Keck Adaptive Optics Note 714

Keck Next Generation Adaptive Optics:
AO Control System Design Document

Doug Morrison, Kevin Tsubota, Jimmy Johnson
April 6, 2010

Revision History

Revision	Date	Author(s)	Remarks
1.0		Douglas Morrison Kevin Tsubota	Initial release.

Table of Contents

1 Introduction 7
2 Glossary 7
3 Document Overview 8
3.1 Subsystem Template 8
3.1.1 Description 8
3.1.2 Interfaces 8
3.1.3 Control Loops 8
3.1.4 Devices 8
4 Controls Infrastructure 9
4.1 Framework 10
4.2 Common Services 12
4.2.1 Connection Service 12
4.2.2 Event Service 13
4.2.3 Health Monitoring 13
4.2.4 Logging 14
4.2.5 Archiving 14
4.2.6 Alarming 14
4.2.7 Configuration 15
4.3 Component Architecture 15
4.3.1 Controllers 15
4.3.2 Composites 16
4.3.3 Device Control Interface 16
4.4 Low Level Motion Device Control 18
4.4.1 Introduction 18
4.4.2 Discrete Commanded Devices 19
4.4.3 Non-tracking Motion Devices 20
4.4.4 Tracking Motion Devices 21
4.5 Low Level Non-Motion Device Control 22
4.5.1 Introduction 22
4.5.2 Remote Power Control 23
4.5.3 Camera / Detector Control 24
4.5.4 Video 24
4.5.5 RS-232 and USB 24
5 AO System 25
5.1 Introduction 25
5.2 Low Order Relay 25
5.2.1 Description 25
5.2.2 Control Loops 26
5.2.3 Devices 27
5.3 Acquisition System 28
5.3.1 Description 28
5.3.2 Interfaces 28
5.3.3 Control Loops 28
5.3.4 Devices 28
5.4 LGS Wavefront Sensors 29
5.4.1 Description 29
5.4.2 Interfaces 30
5.4.3 Control Loops 32
5.4.4 Devices 34
5.5 Low Order Wavefront Sensors 34

5.5.1 Description ... 35
5.5.2 Interfaces ... 35
5.5.3 Control Loops ... 37
5.5.4 Devices ... 37
5.6 High Order Relay ... 38
5.6.1 Description ... 38
5.6.2 Control Loops ... 39
5.6.3 Devices ... 39
5.7 NGS Wavefront Sensor ... 40
5.7.1 Description ... 40
5.7.2 Interfaces ... 41
5.7.3 Control Loops ... 42
5.7.4 Devices ... 42
5.8 Atmospheric Dispersion Corrector ... 43
5.8.1 Description ... 43
5.8.2 Interfaces ... 44
5.8.3 Control Loops ... 45
5.8.4 Devices ... 45
5.9 RTC Interface ... 46
5.9.1 Description ... 46
5.9.2 Interfaces ... 46
5.9.3 Control Loops ... 46
5.9.4 Devices ... 46
5.10 External Offloading ... 46
5.10.1 Description ... 46
5.10.2 Interfaces ... 47
5.10.3 Control Loops ... 48
5.10.4 Devices ... 49
5.11 Calibration Simulation ... 50
5.11.1 Description ... 50
5.11.2 Interfaces ... 50
5.11.3 Control Loops ... 50
5.11.4 Devices ... 50
5.12 AO Enclosures ... 50
5.12.1 Description ... 50
5.12.2 Interfaces ... 50
5.12.3 Control Loops ... 51
5.12.4 Devices ... 51
5.13 AO Control Sequencer ... 53
5.13.1 Description ... 53
5.13.2 Interfaces ... 56
5.13.3 Control Loops ... 56
5.13.4 Devices ... 56
6 Laser System ... 58
6.1 Introduction ... 58
6.2 Laser Switchyard ... 58
6.2.1 Description ... 58
6.2.2 Interfaces ... 59
6.2.3 Control Loops ... 61
6.2.4 Devices ... 62
6.3 Laser Beam Generation System ... 63
6.3.1 Description ... 63
6.3.2 Interfaces ... 64
6.3.3 Control Loops ... 66

6.3.4 Devices 68
6.4 Laser Safety System 69
6.4.1 Description 69
6.4.2 Interfaces 70
6.4.3 Control Loops 72
6.4.4 Devices 72
6.5 Laser Service Enclosure 72
6.5.1 Description 72
6.5.2 Interfaces 73
6.5.3 Control Loops 73
6.5.4 Devices 73
6.6 Laser Interface 74
6.6.1 Description 74
6.6.2 Interfaces 75
6.6.3 Control Loops 76
6.6.4 Devices 76
6.7 Laser Control Sequencer 76
6.7.1 Description 77
6.7.2 Interfaces 77
6.7.3 Control Loops 79
6.7.4 Devices 80
7 Data Server 81
8 Risk Analysis 81
9 Issues Requiring Resolution During DD Phase 81
A1. Functional Requirements 82

List of Figures

Figure 1: A Block diagram of the NGAO system viewed abstractly as a distributed controls system 9
Figure 2: A block diagram of the NGAO controls infrastructure 10
Figure 3: Middleware communication with multiple devices. 11
Figure 4: Generic KCSF Device Interface Layer 23
Figure 5: AO Control Block Diagram 25
Figure 6: Low-Order Relay Collaboration Diagram 26
Figure 7: Image Rotator Control Loop 27
Figure 8: Field Rotation Offset Control Loop 27
Figure 9: Acquisition Subsystem Collaboration Diagram 28
Figure 10: LGS WFS Collaboration Diagram 30
Figure 11: WFS Focus Control Loop 32
Figure 12: Tomography LGS WFS Offloading 33
Figure 13: Point-and-Shoot LGS WFS Offloading 34
Figure 14: Low-Order WFS Collaboration Diagram 35
Figure 15: High-Order Relay Collaboration Diagram 39
Figure 16: ADC Tip-Tilt Error Correction Control Loop 39
Figure 17: NGS WFS Collaboration Diagram 40
Figure 18: ADC Collaboration Diagram 43
Figure 19: ADC Tracking Control Loop 45
Figure 21: External Offloading Collaboration Diagram 47
Figure 22: Telescope Offload Control Loop 48
Figure 23: AO Offset Control Loop 49
Figure 24: AO Enclosure Collaboration Diagram 50
Figure 25: AO Control Sequencer Collaboration Diagram 54
Figure 26: AO Sequencer State Diagram 55
Figure 27 Laser Control Block Diagram 58

Figure 28: Laser Switchyard Collaboration Diagram 59
Figure 29: Laser Steering Control Loop 61
Figure 30: Laser Polarization Control Loop 62
Figure 31: Beam Generation System Collaboration Diagram 64
Figure 32: Laser Focus Tracking Control Loop 67
Figure 33: Asterism Rotator Tracking Control Loop 67
Figure 34: Flexure Compensation Control Loop 67
Figure 35: Asterism Validation Control Loop 68
Figure 36: Laser Safety System Block Diagram 70
Figure 37: Laser Service Enclosure Collaboration Diagram 72
Figure 38: Laser Interface Collaboration Diagram 74
Figure 39: Laser Control Sequencer Diagram 77

1 Introduction

This document describes the system design for the Non-Real-Time Controls portion of the NGAO system for the Preliminary Design Phase.

We begin by discussing the layout and format of the document, and cover some of the common terms that will be used throughout. An overview of the distributed controls architecture is discussed, which includes development of control components and use of common services. Section 4 covers the software design for the motion control hardware system, followed by the software controls architecture design in sections 5 and 6. The design document closes with a summary of existing risks and detailed design phase goals. An abbreviated form of the identified software functional requirements is listed in the appendix.

2 Glossary

The following terms and acronyms will be used throughout this document:

Acronym	Definition
AO	Adaptive Optics
BGS	Beam Generation System
DCS	Drive Control System
DM	Deformable Mirror
IF	Interferometer
IFS	Integral Field Spectrograph
KAON	Keck Adaptive Optics Note
KAT	Keck Angle Tracker
KCSF	Keck Common Service Framework
LGS	Laser Guide Star
LOWFS	Low Order Wavefront Sensor
LLT	Laser Launch Telescope
LSS	Laser Safety System
LTO	Laser Transport Optics
MEMS	Micro-Electro Mechanical Systems
MSCS	Multi-System Command Sequencer
NGAO	Next Generation Adaptive Optics
NGS	Natural Guide Star
OA	Observing Assistant
OOCD	Observing Operations Concept Document
PA	Position Angle
PNS	Point-and-Shoot
PSF	Point Spread Function
TT	Tip-Tilt
TTFA	Tip-Tilt Focus and Astigmatism (LOWFS)
TWFS	Truth Wavefront Sensor
VA	Vertical Angle
WFS	Wavefront Sensor

3 Document Overview

This document is divided into five main sections:

- *Controls Architecture Overview* – This section presents an overview of the NGAO distributed controls architecture as discussed in KAON 679: Controls System Software Architecture, focusing primarily on the component-container distributed model and the KCSF common services.
- *Motion Control Software Design* – Discusses the motion requirements, controller types, and architecture for low-level device control.
- *AO System Software Design* – Details the down-link control process of the AO system.
- *Laser System Software Design* – Details the up-link control process of the LGS system.
- *Risks and Issues* – Summarizes the remaining risks and issues that need to be resolved.

Each of the software design sections will give an overview of each subsystem as well as list their expected inputs and available outputs to the system. The AO and Laser software design sections will also detail the device hierarchies managed by each software controller, as well as the active tracking and offloading control loops. The following template is used to detail each controls subsystem.

3.1 Subsystem Template

Each subsystem of the NGAO controls architecture will be described in detail, and follow a consistent template as described here.

3.1.1 Description

An overview of each subsystem is given by describing its responsibilities, component composition and external collaborators.

3.1.2 Interfaces

The *Interfaces* section details the mechanisms provided by the subsystem for input and output with external components.

3.1.2.1 Inputs

The primary mechanism for commanding and interfacing with a subsystem is provided through the KCSF command interface. Methods are available to get and set attributes, execute commands, and monitor changes in component state. This section details all of the available actions recognized by the subsystem.

3.1.2.2 Outputs

NGAO subsystems will provide status and device information in the form of KCSF events. Any KCSF component can subscribe to a subsystem's published events. This section details each of the subsystem's event streams.

3.1.3 Control Loops

A subsystem or component may implement asynchronous background control tasks. These tasks could be responsible for device tracking, error offloading, and health monitoring. This section details all of the background tasks that will be performed by the subsystem and its controllers.

3.1.4 Devices

A subsystem is a logical organization of a set of hardware devices and their associated software controllers. This section will detail each of the devices managed by the subsystem and how they are used to satisfy the observing use-cases and requirements.

4 Controls Infrastructure

The entire NGAO system can be viewed as a distributed controls system, in which many different components communicate with each other in various client/server or master/slave relationships using possibly several communication networks and protocols. We have represented a high-level abstraction of the NGAO system in the block diagram shown below in Figure 1.

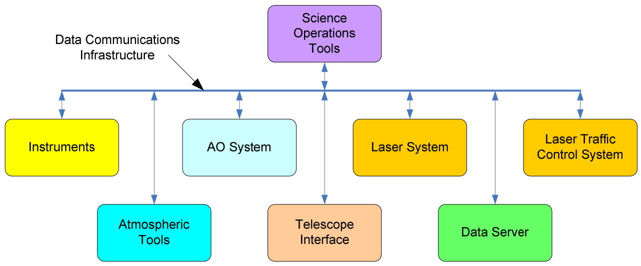

Figure 1: A Block diagram of the NGAO system viewed abstractly as a distributed controls system

Each block in the diagram represents a set of control functions unique to a particular system or subsystem that can logically be grouped together. The grouping is an abstraction, as the various functions represented by each block are actually to be implemented using multiple software and hardware components that may also be distributed.

The software architecture we have chosen to build the NGAO controls system is a framework based on the Advance Technology Solar Telescope's (ATST) Common Services Framework (CSF), described in the next section. We have dubbed the architecture the "Keck Common Services Framework", or KCSF. The KCSF provides the base infrastructure needed to develop a distributed control system.

In a distributed system, components will need to communicate with each other, perform certain actions, and work together in a pre-defined way to form a larger system. The framework enables this by providing common communications, common data types, common commands and common tools throughout. The solution uses a component based development model that supports methods, attributes and properties. The framework provides a kernel API which hides all the details of network access and provides object browsing, discovery, common services and more.

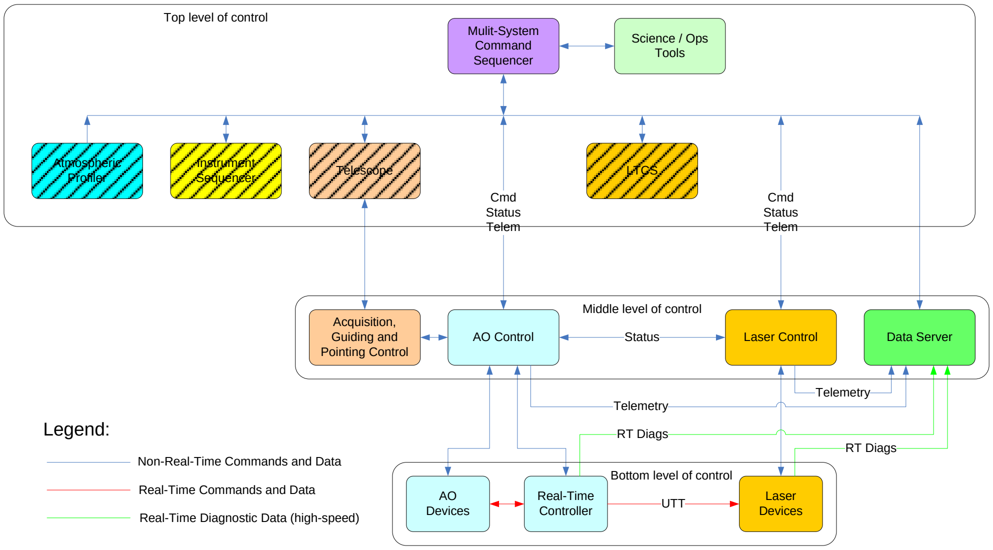

Figure 2: A block diagram of the NGAO controls infrastructure

This diagram depicts the controls system as a software hierarchy. At the top level are the main interfaces to the various subsystems (some of these interfaces are referred to as sequences in the diagram). All user commands to the subsystems pass through these top level interfaces. The middle level of the hierarchy represents an abstraction of more complex lower level control tasks, namely the basic control functions for that subsystem. Users do not access the system at this level except for engineering or troubleshooting purposes. Finally, at the bottom level of the hierarchy are the devices controlled by the control system themselves. The telescope, LTCS, instrument, and atmospheric profile are outside of the direct control of the NGAO system.

4.1 Framework

We have chosen to use a Component-Based Development (CBD) model. This model focuses on removing dependencies between objects by designing components with strict and well defined interfaces. As part of this design a narrow interface model will be employed to reduce component interdependencies. A narrow interface essentially reduces and restricts a components public signature to a well-know and expected set of general methods, thus allowing otherwise unfamiliar clients to interact with a distributed component. Specific functionality is defined through a set of configurable parameters passed to the component interface. This allows developers to add or remove functionality from a component with minimal impact to the existing infrastructure and design. The goal of the component development process is to be able to isolate the functional implementation of a task within a single plug-and-play object that can be modified or swapped with an alternative implementation without impacting the rest of the system. A user should be able to replace a component that satisfies a similar use-case without creating or breaking any dependencies in the system. Refer to KAON 671 for more details.

Supporting the component based model will be a data communications infrastructure that is adopted from the ATST Common Services Framework. Much of the implementation of the communications infrastructure is based upon services and support features found in third-party communications middleware packages. However, the framework isolates the dependence on third-party middleware from the rest of the KCSF software system so that replacing the middleware is always a viable option. We have successfully prototyped three different communication middleware packages (ICE and two versions of DDS). In addition, different middleware can be transparently adapted for different services if they prove to be better in one particular area (i.e., middleware can be mixed and matched as needed). The connection service that encapsulates the chosen middleware essentially forms a software bus to which components can connect. This is illustrated in Figure 3: Middleware communication with multiple devices., which shows many components attached to an abstract software bus. The components all use a simple abstract interface (the software bus) to communicate with each other and are oblivious to the underlying communications mechanism.

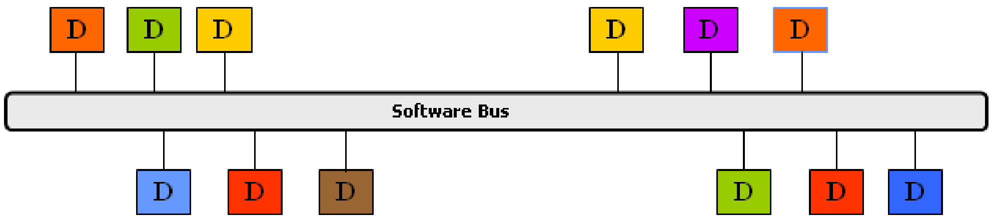

Figure 3: Middleware communication with multiple devices.

The framework takes care of the following:

- Automatically registering all components when they are created so that they become known on the bus.
- Automatically un-registering all components when they are destroyed so that they are removed from the bus.
- Allows any component to be referenced by its fully qualified name, regardless of its location.
- Allows operations to be performed on a remote component as though it were local to the caller.
- Transparently handles network problems such as packet drops etc.

Moreover, the communications infrastructure must be capable of high throughput, varied data size and diverse message types, using different messaging paradigms. KCSF supports this by allowing different communication middleware to be used without affecting the application developer and by using a middleware that supports:

- Peer-to-peer *command* messaging, allowing arbitrarily complex messages to be sent directly from one component to another.
- Publish-subscribe *event* messaging, allowing the generation and reception of messages by components without regard to the intended message recipients or sources.
- Simple connection support. All that is needed for an application to establish a peer-to-peer connection to another application is the *name* of the target application. The communications infrastructure will locate the target application (possibly starting it if it is not running).
- Heartbeat monitoring. Applications are watched and an alarm is raised if an application unexpectedly stops responding.
- Distributed systems. Distributed systems are easily supported, allowing seamless integration of third party modules.

By combining the communication middleware with component-based development, services and a technical architecture, a number of benefits can be realized:

- System details are hidden from the developer and user. This allows application developers to concentrate on applications, not the underlying infrastructure and allows system developers to make infrastructure changes with out affecting deployed applications.

- By using a narrow interface, components can be updated as needed without having to worry about version mismatch on interfaces or methods. The KCSF API is very simple and is designed such that all third party communication middleware is completely isolated from the API. All major services are provided through standard interfaces promoting simplicity and consistency. Dependencies between objects are eliminated, resulting in improved ease of maintenance. As a result components may be swapped out with alternate implementations without impacting the rest of the system. Users can replace a component with another that satisfies a similar use-case without creating or breaking any dependencies in the system. The component model makes it very easy to use components from different development groups in an application.

- The technical architecture takes care of deploying components. Components can easily be mixed and matched and even be added or removed during run time, as can services, effectively changing the system function on the fly. The implementation of the CCM provided by KCSF allows components to be installed, uninstalled, started, stopped or updated without bringing the entire system down. For deployment, the system simply needs to be configured by the engineer to tell it which components to deploy in which containers and on what machines.

Once a component is deployed, the API provides access to its internal state as well as how it is connected to other entities. Parts of the applications can be stopped to debug a particular problem, or diagnostic components can be brought in. Instead of staring at hundreds of lines of logging output and enduring long reboot times, applications can often be quickly debugged using a live command shell. Any entity can be tracked to see its current health and if it is registered.

4.2 Common Services

Common services are infrastructure utilities that provide a black box implementation of typical, repetitive, or convenient functionality need by a software system. KCSF Common services are the backbone of controls architecture, and provide an abstract layer between the controls system and the underlying middleware infrastructure. As a distributed architecture, KCSF offers services to find and connect to remote processes, communicate between individual components in a system, and interface with other system outside of the KCSF infrastructure. Additional services are provided to manage software configurations; monitor the health and status of the system; respond to alarms; and archive system data and events.

The following sections give an overview of the KCSF common services. Additional details can be found in the *Common Services* section of KAON 679: NGAO Software Architecture. Any NGAO specific functionality or considerations related to the use of the common service will be discussed here.

4.2.1 Connection Service

KCSF provides a utility to find and create peer-to-peer connections to remote components in a distributed system through the Connection Service. This service is implemented as a black box to hide the details of the underlying middleware infrastructure, and provide an abstract, simple interface to clients. The following functionality is provided by the Connection Service:

- Register – A distributed component is associated with a unique name and is made public to the distributed control system. Once registered other components can find this object across the network by its unique name.
- Remove – An object is removed from the registry. The object name is no longer a valid endpoint on the network.
- Connect – Allows a component to find and connect to other components in the system based on their unique registered names. This creates a direct peer-to-peer connection through a proxy interface.
- Disconnect – A peer-to-peer connection with a remote component is terminated. The proxy interface no longer has a reference to the remote client.

The registration and removal process are handled automatically by the KCSF middleware: developers do not need to explicitly perform these tasks in their code. A component developer only needs to be concerned with opening and closing connections to distributed components to perform the functions of the controls system.

NGAO will not require any special implementation or any additional functionality that is not currently available with the standard KCSF Connection Service.

4.2.2 Event Service

The majority of broadcast and non-command messaging is satisfied using events. Events are based upon the publish-subscribe design pattern, and allow a component to make information available to the entire distributed system without requiring an explicit connection between clients. Events are designed to provide fast peer-less communication channels using the capabilities of the underlying middleware. KCSF provides access to the event system through the Event Service.

The KCSF Event Service has the following properties:

- An event stream (a.k.a *topic*) represents a many-to-many mapping: events may be posted to the topic from more than one source and received by zero or more targets. Typically, however, most event topics will have a single source.
- Events posted by a single source into an event topic are received by all targets in the same order as they were posted.
- Delivery of events to one subscriber cannot be blocked by the actions of another subscriber.
- Events are not queued by the service. A "late" subscriber will not see earlier events.
- The event service does not drop events. A published event will be delivered to all subscribers.
- Event topics are identified by a unique name.
- The Event Service supports arbitrary topic names.
- Events are automatically tagged with the source and a timestamp.

The Event Service provides a simple API to interface with event topics:

- Subscribe – Start listening to events published on a particular topic.
- Unsubscribe – Stop listening to events on a particular topic.
- Post – Publish data to a topic.

Events are received by a component by attaching a callback to a subscription. The event service, upon receipt of an event, invokes this callback in a separate thread. All events received from the same subscription use the same thread: delivery order is preserved within the callback processing. If events are received faster than the callback can process them, the unprocessed events are locally queued within the event service. This is a potential problem, but represents a trade-off of mutually exclusive goals. Component developers are encouraged to write callbacks that process quickly. Numerous approaches are available to handle the case where the required action cannot be performed quickly - the best approach to use is dependent upon the nature of the specific task and is thus the responsibility of the component developer.

4.2.3 Health Monitoring

The ability to quickly ascertain device health throughout the NGAO control system is very important. KCSF supports this by providing a health monitoring service. The health of each component is monitored by its parent container. The various health states for a component are: good, ill, bad, and unknown, and are defined as follows (in order of worsening health):

- Good: No problems have been detected by the component, it is fully operational.

- Ill: Problems have been detected, but they do not prevent observing. Data quality, however, may be affected. It may also be the case that operation of the component will fail soon if corrective action is not taken. The component is partially operational.
- Bad: Severe problems have been detected. The component is unable to operate correctly. Corrective action is required.
- Unknown: The component is not responding. It may or may not be operating. This health value is not set by the component (obviously) but may be set by the health service.

The health service automatically posts an event showing changes to the component health and logs a warning on worsening health and a note on improving health. When a health condition worsens to bad or unknown the log message severity switches from warning to severe.

All components define an abstract *performCheckHealth* method, which must be implemented by the component developer. This method is responsible for determining the health of the component and returning its health status. For most NGAO controllers the health status will be based on feedback from the device or motion controller, and will typically be in either a *good* or *bad* state. Higher level composite components and sequencers will have to calculate health based on the overall status of their individual controllers. For these components determining health will be based on the importance of specific devices over others in the control system, where many failure scenarios can be categorized as either *bad* or *ill*.

4.2.4 Logging

Logging is the capability of writing messages to create a record of system events. Typically the content of a log message isn't intended for immediate review or action, and is instead intended to trace the execution of tasks and provide feedback of system performance. Log messages can typically be written to a variety of output streams for temporary viewing or long-term preservation. KCSF provides logging functionality through the Log Service.

The Log Service offers clients the ability to write messages to one of three types of outputs: console, file, and database. A log message is categorized by a specific log level, indicating the type of message and its importance. The KCSF Log Service defines a total of seven log levels ranging from a simple trace message to an emergency system event. When a message is issued the Log Service will automatically append timestamp and source information and format the message for consistency. The Log Service provides convenience methods for each of the defined log levels to simplify component development. The KCSF Log Service is described in detail in KAON 673: NGAO Software Architecture: Logging Service.

In addition to the Log Service, KCSF also provides a graphical log viewer. This tool allows users to inspect persistent log files saved to a database. Logs can be searched by specifying a range of options, including date, log levels and sources. All log events that fit the search options will be displayed in a simple table organized by date-time, level and source.

The KCSF Log Service and viewer will be used unchanged for NGAO.

4.2.5 Archiving

To be supplied.

4.2.6 Alarming

Alarms are reports of component failure (software or hardware) or other abnormal behavior within the system. Alarms occur asynchronously in a random fashion. Some alarm conditions may clear themselves and others may require operator intervention. The KCSF Alarm Service provides capabilities for components to set and clear alarms, and provides a managing system to detect the occurrence of alarms and initiate the appropriate system response. The Alarm Service is part of the larger NGAO alarm system.

The Alarm System is described in detail in KAON 677: NGAO Software Architecture: Alarm System

4.2.7 Configuration

The configuration service is just one part of the larger configuration system in KCSF, which in general provides the ability to manage the configurable properties of each component in the NGAO system. It consists of a configuration database, the associated tools to manage the database, and the configuration service, which provides a client interface to each component to request its own configuration information from the database.

During initialization, each component calls the configuration service to access its specific configuration data. There is a specific method for Container Managers, Containers, and Components that accept a fully-qualified name used to lookup the information relevant to each of these types. The Configuration Service utilizes attribute lists to return data to the invoking object. The details of this process are handled automatically by the parent container with no action required on the part of the user.

4.3 Component Architecture

The CCM defines a distinct separation between the technical and functional requirements of a task. The model is based on the utilization of two distinct software modules: *containers* and *components*. Containers provide a structural or logical organization to software objects (components), and are responsible for their lifecycle and management. This is illustrated in **Error! Reference source not found.**.

A ***container*** will create components, start them running, shut them down, and remove them from the system. Containers provide a uniform method of system management and allow component developers to ignore the majority of the technical requirements of individual objects. Containers, component interfaces and abstract implementations are provided as part of the software infrastructure.

A ***component*** is a piece of software that implements the functional requirements of the system and conforms to a well defined interface or set of interfaces. The interface abstracts away the actual implementation of a component, and from the perspective of the system, there is no difference between one implementation and another. In this way, a developer can swap out multiple implementations of a component for a specific use case without affecting the system as a whole. This is most evident when you compare the maintenance and upgrade of a component-based and a non-component-based system.

Implementation of the NGAO Controls system will focus on the design and development of component subclasses. KCSF provides two component implementations that can be used to build a controls system. They are the *controller* and *composite* base classes. From these classes a controls system can derive all of the device and software control functionality required by the system.

4.3.1 Controllers

Controllers are the lowest level software component in a control system and interface directly with the motion control and device hardware. Controllers provide functionality to set or get properties; execute synchronous or asynchronous tasks; and monitor state with callbacks. The controller development process is focused on overloading the stubbed *perform* methods to recognize and execute actions and tasks issued by clients. The actual receipt and dispatching of actions is handled by the controller's technical architecture, and is completely invisible to the developer.

The NGAO controller design will be based on a state driven architecture. Actions will be used to issue transition between the different run-time states of the component. The state machines for each controller will vary but typically follow the standard state pattern for motion controllers: home, initialize, slew, inpos, track, halt, and disable / shutdown. State machines will be built using the State Machine Compiler (SMC) utility. This tool allows developers to define states and transitions by creating a dispatching wrapper that executes tasks on the driver class (in this case the Controller). A low-level state controller will be created for each device type (as outlined in section Low Level Motion Device Control). The developer will only need to override the task implementations in the subclasses to gain the desired functionality. (Additional details can be found in the *Sequencers* section of KAON 679: NGAO Software Architecture.)

4.3.2 Composites

Composites are similar to controllers in basic functionality: both provide a state driven software component with set, get, and execute methods. However, they are different from controllers in the role they fill in the controls hierarchy. As the name implies, a composite is a collection of components. Specifically, composites manage the life-cycle and coordinate the execution of one or more low-level controllers and sub-composites. Composites themselves do not directly interact with hardware. Instead they build the software hierarchy by dividing the system into discrete units. Each composite and its subordinate components are intended to be fully self-contained, allowing clients to treat a particular portion of the system as a black-box.

The NGAO Controls system will be designed as a hierarchy of composites, where each layer abstracts away the underlying detailed functionality of the sub-components. At each layer, the client is provided with a high-level interface to command the execution of common sequences required by the controls subsystem. As the composite move farther away from the low-level devices the interfaces become more general and identify less of the underlying infrastructure. At the highest levels a set of global commands are offered to initialize, acquire, track, and halt the system.

4.3.3 Device Control Interface

The following section details the common input and output interfaces available to most NGAO components. In the AO and Laser system portions of this document the individual subsystem interfaces are detailed. Each subsystem will expand on these interfaces and include any device specific functionality involved in processing these actions and events. Subsystem may implement additional interface capabilities or omit specific functionality entirely. See each subsystem for more information.

4.3.3.1 ***Inputs***

Home

The *Home* action is responsible for homing the component's related hardware devices. Typically this action is only applicable for motion control devices, and involves slewing to one or more limits and zeroing the encoder position. Additional tasks may include configuring internal control loops, loading configuration files, and refreshing driver connections.

Initialize

The *Initialize* action is responsible for initializing a component and its related hardware devices. Software initialization consists of refreshing proxy connections, connecting to event stream, and clearing existing monitoring callbacks. Essentially the initialize action should return the software component to its starting configuration.

Hardware initializations cover motion control and static devices. As static devices, such as sensors, do not have any moving components an initialization usually performs a device refresh or some form of internal configuration (and may be similar to the *Home* action.) Motion control initializations, in addition to performing any required internal reconfigurations, will typically involve slewing to a known limit and returning to the home switch or a specific encoder position (e.g. zero). The initialization process for a device should be faster than homing, and must not invalidate its position counter. This allows an operator to accurately return an initialized device to a previously defined position.

SetMode

The *SetMode* action is responsible for configuring a component for a particular observing configuration. Every component in the NGAO control system is designed to recognize and understand its role in a particular observing mode, and will configure itself appropriately. The configuration process can include slewing devices, configuring device properties, making connections to remote components, disabling device controller, and powering down hardware. On completion of a *SetMode* action the component and its devices should be ready for observing in the desired mode.

Slew

The *Slew* action is common to all motion control devices are is responsible for commanding the device to a specified position. The specific format of the data is dependent on the type of device (tip-tilt, linear stage, etc.) and the number of degrees of freedom. Most controllers also permit the use of named positions allowing clients to specify an enumerated value in place of device counts or physical positions. On completion of the *Slew* command the controllers associated devices should reach their commanded position.

Track

The *Track* action is available to all motion control tracking devices. A tracking device is any device that runs in an event driven (or periodic) mode based on system events -- also known as closed loop motion control. The *Track* action commands the component and associated hardware to enter closed loop control to respond to one or more stimuli. For example, a focus stage that responds to focusing errors measured by a sensor. A component in track mode will stay in track until it receives a command to *Halt* or a problem is encountered.

Acquire

The *Acquire* action is a high-level operation available to composites and subsystems. This action is responsible for implementing the component and device sequence to acquire an observing target(s) (e.g. a tip-tilt NGS). Typically this action accepts target coordinates and properties that are used to calculate device positions, loop rates, and other system configurations. Depending on the responsibilities of the subsystem an acquire may cause devices to slew, sensors to reconfigure and control loops to close. On completion of this action the subsystem should be properly configured for observing.

Halt

The *Halt* action is available to all NGAO components and is designed to immediately open all tracking / offloading loops and cut power to all devices managed by the component. This action is executed to terminate a sequence -- either in response to a system failure or the end of an observation. The *Halt* action should leave all devices and components in a useable state: they should not require an initialization or special recovery sequence; although the acquisition sequence may not to be restarted before observing can continue.

Recover

The *Recover* action is available to all NGAO components, and is used in the context of selective initialization of one or more devices during a system fault. During observing, devices may encounter problems that cause the hardware motion controller and NGAO component to enter their FAULT state. Rather than being forced to terminate the observing sequence to resolve the problem, it may be desirable to continue observing, although in a potentially degraded capacity. In such an event the operator will be able to issue the *recover* action at the subsystem level.

When a *recover* is issued to a subsystem it is propagated to all of its child components, even if they report that they are functioning properly. Essentially the recover action is sent blindly to the entire control subsystem hierarchy. As each component receives the recover action it will check to see if it is currently in a faulted state. If it is, an initialization is performed to attempt to clear the fault, and if successful, the device is repositioned and loops are closed to bring the device(s) back into the proper operating state. (Note: device acquisition is only performed if the device had faulted while in the TRACK state.)

If the device can not be recovered it will remain in its faulted state, and direct operator intervention may be required at the device level. If a device fault is cleared at the device level (e.g. through an engineering GUI), the control system will still need to be told to recover from the fault by issuing the recover action: a component will not clear its fault just because the device reports “in-position”. This is to ensure that all control system dependencies have been properly informed of the recovery and have responded as needed.

Disable

The *Disable* action is designed to put a component into a silent, low-power, low-resource state. While disabled the component will not be publishing telemetry nor monitoring and responding to system events. Disable can only be performed when the controller is in the HALT state.

The intended purpose of this action is to reduce system resources and device interference when the component is not needed. An example of this situation would be during system testing and calibration when the operator only wants to interact with a portion of the control system. Disabling extraneous devices reduces the chances of interference and the risks to personnel that may be in the AO or Laser facilities.

Another example where devices will regularly be disabled will occur during system reconfiguration while setting an observing mode (see *SetMode*). When switching between the various observing modes certain devices will not be needed, and may in fact interfere with observing if they were to be accidentally issued a command. Disabling the associated controller will prevent the accidental commanding of these devices.

In order to re-enable a disabled component for use a *Home* action must be performed. As with a standard home operation the device will go through an initialization process and zero its encoder positions. Once the home and initialization is complete the device can be commanded normally.

Outputs

To Be Supplied

4.4 Low Level Motion Device Control

4.4.1 Introduction

The devices to be controlled by AO Controls are many, and are detailed in the configuration controlled KAON 682: NGAO Master Device List spreadsheet. As of this writing there are a total of 83 motion devices, 29 of which are laser devices. There are also a substantial number of non-motion devices that is covered under the AO Control System. One thing to note is that AO Controls does refer to real-time control of devices, as in realtime control of deformable or tip-tilt mirrors, but to the basic control of the devices themselves. This includes power control, initialization, basic parameter control, etc…

Motion control refers to the control of any device that has to be positioned before it can be used. This control depends on the type of motion device being controlled. The particular devices are being designed in parallel and are described in KAON 715: NGAO Preliminary Motion Control Design.

Although we haven’t converged on a particular hardware control system, we are very familiar with Delta Tau PMACs as they are used extensively in our current AO and Interferometer systems. Our current thought is to continue to use PMACs, especially given its programmability and scalability. We are however considering moving away from the VME design to a smaller PXI/CPCI form factor. Also, for low precision intermittent motion devices we may use smart motors for simplification of cabling as they can be chained together and would be controlled over a serial interface.

The low level motion control functions consist of one or more device drivers that implement an API for interacting with the motion control system to provide the most basic level of control. This device driver will most likely be supplied by the manufacturer of the motion control electronics. An abstraction layer will be added above the low level driver as part of AO Controls. This layer would hide all the hardware specifications and implementation of the actual motion control hardware. At the top of this control block is a generic state machine that implements the basic coordinated functions required for controlling a motion control device with an arbitrary number of degrees of freedom. This concept is used in the existing Keck AO and Interferometer systems.

KAON 715 identifies all of the current device types / categories required by NGAO. In order from least complicated to most demanding, they are:

- Shutters
- Low precision non-tracking devices
- Medium precision non-tracking devices
- High precision non-tracking devices
- Tracking devices
- Extremely high precision tracking and non-tracking devices
- Coordinated high precision tracking and non-tracking devices

From a software perspective, these can be grouped into the following generic categories:

- discrete commanded devices
    - shutters
- non-tracking devices
    - low precision non-tracking devices
    - medium precision non-tracking devices
    - high precision non-tracking devices
    - extremely high precision non-tracking devices
    - coordinated high precision non-tracking devices
- tracking devices
    - tracking devices
    - extremely high precision tracking devices
    - coordinated high precision tracking devices

The intent would be to develop each type once and instantiate it for all the necessary devices. This will simplify implementation and allow for quick and easy changes or addition of new devices. Each of these types is discussed in the following sections.

4.4.2 Discrete Commanded Devices

4.4.2.1 Description

This is the simplest of device types and consists of in/out devices with loose positional requirements; actuators other than motors (e.g., solenoid, pneumatic, etc); and devices with switches or hard stops used to define positions.

Software controls for this device type will involve sending a discrete output signal to a digital I/O controller board and status information will be returned in the same manner. More than likely the status will consist of limit switch readings where only one limit can be active at any given time. When no limit switches are active, this indicates that the device between positions and thus assumed to be “travelling”. When both limit switches are active, this indicates an error.

4.4.2.2 **Inputs**

Commands to this device type will be simple boolean 0 or 1 commands where 0 means one state and 1 means the opposite state. For instance open/close or in/out.

4.4.2.3 **Outputs**

Since these are simple discrete devices, the outputs will consist of two boolean state readback values which correspond to opposite positions, i.e. open/close, in/out, or cw/ccw, etc… A valid status will be when only one state is active and the other is open. When both states are open, this indicates that the device is travelling between states. When both states are active, this will indicate a fault condition.

4.4.3 Non-tracking Motion Devices

4.4.3.1 **Description**

Non-tracking motion devices are devices that can be positioned anywhere within a fixed position range. Examples of non-tracking devices are:

- dichroics
- fold mirrors
- focus stages
- lenslet stages

These devices are positioned during configuration or acquisition and then left in place with its servo loops open, i.e. no power. They are not moved during an observation. There will be varying degrees of precision: low, medium, and high, all based on encoder resolution with matching DAC output control. However from a software perspective these are all treated the same. They must all be initialized, which means that their encoders must be homed. This involves driving to a home switch, which could be a limit switch, and loading a particular position once the switch is sensed then applying a home offset to position the device to a known position. Depending on the device, this could be a specific position or some where in the middle of travel, which we call encoder position 0. This homing process is built into the PMAC and can be configured on a motor by motor basis as to which direction to home, what to use for a home switch, and the specific home offset. In most cases this is a straight forward operation meaning that the device axis can be freely homed without any limitations. However in the case of the LOWFS OSM (object selection mechanism), care will need to be take to prevent them from colliding into one another during the initialization process. The actual homing of the OSM pickoff arms will also be tricky and require a customized homing process.

Once the device is homed, it will then need to be positioned to specific locations such as in beam or out of beam. This will be done by using named positions instead of actual position values or encoder counts. The named position will be mapped to a specific encoder position which can change as needed but the named position used will always be the same, i.e. in-beam, out-beam, etc.

For some devices, i.e. focus stage or a lenslet stage, positioning will need to be based on phsical positions then adjusted throughout the night. The initial position, in the case of a focus device, will be pre-calibrated and can have a named position, such as initial-focus, but subsequent moves / adjustments will need to be commanded in some form of engineering unit that is logical to the device. So instead of adjusting focus by some number of encoder counts, you would specify a move in terms of millimeter of focus, for instance.

4.4.3.2 **Inputs**

- Home / Initialize command
- Abort command
- Move to named position command
- Move to logical unit position command
- Move to raw encoder count command

4.4.3.3 **Outputs**

- Current position in
    - raw encoder counts
    - equivalent logical unit position
    - applicable named position
- In position status
- Home status
- Servo loop status
- DAC output
- Following / position error
- Limit conditions

4.4.4 Tracking Motion Devices

4.4.4.1 **Description**

Tracking motion devices are devices that are constantly commanded to a new position during an observation. The actual position is based on and synchronized with an external source, i.e. telescope az or el, and is updated at a periodic rate. Like non-tracking devices, tracking devices require various levels of precision as well as coordination of moves with other DOFs, i.e. steering mirrors, and in some cases spatial position constraints to avoid collisions and occlusion, i.e. pickoff arms.

From a software perspective, tracking devices are basically non-tracking devices with the added capability of being able to handle continuous position demands by keeping its servo control loop closed until explicitly opened by an explicit command. Ideally the controller will be able to handle velocity and position loops to help keep motion smooth and constant.

4.4.4.2 **Inputs**

- Home / Initialize command
- Abort command
- Move to named position command
- Move to logical unit position command
- Move to raw encoder count command
- Continuous tracking demanded position commands

4.4.4.3 **Outputs**

- Current position in
    - raw encoder counts
    - equivalent logical unit position
    - applicable named position
- In position status
- Home status
- Servo loop status
- DAC output
- Following / position error
- Limit conditions

4.5 Low Level Non-Motion Device Control

4.5.1 Introduction

As defined by KAON 668: NGAO Device Control Architecture, non-motion device control refers to the control of any device or subsystem that is not creating motion and is not one of the devices requiring real-time control for wavefront sensing and correction. This includes power control, environmental control, camera control, mirror control, general sub-system control (e.g., the laser units, the RTC) and the controls infrastructure.

Non-motion device control is, in general, low speed without tight timing constraints. The commands are frequently single events (e.g., GUI button press) and are expected to be executed on the timeframe of seconds. Ethernet will be the primary means of communicating with devices for control. If a device does not have an Ethernet interface there may be simple Ethernet solutions which exist (e.g., terminal servers to communicate with RS-232 based devices, or digital I/O modules with an Ethernet interface to generate digital I/O signals). To reduce software integration efforts, Ethernet devices should support an industrial protocol (Modbus TCP/IP, Fieldbus, etc) rather than a web (HTTP) based interface whenever possible.

The following sections describe the various types of device control which would be supported by this device layer. Currently the following device interface control methods have been identified in KAON 668:

- Remote power control
- Camera / detector control
- Digital input/output
- Analog input
- Analog output
- Remote reset control
- Video
- RS-232 and USB

As part of the synergy with the TCS Upgrade project we hope to use the same generic IDaqDevice interface currently under development by TCSU.

Figure 4 below, illustrates the concept of a generic IDaqDevice interface taken from a TCS Upgrade example. The DAQ specific controller would be created to match the IDaqDevice interface. In this case we show the use of the NIDAQ Device Controller. This device controller uses the NI-DAQmx API from Nation Instruments and supports the ICE IDaqDevice interface. The NI-DAQMmx drivers supports a number of device types:

- Multifunction DAQ
- Analog output
- Digital input/output
- Counter/timer
- High speed digitizer
- Dynamic signal acquisition

A DAQ Client will run as part of the application and will be responsible for setting up the IO, alarming and reading/writing to the devices via the ICE interface

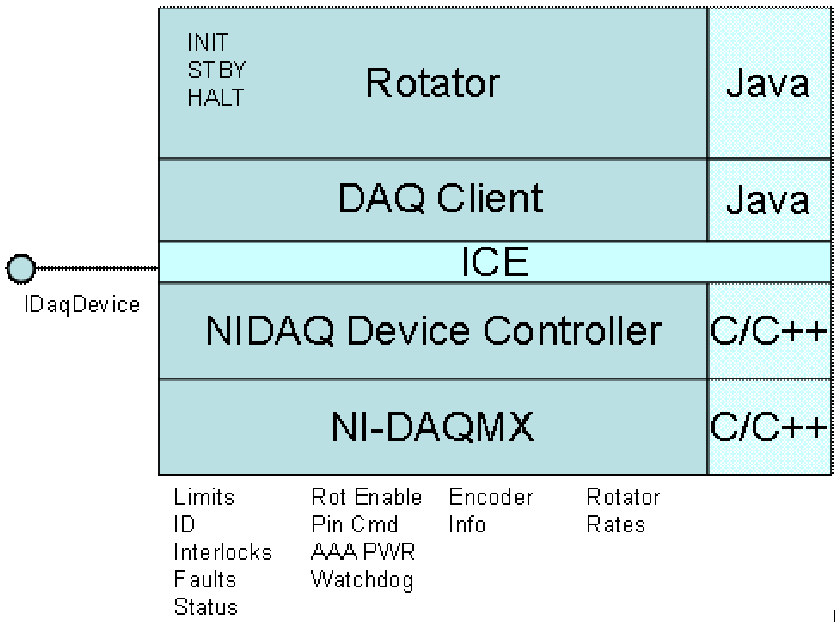

Figure 4: Generic KCSF Device Interface Layer

Serial and Ethernet interface device types, which are not supported by NI-DAQmx, as well as special camera or detector controls, will be handled as part of the existing KCSF device/controller infrastructure. For more details refer to KAON 679: NGAO Control Software Architecture.

Non-motion device control not covered by the above IDaqDevice interface will be described case by case in the following sections. However, they should all be able to fall under the KCSF device infrastructure and be designed with the generic IDevice interface. The IDevice interface allows a device to access the *get, set, execute* and *xxxMonitor* methods which should be able to support the full command set for each device.

4.5.2 Remote Power Control

Remote power control is used to power on / off devices remotely through COTS power controller device. The observatory currently uses two different supplied power controllers (Pulizzi and APC) both of which are serial device types. Serial device types would be custom implemented using KCSF's device/controller concepts.

Should a more customized solution based on solid state relays and digital controls be necessary, it would easily fit into KCSF's generic iDaqDevice interface.

4.5.2.1 **Inputs**

Execute command attributes:

- power on <port|# | all>
- power off <port# | all>

4.5.2.2 **Outputs**

Get command attributes:

- port status

4.5.3 Camera / Detector Control

There will be a number of cameras used in the NGAO system. Most will be controlled by AO Controls but there will be some that have control interfaces as part of the data readout interface, as is the case with Camera Link. For these cameras, the control functions will be implemented directly by the RTC, but the RTC is required to make specific control commands available to the NGAO control system for initiation of these functions.

For the other cameras and detectors, we will use the KCSF device infrastructure to implement a generic IDevice interface. The IDevice interface allows a device to access the *get, set, execute* and *xxxMonitor* methods which should support the full command set.

4.5.3.1 **Inputs**

Set command attributes:

Execute command attributes:

- power on
- power off
- set frame rate
- expose

4.5.3.2 **Outputs**

Get / monitor command attributes:

- status
- frame rate
- frame count

4.5.4 Video

Video servers will be used to monitor beam positions for use in alignments as well as a remote access tool to verify beam position. The Interferometer project currently uses AXIS video servers which does not need much control once it's configured via a web interface.

4.5.4.1 **Inputs**

Set command attributes:

Execute command attributes:

4.5.4.2 **Outputs**

Get / monitor command attributes:

- centroid position
- fwhm
- intensity
- image

4.5.5 RS-232 and USB

a goal is communicate with every controllable device via Ethernet. To accomplish this, some equipment will require a protocol converter. In the case of RS-232 and RS-422/485 devices, this "protocol converter" can be implemented by the use of a terminal server hooked up to the Eternet. The observatory currently uses a number of Lantronix terminal servers. Lantronix also produces a USB server if our only option for specific devices are USB interfaces. As of this writing, it is unclear what type of devices these would be and is thus difficult to design to.

4.5.5.1 **Inputs**

Set command asttributes:

Execute command attributes:

4.5.5.2 **Outputs**

Get / monitor command attributes:

5 AO System

5.1 Introduction

For the preliminary design we've chosen to diverge from the original concept presented in [KAON 569: NGAO System Design Report, Non-Real-Time Controls.]() We are proceeding with a more subsystem view approach to help us better visualize and map the control system to the way the hardware is actually laid out. the original concept had us treating each device as a standalone entity sitting off by itself. Software wise the same functionality will still be developed but by visualizing things from a system perspective it helps us to better understand how the pieces fit together and ultimately how the system will function as a whole. This new view is show in **Error! Reference source not found.** below.

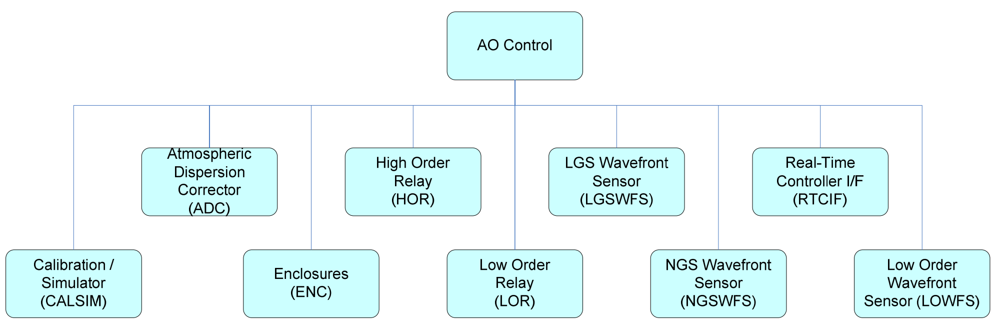

Figure 5: AO Control Block Diagram

You can consider each block as a high level control state machine that implements the complex coordinated control required by the devices in that subsystem. They accept high level commands from the AO sequencer and issue the appropriate commands to the low-level motion control blocks. The sequences will be implemented using a state machine compiler (SMC). The high level motion control sequences identified above are described in detail in the following sections.

5.2 Low Order Relay

5.2.1 Description

The Low-Order Relay covers the low-order atmospheric correction hardware, and miscellaneous first-relay devices. The Low-Order Relay subsystem is a logical grouping of these hardware devices, but is not actually implemented as a standard KCSF component due to the very different management and control requirements of the individual devices.

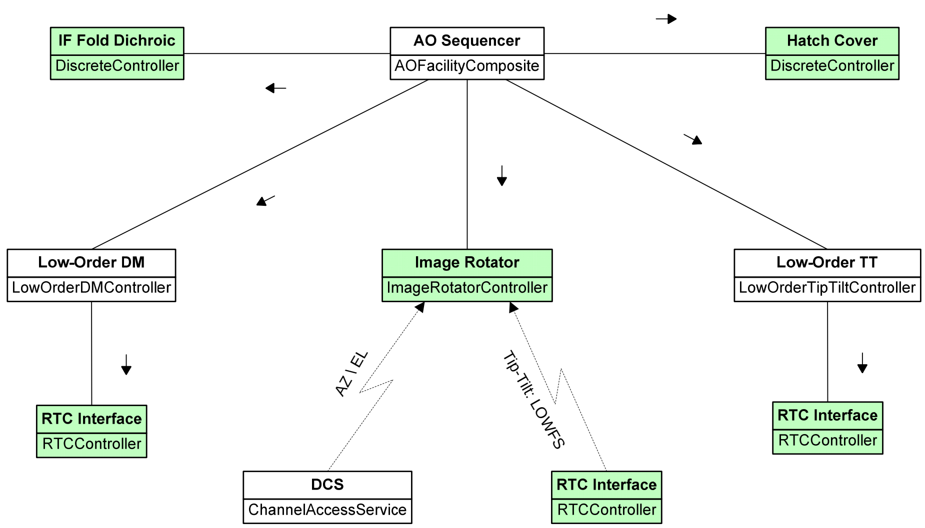

Figure 6: Low-Order Relay Collaboration Diagram

The low-order DM and TT stage are under the control and management of the real-time control system. The NGAO Controls system provides KCSF component interface classes to initialize and configure the RTC devices. During regular operations the operation of the DM and TT will be handled entirely by the RTC control loops.

The IF fold dichroic, hatch cover, and image rotator are managed by the NGAO Controls system. The dichroic and cover are simple discrete devices. The fold mirror feeds the interferometer DSM with light and is only used on IF nights. The hatch cover protects the AO system input and will typically be left open for the duration of the observing period. The image rotator is the only tracking device in the low-order relay under the NGAO control system management, and is responsible for maintaining a fixed-field or fixed pupil orientation with respect to the science instrument.

5.2.2 Control Loops

The following sections detail the NGAO controls loops implemented by the Low-Order Relay devices.

5.2.2.1 ***Image Rotation***

Due to the design of the Keck telescopes, as a target is tracked across the sky it will appear to rotate on the science instrument. Similarly, as the telescope elevation changes, additional rotation is added to the field and pupil due to the changing orientation of the tertiary mirror. The image rotator is designed to correct for both of these rotations to keep the field or pupil stable relative to the instrument.

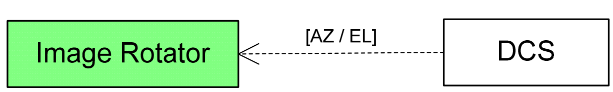

Figure 7: Image Rotator Control Loop

In fixed-field observing we want the field to remain at a fixed orientation relative to the detector. In this mode the rotator should track the position angle of the target (the apparent angle of rotation on sky) which will be constantly changing as the telescope tracks the target. While in fixed-field the pupil is allowed to rotate freely.

In fixed-pupil we want the pupil to remain at a fixed orientation relative to the detector. To maintain a stable pupil the rotator will track the telescope vertical angle. The vertical angle changes as the telescope slews in elevation. While in fixed-pupil mode the field is allowed to rotate freely.

*5.2.2.2 **LOWFS Rotation Correction***

During LGS observing the LOWFS sensors will be tracking the tip and tilt of the natural guide stars. From this information the system can determine the appropriate tip-tilt correction to apply to TTM1, and determine if there is any residual field rotation not corrected by the *Image Rotation* control loop.

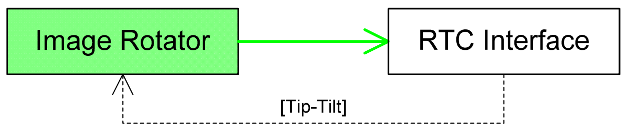

Figure 8: Field Rotation Offset Control Loop

The rotator will be listening to LOWFS tip-tilt telemetry from the RTC control interface, and will determine if any field rotation is seen by the sensors. The rotator will correct for field rotation, which will be seen by the LOWFS as centering the TT targets.

5.2.3 Devices

The following devices are managed under the Low-Order Relay.

*5.2.3.1 **Hatch Cover***

The hatch cover is a simple discrete device positioned at the input of the AO system. When the AO system is not use the hatch cover is closed. During observing the hatch must be open.

*5.2.3.2 **Image Rotator***

The image rotator is a 1-DOF rotational tracking device responsible for maintaining a fixed field / fixed pupil orientation as the telescope follows a target during observing. The rotator controller will be responsible for calculating the required position of the device based on telescope azimuth and elevation. The rotator will also receive rotation corrections based on outputs from the LOWFS RTC.

*5.2.3.3 **IF Fold Dichroic***

The IF fold dichroic is a beamsplitter fold mirror mounted on a discrete motion stage that is designed to feed the IR AO corrected beam to the interferometer. The visible and near-IR light is propagated to the NGS WFS for processing.

5.2.3.4 Low-Order Relay DM and TT
The low-order relay DM and TT are managed and controlled directly by the RTC system. The NGAO Controls system will be responsible for powering on the devices; loading reconstructors; setting properties; and initiating RTC servo loop commands.

5.3 Acquisition System
T.B.S

5.3.1 Description

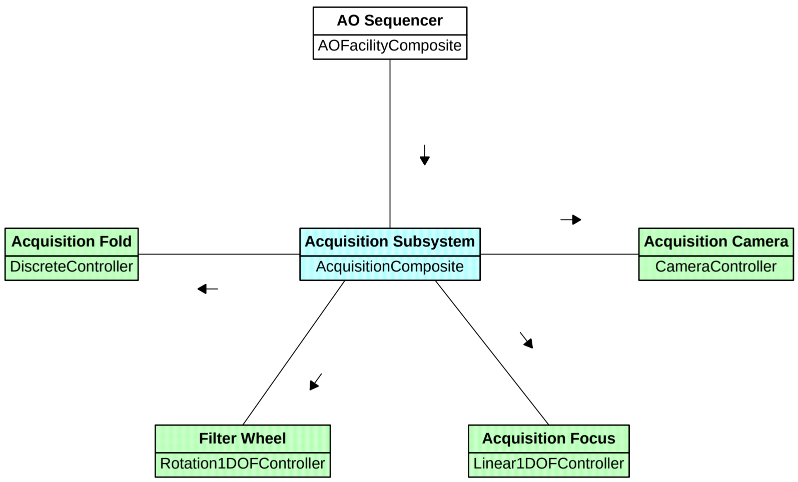

Figure 9: Acquisition Subsystem Collaboration Diagram

5.3.2 Interfaces

5.3.2.1 Inputs
T.B.S

5.3.2.2 Outputs
T.B.D

5.3.3 Control Loops
The Acquisition System does not implement any control loops.

5.3.4 Devices

5.3.4.1 ***Acquisition Fold***

T.B.S

5.3.4.2 ***Filter Wheel***

T.B.S

5.3.4.3 ***Acquisition Focus Stage***

T.B.S

5.3.4.4 ***Acquisition Camera***

T.B.S

5.4 LGS Wavefront Sensors

The Laser Guide Star (LGS) Wavefront Sensors (WFS) are used by the RTC to determine the appropriate corrections to apply to the low-order deformable mirrors (woofer and LOWFS) to correct for atmospheric turbulence.

There are a total of seven LGS WFS: four are used to perform tomographic correction; the remaining three are used to perform localized corrections for each of the three tip-tilt stars. The four tomographic WFS are stationary within the LGS assembly, and are used by the RTC to calculate the three dimensional atmospheric profile along the telescope seeing. The real-time corrections are applied directly to the low-order DM. The wide-field PnS WFS each have their own 2-DoF pickoff arm that allows the LGS subsystem to acquire the PnS LGS beacons from anywhere within the field. A caveat to this is that the pickoff arms can not cross the central 10 arcsecond of the field (science instrument FoV), and can not occlude any of the other pickoff arms. The wide-field WFS information is used to correct the atmosphere at each of the tip-tilt (2) and focus and astigmatism stars as seen by the LOWFS. The DMs for each tip-tilt star is located within the LOWFS assembly.

5.4.1 Description

The LGS Wavefront Sensor Subsystem is responsible for the management and control of the LGS wavefront sensors and related hardware. This includes the WFS pickoffs and LGS focus stage.

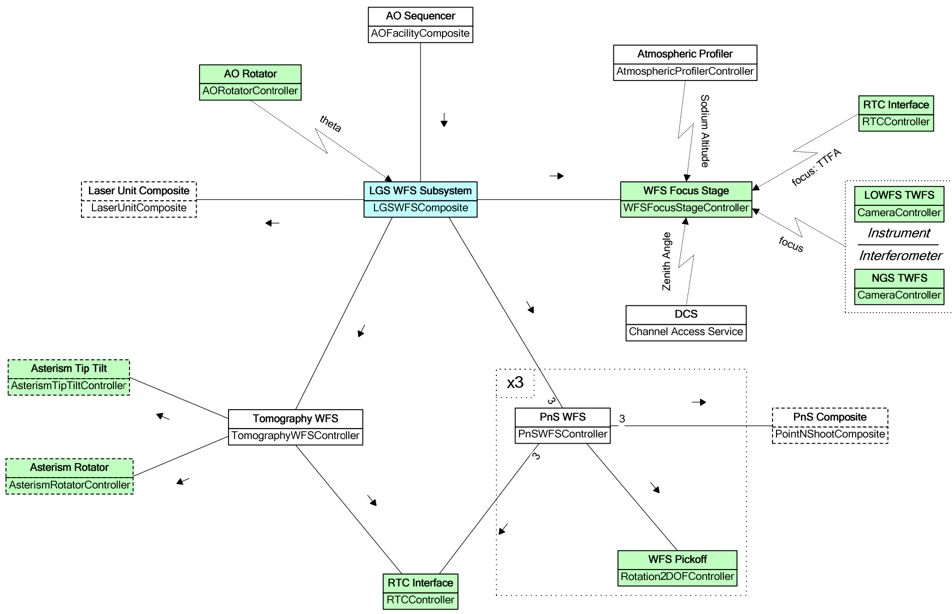

Figure 10: LGS WFS Collaboration Diagram

The LGS WFS Subsystem provides the initialization and acquisition functionality to position the wide-field pickoffs; align the PnS and tomography LGS beacons; configure the WFS RTC cameras; and manage the focus for the entire LGS assembly. The LGS WFS Subsystem also provides capabilities to perform background calibrations and respond to alignment errors by offloading to the laser facility up-link tip-tilt and asterism rotator.

5.4.2 Interfaces

The following sections detail the input and output interface to the LGS WFS subsystem controller.

*5.4.2.1 **Inputs***

The primary inputs to the LGS WFS composite are implemented as KCSF actions. Actions are defined through an attribute list and may contain parameters. The following define all of the recognized actions available to the WFS composite.

Home
T.B.S

Initialize
The *Initialize* action is responsible for preparing the LGS WFS subsystem and devices for observing.

SetMode
The *SetMode* action is responsible for configuring the LGS WFS Composite and subcomponents for a particular observing mode. As the LGS WFS are only used during LGS observing, the LGS WFS

Composite and subcontrollers will be disabled during NGS observing. The following configurations are used in laser guide star mode:

Fixed-Field LGS (Instrument)

- Enable WFS Focus Stage
- Enable Tomography WFS Controller
- Enable PnS WFS Controllers

Fixed-Field LGS (IF)

- Enable WFS Focus Stage
- Enable Tomography WFS Controller
- Disable PnS WFS Controllers

Fixed Pupil LGS (All)

- Enable WFS Focus Stage
- Enable Tomography WFS Controller
- Disable PnS WFS Controllers

The system must be configured each time the observing mode changes, and must be performed prior to target acquisition and observing.

Background

The *Background* action is responsible for initiating a WFS background calibration. The WFS Composite will open the RTC control loops (if currently tracking); tune the lasers off of the 589nm wavelength; initiate WFS background calibrations and tune the lasers back on.

This action can be performed at any time post-initialization, and can be performed as a system wide performance optimization during observing (this will interrupt the observing sequence). If performed during an observation the WFS RTC control loops will have to be closed through a call to *Acquire* when complete.

Acquire

The *Acquire* action is responsible for aligning the LGS asterism with the WFS, and closing the AO and offload control loops. There are a total of seven WFS: four tomography sensors managed under the *TomographyWFSController*; and three point-and-shoot sensors, each managed by a *PnSWFSController*. Alignment of the LGS beacons will be performed with the WFS as the reference, by commanding the BGS tip-tilt, rotator and point-and-shoot arms. The WFS were selected as the preferred reference because the pickoff devices should be positioned with higher precision and will not be affected by telescope flexure, unlike the BGS which is mounted on the telescope secondary socket.

During the LGS acquisition process the WFS Composite will be provided with the telescope pointing target and the positions of up to three laser beacons. The composite will be responsible for converting these targets into image plane coordinates, taking into account telescope and rotator orientation, and calculating the required positions of the WFS pickoffs. After the devices have been positioned the device controllers will be commanded to acquire their respective laser beacons, and if successful, enter track to close the RTC and offloading control loops.

Halt

The *Halt* action is responsible for opening the control loops and halting the WFS. Motion devices such as the pickoff arms and focus stage will be powered down and RTC will be told to open the AO control loop and idle the sensors. The devices and control loops can be reacquired with a call to *Acquire*.

Recover

The *Recover* action is a state dependent process that is responsible for recovering the WFS Composite and controllers in the event of a fault. During observing if the pickoff arms, focus stage, or WFS sensors enter a fault state observing may be interrupted. Rather than requiring a full subsystem initialization to clear the fault – consequently ending the science sequence – a recover can be issued, which will selectively reinitialize the faulted devices only. If the fault can be cleared the device(s) will be put back into its previous state prior to the fault, and observing should be able to continue, uninterrupted.

Disable

The *Disable* action will power down the RTC WFS and devices, and put the controllers into an idle (low network / resource usage) state. Disable is typically performed at the end of a nights observing or when the device is no longer need, for example when switching to NGS mode. While DISABLED the controller will not be publishing telemetry nor monitoring and responding to system events. A *Home* must be performed to re-enable the device and controller for use. Disable can only be performed when the controller is in a halted state.

5.4.2.2 **Outputs**

T.B.D

5.4.3 Control Loops

The following sections detail the tracking and offload control loops of the LGS WFS subsystem. Each control loop is implemented in part of in full by one or more of the NGAO controllers.

5.4.3.1 ***WFS Focus***

As a tracking device, the WFS Focus Stage Controller will be monitoring the state of the system and responding to events. Focus information is derived from the atmospheric conditions, LOWFS calculated focus, and the telescope zenith angle.

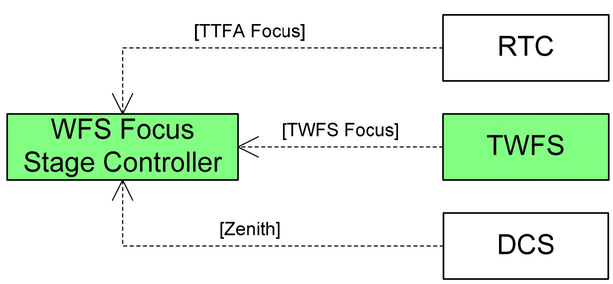

Figure 11: WFS Focus Control Loop

WFS focus tracking is an event driven control loop triggered by telemetry events from the RTC, the TWFS camera, and DCS telescope zenith angle keywords. The RTC provides time-averaged data from the Tip-Tilt Focus and Astigmatism (TTFA) WFS to determine the approximate height of the sodium layer. More accurate focus information is provided at a slower rate (~1Hz) directly by the Truth Wavefront Sensor (TWFS). The TWFS images will be processed by the NGAO controls system to determine focus and centroid offsets and published as telemetry. When the controller receives input from these items it will calculate the LGS WFS focus compensation and apply it to the stage.

5.4.3.2 ***LGS Offloading***

During observing the LGS WFS will periodically have to offload tip-tilt errors to the Beam Generation System to maintain alignment of the LGS beacons. The tip-tilt offload loop is different for the tomography and point-and-shoot (PnS) WFS.

Tomography WFS

The Tomography WFS Controller uses a single offload task to manage the tip-tilt offload for all four WFS. Offloads are performed when the average of all the tomography tip-tilt stages exceeds a specified threshold. This task is responsible for maintaining a balanced tip-tilt average across the entire asterism.

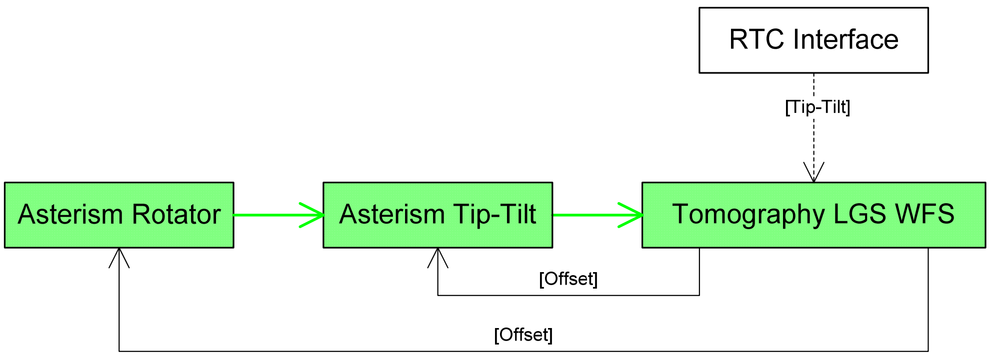

Figure 12: Tomography LGS WFS Offloading

Tip-tilt information from each of the WFS TT mirrors is provided through the RTC telemetry, and is used to calculate average tip-tilt across the tomography sensors. The offload task will determine the offset to apply to the BGS up-link Tip-Tilt and Rotator for asterism shift and rotation, respectively. The BGS offsets will be seen by the WFS as rapid motion away from the increasing tip-tilt. This will result in the RTC compensating by moving the WFS TT mirrors towards their zero point.

Point-and-Shoot WFS

Each of the PnS WFS Controllers implements their own tip-tilt offload control loop. This loop is responsible for periodically offloading the accumulated tip-tilt of the WFS TT mirror (the mirror itself is controlled by RTC) to the BGS PnS steering arm.

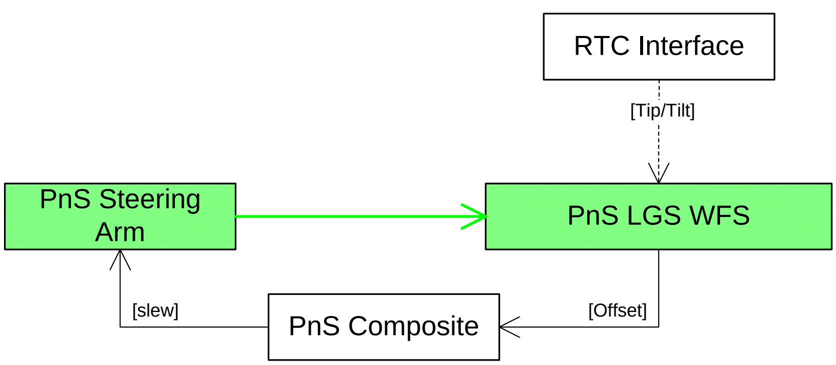

Figure 13: Point-and-Shoot LGS WFS Offloading

When the RTC WFS loop is closed the real-time system will be using the WFS tip-tilt mirror to keep the LGS spot stable on the sensor. Over time the tip-tilt will build up and if not offload can cause the TT mirror to reach its limit, and eventually loose the LGS beacon all together. When the tip-tilt exceeds a defined threshold, the LGS WFS Controller will calculate the offset required by the steering arm to bring the beacon closer to true center (zero tip-tilt). The WFS will see this as the beacon moving rapidly out of center in the opposite direction of drift, and will compensate by moving the TT mirror to bring the spot back to center. This should consequently return the TT mirror to near its center of range of motion.

5.4.4 Devices

The following devices are managed by the LGS WFS subsystem. Specific details, interfaces, and discussions are available in their related design documents.

*5.4.4.1 **Focus Stage***

All seven LGS WFS are mounted on a rigid assembly. Focus for the wavefront sensors are achieved by moving the entire assembly along the z-axis with a linear 1-DOF focus stage. The required position of the stage is based on focus measurements determined by the TTFA LOWFS, and at a slower rate by the TWFS. The TTFA measurements are produced by RTC, while the TWFS focus control loop is implemented directly by the NGAO Controls system.

*5.4.4.2 **Pickoff Arms***

The three wide-field Point-and-Shoot LGS wavefront sensors utilize a 2-DOF rotational pickoff arm to acquire their LGS beacons from the field. The pickoff arms are based on a theta-phi motion control design, allowing each WFS to pickoff a target anywhere in the field. Although the three pickoff arms are stacked along the optical axis to prevent physical collision with each other, motion control and positioning requirements exist to prevent vignetting of the science field and ensure the safety of the system.

*5.4.4.3 **LGS WFS***

Although the seven LGS WFS perform different functions in the NGAO system, they share identical hardware and similar control interfaces. The wavefront sensors are under the direct control of the RTC subsystem. The NGAO Controls system is responsible for initializing and configuring the WFS cameras, and commanding RTC to open or close the control loops. All four tomography sensors are managed by *TomogarphyWFSController*. As there are no operating modes where only a portion of the tomography WFS would be active, the Controller exposes a single abstract interface to command all four sensors simultaneously. The three PnS wavefront sensors are each represented by an instance of the *PnSWFSController*. The PnS Controller interface is similar to the tomography controller, but adds additional functionality for slewing the pickoff arms and acquiring the wide-field LGS beacons.

5.5 Low Order Wavefront Sensors

To correct for atmospheric turbulence you need to know how the light is distorted as it travels through the atmosphere. Local turbulence can be imaged by each of the LGS WFS and applied to the first relay DM. The LGS WFS alone though do not provide information about the focus of the system, nor can they be used to detect the overall field tip and tilt. For this, natural guide stars are needed to calculate the tip-tilt caused by the atmosphere, and the required focus of the AO system. This information and functionality is provided by the low-order wavefront sensor (LOWFS) system.

5.5.1 Description

The Low Order Wavefront Sensor subsystem is responsible for the management and configuration of the low order wavefront sensors (LOWFS) and related hardware. The LOWFS are responsible for providing tip-tilt and focus information to the RTC and AO Controls system. RTC will apply corrections to the low-order relay TTM to correct for full-field tip and tilt. Focus and rotation offloading will be managed by the AO Controls and will be applied to the LGS focus stage and AO rotator.

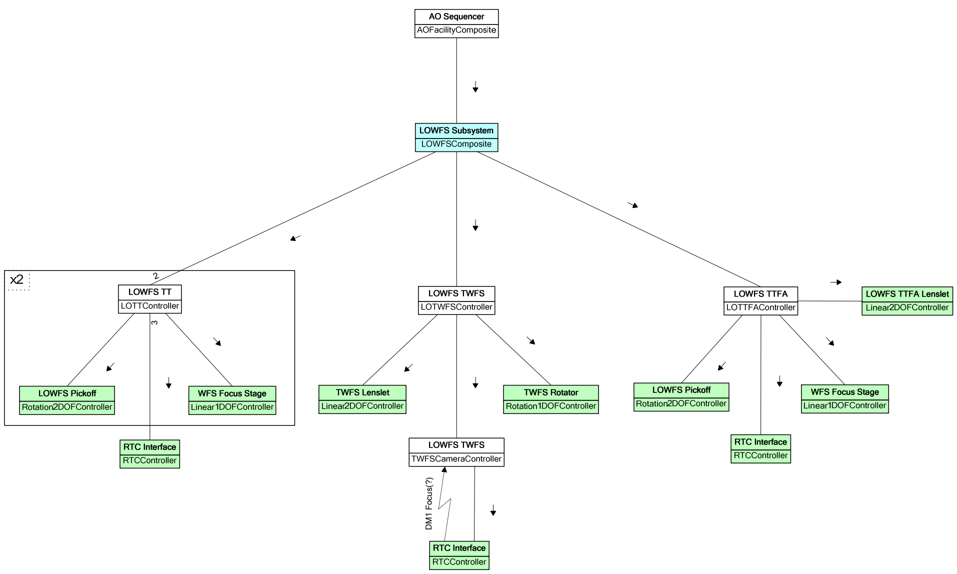

Figure 14: Low-Order WFS Collaboration Diagram

There are a total of three LOWFS assemblies, each with a dedicated deformable mirror, pickoff arm, focus stage, tip-tilt mirror and sensor. In addition to tip and tilt, one of the LOWFS sensors also provides focus and astigmatism information (TTFA) to the RTC. Co-mounted with the TTFA is a truth wavefront sensor (TWFS). This sensor provides more accurate focus information, but at a slower rate than the TTFA. The TWFS sensor is under the direct management and operation of the AO Control system.

All three LOWFS and the TWFS are managed by the *LOWFSComposite*. This NGAO component provides a simple interface to initialize, acquire, and manage the devices during observing. The following sections detail the LOWFS Composite.

5.5.2 Interfaces

The following sections detail the input and output interface to the LOWFS subsystem controller.

5.5.2.1 __Inputs__

The primary inputs to the LOWFS composite are implemented as KCSF actions. Actions are defined through an attribute list and may contain parameters. The following define all of the recognized actions available to the LOWFS composite.

Home

The *Home* action is responsible for creating network connections, powering on devices, and initiating the homing sequences. The home action is a larger process of system initialization and will automatically perform and *Initialize* when complete.

Initialize

The *Initialize* action is responsible for preparing the LOWFS subsystem and devices for observing. This includes setting default RTC sensor parameters; initializing the pickoff arms, focus stages, and lenslet stages; and refreshing component and telemetry connections.

SetMode

The *SetMode* action is responsible for configuring the system and devices for observing in a particular mode. The LOWFS hardware is only used when there are laser corrected tip-tilt stars, which exists only in Fixed Field LGS instrument observing (non-IF). The LOWFS composite and subcontrollers will be disabled during fixed pupil, fixed field IF, and NGS observing.

The system must be configured each time the observing mode changes, and must be performed prior to target acquisition.

Acquire

The *Acquire* action is responsible for aligning the NGS tip-tilt targets with the LOWFS, and closing the AO control loops. There are a total of three LOWFS: two tip-tilts, implemented by the *LOTTController*; and one TTFA, implemented by the *LOTTFAController*. All three LOWFS controllers share the same acquisition sequence and are treated identically from the perspective of the LOWFS Composite.

During the NGS acquisition process the LOWFS Composite will be provided with the telescope pointing target and the positions of up to three tip-tilt targets. The composite will be responsible for converting these targets into image plane coordinates, taking into account telescope and rotator orientation, and calculating the required positions of the LOWFS pickoffs. After the devices have been positioned the controllers will be commanded to acquire their respective stars. This will peak-up NGS alignment and determine the appropriate seeing parameters to apply to the RTC cameras. When the acquisition completes a background will be performed and the controllers will be put into track to close the RTC control loops.

Background

The *Background* action is responsible for initiating a sequential background calibration of all three LOWFS, and can only be performed after the LOWFS have been acquired. Essentially, this action exists to perform servo loop optimizations while observing without interrupting the science sequence. The following actions are performed for each LOWFS controller:

1. Issue a *Background* command. This will open the control loops, offset the pickoff arm, take a background over the region of interest, and reacquire the NGS.
2. Issue a *Track* command. This will close the LOWFS control loops.

This action can be performed at any time post-acquisition, and is typically performed as a system wide performance optimization during observing. If new frame rates are required they should be specified at the LOWFS Controller level before initiating a background. If only one or two LOWFS need to be recalibrated the background should be done individually, rather than being performed from the LOWFS Composite level. Performing a background during observing should be done sequentially to avoid having to interrupt the observing sequence.

Halt
The *Halt* action is responsible for opening the control loops and halting the LOWFS. Motion devices such as the pickoff arms and focus stages will be powered down and RTC will be told to open the AO control loop and idle the sensors. The devices and control loops can be reacquired with a call to *Acquire*.

Recover
The *Recover* action is a state dependent process that is responsible for recovering the LOWFS Composite and controllers in the event of a fault. During observing if the pickoff arms, focus stages, or LOWFS sensors enter a fault state observing may be interrupted. Rather than requiring a full subsystem initialization to clear the fault – consequently ending the science sequence – a recover can be issued, which will selectively reinitialize the faulted devices only. If the fault can be cleared the device(s) will be put back into its previous state prior to the fault, and observing should be able to continue, uninterrupted.

Disable
The *Disable* action will power down the RTC LOWFS and devices, and put the controllers into an idle (low network / resource usage) state. Disable is typically performed at the end of a nights observing or when the device is no longer need, for example when switching to NGS mode. While DISABLED the controller will not be publishing telemetry nor monitoring and responding to system events. A *Home* must be performed to re-enable the device and controller for use. Disable can only be performed when the controller is in a halted state.

5.5.2.2 Outputs

T.B.D

5.5.3 Control Loops

The following detail the control loops managed by the LOWFS subsystem.

5.5.3.1 TWFS Focus and Centroiding

The TWFS Controller will be responsible for processing images to determine high accuracy focus and centroiding errors. A Shack-Hartmann technique is used to estimate the wavefront and calculate the overall focus of the light and its average centroid. The focus information will be published to the LGS WFS focus stage, and centroid information will be sent to RTC to improve AO corrections.

5.5.4 Devices

The following details all of the devices managed by the LOWFS subsystem.

5.5.4.1 LOWFS Pickoffs

Each LOWFS sensor utilizes a 2-DOF rotational pickoff arm to acquire the NGS targets from the field. The pickoff arms are based on a theta-phi motion control design, allowing each LOWFS to pickoff a target anywhere in the field. Although the three pickoff arms are stacked along the optical axis to prevent physical collision with each other, motion control and positioning requirements exist to prevent vignetting of the science field and ensure the safety of the system.

5.5.4.2 Focus Stages

Each LOWFS unit is mounted onto a 1-DOF linear focus stage. Focus for each LOWFS must be conjugate to the science instrument and is affected by the position of the TT target in the field; optics in the science path; and the configuration of the instrument itself. As the focus for the science instrument shouldn't change during observing the LOWFS focus position should only need to be set at the beginning of a sequence. If optics any hardware is added, removed, or reconfigured in the beam train that may affect the focus the LOWFS focus position will need to be recalculated and applied.

5.5.4.3 Lenslet Array

The TTFA and TWFS LOWFS utilize a lenslet array to determine the structure of the arriving wavefront, after low-order corrections by the DM. As the shape of the DM is based on the atmospheric turbulence seen by the LGS WFS, any remaining aberrations in the wavefront indicate what was not able to be removed by the low-order correction. This includes field tip-tilt and LGS focus errors. RTC will directly calculate the focus error as seen by the TTFA. The NGAO Controls system will be responsible for processing the TWFS images to determine focus errors. The RTC and TWFS controller will publish the errors, which will be used by the LGS WFS focus tracking control loop.

5.5.4.4 TWFS Camera

Unlike the other LOWFS sensors, the TWFS is under the direct control and management of the NGAO Control system. The TWFS Controller implements the functional control logic for the sensor, and exposes a simple interface to initialize, configure, and control TWFS image processing. The controller is responsible for providing the slow-rate focus and centroid calculations to RTC to optimize system performance.

5.5.4.5 **TWFS Rotator**

T.B.S

Comment [dm1]: I'm guessing this is to maintain the same orientation of the spot as seen by either the TTFA or LGS WFS, but I'm not sure.

5.6 High Order Relay

The High Order Relay subsystem covers the devices responsible for applying the high order corrections to the science field. These include the MEMS DM and high-order TT stage. The DM is mounted to the TT stage, and together controls the propagation of the field to the NGS WFS and science instrument. The TT stage will be used to perform slow-rate TT / field alignment corrections, and can be used to perform dithering of the science field.

5.6.1 Description

There are two devices that comprise the High-Order Relay: the high-order DM and TT mirror. The DM is under the direct control and management of the RTC, and is used to correct for the remaining atmospheric distortion that could not be removed by the Low-Order relay DM. The high-order tip-tilt device (TTM2) is under the direct control of the NGAO controls system, and acts as the mount for the MEMS DM.

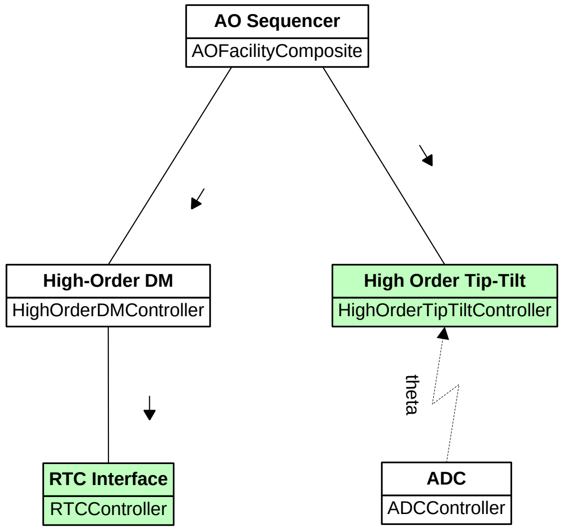

Figure 15: High-Order Relay Collaboration Diagram

As a low-bandwidth device, TTM2 is used primarily for tracking of differential atmospheric refraction (DAR); correcting ADC tip-tilt errors; and offloading for the Keck Angle Tracker (KAT). In addition, TTM2 will be able to execute science dithering routines, either as scripts or built in sequences, to move the field in a coordinated fashion.

5.6.2 *Control Loops*

The following identifies the control loops defined for the High-Order Tip-Tilt controller.

5.6.2.1 ***ADC Tip-Tilt Errors***

It is expected that the Atmospheric Dispersion Compensator (ADC) will cause position translation errors in the science image. Fortunately these errors can be compensated for by the High-Order Tip-Tilt device.

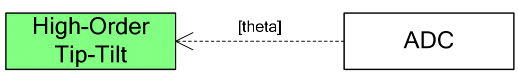

Figure 16: ADC Tip-Tilt Error Correction Control Loop

The High-Order Tip-Tilt will listen to position events from the ADC, and using a table of premeasured tip-tilt error values find the appropriate compensation to apply to the stage.

5.6.3 Devices

5.6.3.1 ***High-Order Tip-Tilt stage***

The High-Order Tip-Tilt stage is implemented as a 2-DOF tip-tilt device.

5.6.3.2 ***High-Order Relay DM***

The high-order relay DM is managed and controlled directly by the RTC system. The NGAO Controls system will be responsible for powering on the device; loading reconstructors; setting properties; and initiating RTC servo loop commands.

5.7 NGS Wavefront Sensor

Just after the high-order relay is the natural guide star WFS system. This system is responsible for calculating the wavefront, tip-tilt, and focus of a guide star in NGS mode, and applying the corrections to the low and high-order relays (depending on instrument). The NGS subsystem can also provide truth and focus information to the Real-Time Control system during laser guide star observing, which is required by the Interferometer.

5.7.1 Description

The NGS WFS Subsystem is implemented by the *NGSWFSComposite*, and is responsible for the management and operation of the natural guide star (NGS) devices. This includes the NGS WFS, focus stage, lenslet array, and field steering mirrors. The Composite presents a simple high level interface to the client to initialize, configure, and acquire guide stars from the field. The NGS WFS Composite is managed directly by the AO Sequencer subsystem.

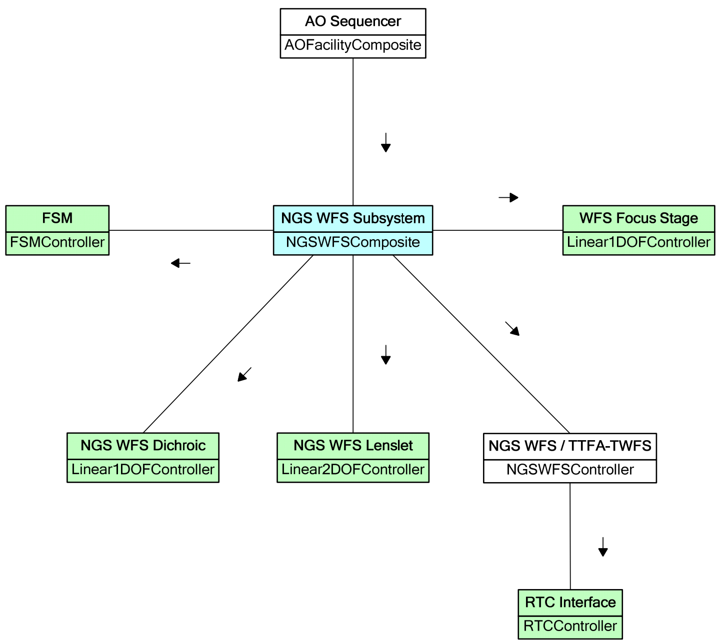

Figure 17: NGS WFS Collaboration Diagram

The NGS WFS is under the direct control of the real-time system. The NGAO controls system is responsible for the initialization and configuration of the sensor, and commanding RTC to open and close the WFS control loops. The other NGS devices are all under the direct control of the Controls System. Outside of the RTC NGS control loops, the NGS Subsystem does not implement any tracking or offload tasks. The majority of device control is performed during configuration and acquisition.

5.7.2 *Interfaces*

The following sections detail the input and output interface to the NGS WFS Subsystem.

5.7.2.1 ***Inputs***

The primary inputs to the NGS WFS Composite are implemented as KCSF actions. Actions are defined through an attribute list and may contain parameters. The following define all of the recognized actions available to the NGS WFS Composite.

Home

The *Home* action is responsible for connecting to the hardware and NGAO components; powering on devices; and performing home sequences. The home sequence will cause the motion control devices to find their home switch and reset their device position to zero. All motion afterward will be relative to the home position.

Initialize

The *Initialize* action is responsible for preparing the NGS WFS Composite subsystem for observing. This involves refreshing network and telemetry connections and returning to the home position.

SetMode

The *SetMode* action is responsible for configuring the NGS Subsystem for a particular observing mode. The NGS hardware is used for all NGS observing modes and LGS with the interferometer. In LGS IF mode (fixed-field and fixed-pupil) the NGS WFS switches into a TTFA / TWFS mode to provide focus information to the LGS WFS focus stage. For all active NGS WFS modes the dichroic beamsplitter will be inserted on-beam. In LGS with an instrument the dichroic will be removed. The NGS subsystem is not used during LGS observing with an instrument, and will be disabled.

The system must be configured each time the observing mode changes, and must be performed prior to target acquisition and observing.

Focus

The *Focus* action is responsible for determining and applying the required focus compensation for the NGS WFS sensor conjugate to the science instrument focal plane. The actual focus position is based on the optics in the science path, and the properties of the science instrument. For any hardware / system configuration change (such as inserting or removing optics, filters, etc.) the NGS WFS focus should be recalculated and applied.

Acquire

The *Acquire* action is responsible for positioning the field steering mirrors to acquire the specified NGS target. The Acquire action accepts a set of coordinates for a target in the field, and will compute the relative position of the target with respect to the optical axis. This information will be sent to the FSM to position the target on the WFS. After peak-up the RTC control loops will be closed on the target.

Zero

The *Zero* action removes any offset applied to the FSMs through the *Offset* attribute. The *Zero* action can be executed from any state.

Halt

The *Halt* action is responsible for opening the RTC control loop and taking the controller out of the TRACK state. Halt can be issued from any of the post-initialization states.

Recover

The *Recover* action is a state dependent process that is responsible for recovering the NGS WFS Controller in the event of a fault. During observing if the sensor enters a fault state observing may be interrupted. Rather than requiring a full subsystem initialization to clear the fault – consequently ending the science sequence – a recover can be issued, which will selectively reinitialize the faulted device. If the fault can be cleared the device will be put back into its previous state prior to the fault, and observing should be able to continue, uninterrupted.

Disable

The *Disable* action will power down the device and put the software controller into an idle (low network / resource usage) state. Disable is typically performed at the end of a nights observing or when the device is no longer need, for example when switching to NGS mode. While DISABLED the controller will not be publishing telemetry nor monitoring and responding to system events. A *Home* must be performed to re-enable the device and controller for use. Disable can only be performed when the controller is in a halted state.

*5.7.2.2 **Outputs***

T.B.D

5.7.3 Control Loops

The NGS WFS subsystem does not directly manage any tracking control loops.

5.7.4 Devices

The following details all of the devices managed by the NGS WFS subsystem.

*5.7.4.1 **Dichroic***

The NGS dichroic is a simple linear 1-DOF device that is responsible for splitting the NGS WFS visible light from the science field to the sensor. During NGS and LGS inteferometry the dichroic will be on-beam. For LGS with an instrument the WFS is not used, and the dichroic will be positioned off-beam.

*5.7.4.2 **Focus Stage***

The NGS focus stage is a linear 1-DOF device that is responsible for maintaining the focus of the NGS WFS. The focus of the WFS is based on the properties of the system and instrument. Once the system is configured the focus should not have to be updated. If any new optics are inserted or removed from the beam path, or the instrument properties are changed, the focus may have to be recalculated.

*5.7.4.3 **Field Steering Mirrors***

The field steering mirrors are a pair of coordinate 2-DOF tip-tilt fold mirrors designed to shift the field to change the apparent view of the WFS NGS. This is the reverse concept of using a pickoff arm to select a point in the field: instead you move the field to acquire the target on the WFS. The FSMs will be able to shift the field ± 20 arcseconds to acquire any target in the science field.

5.7.4.4 ***Lenslet Array***

The lenslet array is a linear 2-DOF device used in producing Shack-Hartmann wavefront calculations. This WFS system allows the RTC to determine focus, tip-tilt, and astigmatism for the natural guide stars. The lenslet array will be positioned based on pre-observing calibration tests determined to produce an optimal flat wavefront, with minimal focus and tip-tilt errors.

5.7.4.5 ***NGS WFS***

The NGS WFS is implemented by the *NGSWFSController* and provides an interface to the RTC NGS wavefront sensor. The controller provides all of the functionality necessary to initialize, configure, and acquire the WFS. The NGS WFS Composite will be responsible for managing the NGS WFS controller during regular observing.

5.8 Atmospheric Dispersion Corrector

T.B.S

5.8.1 *Description*

The *ADCComposite* is responsible for the management and operation of the ADC subsystem. The subsystem comprises the ADC device and in-out stage.

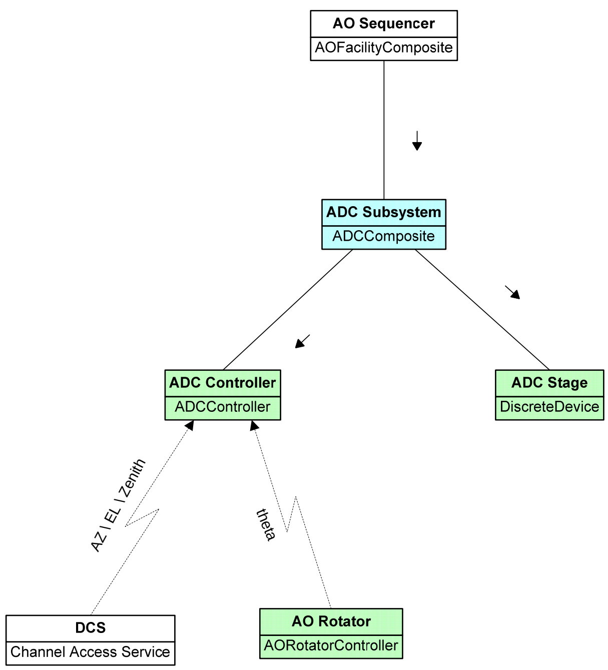

Figure 18: ADC Collaboration Diagram

5.8.2 *Interfaces*

The following sections detail the input and output interface to the ADC subsystem controller.

5.8.2.1 ***Inputs***

The primary inputs to the ADC composite are implemented as KCSF actions. Actions are defined through an attribute list and may contain parameters. The following define all of the recognized actions available to the ADC composite.

Home

The *Home* action is responsible for connecting to the subcontrollers and homing the devices. The home process involves slewing the ADC devices until they encounter their home switch. The devices will then move off the switch a specified distance and set the encoder position to zero. All device motion will be made relative to the homed position.

Initialize

The *Initialize* action is responsible for initializing the ADC composite and controllers. During the initialization process the network and telemetry connections will be refreshed and the ADC device and stage will be sent to their zero position (defined by the *Home* process.) Once complete the ADC subsystem will be ready for observing.

SetMode

The *SetMode* action is responsible for configuring the ADC Composite for observing in a particular mode. The ADC is used during all instrument observing modes (NGS, LGS, fixed-field, and fixed-pupil), and will be disabled during IF observing. During instrument observing the ADC will automatically be placed on beam.

The system must be configured each time the observing mode changes, and must be performed prior to target acquisition.

Track

The *Track* action will close the ADC control loops and put the Controller in the track state. While in the track state the ADC will be actively responding to system events from DCS and the AO Rotator to correct for the expected atmospheric dispersion. The dispersion correction values will be based on theoretical and empirical tests, and maintained in a look-up table. The ADC tracking loop will index this table to compute the appropriate correction to apply.

Halt

The *Halt* action is responsible for halting the ADC and stage controllers, and taking the composite out of the TRACK state. Halt can be issued from any of the post-initialization states.

Recover

The *Recover* action is a state dependent process that is responsible for recovering the ADC Composite and controllers in the event of a fault. During observing if the ADC device enters the fault state, dispersion affects will begin to increase. Although this will not require that the observing sequence be interrupted, it will continue to degrade performance and data quality as the sequence continues.

In the event of a fault a recover can be issued to selectively re-initialize the device and attempt to clear the error. If the error can be cleared the device will return to its original state before the fault occurred. If the fault can not be cleared the device will remain in the fault state, but the composite will still be able to receive commands. If the client attempts to command the faulted device through the composite the composite will in turn enter its fault state, requiring a response from the operator.

Disable

The *Disable* action will power down the device and put the software controller into an idle (low network / resource usage) state. Disable is typically performed at the end of a nights observing or when the device is no longer need, for example when switching to NGS mode. While DISABLED the controller will not be publishing telemetry nor monitoring and responding to system events. A *Home* must be performed to re-enable the device and controller for use. Disable can only be performed when the controller is in a halted state.

5.8.2.2 Outputs

T.B.D

5.8.3 Control Loops

The following detail all of the tracking control loops implemented by the ADC subsystem.

5.8.3.1 ADC Tracking

As a tracking device the ADC Controller will be listening to system event and responding automatically to changes. The ADC dispersion corrector control loop takes as input the telescope zenith angle, zenith direction and rotator orientation. (The science instrument wavelength is also needed, but this is set before the start of a sequence.)

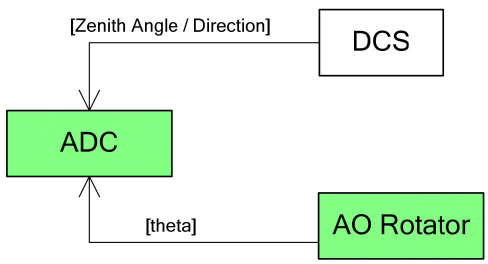

Figure 19: ADC Tracking Control Loop

The input values are used to index a table of premeasured dispersion values and combined to determine the appropriate correction to apply with the ADC. The ADC correction loop is event driven and can potentially run at the rate of the fastest telemetry item. Decimation can be used to reduce the correction rate if appropriate.

5.8.4 Devices

The following lists all of the devices managed by the ADC Subsystem.

5.8.4.1 ADC Stage

The ADC stage is a simple discrete device that will insert or remove the ADC device from the beam path. The position of the stage is implicitly assumed by the current observing mode selection. If the user wishes to manually insert or remove the stage the device controller can be addressed and commanded directly.

5.8.4.2 ***Atmospheric Dispersion Corrector***

The Atmospheric Dispersion Corrector is implemented by the ADCController, a subclass of the NGAO *HybridController*. This device contains a rotational and linear DOF allowing the client to precisely position the ADC in the beam path and control the degree of correction applied. Although the controller allows clients to explicitly position the device for the most part this device will simply be run in closed loop tracking mode, automatically applying correction as required.

5.9 RTC Interface

T.B.S

5.9.1 *Description*

5.9.2 *Interfaces*

5.9.2.1 ***Inputs***

5.9.2.2 ***Outputs***

5.9.3 *Control Loops*

5.9.4 *Devices*

5.10 External Offloading

T.B.S

5.10.1 *Description*

The External Offloading subsystem manages the software controllers responsible for performing the telescope and tip-tilt offload control loops.

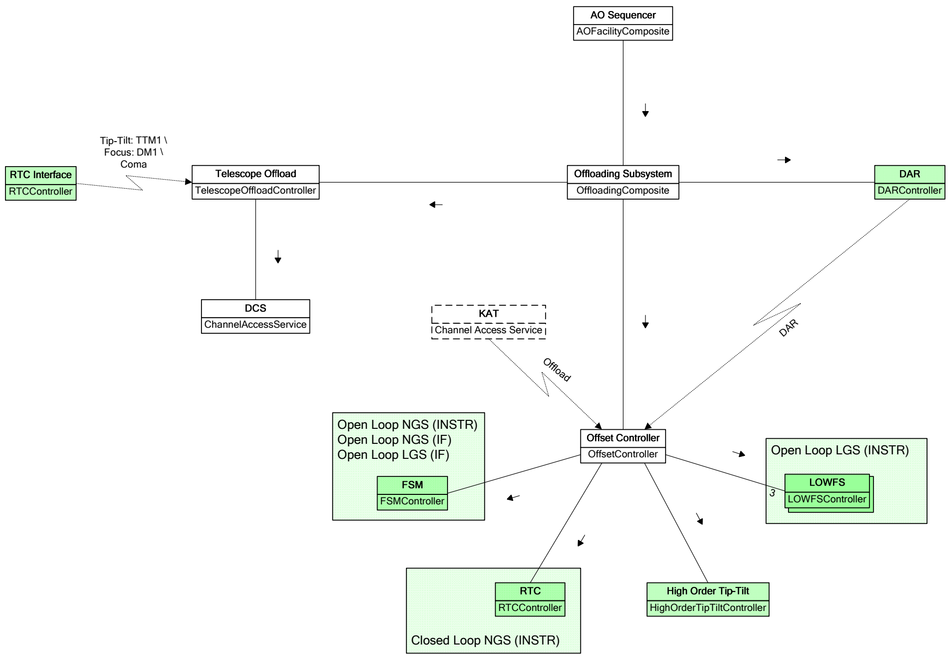

Figure 20: External Offloading Collaboration Diagram

5.10.2 Interfaces

The following sections detail the input and output interface to the External Offloading subsystem controller.

5.10.2.1 ***Inputs***

The primary inputs to the External Offloading subsystem are implemented as KCSF actions. Actions are defined through an attribute list and may contain parameters. The following define all of the recognized actions available to the Offloading subsystem components.

Home

The *Home* action is responsible for configuring the KCSF services, connecting to NGAO components, and establishing the telemetry listeners. As the devices involved in the AO Offset control loop are managed by external subsystems the controller does not need to perform any device configuration or initializations.

Initialize

The *Initialize* action is responsible for refreshing connections to distributed components and telemetry. No device level initializations are required.

SetMode

The AO Control System will be receiving offsets and offloads from a number of sources and is responsible for calculating the appropriate offsets to apply to specific devices. Each observing mode has its own unique offset requirements. The *SetMode* action is responsible for setting the required input and output streams for the AO Offset and Telescope Offload Controller. While in track mode, these controllers will read from the input streams and send offloads to the defined output devices.

Track

The *Track* action is responsible for starting the offloading tasks, and represents the bulk of the subsystems functionality. While tracking, the controller will be listening to events from the Differential Atmospheric Refraction (DAR) controller, the Keck Angle Tracker (KAT; IF observing only), and RTC. Offsets will be calculated based on these input values, and output to the relevant device and telescope.

The composite and controllers will stay in track until the loop is explicitly opened through a call to *Halt.*

Halt

The *Halt* action is responsible for opening the AO Offset and Telescope Offload control loops and taking the subsystem out of the TRACK state. Halt can be issued from any of the post-initialization states.

Recover

The *Recover* action is designed to selectively reinitialize a component in the event of a system fault during observing. As the External Offload Subsystem is comprised of simple software controllers, and does not directly manage any hardware, a recover has essentially the same functionality as performing an initialization.

Disable

The *Disable* action will put the software controllers and subsystem into an idle (low network / resource usage) state. While DISABLED the components will not be publishing telemetry nor monitoring and responding to system events. A *Home* must be performed to re-enable the components for use. Disable can only be performed when the subsystem is in a halted state.

5.10.2.2 Outputs

T.B.D

5.10.3 Control Loops

The Offloading ubsystem does not directly manage any tracking control loops.

5.10.3.1 Telescope Offload

The *TelescopeOffloadController* is responsible for offloading tip-tilt, focus, and coma from the low-order relay to the telescope. During observing focus and field tip-tilt will be corrected by the low-order DM and TTM1 stage, respectively. Periodically these quantities will need to be offloaded to the telescope to ensure the AO devices have sufficient range to correct for the atmosphere. Tip-tilt is offloaded to the telescope AZ / EL position of the telescope, and focus / coma is offloaded to the secondary piston.

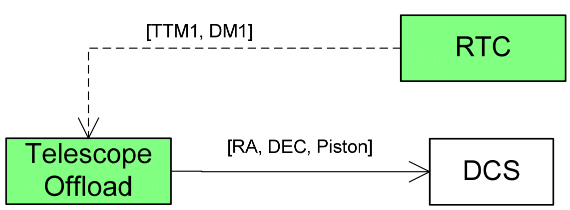

Figure 21: Telescope Offload Control Loop

The *Track* action is responsible for starting the telescope offload task, and represents the bulk of the controller's functionality. While tracking, the controller will be listening to events from the low-order relay DM and Tip-Tilt Mirror (TTM). If the controller determines that the focus applied to the DM or the position of the TTM exceed configurable thresholds, the controller will take action to offload the buildup to the telescope.

5.10.3.2 __AO Offsets__

The NGAO system requires a complex offset control process to correct for errors introduced by the many optics and devices in the system. Depending on the specific observing mode and configuration, these error corrections are applied as offsets to one or more of the NGAO motion control devices. The management, calculation, and routing of offsets is performed by the *AOOffsetController*.

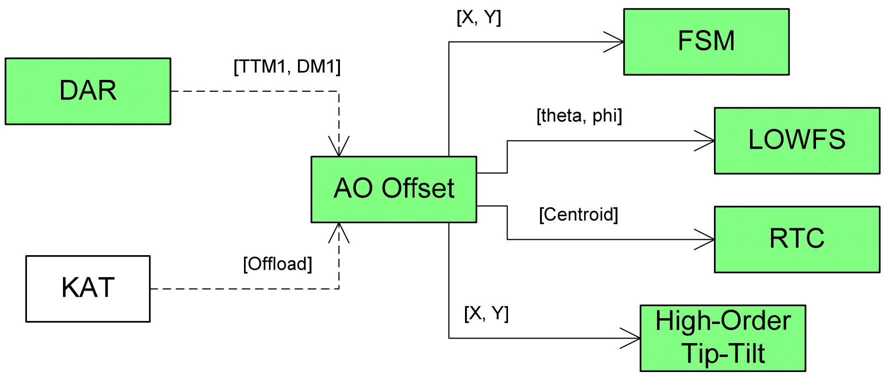

Figure 22: AO Offset Control Loop

Offset corrections cover tip-tilt, refraction, and rotation errors that will naturally occur as the telescope and AO system tracks targets during observing. These offsets must be applied to one of the beam train optics to counteract these aberrations. The four devices currently selected to receive offsets are the NGS Field Steering Mirrors (FSM), Real-Time Control (RTC) system, High-Order Tip-Tilt, and the Low-order Wavefront Sensors (LOWFS). Only one of these devices will receive offsets at any one time, depending on the current observing mode (see diagram above). The Offset Controller will determine how to apply corrections to the active offset device, and can quickly be reconfigured to send offsets to an alternate device if the observing mode changes.

Although the AO Offset Controller interacts with a number of devices, they are managed by other NGAO subsystems. The Offset Controller is not responsible for initializing, configuring or acquiring the devices. The Offset Controller assumes the devices are in the proper operating state and capable of accepting offsets when it is actively producing corrections.

5.10.4 Devices

Although the offload control subsystems communicate with a number of devices they do not actively manage their life-cycle or state. See the relevant subsystems for more information about the offload devices.

- NGS FSM
- High-Order Relay
- LOWFS
- RTC Interface

5.11 Calibration Simulation

T.B.D

5.11.1 Description

5.11.2 Interfaces

5.11.2.1 Inputs

5.11.2.2 Outputs

5.11.3 Control Loops

5.11.4 Devices

5.12 AO Enclosures

T.B.S

5.12.1 Description

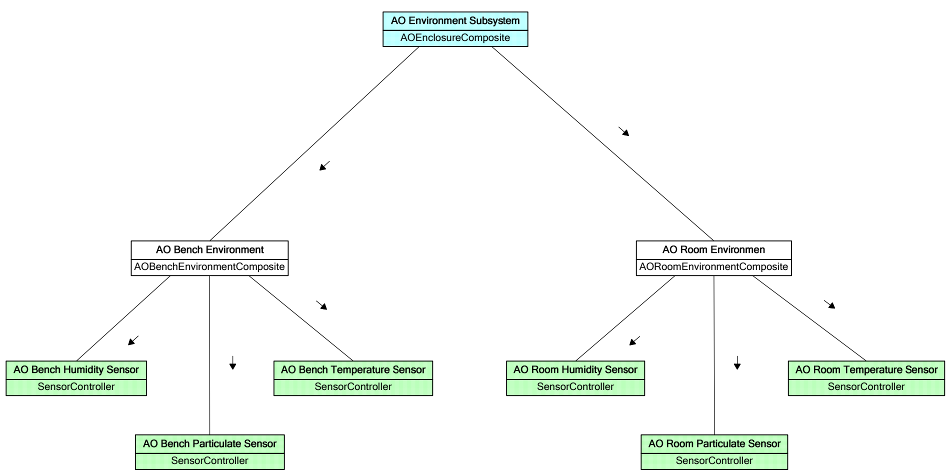

Figure 23: AO Enclosure Collaboration Diagram

5.12.2 Interfaces

The following sections detail the input and output interface to the AO Environment subsystem controller.

5.12.2.1 *Inputs*

The primary inputs to the AO Environment Enclosure subsystem are implemented as KCSF actions. Actions are defined through an attribute list and may contain parameters. The following define all of the recognized actions available to the Offloading subsystem components.

Home

The *Home* action is responsible for configuring the KCSF services; connecting to device drivers and components; and establishing telemetry topics.

Initialize

The *Initialize* action is responsible for refreshing connections to distributed components and telemetry, and starting the enclosure sensors. Once initialized, the sensors will continually publish readings through their corresponding telemetry topics. Clients can subscribe to these topics to get the current value of the sensor.

Recover

The *Recover* action allows the client to selectively reinitialize a faulted device or subsystem without interrupting the operation of other devices managed by the composite.

Disable

The *Disable* action will put the software controllers and subsystem into an idle (low network / resource usage) state. The sensors themselves will be powered down. While DISABLED the components will not be publishing telemetry nor monitoring and responding to system events. A *Home* must be performed to re-enable the components for use. Disable can only be performed when the subsystem is in a halted state.

5.12.2.2 *Outputs*

T.B.D

5.12.3 Control Loops

As the AO Environment Subsystem is outside of the NGAO control system, and does not manage or interact with any motion control devices, no control loops are implemented.

5.12.4 Devices

The following sensor devices are managed by the AO Environment Subsystem. Each sensor instance will be monitored by the Alarm Handler to provide operators with immediate feedback and alarm notification in case any of the sensors exceed a defined threshold. Thresholds are based on the LOLO to HIHI alarm categorization, and can be defined in the configuration database.

5.12.4.1 *Temperature Sensor*

The AO Environment Subsystem manages two temperature sensors: one on the AO bench, the other in the AO room enclosure. The temperature sensors will continually report the current temperature of the local environment. Large fluctuations in temperature can indicate failure of one or more cryo units, and run the highest risk in causing damage to the system. High temperatures can also significantly degrading system performance.

5.12.4.2 *Humidity Sensor*

High humidity in the AO system can cause problems for the electronics, and even more so for the optics and AO performance. As there is a variance between the ambient and cryo temperatures, condensation can also form and collect if the humidity in the environment is not controller.

*5.12.4.3 **Particulate Sensor***

As with the temperature and humidity sensors the AO facility also employs two particulate sensors to monitor the purity of the air in the AO enclosure. A high percentage of dust and debris in the air will reduce the throughput and quality of light in the AO system, and will require more frequent cleanings to bring the system back up to peak performance. A surge in particulates can also indicate smoke from electronic failures or friction between surfaces that may not register a high enough heat to be noticed by the temperature sensors.

5.13 AO Control Sequencer

NGAO is a complex system that will require a controls architecture capable of coordinating and managing the execution of dozens of devices. Each device must be configured and operated in a coordinated and synchronized fashion to ensure the optimal performance and safety of the system. A key to defining this device coordination is through the development of control sequences.

A sequence is a step-by-step breakdown of events and operations that a control system must perform to accomplish some task. Each step in a sequence represents a distinct system command. In many cases a step can itself be divided into sub-steps allowing the developer to design a hierarchical command sequence. These hierarchies are critical for discovering the collaboration and communication between components within the system.

Steps within a sequence can be executed serially, parallel or a combination of the two. A sequence or an operation within a sequence can be defined synchronously or asynchronously, where the execution of the sequence may depend on a specific system event.

The NGAO system can be divided into two main sequences: up-link and down-link. The up-link sequence pertains to the tasks and operations that must be performed by the Laser facility to generate, align, and project the laser asterism on the sky. The down-link sequence defines all of the steps that must be performed by the AO facility to acquire, calibrate, and perform atmospheric correction. The terms down-link and up-link indicate the direction of travel the light takes in and out of the system, respectively.

The following sections detail the implementation of the down-link control sequence for NGAO.

5.13.1 Description

The AO control sequence is implemented by the *AO Control Sequencer*, which is the main interface between the various AO controls functions and the rest of the NGAO system.

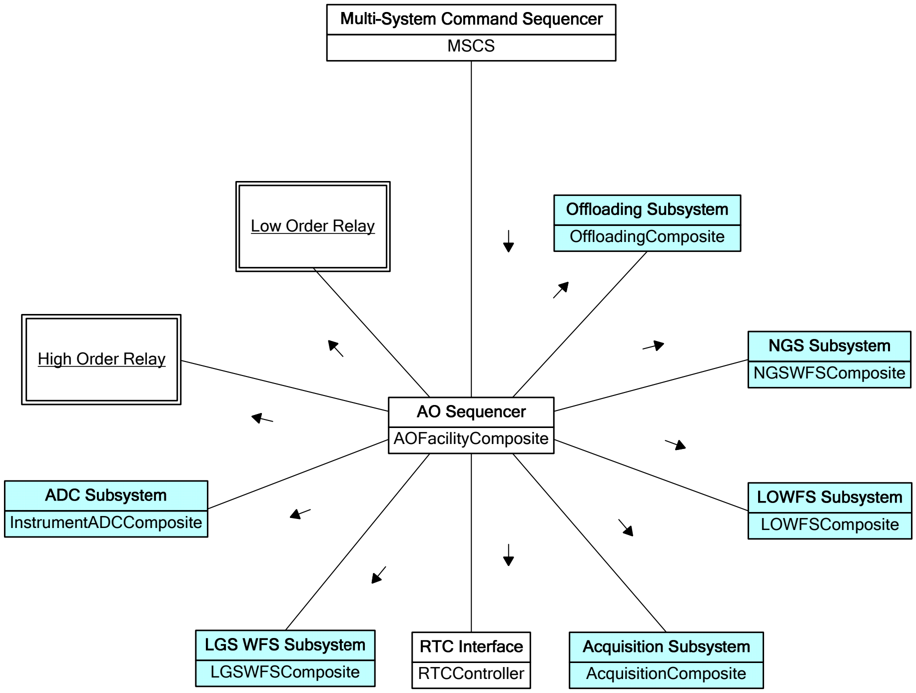

Figure 24: AO Control Sequencer Collaboration Diagram

5.13.1.1 *AO Sequence*

The AO control sequence covers everything from bringing up the software system; to acquisition; and finally shutting down for the night. The sequence itself can be divided into phases each covering a specific state of the system during a typical night's observing. These phases are (listed in order of execution):

- Startup
- Initialize
- Configure
- Acquire
- Observe
- Halt
- Disable
- Power-down

For the most part these phases can be executed straight through to perform NGAO correction for a science target. However there is flexibility allowing the sequence to return to a previous phase or jump forward to a latter phase. The following state diagram describes the relationship between each phase in the AO sequence:

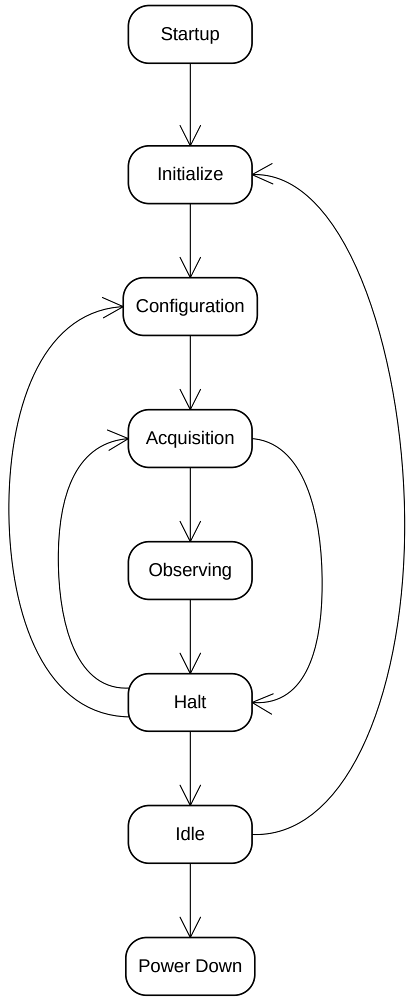

Figure 25: AO Sequencer State Diagram

The main branch of the sequence proceeds through each of the phases as outlined. At specific points in the sequence the control can be sent back to a previous step, such as going from *Halt* to *Acquisition*. This flexibility allows the system to be reconfigured multiple times throughout a night for greater flexibility. (Note: this diagram is a simplified version of the actual AO control sequence. The complete sequence employs a more complex state machine, with some details left out here for brevity. The full AO Sequencer state machine is documented in the NGAO AO Facility Component Design Document.)

Comment [dm2]: Link

A detailed review of the AO sequences is described in the NGAO AO Sequencer companion document.

Comment [dm3]: Link

5.13.2 Interfaces

T.B.S

*5.13.2.1 **Inputs***

Home

T.B.S

Initialize

T.B.S

SetMode

T.B.S

Acquire

T.B.S

Track

T.B.S

Halt

T.B.S

Recover

T.B.S

Disable

T.B.S

*5.13.2.2 **Outputs***

T.B.D

5.13.3 Control Loops

T.B.D

5.13.4 Devices

The AO Control Sequencer is the highest level component in the AO hierarchy, and as such, has the responsibility of managing the overall state of the entire AO facility. Most devices, however, are managed by at least one intermediate subsystem composite, which in turn may define its own component hierarchy: these subsystems have been detailed in the previous sections of this document. Although most of the AO facility device control is delegated to subsystems, a few devices are under the direct control and management of the AO Control Sequencer.

*5.13.4.1 **RTC Interface***

T.B.S

6 Laser System

6.1 Introduction

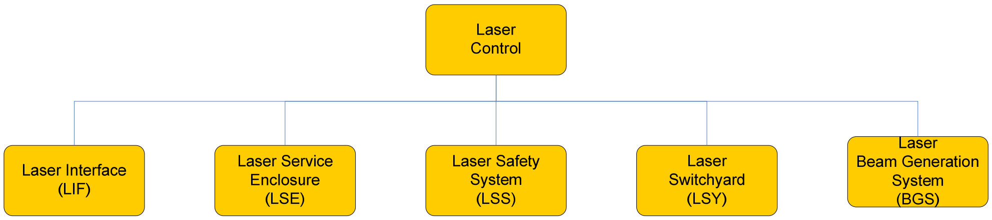

Figure 26 Laser Control Block Diagram

6.2 Laser Switchyard

The NGAO laser system is designed for center mounted propagation, with the laser asterism launched from the back of the telescope secondary. As there are size and weight restrictions on the secondary socket, the actual generation of the lasers will have to be done elsewhere. This will require a system of optics to transport the laser beams from the laser facility to the launch telescope. The transport optics and related hardware are all part of the laser switchyard (the laser units themselves are considered a separate subsystem), and are found in the laser enclosure on the nasmyth platform.

The transport optics consists of tip-tilt mirrors and an associated laser beam positioning sensor. Together the steering mirrors and sensors allow the system to control the path of the lasers as they travel to the secondary. The switchyard also includes a set of polarizers for each laser; a low-power fold mirror that can be used for alignment and acquisition; and the fast safety shutter which is controlled by the Laser Safety Subsystem.

The laser switchyard optics are controlled and managed by the Laser Switchyard Subsystem.

6.2.1 Description

The Laser Switchyard control system is implemented by the *LaserSwitchyardComposite*. This component is responsible for controlling the individual devices and distributed software controllers that implement the switchyard functionality. This includes the low-power fold mirror, steering mirrors, beam polarizers, and beam positioning sensors.

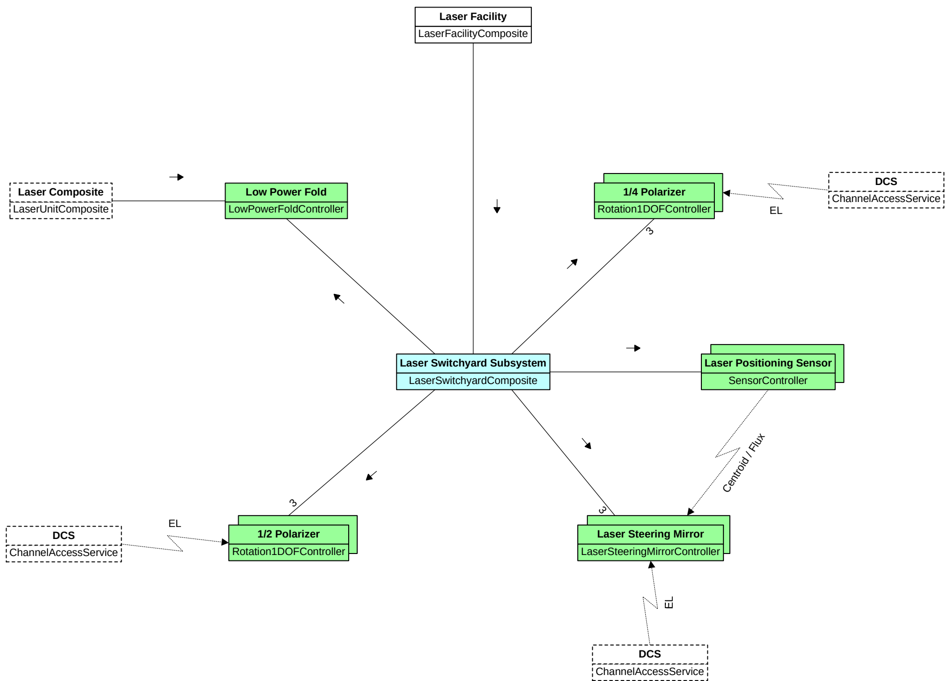

Figure 27: Laser Switchyard Collaboration Diagram

The Switchyard Composite is managed directly by the Laser Facility Composite, and presents a simple high-level interface to initialize, configure, acquire, and disable the switchyard devices. The Composite's primary responsibility is to prepare the switchyard devices for closed loop observing. The majority of the Switchyard Composite functionality exists as part of the configuration and acquisition sequence. During observing the tracking components are independently controlled or run closed-loop with the rest of the system.

The Laser Switchyard Subsystem implements a total of nine tracking control loops: one for each of the quarter and half polarizers, and for each laser steering mirror. The half and quarter polarizer controllers track the required polarization of the laser as a function of the telescope elevation. The Laser Steering Mirror Controllers are responsible for maintaining beam alignment with the BGS, and track with a combination of the telescope elevation and centroid information from the positioning sensors.

6.2.2 Interfaces

The following sections detail the input and output interface to the Switchyard subsystem controller.

6.2.2.1 Inputs

The primary inputs to the Laser Switchyard composite are implemented as KCSF actions. Actions are defined through an attribute list and may contain parameters. The following define all of the recognized actions available to the Switchyard composite.

Home
T.B.S

Initialize
The *Initialize* action is responsible for initializing the Switchyard Composite and controllers.

SetMode
The *SetMode* action is responsible for configuring the Laser Switchyard composite and subcomponents for a particular observing mode. Mode configuration will enable / disable and reposition devices as required by the observing use cases.

Fixed field and fixed pupil modes only differ in the use of the wide field LGS beamtrain. In fixed pupil observing the point and shoot LGS beacons are not used and the PnS laser unit is shuttered. The PnS laser polarizers (half and quarter) can be disabled to reduce resource usage. For safety reasons however the associated laser steering mirror and beam positioning sensor must remain active. (If the laser was to be accidentally opened we wouldn't want the laser to be sent off into hardware.)

In fixed field mode up to three wide field beacons will be managed. The PnS laser unit, switchyard hardware, and transport optics must all be enabled and tracking.

Mode configuration is performed internally by each component. The specific configuration details are hidden from view of the other control system components. This allows each system component to configure its self, without requiring higher level controllers to understand the characteristics of a subcomponent. The Switchyard Composite can simply forward the target mode configuration to each of its components and be confident they will configure correctly.

The system must be configured each time the observing mode changes, and must be performed prior to target acquisition and observing.

Acquire
The *Acquire* action is responsible for commanding the laser switchyard components to prepare the switchyard for laser propagation and transport.

This action has the following effect on the switchyard devices:
- Track the active laser steering mirrors.
- Track the active ¼ polarizers.
- Track the active ½ polarizers.

The composite will wait for each device to complete the acquisition process. The collective result of the acquisition process will be returned by the action.

Zero
The *Zero* action is designed to remove any offset applied to the steering mirror or laser polarizers. This action can be executed at anytime post initialization.

LowPower
The *LowPower* action is responsible for configuring the switchyard system for low power laser propagation. This is performed by inserting the Low-Power fold mirror into the laser beampath redirecting 99.9% of the laser light to a dump: allowing the leakage to continue through the system. As a requirement, the Laser shutters must be closed before the fold can be inserted. If the lasers are not shuttered the action will fail. This validation step is performed by the Low Power Fold Controller.

HighPower
The *HighPower* action is responsible for configuring the switchyard system for high power laser propagation. This is performed by removing the Low-Power fold mirror from the laser beampath allowing the laser to propagate at full strength. As a requirement, the Laser shutters must be closed before the fold

can be removed. If the lasers are not shuttered the action will fail. This validation step is performed by the Low Power Fold Controller.

Halt

The *Halt* action is responsible for halting device motion and opening the tracking loops. This action will cause the laser steering mirrors, polarizers and beam positioning sensors to all enter their *HALT* state. As these devices are no longer tracking, flexure compensation will not be performed. If the switchyard is not tracking the laser units or safety shutter should be closed.

Recover

The *Recover* action is responsible for selectively initializing faulted sensors devices by Laser Switchyard Subsystem. When the subsystem receives the recover action it will send a recover command to each of the switchyard controllers. If any of the controllers are faulted they will attempt an initialization, and clear the fault if the device reports the initialization was successful. . If the recover is successful the switchyard composite will clear its own fault state. The recover action can be issued at any time, and will not affect healthy components.

Disable

The *Disable* action is responsible for putting the Switchyard composite in a low-power, low-resource state.

6.2.2.2 Outputs

T.B.D

6.2.3 Control Loops

The following sections detail the tracking and offload control loops of the Switchyard subsystem. Each control loop is implemented in part of in full by one or more of the switchyard subsystem controllers.

6.2.3.1 Laser Transport

The laser transport control loop is responsible for maintaining the proper alignment of the lasers from the laser enclosure to the BGS. This control loop is needed to compensate for flexure as the telescope tracks a target; and uses a combination of the telescope elevation and feedback from the beam positioning sensors to calculate the appropriate position of the laser steering mirrors.

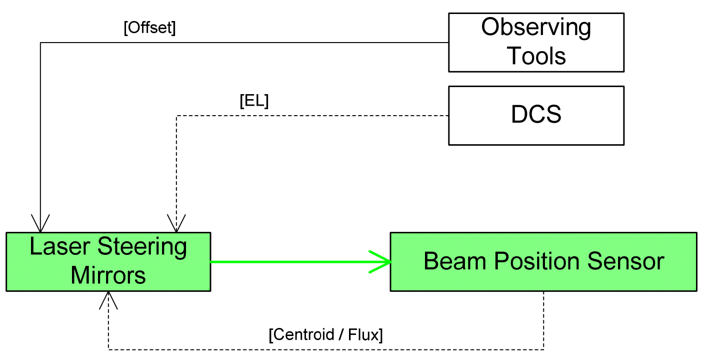

Figure 28: Laser Steering Control Loop

Although the laser transport tracking loop (implemented by the Laser Steering Mirror controller) uses inputs from DCS and the positioning sensors, it does not use both of them at the same time. The laser facility can propagate the lasers in high-power and low-power modes, depending on the position of the low-power fold mirror. When the fold mirror is in the beam path, all but a small percentage of the laser light is diverted to a beam dump. The leakage is allowed to propagate through the system and can be used for alignment and tests with little risk to hardware or personnel. When the fold mirror is retracted the full beam is allowed to propagate.

The laser transport control loop will use the flux value from the beam positioning sensors to determine if the system is configured for low or high-power propagation. If the flux is below a certain threshold the control loop will use the current telescope elevation from DCS. The elevation will be used to index a simple table to determine the appropriate flexure compensation to apply to the steering mirrors. These values will have been derived from tests empirical tests and will be interpolated to find the appropriate tip-tilt to apply to the stage.

If the positioning sensor threshold reports sufficient laser light centroid information will be obtained from the detector and will be used to calculate the precise compensation required to keep the laser aligned with the BGS.

In either low or high-power track mode the operator can apply an offset to the steering mirrors to redefine the ideal centroid position. The Laser Steering Mirror tracking loop should be closed whenever the lasers are propagating.

6.2.3.2 **Laser Polarization**

During observing the laser asterism needs to have the proper polarization to be seen by the AO wavefront sensors. Fortunately this polarization is directly related to the elevation of the telescope and incidence angle of the laser with the sodium layer. The control loop to maintain the desired polarization is implemented by each of the polarizer controllers.

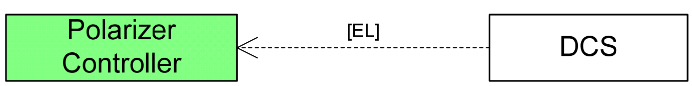

Figure 29: Laser Polarization Control Loop

The polarizer controller tracking loop is event driven and responds to telescope elevation events from DCS. When an EPICS event is received the controller will lookup the required orientation of the polarizer in a table based on the current elevation. The table will be defined through empirical tests at regular telescope elevations. The actual polarizer orientation will be interpolated based on these values.

The Half and Quarter Polarizer tracking loop can be remain closed throughout a night’s observing.

6.2.4 Devices

The following devices are managed by the Laser Switchyard subsystem, and are listed in the order the laser light propagates through the switchyard. Specific details, interfaces, and discussions are available in their related design documents.

6.2.4.1 Half & Quarter Polarizers

Each laser unit has a dedicated half and quarter polarizing lens. The polarizers are responsible for maintaining the proper polarization of the lasers for observing. The required polarization is based on the angle of incidence of the lasers and the sodium layer, which can derived from the current telescope elevation.

6.2.4.2 ***Low-Power Fold Mirror***

The low-power fold mirror is designed to redirect all but a small fraction of the laser light into a cold dump. The leakage from the mirror is allowed to propagate through the system to the BGS. At this power level the laser light can not cause any damage to the system during alignment, calibration or testing. The low-power fold mirror is common for all three laser beams.

6.2.4.3 ***Laser Steering Mirrors***

There are a total of three tip-tilt laser steering mirrors – one for each laser – responsible for propagating the lasers from the switchyard to the BGS. The steering mirrors will compensate for flexure as the telescope moves by using pre-measured flexure values, and feedback from beam positioning sensors on the BGS.

6.2.4.4 ***Beam Position Sensors***

The three beam positioning sensors work closed loop with the laser steering mirrors to correct for telescope flexure. As the telescope changes in elevation the gravity vectors will exert forces on the telescope structure causing flexure. The sensors will see this as a change in the beam position for which the laser steering mirrors will then compensate.

6.3 Laser Beam Generation System

NGAO is a multi-laser adaptive optics system designed to correct for atmospheric turbulence over a large area. As part of the science requirements of the system, a constellation of lasers will need to be produced and positioned on sky to perform tomography on the seeing column. In addition, up to three wide-field natural guide stars may also be used to provide full-field tip-tilt and focus correction.

To satisfy these requirements the laser system will make use of four fixed laser guide star (LGS) beacons for tomography, and three roaming LGS beacons to individually sharpen the wide-field tip-tilt stars. The entire laser constellation must be flexible enough to be positioned and orientated anywhere in the field of regard, and must be able to achieve any wide-field tip-tilt configuration.

All of the hardware needed to produce this laser asterism are part of the laser beam generation system (BGS), and is located on the back of the telescope secondary mirror. The BGS is controlled and managed by the Laser Beam Generation Subsystem.

6.3.1 Description

The *BeamGenerationSystemComposite* is a high-level multi-device manager that is responsible for acquiring and configuring the hardware on the BGS. These devices include the point-and-shoot steering arms, beam splitters, asterism rotator, tip-tilt mirror, and beam expander focus.

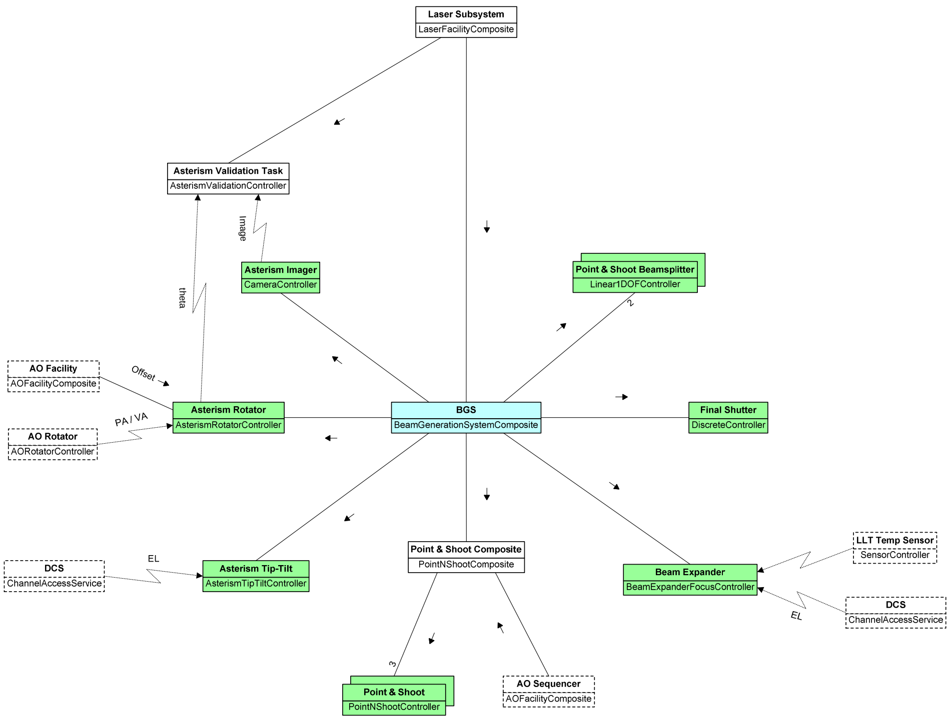

Figure 30: Beam Generation System Collaboration Diagram

The BGS Composite is managed directly by the Laser Facility Composite, and presents a simple high-level interface to initialize, configure, acquire, and disable the BGS devices. The majority of the BGS Composite functionality exists as part of the configuration and acquisition sequence. During observing the majority of the subcomponents are independently controlled or run closed-loop with the rest of the system.

The BGS subsystem is responsible for configuring and positioning the hardware to obtain the desired laser constellation. The constellation is based on the number and position of the wide-field targets, the position of the telescope, and the desired image orientation as seen from the science detector. The BGS subsystem will take this information and calculate the required positions for each of the associated devices.

The BGS subsystem also manages the focus and propagation of the lasers and implements safety processes to validate the health of the laser generation system. Before the lasers can be propagated on sky a validation is performed to ensure the proper structure of the laser asterism. This step is performed periodically throughout observing and is used to verify that the lasers are working properly and are not at risk of causing damage to the system or personnel.

The following sections detail the BGS subsystem and devices.

6.3.2 <u>Interfaces</u>

The following sections detail the input and output interface to the BGS subsystem controller.

6.3.2.1 Inputs

The primary inputs to the BGS composite are implemented as KCSF actions. Actions are defined through an attribute list and may contain parameters. The following define all of the recognized actions available to the BGS composite.

Home

The *Home* action is responsible for homing the BGS Composite and subcontrollers. On completion all of the devices on the BGS should be properly homed and encoders zeroed.

Initialize

The *Initialize* action is responsible for initializing the BGS Composite and controllers.

SetMode

The *SetMode* action is responsible for configuring the BGS software controllers and devices for a particular observing mode. BGS mode configuration will enable / disable and reposition devices as required by the observing mode. The following actions are taken for LGS observing:

Fixed-Field LGS

- Enables all BGS controllers.
- Configures Asterism Rotator for fixed-field (parallactic angle) tracking.

Fixed-Pupil LGS

- Disables Laser Beam Splitter Controllers and Point-and-Shoot Composite.
- Configures Asterism Rotator for fixed-pupil (vertical angle) tracking.

The system must be configured each time the observing mode changes, and must be performed prior to target acquisition.

Acquire

The *Acquire* action performs the target acquisition of the BGS system. This action requires the pointing target for the telescope (including any telescope offsets) and up to three LGS targets, all in sky coordinates: the target coordinates will be used to position the point-and-shoot laser beacons. The BGS Composite will calculate the offsets of the three targets relative to the optical axis in arcseconds, and command the Point-and-Shoot Composite to acquire the targets. The number of LGS targets will also be used to determine the appropriate configuration for the PnS Beam Splitters.

The Acquire action is also responsible for putting the BGS devices into the proper state for observing:

- Asterism Tip-Tilt Controller is put into TRACK.
- Asterism Rotator is put into TRACK.
- Asterism Imager Controller is put into free-run mode.
- Beam Expander Focus Controller is put into TRACK.

If any of the devices fail to enter their required states, or there is a problem acquiring the wide-field targets, the Acquire action will fail.

Close

The *Close* action commands the BGS Composite to close the final shutter, and halt laser propagation. This action will typically be executed during a configuration mode change and when an observing sequence has been halted.

Open

The *Open* action commands the BGS Composite to open the final shutter and is performed as the last step of laser acquisition and propagation.

Zero

The *Zero* action is responsible for returning the BGS devices to their nominal zero position (this may or may not be true zero as related to range of motion or encoder count). Effectively this is the same as removing any offsets added to the devices during observing or acquisition. This action has the effect of:

- Positioning the Asterism Tip-Tilt mirror to its nominal (flexure compensated) position.
- Return the Asterism Rotator to its nominal (PA / VA defined) position.
- Evenly distributing the Point and Shoot arms in the wide-field area of regard.
- Position the Beam Expander Focus stage to its nominal position.

Since the Zero action can cause the point-and-shoot steering arms to perform a large slew, it can not be executed while the lasers are propagating at full power. The laser shutters must be closed, or the low-power fold inserted before the Zero action can be executed.

Halt

The *Halt* action is responsible for halting all devices on the BGS and is typically issued to interrupt or terminate a sequence. Any devices currently in motion will be immediately stopped, and all loops will be opened: this includes stopping automated asterism validation (Validation Controller) and taking the asterism imager out of free-run.

Halt can be issued from any of the post-initialization states, and from the HALT state the controller can be reinitialized, disabled, or perform a new acquisition.

Recover

The *Recover* action is responsible for selectively initializing faulted devices managed by the Beam Generation System. When the subsystem receives the recover action it will send a recover command to each of the BGS controllers. If any of the controllers are faulted they will attempt an initialization, and clear the fault if the device reports the initialization was successful. If the recover is successful the BGS will clear its own fault state. The recover action can be issued at any time, and will not affect healthy components.

Disable

The *Disable* action is responsible for putting the BGS devices and controllers in a low-power, low-resource state.

6.3.2.2 ***Outputs***

T.B.D

6.3.3 Control Loops

The following sections detail the tracking and offload control loops of the BGS subsystem. Each control loop is implemented in part of in full by one or more of the NGAO controllers.

6.3.3.1 ***Focus Tracking***

During observing the BGS subsystem will have to track the changing focus of the LGS asterism. Focus is primarily based on the current telescope elevation and air temperature. As the telescope elevation changes the beam length of the laser from the telescope to the sodium laser will fluctuate. As the apparent distance to the sodium layer changes the beam expander will have to compensate to maintain LGS focus. The temperature of the air at the launch telescope also affects the focus of the laser. This information is derived from the environmental sensors within the LLT.

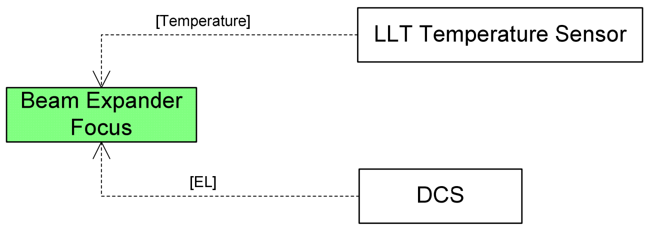

Figure 31: Laser Focus Tracking Control Loop

Beam expander tracking is an event driven control loop triggered by telemetry events from the LLT environment sensors and DCS telescope elevation keywords. When the controller receives input from these items it will calculate the LGS asterism focus compensation and apply it to the stage. A lookup table should be sufficient to define focus compensation based on telescope elevation and temperature.

*6.3.3.2 **Rotator Tracking***

During observing the laser asterism will need to track the LGS targets as the telescope rotates. In fixed field mode this is based on the parallactic angle (PA). In fixed pupil mode the rotator will track the vertical angle (VA). In addition, the asterism rotator will need to maintain a consistent orientation in regards to the AO image rotator.

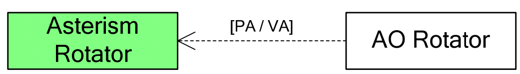

Figure 32: Asterism Rotator Tracking Control Loop

Rotator tracking and offloading is event drive. The Asterism Rotator Controller will receive field and pupil angle information from AO rotator telemetry, as well as any arbitrary changes to the AO rotator orientation. The asterism rotator will process this information and track with telescope and field rotation.

*6.3.3.3 **Flexure Compensation***

Changes in telescope elevation will induce flexure on the BGS and launch optics. As the telescope tracks a target the asterism tip-tilt will need to compensate for flexure to keep the asterism fixed with the targets.

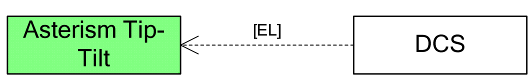

Figure 33: Flexure Compensation Control Loop

The Asterism Tip-Tilt Controller implements an event driven tracking loop. Primary input to the control loop is the telescope elevation and is made available from DCS through the Channel Access service. A simple look-up table can be used to interpolate the required flexure compensation for the current elevation.

6.3.3.4 Asterism Validation

A key requirement during observing is to ensure the safe propagation of the lasers and the proper structure of the asterism. This safety issue is managed by the Asterism Validation task.

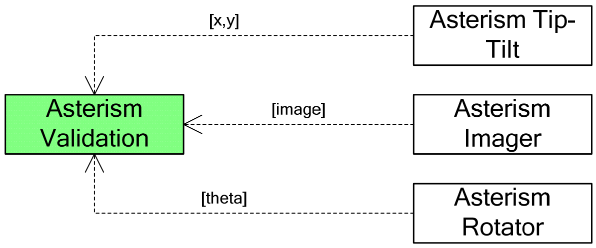

Figure 34: Asterism Validation Control Loop

This task is designed to perform asterism validation and error handling during the acquisition and observing sequence. The control loop implements an image processing routine that takes images from the Asterism Imager Camera and compares the laser beacons against their expected positions in the field. These positions are based on the coordinates of the tip-tilt targets, position of the asterism tip-tilt mirror, and the orientation of the rotator. If there are any major changes to the asterism structure, such as the loss of an LGS beacon, the validation controller can take an automated response to ensure the safety of facility.

6.3.4 Devices

The following devices are managed by the BGS subsystem, and are listed in the order the laser light propagates through the BGS. Specific details, interfaces, and discussions are available in their related design documents.

6.3.4.1 Beam Splitter

The BGS subsystem manages two linear 1-DOF beam splitters. These devices are used to split the wide-field LGS laser into the required number of roaming LGS beacons. The beam splitters can be configured to produce one, two, or three beacons: off-beam, one on-beam or both on-beam, respectively. The specific configuration will depend on the number of tip-tilt guide stars used. The beam splitters are controlled through KCSF Linear 1-DOF Controllers.

6.3.4.2 Point & Shoot

The point and shoot steering arms allow the BGS to position the wide-field beacons anywhere in the permitted range of the LGS asterism. There are a total of three point and shoots, each implemented as a linear 2-DOF actuated arm. Each point and shoot can be positioned arbitrarily in the field; with software restrictions preventing them from being within the central asterism and from occluding one another. The three point and shoot steering arms are managed by the Point & Shoot Composite which is responsible for validating target acquisition, preventing collisions between arms, and managing the point and shoot interaction. The point and shoot devices are controlled through the KCSF Linear 2-DOF Controller.

6.3.4.3 Asterism Rotator

The asterism rotator is a rotational 1-DOF K-mirror device responsible for the orientation of the entire laser asterism. The asterism rotator is designed to keep the laser asterism aligned to a specific orientation with respect to the sky (fixed field) or detector (fixed pupil). This tracking device receives inputs from the AO facility as offsets and must maintain a relative orientation to the AO image rotator. The asterism rotator is controlled through the NGAO Asterism Rotator Controller.

6.3.4.4 ***Asterism Tip-Tilt (up-link)***

The asterism tip-tilt mirror is a 2-DOF device that allows the AO controls system to shift the entire laser asterism on the sky. This common mirror is used during the acquisition and observing sequences for LGS WFS alignment and flexure compensation. The tip-tilt mirror will compensate for flexure by tracking the telescope elevation, and any additional offsets issued by the control system. The tip-tilt mirror is controlled through the NGAO Asterism Tip-Tilt Controller.

6.3.4.5 ***Beam Expander***

The beam expander stage is a linear 1-DOF device that is responsible for maintaining the required LGS focus. This tracking device responds to changes in the telescope elevation and the environmental conditions in the LLT. The beam expander focus stage runs in a nearly continuous automated fashion, and is controlled through the NGAO Beam Expander Focus Controller.

6.3.4.6 ***Asterism Imager***

The asterism imager is a slow-rate diagnostic camera that provides a view of the LGS asterism prior to the launch telescope. This camera picks up the leakage from the LLT fold mirror on the BGS, and shows the real-time structure of the asterism after the rotator, tip-tilt and steering arms. The asterism imager camera is controlled through a KCSF Camera Controller and is primarily used by the Asterism Validation task.

6.4 Laser Safety System

The *LaserSafetySystemController* is a subclass of the Controller definition, and provides an interface to the Laser Safety System (LSS): a subsystem responsible for monitoring the state of the laser system and ensuring safe operating conditions. A number of devices are under the complete control of the safety system, and can not be controlled directly by the software system. In order to offer a level of control to these devices (such as the beam dump and BGS fast shutter) functionality will exist in the safety system to accept control requests. If the requests do not risk the safety of the system or personnel they will be performed.

For the most part the laser Safety System is a self-contained system. The LSS controller provides an interface to a basic set of safety system controls and hardware. The controller will be responsible for providing permissive data to the safety system, and for publishing alarm events detected by the laser safety system.

6.4.1 Description

The control of the laser safety system functions is shown below in Figure 13. This section is not meant to describe the design of the laser safety control system, which is covered in the laser portion of the system design, but the control interface to the laser safety control system. The control interface to the laser system consists of an EPICS device driver to provide an EPICS interface to the system, and possibly some high-level coordination sequences that would be part of the laser sequencer to coordinate sequences of complex safety system commands.

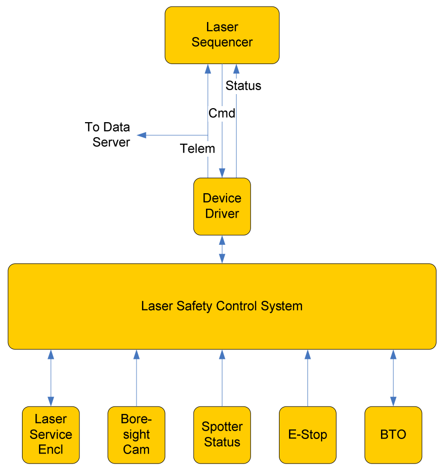

Figure 35: Laser Safety System Block Diagram

6.4.2 Interfaces

6.4.2.1 Inputs

Home
T.B.S

Initialize
The *Initialize* action is responsible for preparing the LSS controller for LGS observing. The initialization process will connect to the laser safety system, create monitors and telemetry topics, and launch the safety event handler task. This task is responsible for notifying the rest of the NGAO control system of any errors or warnings that may impact observing, allowing the higher level controllers to respond appropriately (including halting the current sequence). See the Remarks section for more details.

SetPermissive
The *SetPermissive* action is responsible for setting the state of one or more LSS permissives. This action can only be executed from within the post-initialization states.

A permissive is defined by a string and a boolean value, and can be any one of the recognized names available to the LSS. The permissives include, but are not limited to, the following:

- Observing Assistant Acknowledgement: The final permissive from the OA to permit laser propagation on-sky.
- Laser Traffic Control System: Confirms there is no laser or observing collisions between the other Mauna Kea observatories.
- Spotter Acknowledgment: Confirmation from the spotters that it is safe to propagate.
- Telescope Control Acknowledgement: Confirmation that the telescope is tracking and in the safe telescope elevation window for laser propagation.
- Satellite Safety Acknowledgment: Confirmation from space traffic control that there are no imminent satellite crossings over the laser path.

- Laser Transport Acknowledgement: Software confirmation that the lasers are being transported to the BGS properly. (This can double for a valid asterism confirmation.)
- Reduced Performance Acknowledgment: Permits (or prevents) the system from running with a reduced set of laser units. This may be needed in the event that one of the lasers is malfunctioning, but observing should continue.

These permissives cover the majority of system state properties that are not directly visible by the Laser Safety System. The full set of software supplied permissives will be finalized after the laser vendor has been chosen.

Close

Closes the shutter specified in the attribute *shutter*. This can be any of the subordinate shutters that comprise the laser safety system, and are specified by a partial or fully qualified KCSF name. The actual shutter name will depend on the values set in configuration but cover the Beam Dump and BGS Fast Shutter.

Open

Opens the shutter specified in the attribute *shutter*. This can be any of the subordinate shutters that comprise the laser safety system, and are specified by a partial or fully qualified KCSF name. The actual shutter name will depend on the values set in configuration but cover the Beam Dump and BGS Fast Shutter.

FullStop

This action tells the Laser safety System that it should immediately put the system into an emergency safe state. This includes closing shutters, powering down lasers, and taking any additional full-stop safety precautions.

Acknowledge

The *Acknowledge* action is used to force the LSS Controller out of an alarm state by acknowledging the existence of an alarm condition. Ideally alarm conditions will be satisfied by either correcting the problem identified by the LSS or setting the appropriate permissives. This should then cause the LSS Controller to return to the READY state. In the case where the problem can not be immediately handled and observing is to continue the acknowledge action can confirm receipt of problem, and put the controller back in the READY state allowing it to respond to other alarm conditions that may occur.

Note: the acknowledge action does not change the state of the LSS, it simply silences the current alarm state from within the NGAO control system. If the LSS issues the same error condition again it will result in reentrance of the ALARM state.

Halt

The *Halt* action is responsible for putting the LSS Controller into the *HALT* state. As the LSS Controller does not have any motion control requirements this action is implemented as a no-op routine, and exists primarily for consistency with the NGAO Controller design.

Recover

…

Comment [dm4]: Is there any recovery technique for the LSS?

Disable

The *Disable* action will cause the controller to suspend the event handler task and enter a low resource usage state. While *DISABLED* the controller will not be able to notify the higher level sequencer of safety events or be able to set permissives for the laser safety system.

This action should only need to be executed during shutdown of the NGAO control system, and can only be executed from the HALT state. The controller can be reinitialized and safety event handler restarted by performing a *Home*.

6.4.2.2 Outputs

T.B.D

6.4.3 Control Loops

T.B.D

6.4.4 Devices

The following devices are managed by the Laser Safety Subsystem. Specific details, interfaces, and discussions are available in their related design documents.

6.4.4.1 Laser Safety System

T.B.S

6.5 Laser Service Enclosure

T.B.S

6.5.1 Description

The *LaserEnclosureComposite* manages the environmental sensors found in the laser unit enclosure. These sensors are responsible for providing the current temperature, humidity, and particulate level.

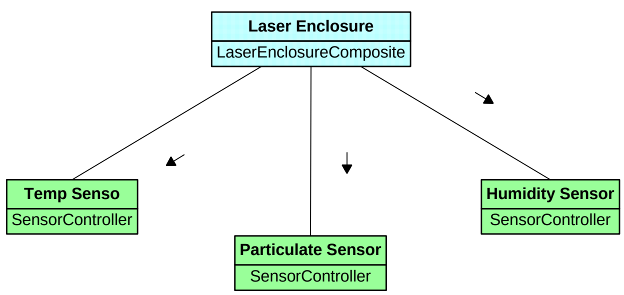

Figure 36: Laser Service Enclosure Collaboration Diagram

As sensors are simple on / off devices without motion control requirements, the composite itself is very simple, and provides the basic capabilities to initialize the system for observing. The composite will collect data from each of its sensors and will provide a summarized health status as telemetry and alarming.

6.5.2 Interfaces

6.5.2.1 Inputs

Home
T.B.S

Initialize
The *Initialize* action is responsible for initializing the Laser Enclosure Composite and controllers. As part of the initialization process the composite will create all of its telemetry topics and refresh connections to subordinate controllers. The Initialize action will cascade down to each of the sub-controllers, which will in turn prepare the devices for use. (See the subordinate controllers for more information).

Halt
As there are no motion control requirements for the Laser Enclosure Composite the *Halt* actions exists primarily for consistency with the NGAO composite design pattern. The halt action is effectively a no-op routine, and can be issued from the INIT or READY states. The halt action is required prior to disabling and shutting down the composite and sensor controllers.

Recover
The *Recover* action is responsible for selectively initializing faulted sensors managed by the Laser Service Enclosure. When the subsystem receives the recover action it will send a recover command to each of the sensor controllers. If any of the controllers are faulted they will attempt an initialization, and clear the fault if the device reports the initialization was successful. The recover action can be issued at any time, and will not affect healthy components.

Disable
The *Disable* action is responsible for putting the laser enclosure composite in a low-power, low-resource state. Disable is typically performed at the end of a nights observing or when monitoring of the laser enclosure is no longer need, for example when the lasers are powered down. While DISABLED the Laser Enclosure Composite and sensors will not be publishing telemetry nor monitoring and responding to system events. A *Home* must be performed to re-enable the composite and controllers for use. Disable can only be performed when the controller is in the HALT state.

6.5.2.2 Outputs

T.B.D

6.5.3 Control Loops

As the Laser Enclosure subsystem does not manage any motion control devices, and the environmental information of the laser enclosure is not used in a controls context, this subsystem does not implement any control loops.

6.5.4 Devices

The following devices are managed by the Laser Enclosure subsystem. Specific details, interfaces, and discussions are available in their related design documents.

6.5.4.1 ***Environmental Sensors***

The Laser Enclosure subsystem manages the temperature, humidity, and particulate sensors found in the laser enclosure. These sensors run continually, even outside of the regular NGAO control system, and provide environmental information to the operators through the observing tools. Although not used in a direct controls context, the environmental sensors will report excessive environmental conditions to the NGAO alarm system, allowing the operator to take action if needed.

6.6 Laser Interface

The NGAO laser system is designed to produce up to seven laser beacons using three laser sources. Two of the laser units are responsible for producing the central asterism of four beacons (each split 50 / 50). The third laser unit is responsible for producing the three wide-field PnS beacons (split 33 / 33 / 33). Depending on the observing mode all three lasers may be used, just the central asterism lasers, or no lasers (NGS observing). The laser units are all managed under the Laser Interface Subsystem.

6.6.1 Description

The Laser Interface Subsystem is implemented by the *LaserUnitComposite*, and manages the three laser units. Although the lasers are under the direct control of the LSS, separate NGAO components will be created to interface with each laser units, providing a more intuitive and convenient method of distributed control.

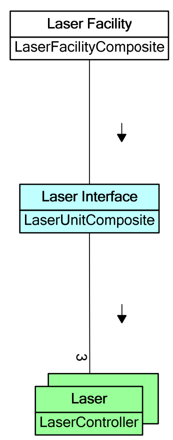

Figure 37: Laser Interface Collaboration Diagram

Above the *LaserControllers* will be the laser interface subsystem. This composite provides a single unified command and status interface to the lasers. Clients can treat the laser sources as a single instance, allowing them to issue a single command in place of three. Additionally, health and status information for all three laser units will be consolidated and made available to the control system for easy reference.

The Laser Unit Composite is managed directly by the Laser Facility Composite.

6.6.2 *Interfaces*

*6.6.2.1 **Inputs***

Home

The *Home* action is responsible for preparing the laser units for observing, specifically initiating laser calibration. The LaserUnitComposite will stay in the *HOME* state until the laser calibrations and warm-up have completed.

Initialize

The *Initialize* action is responsible for closing the laser shutters and putting the software composite into the in-position state.

SetMode

The *SetMode* action is responsible for configuring the Laser Switchyard composite and subcomponents for a particular observing mode. Mode configuration will enable / disable and reposition devices as required by the observing use cases.

Fixed field and fixed pupil modes only differ in the use of the wide field LGS beamtrain. In fixed pupil observing the point and shoot beam train is not used, and the PnS laser unit can be disabled. In fixed field mode the wide-field beacons are available, and all three laser units are enabled and initialized for use.

The system must be configured each time the observing mode changes, and must be performed prior to target acquisition and observing.

Close

The *Close* action is responsible for shuttering the active laser units. While shuttered no laser light will enter the beam train.

Open

The *Open* action is responsible for opening the active laser unit shutters. While open the laser light will propagate through the beam train.

To ensure the safety of the system, this command will fail if any of the transport or alignment devices are not tracking and in position. Specifically all of the following must be true before the laser shutters can be opened:

- Laser Steering Mirrors are tracking
- Beam Splitters are in position.
- Point and Shoot Steering arms in position.

TuneOn

The *TuneOn* action is responsible for tuning the active lasers on to the 589nm wavelength. This must be performed for the AO wavefront sensors to operate properly.

TuneOff

The *TuneOff* action is responsible for tuning the active lasers off of the 589nm wavelength. The actual wavelength will be defined and preset in the unit. This action is performed when taking measurements of the background signal.

Halt

The *Halt* action is responsible for putting the laser units in a halted / safe state. Specifically this action will shutter the laser units to prevent laser propagation. The halt action is required prior to disabling and shutting down the composite and controllers.

Recover
The *Recover* action is responsible for selectively initializing faulted laser devices and controllers managed by the Laser Interface Subsystem. When the subsystem receives the recover action it will send a recover command to each of the laser controllers. If any of the controllers are faulted they will attempt an initialization, and clear the fault if the device reports the initialization was successful. The recover action can be issued at any time, and will not affect healthy components.

Disable
The *Disable* action is responsible for disabling the Laser Controller. While disabled network and resource usage will be at a minimum. The laser unit itself will be shuttered, but its state will remain operational and ready for use. The controller must be re-homed before the laser can be used again.

6.6.2.2 Outputs

ShutterState
This attribute returns the current global shutter state of the active laser units. If the active laser shutters are in different states *Open* will be returned.

TuneState
This attribute returns the current global tuned state of the active lasers. If the active lasers are in different states *Off* will be returned.

6.6.3 Control Loops

The Laser Subsystem does not implement any control loops.

6.6.4 Devices

The following devices are managed by the Laser Interface subsystem. Specific details, interfaces, and discussions are available in their related design documents.

6.6.4.1 Lasers

The Laser Interface Subsystem manages the controllers for the three laser units. These controllers provide the capabilities to initialize, calibrate, and propagate the lasers. The controllers themselves do not actually interface directly with the lasers, but acts as a wrapper around the LSS to provide the set of laser commands available to clients.

6.7 Laser Control Sequencer

NGAO can be divided into an up-link (laser) and down-link (AO) system. The up-link system is responsible for producing and propagating the laser asterism on sky. The down-link system is responsible for correcting distortion in the field image caused by atmospheric turbulence. As there is such a clear and distinct functional separation between the two systems each can be modeled with their own control system hierarchy.

The up-link system is implemented by the Laser Facility System. This hierarchical control system implements a number of tracking and non-tracking devices with varying motion control requirements. At the highest level of the Laser Facility System is the Laser Control Sequencer. This KCSF Composite provides a high-level interface to the entire laser facility, and is responsible for executing the main acquisition and observing sequences; and managing the operation and state of the laser subsystems. The following sections detail the Laser Control Sequencer.

6.7.1 Description

The Laser Control Sequencer is implemented by the *LaserFacilityComposite*, and is responsible for managing and configuring the entire laser facility, and executing the various sequences. The Composite sits above all of the other laser system components, and coordinates their state and task execution during observing. The Laser Sequencer is under the direct control of the MSCS, and is the primary interface to the laser system.

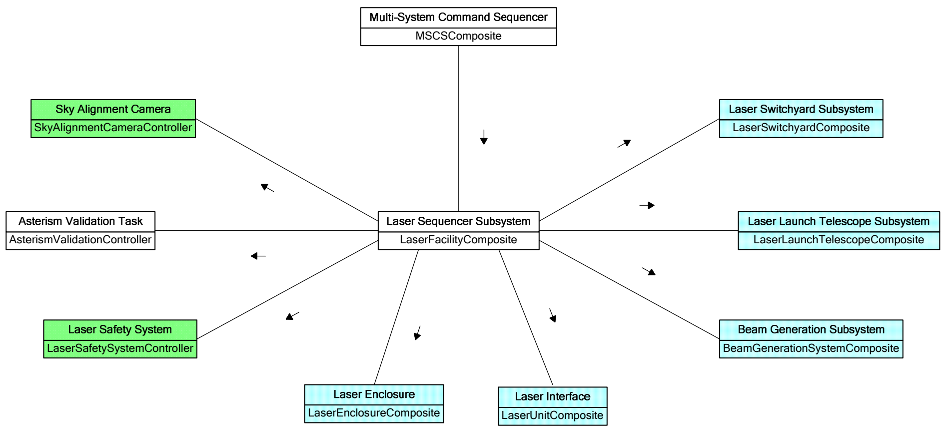

Figure 38: Laser Control Sequencer Diagram

The Laser Sequencer exposes a simple set of high level actions to the MSCS. These actions provide the Command Sequencer with the ability to initialize and configure the laser facility; acquire LGS targets; and shutdown the system. The composite in-turn takes the action data and distributes it among the relevant subcomponents; invoking their corresponding initialization, configuration, acquisition, and shutdown actions.

As with other composites, the Laser Sequencer will reflect the overall state of its subcomponents: effectively the health and state of the entire laser system. The MSCS will monitor the state of the composite to ensure its continued safe operation, and to respond to any errors that may occur.

6.7.2 Interfaces

The following details the Laser Control Sequencer interface.

6.7.2.1 Inputs

Home
T.B.S

Initialize
T.B.S

SetMode

The *SetMode* action is responsible for configuring the laser facility composite and subcomponents for a particular observing mode. Mode configuration will enable / disable and reposition devices as required by the observing mode.

In fixed-field observing the entire laser facility is enabled and prepared for use. In this mode up to three LGS PnS beacons are used to correct for atmospheric effects on the wide-field tip-tilt and focus stars. As the telescope rotates to track the science target the LGS asterism must also rotate on sky to compensate for the changing parallactic angle. This effectively fixes the laser constellation with the star field as seen from down stream.

In fixed-pupil mode the field is allowed to rotate freely, but the pupil must remain fixed. In this mode the PnS LGS beacons are not used. As a result, the majority of the PnS laser beamtrain can be disabled. To stay fixed will the pupil the laser asterism will only need to match the orientation of the telescope's vertical angle.

Mode configuration is performed internally by each component. The specific configuration details are hidden from view of the other control system components. This allows each system component to configure its self, without requiring higher level controllers to understand the characteristics of a subcomponent. The composite can simply forward the target mode configuration to each of its components and be confident they will be configured correctly.

The system must be configured each time the observing mode changes, and must be performed prior to target acquisition and observing.

Acquire

The *Acquire* action is responsible for commanding the laser facility components to acquire a set of LGS targets. Ultimately the laser asterism will be configured to match a set of sky coordinates based on the orientation of the field and position of the telescope. The acquisition process is typically initiated during the slew to the next pointing target.

This action requires the current pointing information for the telescope and up to three LGS targets, all in sky coordinates. The action arguments will be used to configure and command devices within the laser facility; specifically:

- Prepare for low-power laser propagation
- Select the appropriate PnS Beam Splitter configuration,
- Position the PnS steering arms,
- Track the switchyard devices (steering mirrors and polarizers),
- Track the BGS devices (focus stage, asterism rotator and asterism tip-tilt).

The composite will wait for each device to complete the acquisition process. The collective result of the acquisition process will be returned by the action.

Propagate

The *Propagate* action is responsible for validating the asterism and projecting the high-power laser beams on sky. This action can only be executed when the telescope is tracking, and is typically performed immediately after the slew to the science target.

When initiated, this action will perform a set of steps to validate the asterism before and after switching to high-power laser propagation. If the asterism structure appears correct the final shutter will be opened and the lasers will be allowed to propagate on sky. Once this action completes the entire laser facility will be properly configured for observing and all tracking loops will be closed.

The combined result of the propagate action will be returned.

Halt

The *Halt* action is responsible for halting all devices in the laser facility, and is typically issued to interrupt or terminate a sequence. Any devices currently in motion will be immediately stopped, and all loops will be opened. The laser facility shutters will be put into a safe state, and propagation will be stopped at the switchyard. Note: as the AO system relies on the laser light to perform atmospheric correction the AO Facility controller must be halted as well. If performed through the observing tools, the MSCS will automatically take care of halting both systems.

Halt can be issued from any of the post-initialization states. From the HALT state the laser facility can be reinitialized, disabled, or perform a new acquisition.

Recover

The *Recover* action allows the client (typically MSCS) to initiate a selective system re-initialization if a fault or error occurs. The recover command will be passed down the laser control system hierarchy to every NGAO component. If one of the components is in the fault state an initialization and recovery sequence will be executed to attempt to return the device to an operation state. If the recovery succeeds the device will be returned to its previous run-time state (e.g. TRACK). If the recovery fails the client may still be able to continue the sequence; however observing performance may be degraded. If the sequence has to be halted a full initialization should be performed in place of a recover.

Disable

The *Disable* action is responsible for putting the laser facility devices and controllers in a low-power, low-resource state. Disable is typically performed at the end of a night’s observing or when the laser system is no longer need, for example when switching to NGS mode. While DISABLED the Laser Facility Composite and sub-controllers will not be publishing telemetry nor monitoring and responding to system events. A *Home* must be performed to re-enable the composite and controllers for use. Disable can only be performed when the controller is in the HALT state.

6.7.2.2 Outputs

T.B.D

6.7.3 Control Loops

The following sections detail the control loops of the Laser Control Sequencer. Each control loop is implemented in part of in full by one or more of the NGAO components.

6.7.3.1 Asterism Validation

NGAO makes use of high-intensity lasers to produce the LGS beacons. If the lasers are not properly aligned they can cause significant damage to the laser facility hardware and injury to personnel. As such, the laser propagation must be periodically validated to ensure the proper alignment and performance of the system. An asynchronous task will be created to monitor the health of the laser system. If the task identifies any major changes to the asterism structure, such as the loss of an LGS beacon, an automated response will be taken to ensure the safety of the laser facility.

The *AsterismValidationController* is designed to perform asterism validation and error handling during the acquisition and observing sequence. This controller subclass implements an image processing routine that takes images from the Asterism Imager Camera and compares the laser beacons against their expected positions in the field. These positions are based on the coordinates of the tip-tilt targets and position of the BGS hardware.

Three failure modes have been identified for the laser asterism and will be monitored by the validation controller. The failure modes are:

- Loss of a laser or beacon – If the asterism validation identifies that there are fewer than the expected LGS beacons in the image the sequence will be halted and the laser system will be put into a shuttered safe state.
- Additional beacons – Potentially the laser system could be configured incorrectly to produce additional beacons when a reduced subset is desired (e.g. fixed-pupil or a single TT target). If the validation task identifies more than the expected number of LGS beacons the sequence will be halted and the laser system will be put into a shuttered safe state.
- Incorrect beacon position – Potentially caused by a number of factors (offloading, flexure compensation, etc.) the laser beacons may drift over time. As all of the beacons are accounted for, only displaced, there is no imminent safety concern for the system. The validation task will assume the system is performing correctly, but will warn the operator that the asterism has changed and that performance could suffer.

The validation task is event driven and will be executed for each new frame produced by the asterism imager camera. The initial positions of the beacons will be calculated when the validation task is started based on the sky position of the tip-tilt targets. The positions of the beacons will be recalculated whenever the BGS rotator or tip-tilt mirror is moved, and should only require a translation and rotation matrix transformation. The validation task is started at the beginning of the science sequence and halted at the end of integrations.

6.7.4 Devices

As a high-level subsystem, the Composite relies on the low-level subsystems for the majority of hardware control. Although the Laser Control Sequencer is capable of directly interacting with any hardware in the laser facility, there are only a few hardware devices it manages directly. Specific details, interfaces, and discussions are available in their related design documents.

6.7.4.1 Laser Safety System

The Laser Safety System (LSS) acts as a constant supervisory safety controller for the laser system, and must be running at all times the laser units and hardware is active. The LSS monitors a number of system and device states to ensure the system is operating within the defined safe limits. All real-time and safety critical device information is connected directly to the LSS allowing the system to immediately respond to protect the system and personnel from hazards. Non-critical and external system information is provided in the form of software permissives, and includes events from LTCS and operator acknowledgements.

See the Laser Safety Subsystem for more information on interfacing with the LSS.

6.7.4.2 Sky Alignment Camera

The Sky Alignment Camera exists primarily for system testing and laser alignment: this camera is not used during regular observing operations. The camera is physically located in the BGS, just after the beam expander, and can be used in either an up-stream (viewing asterism) or down-stream mode (viewing sky). To use the camera, a fold mirror is inserted into the beam path configured to pickoff the light from the lasers or to view the current pointing alignment of the telescope.

The Laser Sequencer Subsystem is responsible for initializing the sky alignment camera. Control and configuration of the device is performed through the observing tools.

7 Data Server

Although it is part of the AO Controls System, the Data Server can be and is being worked independently. As such, the details of the Data Server design can be found in the separate document: KAON 732: NGAO Data Server Preliminary Design.

8 Risk Analysis

The largest risk to the non-real-time controls system design is still the lack of specificity regarding the many devices to be controlled, both motion control devices and otherwise. In most cases we understand the type and number of devices to be controlled, but we still have not decided on and specific hardware device controllers or platforms. As a result, all the work estimates must have a high level of risk assigned until more information is available and the design can be specified. However, since the System Design phase, we have tried to simplify the design by consolidating and specifying the minimum number of controller types for the AO controls and Laser controls, thus minimizing the work estimates as much as possible.

Also affecting the work estimates is the lack of details for calibration procedures for most aspects of the system. Without enough details, good work estimates cannot be made.

Similarly, a choice still needs to be made on the machine architecture and operating system that we will use to host the NGAO control system. Currently we are considering Solaris and Linux machines, but there may still be requirements for using VxWorks or other platforms. Although the KCSF architecture should be able to operate on any OS and platform compatible with Java there are still questions about the compatibility of the underlying communication middleware (ICE or DDS) with certain architectures. This decision will impact which middleware architecture we will want to use, and which version (community or commercial) we will need to purchase.

Another fairly large risk is posed by the fact Erik Johannson (AO Controls Lead) has left WMKO and with him went a wealth of AO knowledge. This has left a huge void for the AO Control Team and must be filled by the remaining team members coming up to speed on AO, which is difficult while continuing to work on the software design. The NGAO project as a whole will also need to provide much more details when specifying tasks and algorithms for the AO Controls Team. Also, Erik was allocated at the 80 - 90% level, while his replacement, Kevin Tsubota is supposed to be working up to the 50% level by the end of PD. WMKO is currently in the process of hiring a Control Systems Engineer which will help with man power but none of the existing candidates have any AO experience.

9 Issues Requiring Resolution During DD Phase

The following items require attention during the DD design phase:

- Device controllers, workstations, operating systems, and architecture need to be defined before an overall device control philosophy and architecture can be better specified.
- The middleware (DDS or ICE) must be selected.
- Subsystem ICDs need to be defined and documented.
- AO and Laser Controls telemetry streams needed by other systems need to be defined and documented.
- Calibration control and sensor calibration measurement algorithms need to be specified and documented for the AO Controls software team.
- The laser software interface must be defined (this is the software interface to the laser device delivered by the vendor).
- The RTC interface needs to be detailed.
- Fixed-pupil observing mode control loops must be defined.
- Configuration system must be designed.
- Develop KCSF prototypes to prove NGAO controls system design.

A1. Functional Requirements

The following table is a snapshot of the Controls Infrastructure Functional Requirements included as a convenience for the reader. The see the most current and complete set of requirements refer to the Contour Database.

ID	Short Name	Description
FR-456	Definition	The overall controls system infrastructure is defined as all the infrastructure control functions that will be used throughout the AO Controls and LGS Controls system. Throughout the remainder of these requirements, the overall controls system infrastructure control functions will be referred to simply as "Controls Infrastructure". Controls Infrastructure shall provide the following basic functions for the AO and LGS Controls system: 1. Device control for all controllable devices. 2. Motion control for all controllable opto-mechanical devices not directly concerned with the high speed measurement and correction of atmospherically distorted wavefronts. 3. Configuration and control of the RTC. 4. Support for system configuration, calibration and operations. 5. Monitoring the health of all subsystems and providing notification to other systems as required. 6. Control of the AO and Laser enclosure environment.
FR-1839	Health Monitoring	Controls Infrastructure shall provide a health monitoring function that periodically checks all the AO and LGS subsystems for status. If an error is detected, Controls Infrastructure will provide notification to the other NGAO subsystems and users.
FR-1840	Acquisition Control	AO Controls will provide an automated acquisition capability to acquire all guide stars and science targets with limited operator oversight.
FR-1841	Configuration Control	The Controls Infrastructure will provide an automated capability to configure the AO and LGS system for all modes of the system: startup, operational, calibration, and test. This includes configuring all the required cameras with the appropriate default parameters, configuring the appropriate devices on the optics bench, configuring the RTC, and configuring the AO related functions of the IFS (if necessary). The specifics of these configurations will be documented as use cases in the operations concept document.
FR-474	Acquisition and Offload Control	AO Controls will provide functionality to interact with the telescope control system (currently DCS) to implement control for: Acquisition (possibly an interface to MAGIQ or son-of-MAGIQ) Offload (tip-tilt, focus and coma offloading)
FR-1828	General Device Control	Controls Infrastructure shall provide a control function for all non-motorized devices in the system capable of computer control. The devices to be controlled include: detectors, mirrors, the RTC, and the environmental system for the both AO and LGS enclosures. The control function shall provide, at a minimum, power control, device configuration and basic device control, unless these functions are not available. The motorized devices are addressed in a separate requirement on motion control.

ID	Short Name	Description
FR-1834	Motion Control	Controls Infrastructure shall provide a motion control function for all opto-mechanical devices requiring remote computer control. The control function shall include basic device control, configuration control, position control, tracking control for those devices that require it, and support coordinated moves of multiple devices.
FR-1832	Environmental Control	Control Infrastructure shall provide a configuration control function for all the environmental control devices in the systems that require remote computer control.
FR-1812	Acquisition System data products	One hundred per night (TBC) acquisition images shall be stored on the NGAO data server. Images shall be in FITS file format and have standard FITS header information. The specific contents of the header are TBD. The data products from the acquisition system and the information from available catalogs or literature shall be recorded in the same photometry system and with comparable spatial resolution.
FR-1913	Image Size and Profile Measurements	The acquisition system shall include tools to measure the FWHM and ellipticity of images, and to provide a profile of these images.
FR-3360	Logging	The Controls Infrastructure shall provide logging of important system events, parameter changes, etc., to one or more log files.
FR-3361	Alarms	Controls Infrastructure shall provide an alarm system which allows both the reporting of and response to abnormal system conditions. It shall provide both automated and manual response options, multiple levels of alarm severity, a centralized alarm manager, and flexible alarm system GUIs which implement sorting capabilities.
FR-3362	Data Recording	The Controls Infrastructure systems shall support recording of parameter data, telemetry and diagnostics to the Data Server.
FR-3363	Motion Control Tracking Rates	Controls Infrastructure shall support minimum sustained input rates of 40 Hz for tracking motion control applications.
FR-3364	Data Types	The Controls Infrastructure shall handle transmission of arbitrary data types, both scalars and arrays, with data sizes up to at least 1 Mb. The control system must support arbitrary data types, including structures. Many RTC parameters are arrays, which are also large in size. The maximum array size used by the RTC is estimated to be 1 Mb.
FR-3121	Data Server	The NGAO Data Server is defined as the system that provides all the data recorder and server capabilities required by the NGAO system. This includes all database support, logging, archiving, general storage and retrieval. The data server must support all the data storage and retrieval requirements of the NGAO system.
FR-3098	Data Server - Definition	The NGAO Data Server is defined as a computer in the network, with associated software, that is dedicated to the data storage and retrieval needs required by the AO Controls System. It holds the database management system, the databases and is part file server capable of housing non-database files such as FITS. In addition to providing data persistence, retrieval and data management there will be a telemetry data recorder, an ad-hoc data recorder, an alarm recorder, general logger, and configuration.
FR-3132	Data Storage Capacity	At a minimum the Data Server shall store at least three month's worth of collected data [TBC] with a buffer of 1 month's worth. Once it reaches this limit, the oldest data of a configurable timeframe, i.e. 1 day, 1 week, 10 days, etc., will be automatically deleted.
FR-3367	Data Recording	All data recording will be performed through supplied APIs. It is expected that the underlying control system framework will provide services to support data persistence to the various data stores.

ID	Short Name	Description
FR-3368	Time Stamping	All data stored in the data server should be time-stamped in a consistent manner. The recommendation is that UT is used for all time stamps. The timestamp should have at least a millisecond resolution or better. It is expected that the data server be clock synchronized with the rest of the AO Controls system.
FR-3099	Support systems and services	The data server shall provide the database support for the following systems and services: a. The configurator b. The general logger c. The telemetry recorder d. The ad-hoc recorder e. The alarm recorder
FR-3369	Telemetry Recorder	The Data Server shall provide a telemetry data recorder capable of recording time-stamped structured data. Grouped information will be stored and retrieved in an atomic and consistent manner. Ideally the telemetry recorder should not require the manual use or creation of meta data in order to function. The maximum burst and sustained rates for the telemetry recorder are TBD.
FR-3370	Ad-hoc Recorder	The Data Server shall provide the capability to record any requested set of published AO Controls System data at the fastest rate that the data is published for any specified amount of time. It is expected that data can be archived at a particular scan rate or based on some configurable event.
FR-3371	General Logger	The Data Server shall provide general AO Controls System logging. It will record all messages in a persistent store. The recording will include automatic timestamp and source identification There should be separate categories for any message type. Logging should be capable of being enable/disabled programmatically per component and should be capable of changing its logging level programmatically (log level can be used to mediate density).
FR-3372	Alarm Recorder	The Data Server shall have the capability to record all published alarm data. An alarm recorder will be available to record all generated alarms and alarm transitions. The recorder can be enabled and disabled as appropriate.
FR-3373	Configurator	The Data Server shall provide for the storage and retrieval of system configuration information. Calibration data should be made available for an indefinite period
FR-3374	Auto Recovery	The Data Server shall auto recover from disk failures so an observing nights data is not lost. This could be through the use of a RAID system, disk duplication, etc…
FR-3375	Scalable Data Storage	The data store shall be capable of scaling to increase the amount of data storage as needed.
FR-3376	Permanent Archival Storage of Data	The data server shall support permanent archival storage of both raw and processed data. A permanent archive shall include removable media. The archival mechanism must be able to write to a removable media (tapes, hot-swap disks, or a future transportable media) and possibly HQ through its network interface.
FR-3377	Extract and Save	The data server shall have the ability to save off portions of a database to external files for later processing.
FR-3103	Simultaneous queries	The data server shall support simultaneous high-speed storage and recall (queries).

ID	Short Name	Description
FR-3107	Error handling	The Data Server shall provide error logging, alerts, and messages. It should also make its alarm visible to an operator during operations.
FR-3109	Reliability	The Data Server shall have a MTBF of 10hrs (TBC) or a minimum of a full observing night. An unfortunate loss of the Data Server should not prevent the AO Controls System from continuing to run.
FR-1842	Calibration Control	AO Controls will provide a semi-automated capability to calibrate the AO system, to include computation of DM actuator gains, DM to WFS registrations, centroid gains, interaction matrices, WFS calibrations, etc. The term semi-automated is used here to indicate that the calibration capability shall minimize the amount of user interaction required.
FR-470	Sensor Calibration Measurements	AO controls will provide an automated capability to compute the dark frames, sky frames, and flat fields for all the various sensors in the AO system as required.
FR-1843	Telemetry	AO Controls shall provide appropriate telemetry streams for the other AO subsystems, instruments, telescope control system, and top-level MCS, as required. The list of telemetry streams required by these systems is currently unknown; as they become defined, individual requirements will be written for each stream to be generated.
FR-479	Atmospheric Profiler	AO Controls shall publish atmospheric turbulence profile (Cn2) information from the atmospheric profiler.
FR-1829	Detector Control	AO Controls shall provide a configuration control function for all the detectors in the system. In addition to the basic device control functions, the detector control shall include the setting of exposure times and frame rates, bias fields and flat fields, filter and gain settings, and shutter control, if applicable.
FR-1831	RTC Control	AO Controls shall provide a configuration control function for the RTC. In addition to the basic control functions, the RTC control function shall include the setting/readback of all RTC parameters, such as reconstructor arrays, centroid offsets, centroid gains, loop gains, etc., and loop control for all the AO correction loops in the system, includes both go-to and feedback loops.
FR-467	Altitude tracking	AO controls must provide altitude tracking for the LGS WFS (in LGS mode).
FR-461	Truth Wavefront Sensor	AO Controls is responsible for control of the TWFS and calculating low-order WFS focus correction for the RTC.
FR-3188	Field Rotation Offload	The AO Controls shall compute the field rotation error from the average tip-tilt signal of the NGS low order wavefront sensors as output from the RTC telemetry. The field rotation error should be computed at a rate of 10 Hz (TBC). The AO Controls system will use this information along with Telescope position information to update the position of the AO field derotator. Offload data from the RTC includes the separate average tip/tilts from each NGS LOWFS channel.
FR-3365	Loss of Communications	In the event AO Controls fails to establish communication or losses communication with any controller or service, other than a safety system, it shall generate an appropriate alarm and attempt to continue operations to the best of its ability or enter a safe state depending on the condition. The operator, upon receiving the alarm, will be able to take action as needed.
FR-2483	Overall	This folder contains overall requirements that span multiple engineering disciplines

ID	Short Name	Description
FR-2130	Definition	The LGS control system is defined as all the LGS control functions that are not directly concerned with the high speed measurement and correction of atmospherically distorted wavefronts. Throughout the remainder of these requirements, the LGS control functions will be referred to simply as "LGS Controls".
FR-2131	Basic Functionality	LGS Controls shall provide the following basic functions for the LGS facility: 1. A control system, supporting: Simple and complex commands Synchronous and asynchronous commands Command completion monitoring Error detection, handling, notification and recovery User interfaces supporting graphical, command line, and script modes 2. Device control for all controllable devices in the LGS facility. 3. Motion control for all controllable opto-mechanical devices in the LGS facility. 4. Configuration and control of the laser system. 5. Support for LGS facility configuration, calibration and operations. 6. Logging of all telemetry data to the data server. 7. Monitoring the health of all LGS facility subsystems and providing notification to other systems as required. 8. Control of the Laser enclosure environment. 9. Interfaces to the other NGAO and observatory subsystems: AO System LTCS Telescope Control System Data Server Multi-System Command Sequencer
FR-2143	General Device Control	LGS Controls shall provide a control function for all non-motorized devices in the LGS facility capable of computer control. The devices to be controlled include: detectors, mirrors, and the environmental system for the Laser enclosure. The control function shall provide, at a minimum, power control, device configuration and basic device control (initialize, standby, start, stop, power up/down, read status, etc.), unless these functions are not available. The motorized devices are addressed in a separate requirement on motion control.
FR-2144	Detector Control	LGS Controls shall provide a configuration control function for all the LGS facility detectors. In addition to the basic device control functions, the detector control shall include the setting of exposure times and frame rates, bias fields and flat fields, filter and gain settings, and shutter control, if applicable.
FR-2146	Laser System Control	LGS Controls shall provide a configuration control function for the laser system. In addition to the basic control functions (initialize, standby, start, stop, etc.), the laser control function shall include the setting/readback of all laser parameters.
FR-2147	Environmental Control	LGS Controls shall provide a configuration control function for all the environmental control devices in the laser enclosure that require remote computer control.

ID	Short Name	Description
FR-2149	Motion Control	LGS Controls shall provide a motion control function for all opto-mechanical devices in the LGS facility requiring remote computer control. The control function shall include basic device control (initialize, standby, start, stop, etc.), configuration control, position control, tracking control for those devices that require it, and support coordinated moves of multiple devices. As of this writing, the opto-mechanical devices requiring motion control are as follows (number of degrees of freedom shown in parentheses): Laser Shutter (1 DOF) Polarization waveplates (3 DOF) BTO bottom mirrors (4 DOF) BTO mid mirrors (4 DOF) BTO top mirrors (4 DOF) Shutter (1 DOF) Point and shoot beam splitter (2 DOF) Star imager pickoff (1 DOF) Point and shoot steering (6 DOF) Asterism rotator (1 DOF) Asterism tip/tilt (2 DOF) Fast shutter (1 DOF) Beam expander focus (1 DOF) LTO cover (1 DOF) LTO polarization sensor (1 DOF) LTO focus lens (1 DOF) As more information is known about the motion control needs for these devices and subassemblies, individual requirements will be added to address them.
FR-2150	Motion control coordination	LGS controls will coordinate the LGS facility motion control and other tasks needed to perform dithering, offsetting and chopping for science observations.
FR-3366	Loss of Communications	In the event LGS Controls fails to establish communication or losses communication with any controller or service, other than a safety system, it shall generate an appropriate alarm and attempt to continue operations to the best of its ability or enter a safe state depending on the condition. The operator, upon receiving the alarm, will be able to take action as needed.
FR-2484	Optical	This folder contains optical requirements

ID	Short Name	Description
FR-2132	Command Processor	LGS Controls shall provide a command processor (sequencer) with the following functionality: 1. Processing of simple commands. An example of a simple command is the setting of a single parameter. 2. Processing of complex commands. An example of a complex command is to configure the LGS facility for a particular mode of operation. This will require setting several parameters, configuring any optical devices, and checking for valid completions of all. 3. Processing of synchronous commands. A synchronous command is one where the latency of the task to be performed by the command is negligible. 4. Processing of asynchronous commands. An asynchronous command is one where the latency of the task to be performed is large enough to require spawning a thread to execute the task and monitor for a command completion callback. 5. The return status of every command (or sub-command in the case of complex commands) must be checked for valid completion or error. If an error occurs, appropriate actions must be taken to process the error, abort the command or task if required, notify the requestor of the error, and take action to recover from the error, protecting the LGS facility as required. 6. A command interface that supports graphical, command-line, and script modes.
FR-2133	Health Monitoring	LGS Controls shall provide a health monitoring function that periodically checks all the LGS facility subsystems for status. If an error is detected, LGS Controls will take appropriate actions to protect the LGS facility and provide notification to the other NGAO subsystems and users.
FR-2135	Configuration Control	LGS Controls will provide an automated capability to configure the LGS facility for all modes of the system: startup, operational, calibration, and test. This includes configuring all the required cameras or sensors with the appropriate default parameters and configuring any optical devices as required. The specifics of these configurations will be documented as use cases in the operations concept document.
FR-2136	Calibration Control	LGS Controls will provide a semi-automated capability to calibrate the LGS facility, to include alignment of all movable LGS optical devices. The term semi-automated is used here to indicate that the calibration capability shall minimize the amount of user interaction required.
FR-2138	Telemetry	LGS Controls shall provide appropriate telemetry streams for the other AO subsystems, instruments, telescope control system, and top-level MCS, as required. The list of telemetry streams required by these systems is currently unknown; as they become defined, individual requirements will be written for each stream to be generated.
FR-2141	Copy of Interface to LTCS	LGS controls will respond appropriately to protect the LGS facility and recover gracefully from a laser traffic control event.