sent to UMAC to rotate first axis. One of timer functions will get axis real position and operating status per second. Two sets of PID parameters are set up for different velocity. Other auxiliary function includes enable or disable position loop, homing, parking, software limit position protection, saving tuning data, executing and replying OCS command. Other several axis modules have similar function.

4 TRACKING ALGORITHM

4.1 UMAC motion program

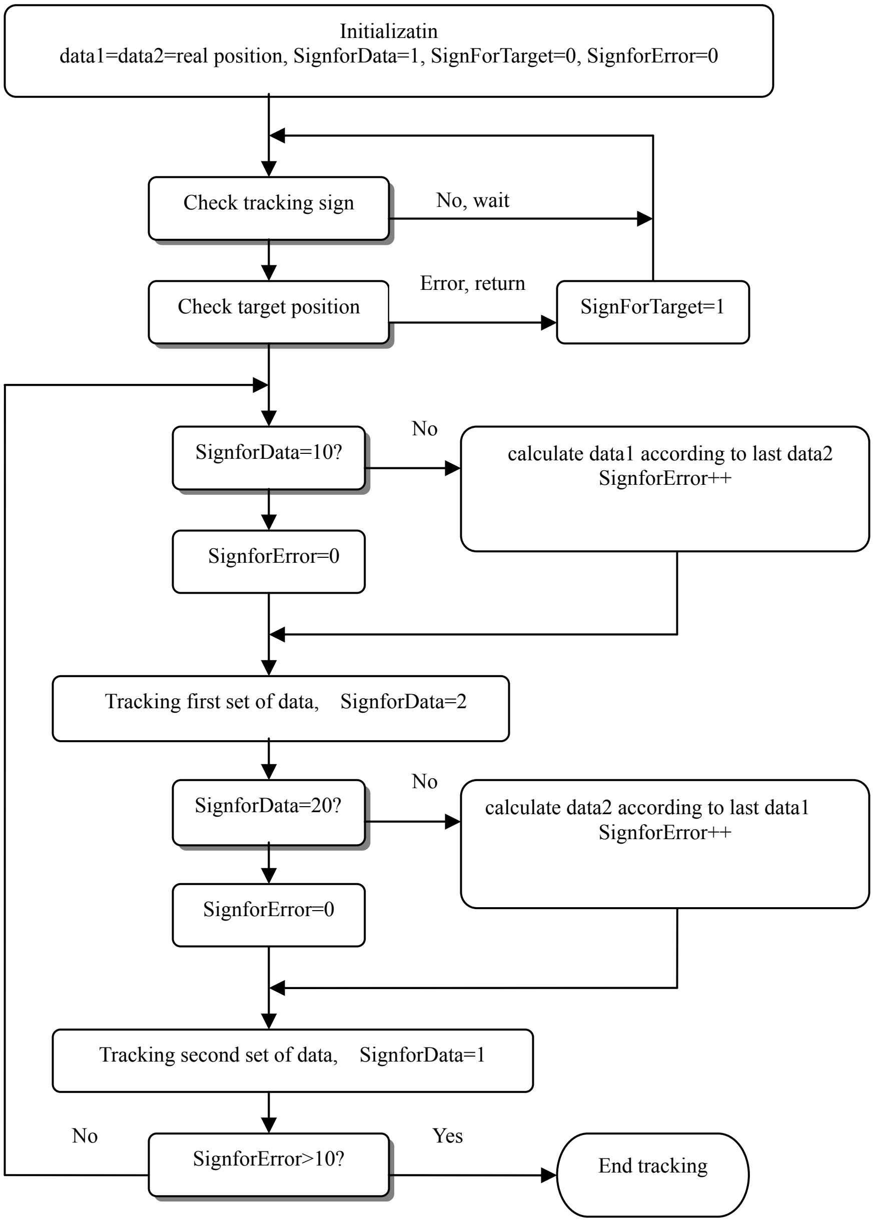

Figure 2. Flow chart of UMAC tracking motion program

Tracking algorithm is comprised of two parts. One is UMAC motion program, another is tracking thread in PC program. UMAC has many motion modes such as Linear, Circular, Rapid, Spline and Position Velocity Time (PVT) etc. Different modes can be used in different occasion. For SONG telescope PVT mode is selected.

PVT mode provides excellent contouring capability, because it takes the interpolated commanded path exactly through the programmed points. It creates a path known as a Hermite Spline. Linear and Spline modes are 2nd and 3rd-order B-splines, respectively, which pass to the inside of programmed points. Compared to Spline mode, PVT produces a more accurate profile.

This mode requires more calculation in the PC. For each piece of a move, the end position and velocity are calculated according to tracking trajectory. Different piece time 50ms, 100 ms, 200 ms, 250ms and 500 ms are compared. For normal velocity small piece time is better, but because encoder resolution is limited and when velocity is close to zero, bigger piece time is preferred. 200 ms is chosen at the end. PC program checks data request sign and sends a set of data to UMAC for next one second at every second. Because the communication between PC and UMAC is not always good, so the motion program adds some protection function and the Figure 2 is motion program flow chart.

4.2 Tracking thread in PC program

Tracking thread is a high accuracy timer function called every 200ms by TCS program. It will calculate theory position and velocity at present and next one second, get real position and velocity, calculate error and correction, get offset from input box or guide CCD, check UMAC sign variable, at last calculate next set of command data and send it to UMAC.
When telescope is behind a star in order tracking a star telescope has to accelerate smoothly from rest to high speed, then maybe has a constant speed period for approaching to the star and then decelerate to the same speed like the star. If telescope is in front of a star telescope will accelerate in reverse direction, then decelerate to zero velocity, and then accelerate to the same speed like the star. The situation is the most complex when star is close to zenith. Basically there are six different situation displayed simply in figure 3. So we need plan the telescope transition trajectory and calculate acceleration, constant speed and deceleration period time.

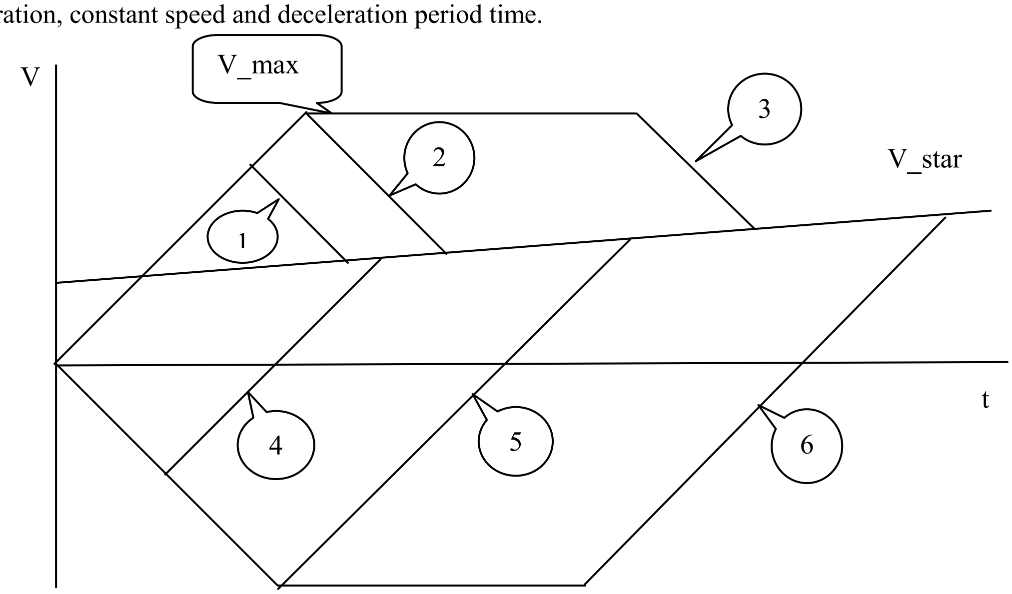

Figure 3. Different transition trajectory

The S-curve is used to eliminate discontinuities in acceleration. To simplify calculation constant jerk S-curves are used. Figure 4 is S-curve profile. List 1 is corresponding formula.

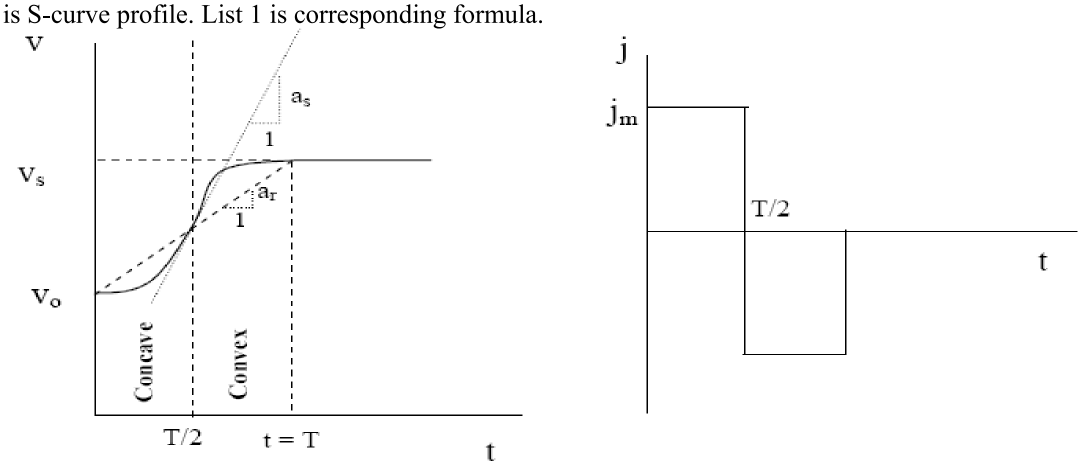

Figure 4. S-curve profile

List 1. Position, velocity and acceleration formula for S-curve acceleration period

Concave period	Convex period
$s(t) = v_0 t + j_m t^3 / 6$	$s(t) = v_1 t + a_s t^2 / 2 - j_m t^3 / 6$
$v(t) = v_0 + j_m t^2 / 2$	$v(t) = v_1 + a_s t - j_m t^2 / 2$
$a(t) = j_m t$	$a(t) = a_s - j_m t$
At t =T, $s = (v_0 + v_s)T / 2 = (v_s{}^2 - v_0{}^2) / a_s, v_s = v_0 + a_s T / 2$	

5 COMMUNICATION WITH OCS

Normally telescope has two work modes track and maintenance. Program only receive remote command in track modes. The main window has a TCP server that listen local area net connecting request. There are 57 commands come from OCS. Each axis has eight basic commands. List 2 is azimuth axis commands and two tracking commands. After receiving a command program will decide how to response according telescope state and can open corresponding module to process the command. Figure 5 is several command process flow chart.

List 2. Azimuth axis command and tracking command

command	process	
TCSCHECKAZ, TCSENABLEAZ TCSDISABLEAZ, TCSSTOPAZ	Reply result immediately	Azimuth module process
TCSHOMEAZ, TCSPOINTAZ	Reply position every 2 seconds	
TCSAZERROR, TCSAZLIMIT	Inform OCS actively	
TCSSTARTTRACKING TCSSTOPTRACKING	Reply position every 1 second	Tracking module process
	Reply result immediately	

OCS has absolute freedom to control telescope via net communication. Remote commands have same priority as local program command. For example program can rotate an axis to a new position after an operator clicks a ‘Go’ button and then can stop the axis after receiving a remote stop command. Telescope need homing first before doing movement. In track mode different axis combination can be chosen, default situation all axes azimuth, altitude, view field rotator and ADC will be switched on. If only azimuth and altitude axes are good and other axes are wrong telescope still will track after program receives ‘TCSSTARTTRACKING’ command and program will reply OCS which axis can’t track.

Telescope every axis normally has several different states and can changes state by executing corresponding command. Figure 6 is just a simple display of state and the actual situation is more complicated. Each axis has a state data structure global variable in the program. When program only has main window opened the state variable is ‘unknown’. Only when some axis module is opened that axis state can be checked.

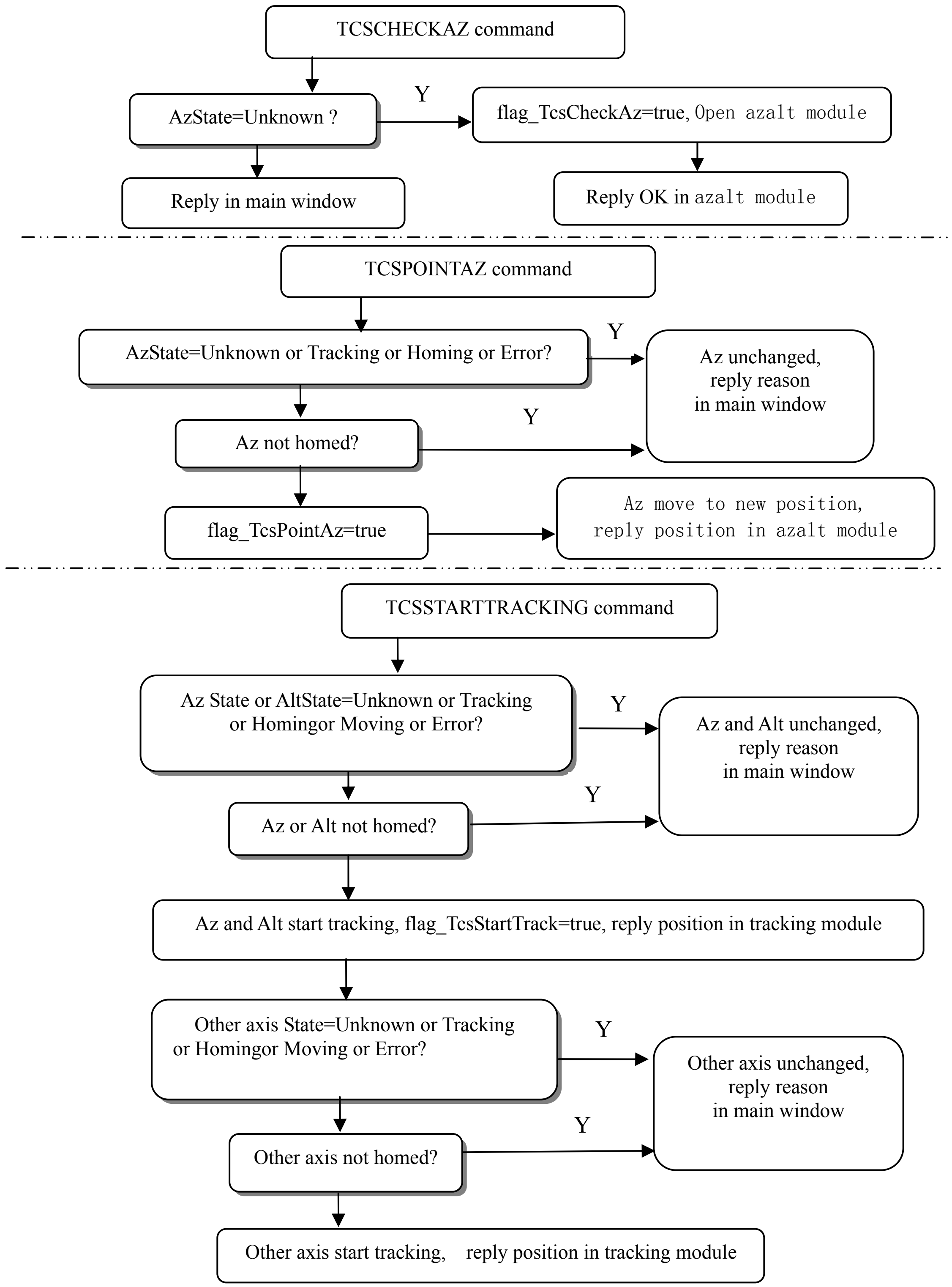

Figure 5. Several command process flow chart

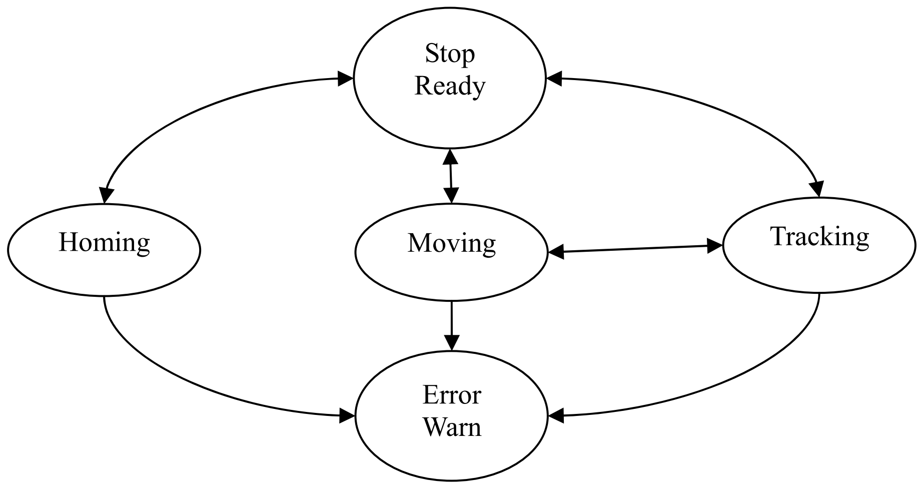

Figure 6. Telescope axis different status

6 DISCUSSION

The telescope has more and more motors to control and total control system become more and more complicated. The communication is not always good when quantity of data is large and transfer frequency is high. The UMAC is very good, but not perfect for telescope control. Program need to be more intelligent to decide how to do when without enough information.

Acknowledgment

We acknowledge fund support from the National Observatory of China. We also thank all colleagues in the NIAOT SONG Group and any other NIAOT staff, who have contributed to the software reported in this paper.

References

[1] F. Carbognani, G. Filippi, “SOFTWARE PRACTICES USED IN THE ESO VERY LARGE TELESCOPE CONTROL SOFTWARE”, Proceedings of ICALEPCS 1999.

[2] Chiozzi, G. et al., “The ALMA Common Software: A developer friendly CORBA based framework” , Proceedings of SPIE 5496, 2004 .

[3] Chiozzi, G. et al., “Application development using the ALMA Common Software”, Proceedings of SPIE 6274, 2006.

[4] Sommer, H. et al., “Container-component model and XML in ALMA ACS”, Proceedings of SPIE 5496, 2004.

[5] Chiozzi, G., “An object-oriented event-driven architecture for the VLT Telescope Control Software” , Proceedings of ICALEPCS 1995 .

[6] Atsushi Shimono, Naoyuki Tamura, Hajime Sugai, Hiroshi Karoji, “The system software development for prime focus spectrograph on Subaru Telescope”, Proceedings of SPIE 8451, 2012.

[7] G.Filippi, “Software Engineering for ESO's VLT project”, Proceedings of ICALEPCS 1993.

[8]Swart, G. P., Bester, D., Brink, J. D., Gumede, C. and Schalekamp, H. J., “Facing software complexity on large telescopes,” Proc. SPIE 5496, 2004.