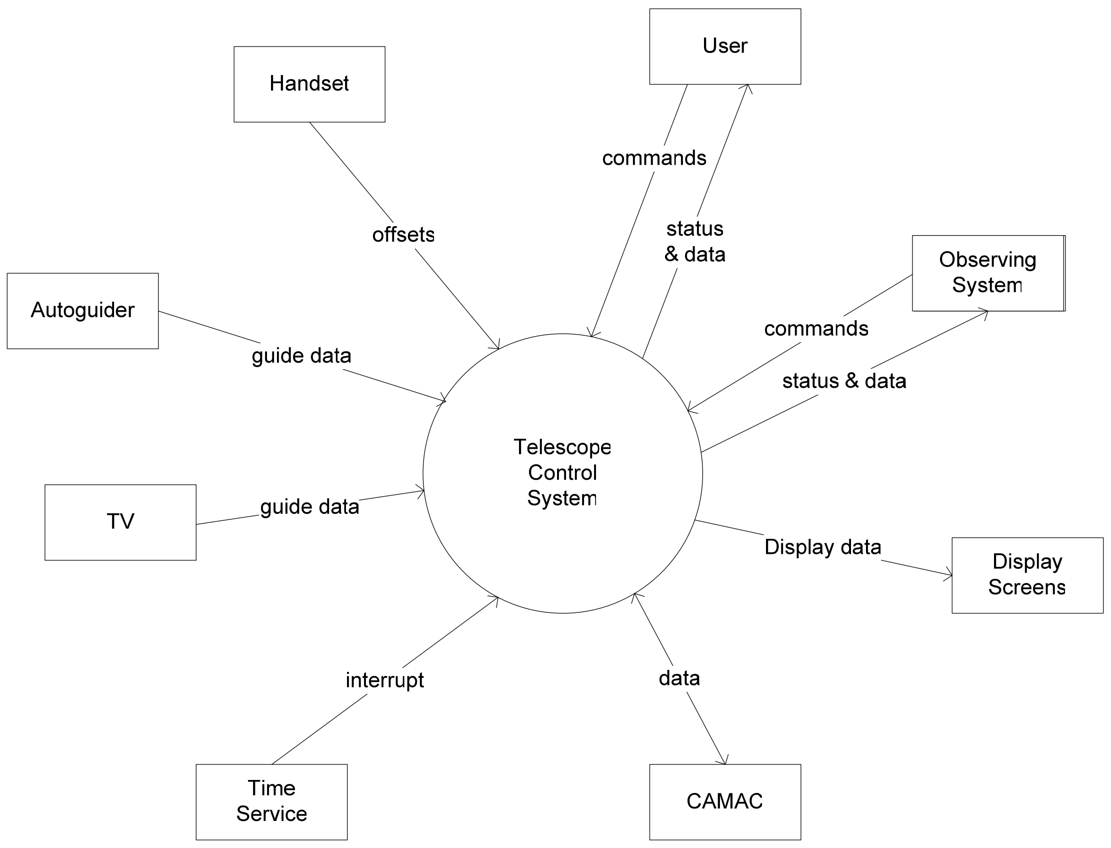
TCS Overview

Statement of Purpose

The purpose of the Telescope Control System is to point the telescope at an astronomical object, to track and focus it accurately and to perform offsets as required during the observation. Information on the state of the telescope system is made available to the observing system.

Context Diagram



Event list

1. Time service generates an interrupt
2. User enters a command at the User terminal
3. User enters a command on the observing system
4. Handset generates an offset
5. Autoguider or TV sends guide data

TCS Processes

There are eight processes that comprise the TCS, they are TCS_MONITOR, TCS_SYNC, TCS_USER, TCS_POINT, TCS_DISPLAY, TCS_TV, TCS_SYSCOMP and TCS_OUTLOG. A DRAMA process, TCS_TELD, is the interface between the TCS and the observing system.

The TCS uses several methods of process communication, both locally to a process and between processes. Asynchronous system traps (ASTs) and other interrupts will transfer control to another routine within a process, or to another process. Common event flags (CEFs) and local event flags (LEFs) are used for synchronisation and control. Data shared between processes are in global COMMON blocks; each COMMON block corresponds to a particular class of data, e.g. mechanism data, encoder data, autoguider data, etc. Data is also passed between processes in self-relative queues.

The TCS processes are usually in a local event flag wait state (LEF state), waiting for a local event flag to be set. The exception is TCS_MONITOR, which is in common event flag wait (CEF state), waiting for the common event flag TCS_TERMINATE. TCS_TELD is in either CEF state or hibernating (waiting for a timer to complete). When an interrupt routine in a process is triggered, for example by an AST or the arrival of a guide packet, the interrupt routine (after a variable amount of processing) will set a local event flag. The main part of the process will reset the flag, perform the required function and then return to the LEF state.

The majority of the code is telescope independent; mount-specific and telescope-specific libraries of subroutines are used to cater for differences between the telescopes. All variables that define how to control a telescope are in the COMMON blocks and are initialised by a telescope specific file (OWNINIT.FOR).

TCS_MONITOR starts all the other processes (except TCS_TELD), then waits until the TCS_TERMINATE common event flag is set, when it forces all remaining processes to close before exiting itself.

TCS_SYNC is the heart of the TCS and runs at 20Hz, triggered by an interrupt from the time service via CAMAC. It is run at priority 17, which ensures it is not interrupted by any other TCS process, nor (under normal circumstances) by any system process. Every cycle it reads the positions of all mechanisms via CAMAC, acts on changes made to the global COMMON blocks by other processes, computes the required position of the mechanisms and writes the resultant data to CAMAC. It writes encoder and related data to a ring buffer for use by TCS_OUTLOG. It triggers TCS_POINT's medium loop (every second), and slow loop (every 5 minutes). It also triggers TCS_DISPLAY, TCS_SYSCOMP and TCS_OUTLOG every second.

TCS_USER parses and validates commands both typed in by the user at the User terminal and received from the observing system via TCS_TELD. Depending on the command it performs one or more of the following actions: update variables in the COMMON blocks; trigger other processes; or display data on the User terminal. TCS_USER can also display a window containing a virtual handset, to allow offsets to be applied interactively to various mechanisms.

TCS_POINT performs pointing calculationsand calculates related position variables in a medium loop (1Hz) and calculates source-independent variables in a slow loop (every 5 minutes).

TCS_DISPLAY displays general tracking information on a default page, there are several other telescope-specific pages containing encoder values, alarms and status information, etc. The pages are refreshed every second.

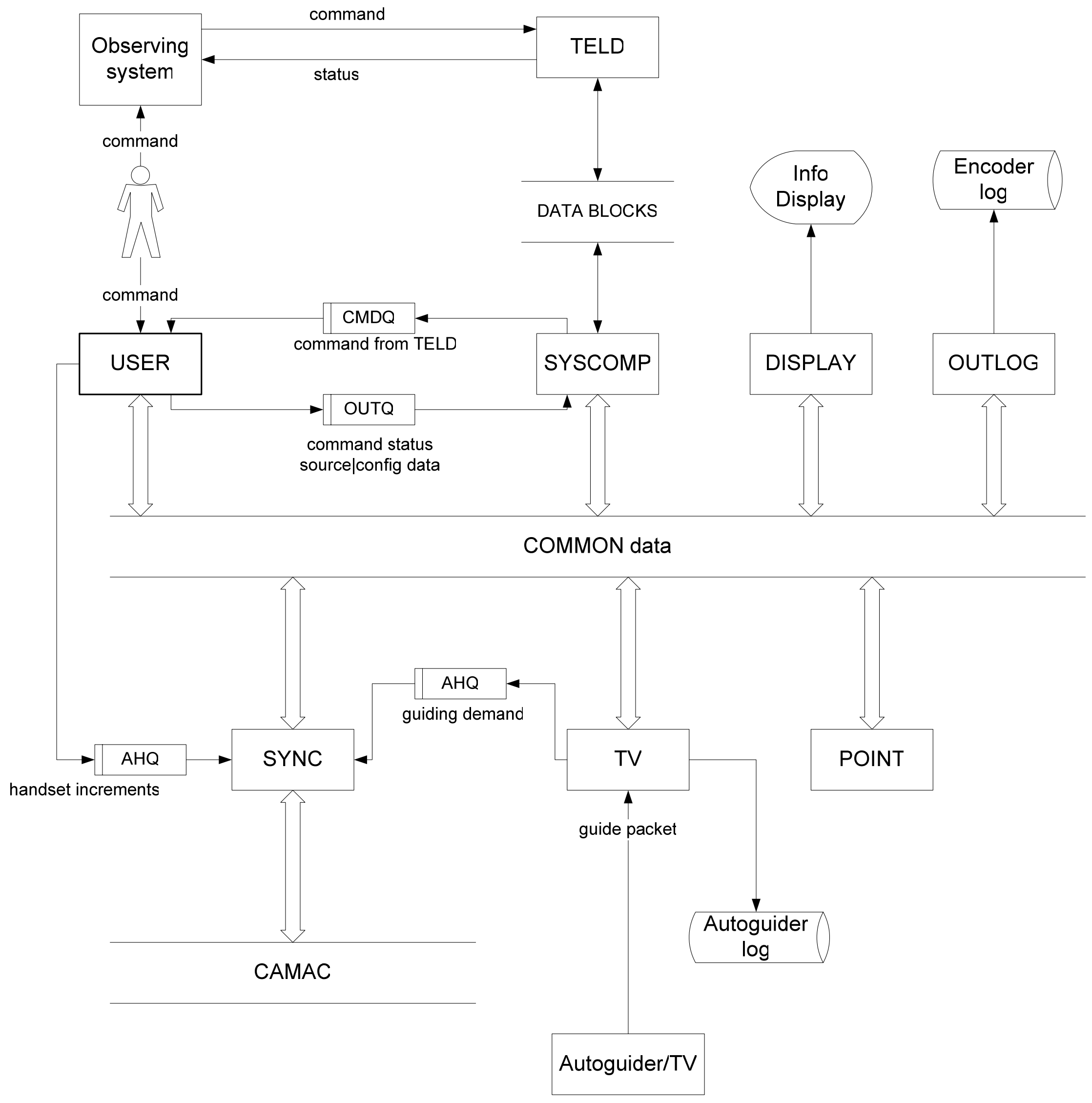
TCS_TV accepts pixel coordinates from the autoguider (or from the TV if one is present), converts them to guiding demands and makes these available to TCS_SYNC.

TCS_OUTLOG writes encoder data and position errors from the internal ring buffer to a direct access ring file every minute. The ring file holds the last 30 minutes worth of data. If requested, the last *n* minutes of data are written out to a log file; also on request, current data will be written to a log file. The log files can be examined with the utility PLOT.

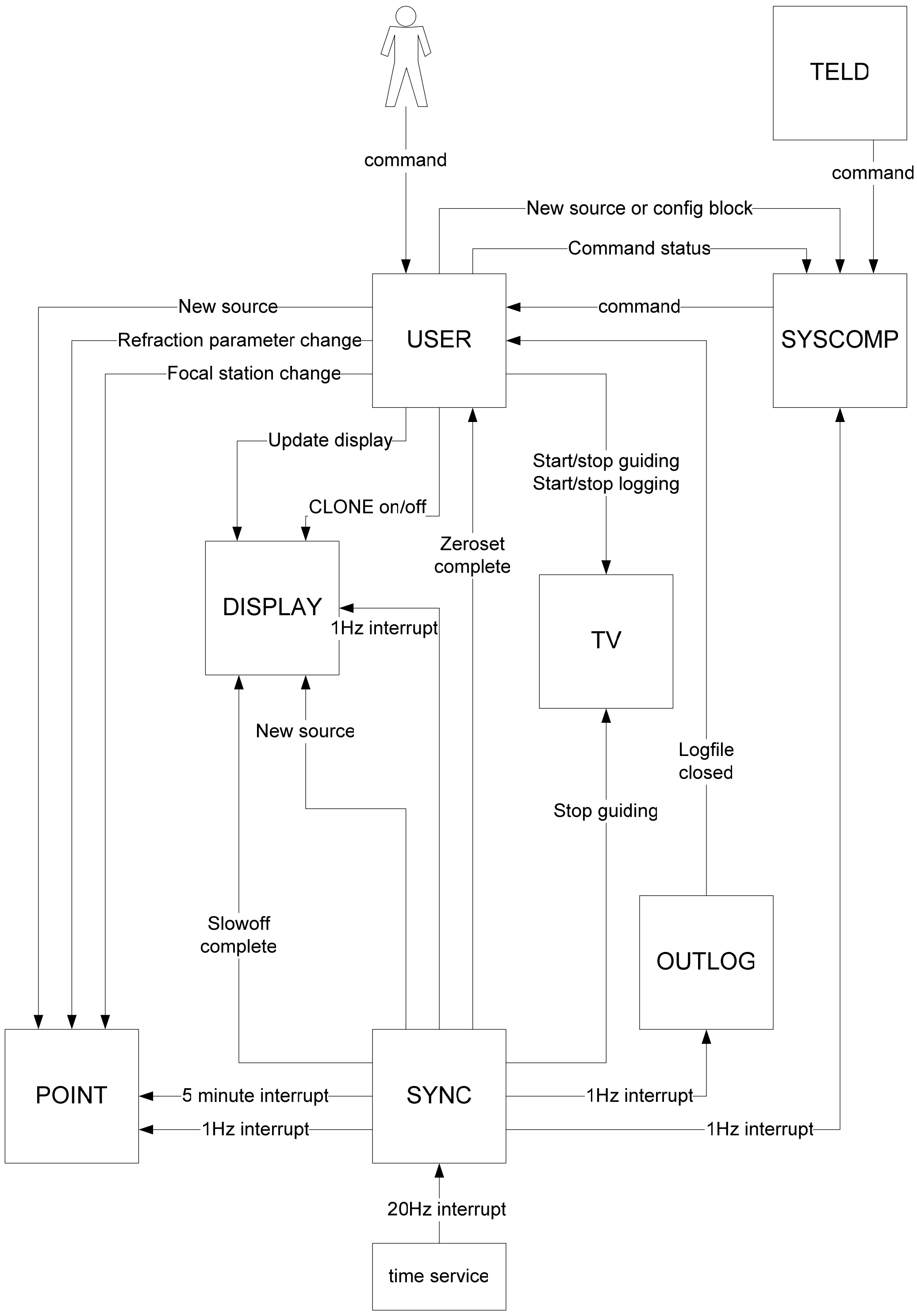
TCS_SYSCOMP is the interface to TCS_TELD. It sends a packet of data containing the current state of the telescope mechanisms once per second to TCS_TELD. Also it sends a packet containing object-related data when the telescope is pointed at a new source, and a packet containing configuration data if the focal station, instrument or autoguider is changed. Each data item in the packets is made available as a DRAMA parameter.
It accepts commands from TCS_TELD originating from the observing system, and sends them on to TCS_USER for validation. It will return an acknowledgement with good status if the command is valid, or with bad status if the command is invalid. If the command causes a mechanism to move, a completion message is sent once the move has completed.

TCS_TELD is a DRAMA process, and is the interface between the TCS and the observing system. It accepts packets of data from TCS_SYSCOMP and makes their contents available as DRAMA parameters. It accepts commands (as DRAMA actions) from the observing system, does some checking of parameters unless the command was the catchall USER command, and sends the command to TCS_SYSCOMP. Once it has received a command acknowledgement or completion message, it signals to the observing system that the DRAMA action is complete, including the returned status. The passing of data packets from TCS_SYSCOMP to TCS_TELD is controlled by CEFs.

Data flows


Observing system
command
TELD
status
command
DATA BLOCKS
Info Display
Encoder log
command
CMDQ
command from TELD
USER
SYSCOMP
DISPLAY
OUTLOG
OUTQ
command status
source|config data
COMMON data
AHQ
guiding demand
AHQ
SYNC
TV
POINT
handset increments
guide packet
Autoguider log
CAMAC
Autoguider/TV

Interrupts



TELD
command
command
New source or config block
Command status
New source
command
Refraction parameter change
USER
SYSCOMP
Focal station change
Update display
Start/stop guiding
Start/stop logging
CLONE on/off
Zeroset
complete
DISPLAY
1Hz interrupt
TV
New source
Logfile
closed
Stop guiding
Slowoff
complete
OUTLOG
POINT
5 minute interrupt
SYNC
1Hz interrupt
1Hz interrupt
1Hz interrupt
20Hz interrupt
time service