google.org

# RE<C:  Pitch/Roll Heliostat Control System Design

## 1. Overview

Our heliostat controller relied on two sets of sensors to accomplish its goals. A reflector mounted 3-axis accelerometer provided rough orientation of the heliostat mirror module, and an on-target light spot position sensing system used multiscopic photometry to measure with high precision the positions of reflected light spots on the target.

In rough orientation mode, the 3-axis accelerometer data is used to estimate the reflector orientation, aligning the reflector with a desired orientation. Orientation error signals are converted into orientation changes for the mirror module, and these orientation changes are converted into actuator commands. In this mode, the heliostat can stow against wind or move sunlight into the vicinity of the target.

In precision control mode, the target spot position sensing system was used to derive error signals for on-target heliostats. These position error signals were converted into desired orientation change rate for the mirror module, and this desired orientation change rate were converted into actuator rate commands for our cable actuators.

The combination of these control modes performed the scenarios required for automatic field operation. The following sections take a deep dive into the supporting math, and control modes and strategies used.

## 2. Design Requirements

For efficient power-conversion, our thermal engines required precise and balanced placement of reflected solar power to reduce thermal stresses and fatigue. We estimated the on-target light spot precision requirement to be on the order of 10cm, or 1 milliradian from the furthest expected heliostat at 100m.

When off-target, heliostats may need to orient a heliostat such that it is in a standby position, able to put its sun on target later in the day or to add some extra heat during a windy period. It may also be necessary to stow the heliostat at night to prevent dew buildup, or to stow it to protect it from high winds. The pointing precision for these operations was not as stringent as for on-target control, but putting sun on target is the most demanding.

To estimate accuracy of a sensor to get a light spot in the target vicinity, we observe that the furthest heliostat in one field design we estimated to be at 100 meters from the receiver target. Our on-target spot position sensing system had a 2 meter capture radius (about twice the spacing of the cameras). Factoring in 2x angle multiplier due to reflection, the angular accuracy of the orientation sensor needed to be:

$$\begin{aligned} allowable\ miscalibration &= 0.5 * (2 * 2) \ / \ 100 \\ &= 0.02\, radians \\ &= 1.2\, degrees \end{aligned}$$

Thus, a 3-axis accelerometer orientation accuracy of approximately 1 degree should be sufficient to place the sun within the target vicinity and capture radius of the on-target spot position sensing system. If this does not immediately succeed, the heliostat orientation could be controlled to move the spot in a spiral search pattern until it is captured by the precision spot position sensing system.

Milliradian accuracy, for on-target positioning required higher accuracy than what was available with the 3-axis accelerometer. We relied on a photometry system that provided 10cm light spot on-target sensing capability, or identified that a light spot was not within its capture radius.
For this reason, there are two dominant modes of control: rough off-target heliostat reflector orientation control, and on-target precision spot location control.

This document starts with analysis leading to heliosat reflector orientation control and then continues further to precise on-target tracking and computational optimizations.

## 3. Alternative Approaches

We did look into other CSP designs as alternatives to using multiscopic photometry. CSIRO in Australia appears to use a calibration target separate from their receiver. Periodic placement of individual heliostats on the calibration target allows them to better predict where the reflected light spot is, given heliostat angle measurements. eSolar CSP power towers, visible in their literature, appear to employ a separate calibration target as well. This approach does not measure how the light spots are positioned while on target.

We believed it was necessary to use receiver-based error measurements when the heliostat is on-target, in order to achieve spot positioning within 10 cm. Kribus et all (2004) demonstrated a multi-camera photometry system that could center a spot on a target using very simple math - essentially a light balance - to center a spot between cameras.

Kribus' approach did not have an explicit mechanism to indicate when the fine position sensor had a valid position signal. We observed that a controller using the sensor-differential error signal worked well near center, but beyond camera radius, would actually move the spot further and further away from the target. This makes it difficult to adopt this technique as-is into the fully automatic control environment needed in CSP power plants. Our multiscopic photometry system included both (x,y) light spot position measurement and capture detection to support automatic transition to and from precise position sensing.

The control law presented by Kribus et al (2004) was admirably simple. We wanted to draw on this approach, but see how simple it could really be for heliostats with different actuation systems, or differences between nearby versus distant heliostats. We also wanted to model and understand the effects of angle-to-target coordinate cross-coupling as a function of the orientation of a heliostat, and including compensation for sun motion in the sky.

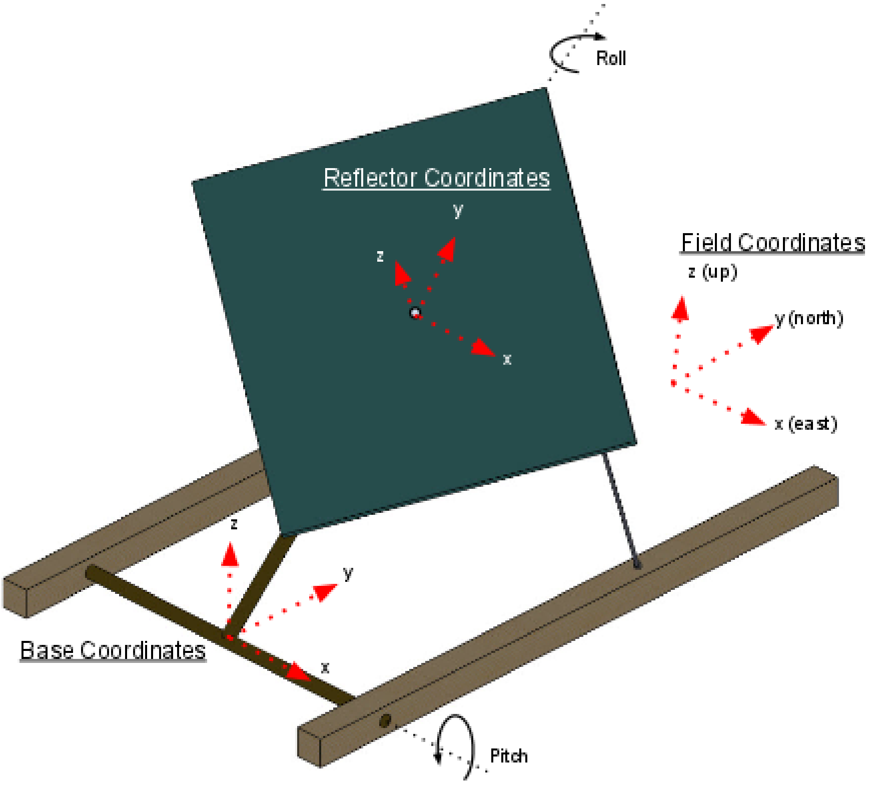
The analysis in this document provides answers to most of these questions, and suggests how to model and compensate for these effects, and then reduce computational overhead of doing so. We have made open source code available that models our controller acting on a simulated heliostat, to aid future research and development of CSP fields.

## 4. Coordinate Systems and Direction Vectors

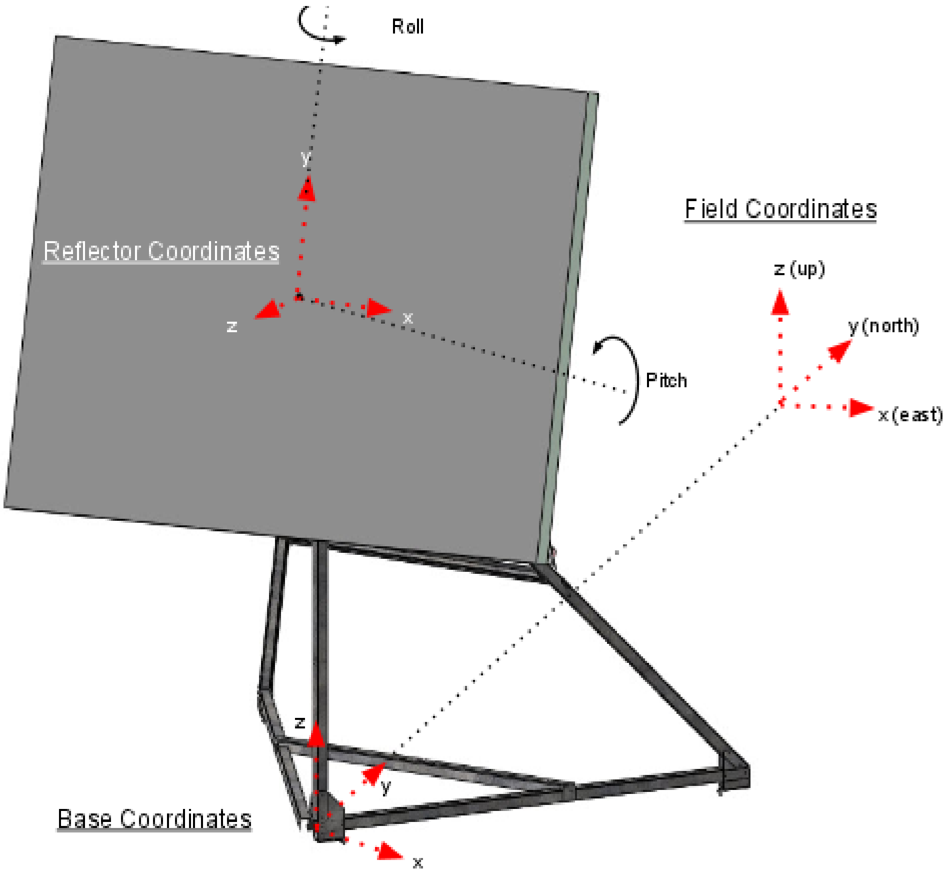
### 4.1. Coordinate Systems

Our heliostats used pitch/roll articulation, which is different from the commonly described azimuth-elevation articulated heliostats. Pitch/roll heliostats first articulate in pitch (from flat to vertical) to move reflected light primarily up or down. They then articulate in roll, which will move reflected light primarily left to right.

Over the course of our experiments, we had several iterations of pitch/roll heliostats. We started with a “catapult” style heliostat, and ended up with a tripod frame with a U-joint articulation. The former and latter are depicted below, with articulation axes and coordinate frames. Fortunately for our computer models, they are mathematically similar.



**Catapult Articulation and Coordinate Frames**



**U-joint Articulation and Coordinate Frames**

Both above figures show three right-handed coordinate systems. The heliostat reflector coordinate system, the base coordinate system, and the field coordinate system.

1. The heliostat reflector coordinate system $\{x_h, y_h, z_h\}$ articulates relative to the base coordinate system via pitch and roll. It aligns with the base coordinate system when pitch and roll angles are zero. The reflector coordinate system has a strong relation to the mirror normal vector $h$. The mirror normal vector $h$ may be equal to $z_h$, but our later heliostats had a mirror normal $h$ at a droop angle offset from the mounting plate of several degrees.[1]
2. The base coordinate system $\{x_b, y_b, z_b\}$ is included for completeness, one could use it to capture imprecise orientation relative to the field coordinates, or heliostats explicitly installed at other than North/South installation alignment. We chose not to model this in this analysis (i.e. intentional alignment of heliostats facing due South or due North). For the derivation we assume perfect alignment between the base coordinates and the field coordinates - and we show later why small errors do not impact performance.

[1] The offset helped impove our accelerometer signal. More information is available in the Heliostat Orientation Using a 3-Axis Accelerometer document.

3. The field coordinate system {$x$ ,$y$ ,$z$ } aligns with { x=East, y=North, z=Up }.

It is important to get the coordinate systems in place for the analysis to make sense.[2] Predominantly vector-based math can keep the computational cost lower than math that relies on a lot of sine and cosine functions. Vector implementation, while not necessarily obvious at first glance, is believed to be less error-prone in coding, easier to debug, and easier to test assertions on during operation.

## 4.2. Reflector Unit Vectors

The unit vectors on the heliostat mirror mount can be expressed in terms of pitch and roll angles relative to the base and field coordinate system. We assumed perfect coordinate system alignment in this derivation (i.e. base coordinates = field coordinates) meaning South or North facing heliostats. Note that perfect installation and calibration are not requirements for operation of this control system.

First we consider unit vectors {$x_1$,$y_1$,$z_1$} after a pitch angle of theta about the x axis from the field coordinate system.

$$x_1 = x$$

$$x_1 = [1\,0\,0] \qquad (4.1a)$$

$$y_1 = [0\ cos(\theta)\ sin(\theta)] \qquad (4.1b)$$

$$z_1 = [0 - sin(\theta)\ cos(\theta)] \qquad (4.1c)$$

Next we express the heliostat coordinate system {$x_h$,$y_h$,$z_h$} by roll rotation about axis $y_1$of an angle $\phi$. Each coordinate system vector is expressed in field coordinates:

$$\begin{aligned} x_h &= cos(\phi)\, x_1 - sin(\phi)\, z_1 \\ &= [cos(\phi)\ sin(\phi)sin(\theta) - sin(\phi)cos(\theta)] \qquad (4.2a) \\ y_h &= y_1 \\ &= [0\, cos(\theta)\ sin(\theta)] \qquad (4.2b) \\ z_h &= cos(\phi)\, z_1 + sin(\phi)\, x_1 \\ &= [sin(\phi) - cos(\phi)sin(\theta)\ cos(\phi)cos(\theta)] \qquad (4.2c) \end{aligned}$$

In practice, the unit vectors are calculated by the orientation sensor system. We discuss in the orientation sensor document (linked from the sensor section below) how this is done without evaluating sine or cosine functions.

The heliostat reflector transformation matrix $H$ can be constructed from the heliostat reflector coordinate system vectors as follows, where prime’ denotes vector transpose:

[2] Many of the derivations employed in the following analysis will be in the vector approach and style promoted by Kane in his 1985 book “Dynamics: Theory and Applications”.

$$H = [\, x'_h \; y'_h \; z'_h \,] \qquad (4.3)$$

The matrix $H$ transforms coordinates in the heliostat reflector coordinate frame into field coordinates.

## 4.3. Angular Rates and Angular Velocity of the Reflector

We will make use of the angular velocity $\omega$ of the heliostat reflector frequently in this document. The angular rate measures for pitch and roll can be expressed along the base (equals field) $x$ unit vector and heliostat $y_h$ unit vectors, decomposing the heliostat mirror module angular velocity vector $\omega$ by projection as follows:

$$\frac{d}{dt}\theta = \omega \cdot x \qquad (4.4a)$$

$$\frac{d}{dt}\phi = \omega \cdot y_h \qquad (4.4b)$$

The angular velocity $\omega$ can be expressed in terms of angular rates - pitch and roll. It is the sum of angular rates along the field $x$ unit vector and heliostat $y_h$ unit vector:

$$\omega = \frac{d}{dt}\theta \cdot x + \frac{d}{dt}\phi \cdot y_h \qquad (4.5)$$

## 4.4. Sensor Data

Our control system used two types of sensor data per heliostat: an orientation sensor of approximately 1 degree precision that could be used to resolve the heliostat reflector basis vectors and mirror normal[3], and secondly, a precision light spot position sensor, good to about 10cm on-target.

**Orientation sensor**

The orientation sensor system uses an accelerometer that gives a direction of gravity. From this it estimates the reflector coordinate system and transformation matrix $H$, and the mirror normal $h$.

**Precision spot position sensing system**

The precision light spot position sensor system provides two pieces of data for each heliostat's light spot:

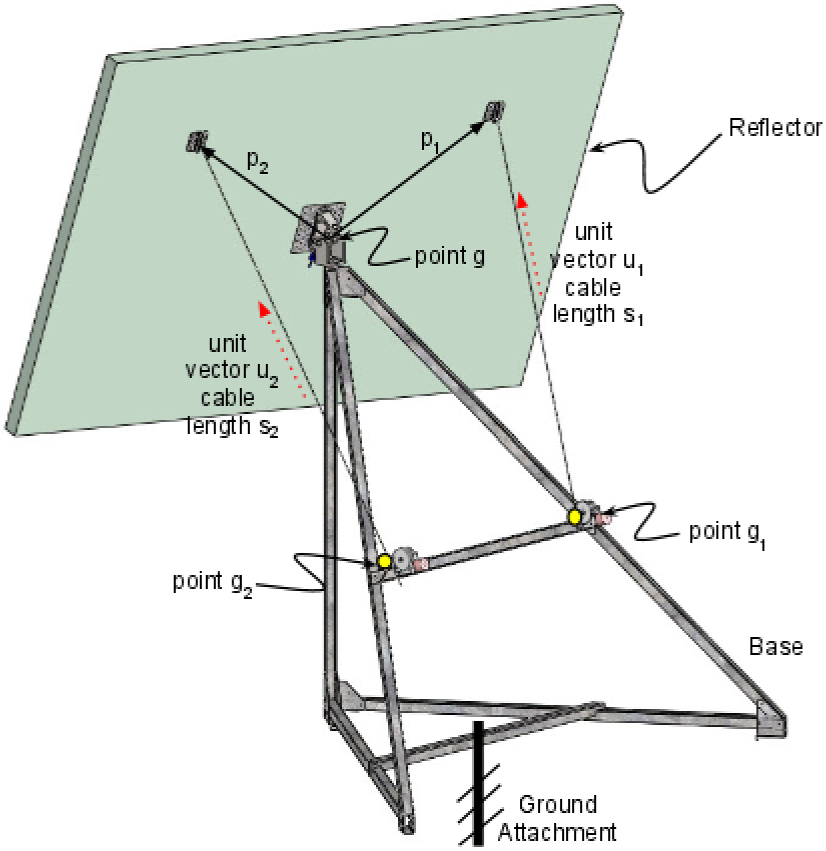
- an indication that this light spot is within the perception range of this sensor, so that we know when the sensor has captured the signal, and,
- an estimate of the position of a light spot on the target plane (e.g. its centroid), expressed as (x,y) coordinates on a target plane.

[3] More information on mirror normal and basis vectors is available in the Heliostat Orientation Using a 3-Axis Accelerometer document.

# 5. Heliostat-to-Actuator Kinematics

## 5.1. From Mirror Angular Rates to Linear Actuator Rates

The linear actuators wind in or play out cables, which reorient the heliostat. As long as the cables are in tension there is a relationship between the linear actuator rates (cable speeds) and the angular rates of motion of the heliostat.


Reflector
p2
p1
point g
unit vector u1 cable length s1
unit vector u2 cable length s2
point g1
point g2
Base
Ground Attachment

**U-joint Heliostat Cable Drive Movement**

The vectors **p1** and **p2** go from the fixed point **g** to the cable attachment points on the heliostat reflector (mirror) module. The coordinates of these points (in field coordinates) can be calculated using the heliostat's fixed attachment point **g**, the heliostat coordinate transformation matrix $H$, and the body vectors $p_{1,h}, p_{2,h}$ in heliostat coordinate space:

$$p_1 = g + H \cdot p_{1,h} \qquad (5.1a)$$

$$p_2 = g + H \cdot p_{2,h} \qquad (5.1b)$$

We define the following cable unit vectors **u** and speeds s for these actuators, in terms of the heliostat angular velocity $\omega$:

$$\frac{d}{dt}s_1 \cdot u_1 = \frac{d}{dt}(g + p_1) = 0 + \omega \times p_1 \quad (5.2a)$$

$$\frac{d}{dt}s_2 \cdot u_2 = \frac{d}{dt}(g + p_2) = 0 + \omega \times p_2 \quad (5.2b)$$

where the cable unit vectors can be computed using the cable attachment points on the heliostat, and where the cable exits the motor drive $g_i$ as

$$u_i = (g + p_i - g_i) \,/\, \| g + p_i - g_i \| \quad (5.3)$$

so by taking the dot product with each cable unit vector $u_i$with (5.3) we can determine actuator speeds s:

$$\frac{d}{dt}s_1 = (\omega \times p_1) \cdot u_1 \quad (5.4a)$$

$$\frac{d}{dt}s_2 = (\omega \times p_2) \cdot u_2 \quad (5.4b)$$

and using the definition of angular velocity from angular rates:

$$\omega = \frac{d}{dt}\theta \cdot x + \frac{d}{dt}\phi \cdot y_h \quad \text{(ref 4.5)}$$

We can express the relationship between linear actuator rates and heliostat angular rates as a matrix transformation using (5.4) and (4.5):

$$[\, \frac{d}{dt}s_1 \; \frac{d}{dt}s_2 \,]' = \chi \, [\, \frac{d}{dt}\theta \; \frac{d}{dt}\phi \,], \quad (5.5)$$

the matrix $\chi$ is a 2x2 matrix, rows separated by semicolon:

$$\chi = [\, x_{1,\theta} \; x_{1,\phi}; x_{2,\theta} \; x_{2,\phi} ] \quad (5.6)$$

the elements are computed using decomposed angular velocity:

$$\chi_{1,\theta} = (\omega \times p_1) \cdot u_1 \big|_{\omega = x} \quad (5.7a)$$

$$\chi_{1,\phi} = (\omega \times p_1) \cdot u_1 \big|_{\omega = y_h} \quad (5.7b)$$

$$\chi_{2,\theta} = (\omega \times p_2) \cdot u_2 \big|_{\omega = x} \quad (5.7c)$$

$$\chi_{2,\phi} = (\omega \times p_2) \cdot u_2 \big|_{\omega = y_h} \quad (5.7d)$$

so:

$$\chi_{1,\theta} = (x \times p_1) \cdot u_1 \quad (5.8a)$$

$$\chi_{1,\phi} = (y_h \times p_1) \cdot u_1 \quad (5.8b)$$

$$\chi_{2,\theta} = (x \times p_2)\cdot u_2 \qquad (5.8c)$$
$$\chi_{2,\phi} = (y_h \times p_2)\cdot u_2 \qquad (5.8d)$$

where the actuator unit vectors $u_i$ are computed as in (5.3). Reflector angular rates can be computed from actuator rates, by inverting (5.5).

$$[\frac{d}{dt}\theta \; \frac{d}{dt}\phi]' = \chi^{-1}[\frac{d}{dt}s_1 \; \frac{d}{dt}s_2], \qquad (5.9)$$

# 6. Rough Orientation Control

Rough orientation control was used to move the mirror to a parking orientation, keep flux off-target, or get the reflected light spot close enough to the receiver target that it could be captured by the precision pointing sensor for precision on-target spot spot position control.

The mirror normal $h$ is used by the control system in orientation control mode to compare with the desired mirror orientation (e.g. stowage pointing, or ideal sun reflection orientation). The control law figures out a corrective angular velocity for the mirror module. Keeping this in vector space rather than in heliostat angle space allows the control system math to focus exclusively on the intended goal (orientation) rather than angles per se - where it is easy to confuse estimated angles with actual angles and imprecisions due to mounting. Orientation can be specified in vector form using a known desired direction, or the halfway direction between the sun unit vector and the target unit vector.

The mirror module (articulation) transformation matrix $H$ is used by the control system for kinematics calculations. For example to get from the desired mirror module angular velocity to linear actuator rates requires a transformation matrix $\chi$ that is a function of the articulation only (pitch & roll *without t*he offset angle of the reflector).

## 6.1. Orientation Control Specified with Angles

Orientation specification using angles is the "traditional way" to control heliostat pointing. Given a desired mirror orientation vector $h$, the target pitch $\theta$ and roll $\phi$ angles for ideal sun pointing can be computed. This follows from equation (4.2c) where $h = z_h$:

$$z_h = [sin(\phi) \; -cos(\phi)sin(\theta) \; cos(\phi)cos(\theta)] \qquad \text{(ref 4.2c)}$$

so

$$\phi = asin(z_{h,x}) \qquad (6.1a)$$
$$\theta = -\,atan2(z_{h,y}, z_{h,z}) \qquad (6.1b)$$

There were several problems with specifying angles for control when using our heliostat design:

1. Our heliostat had no sensors with which to measure the articulation angles $\theta$ and $\phi$, so controlling to these angles is not practical.
2. Our heliostats were imprecisely installed and oriented on the field. The actual angles to control to would be different from the angles determined above due to installation imprecision.

3. In later iterations of our heliostat design, an angle offset $\delta$ was used between the mirror and articulation angles $\theta$ and $\phi$ of the universal joint. This avoided problems with measurements from the 3-axis accelerometer. Equation (4.2c) becomes more complicated if this is factored in.

For these reasons, we expressed our orientation control law using errors in the orientation vector and desired angle rates, rather than desired articulation angles.

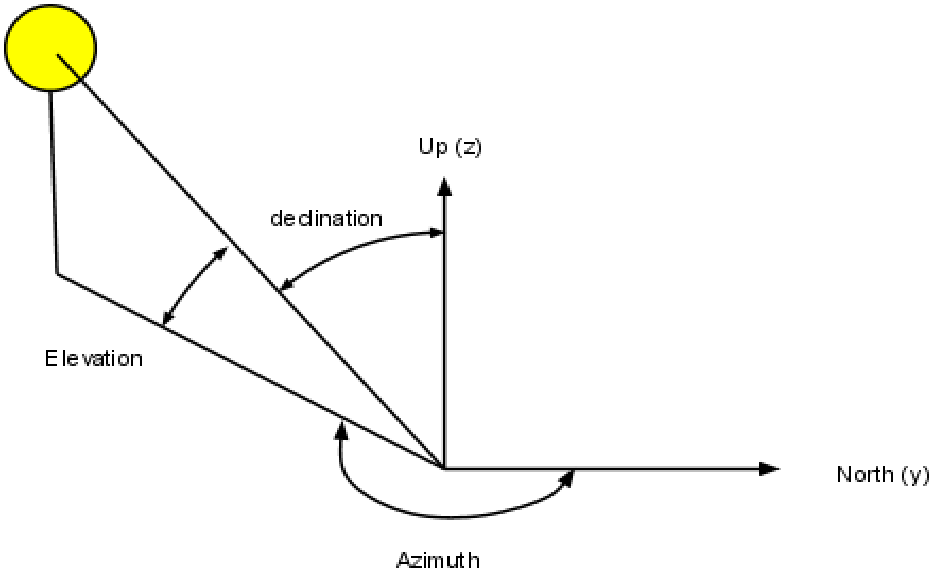
## 6.2. General Pointing Orientation Specification

There are situations where we want to orient a heliostat in a fixed way, to withstand wind, or ease drainage of condensation. These orientations are relatively simple to calculate using geometry. The more complicated case is when we want to orient reflected light to some spot on (or near) the receiver target.

## 6.3. Reflector Orientation Specification

A desired heliostat orientation or mirror reflection can be calculated using knowledge of the sun's direction and knowledge of a desired position on target. The principle is very simple: the mirror normal needs to be placed exactly between the sun ray and the ray to the target.

The sun position computation (a.k.a. ephemeris calculation) has been documented by NREL and an implementation we used was available in the Matlab Central code repository. Also available are a C implementation from NREL, and an open-source Python implementation.



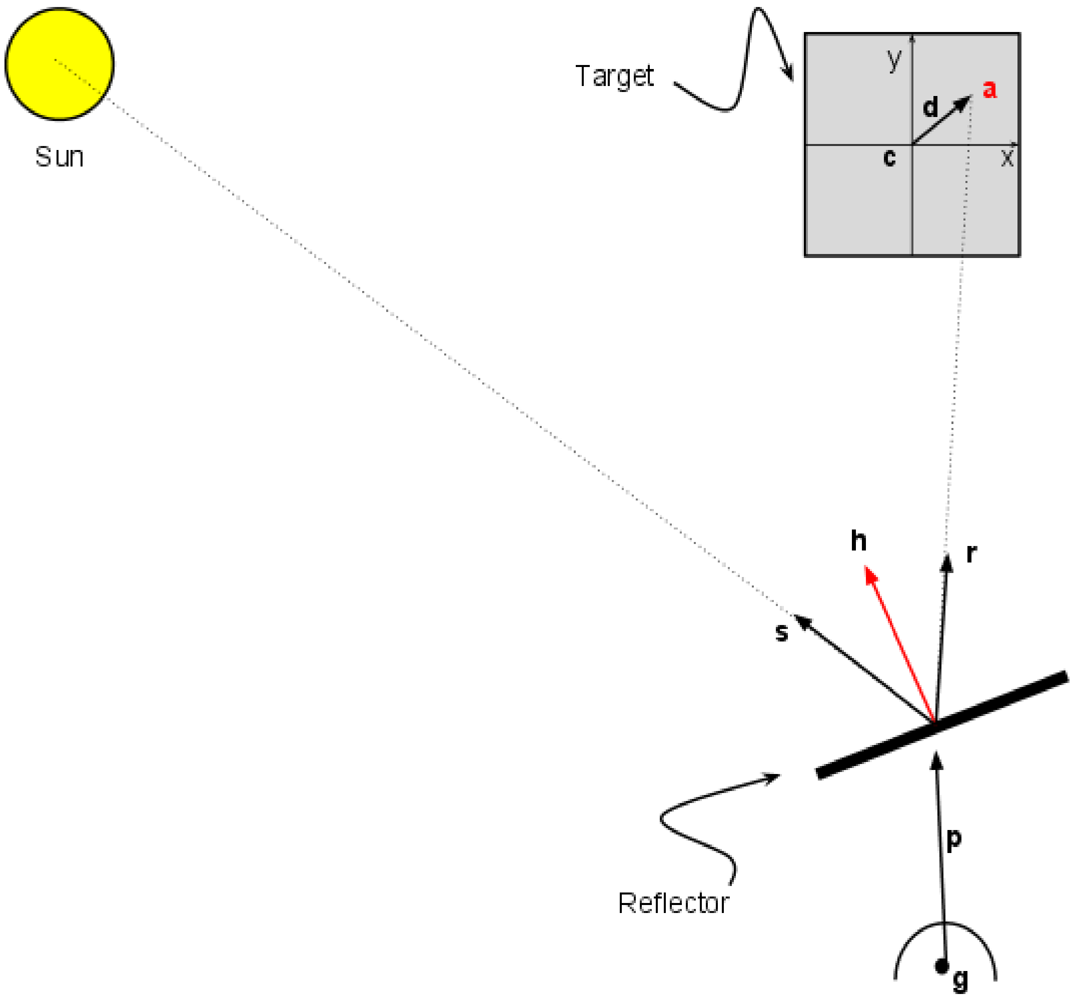
**Sun Azimuth and Elevation/Declination**

Sun position algorithms typically provide sun azimuth $(\alpha_s \, or \, A)$ and elevation $(\theta_s \, or \, h)$ or declination $(\delta \; or \; z)$ (the complement of elevation) angles given location on the planet and time of day. These angles can be used to create a unit vector **s** (in field coordinates) that points in the direction of the sun.

Derivations below use the sun unit vector **s**, which points in the direction of the sun. We express the sun unit vector **s** in the field coordinate system using sun azimuth and elevation:

$$s = [\ cos(\theta_s)sin(\alpha_s)\ cos(\theta_s)cos(\alpha_s)\ sin(\theta_s)\ ] \qquad (6.2)$$

The ideal mirror normal can be specified as exactly halfway between the sun ray and the ray to the desired position on target. The diagram below illustrates the target with center point **c,** reflected light spot center at desired position **a,** the vector **d** from **c** to **a**. If the mirror is pointed at the center of the target, the vector **d** is zero.



**Mirror Reflection Vectors**

The heliostat articulates starting at some fixed point **g**, such as a joint, with vector **p** to the center of the mirror surface. The sun unit vector **s** and ray-to-target unit vector **r** are shown along with the heliostat mirror normal vector **h**. The fixed point **g** by definition is located just before the articulation. On the early “catapult” version, this was near ground level at the center of the pitching axis, and the vector **p** was the pole that pitched up with the mirror. On later U-joint models, the fixed point was at the center of the U-joint and the vector **p** was the short distance from the center of the U-joint to the center of the mirror.

The desired point on target is fed into this formula, where a is the coordinates of that spot (in field coordinate system), the target coordinate transformation matrix T, and the (x,y) coordinates on the target plane d:

$$a = c + T \cdot d \qquad (6.3)$$

The ray to the target point **a** is

$$R = a - (g + H \cdot p) \qquad (6.4)$$

and this is normalized to create the ray unit vector

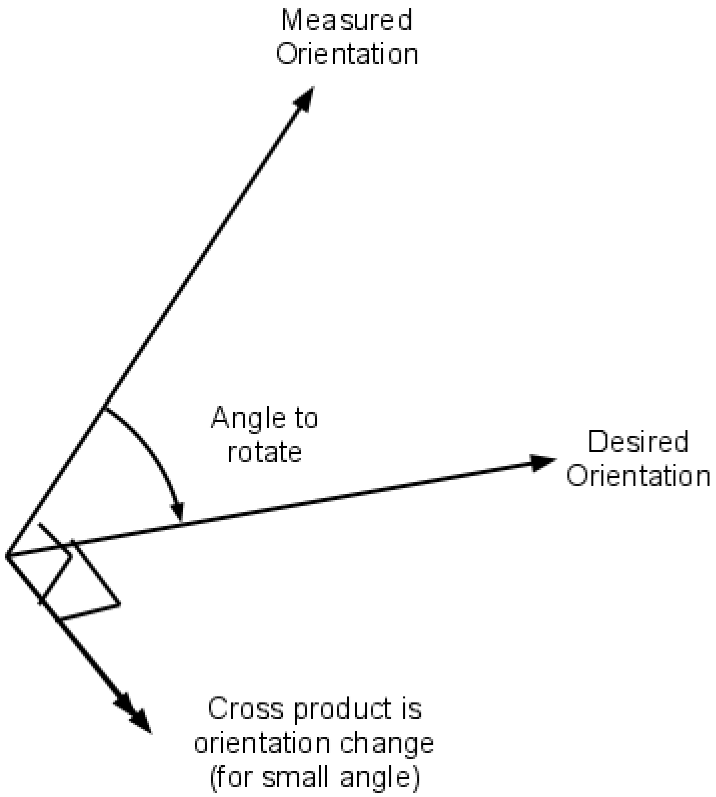
$$r = R \;/\; \|R\| \qquad (6.5)$$

The desired orientation to orient the mirror correctly is where the heliostat mirror normal $h$ is halfway between the sun unit vector $s$ and the target ray unit vector $r$.This can be calculated by taking the average of vectors s and r, and normalizing the result:

$$h_{desired,reflection} = (s + r) \;/\; \|s + r\| \qquad (6.6)$$

### 6.4. Orientation Feedback Control Law

Our reflector orientation sensor subsystem provided a measurement of the mirror normal vector $h_{measured}$ . The desired heliostat orientation $h_{desired}$, whether fixed or target-oriented is specified according to the math in section 6.3. The control law uses the difference (e.g. rotation vector) between the desired orientation and current orientation, to determine a desired angular velocity so as to slew the heliostat to reduce this error to zero, as shown in the diagram below.



**Orientation Correction Expressed as a Rotation Vector**

Our orientation control law translates the orientation correction into a body slew rate magnitude and direction $\omega_{desired}$ via a control gain $k_{gain}$. The selection of $k_{gain}$determines the first order response time characteristic ($\tau$) of the system. For our sample rates of once per second, we used a gain k of 1/10 for a tau for a time constant of about 10 seconds.

$$\omega_{desired} = k_{gain} \cdot (h_{measured} \times h_{desired}) \tag{6.7}$$

Note that this control law expresses a desired behavior (angular velocity) in terms of the mirror normal vector and measured orientation. To achieve this desired angular velocity, we decompose this desired angular velocity into angular speeds using equations (4.4a,b):

$$\frac{d}{dt}\theta_{desired} = \omega_{desired} \cdot x \tag{6.8a}$$

$$\frac{d}{dt}\phi_{desired} = \omega_{desired} \cdot y_h \tag{6.8b}$$

Finally, the angular speeds can be translated into desired actuator rates using the matrix $\chi$ as presented in equations (5.5):

$$[\frac{d}{dt}s_1 \; \frac{d}{dt}s_2] = \chi \; [\frac{d}{dt}\theta_{desired} \; \frac{d}{dt}\phi_{desired}] \tag{ref 5.5}$$

The pointing angle error, in radians, can be approximated for logging or display to the operator. Consider the definition of the magnitude of the vector cross product:

$$\left\| h \times h_{desired} \right\| = \|h\| \cdot \left\| h_{desired} \right\| \cdot sin(\alpha_{error}) \qquad (6.9)$$

so for small angles,

$$\alpha_{error} \approx \left\| h \times h_{desired} \right\| \;/\; ( \|h\| \cdot \left\| h_{desired} \right\| ) \qquad (6.10)$$

and the unit vector magnitudes are 1, so

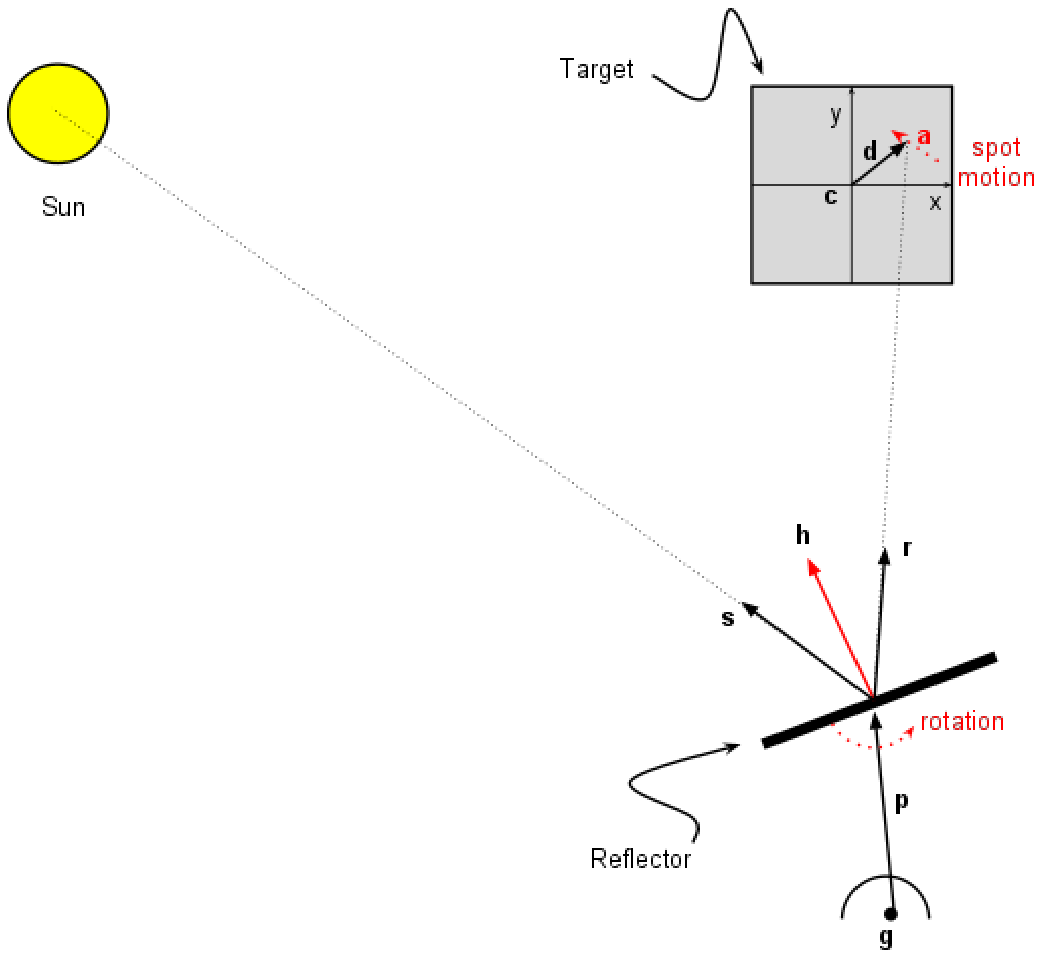
$$\alpha_{error} \approx \left\| h \times h_{desired} \right\| \qquad (6.11)$$

## 7. Heliostat-to-Light-Spot-Position Kinematics

The angular velocity of the heliostat impacts the motion of a spot of light on a target plane. We combined this information with the heliostat-to-actuator kinematics to map desired spot motion on a target plane into desired linear actuator rates.

### 7.1. Heliostat Angular Rates to On-Target Spot Speeds

The position **d** and velocity of the center of a spot from a ray of light from the sun intersecting the target plane, reflected off the heliostat onto the target can be computed by first finding the origin (mirror center) at the end of vector **p** (whose ground base is at point **g**) and direction (unit vector) of the reflected ray **r**. The intersection of that ray with the target **d** (expressed in target (x,y) coordinates can be found using vector math. The diagram below shows the rays and highlights the rotation of the reflector and motion of the spot on target:



**Heliostat Rotation to On-Target Spot Motion Geometry**

The reflected ray unit vector **r** can be determined from the sun unit vector **s** and mirror normal h:

$$r = -s + 2(s \cdot h)h \qquad (7.1)$$

The intersection point **a** of this ray on the target plane can be expressed as vector **d** from target center **c**, as well as distance $l$ along unit vector **r** from heliostat mirror center at the end of vector **p**:

$$a = c + d \qquad (7.2)$$
$$a = g + p + l * r \qquad (7.3)$$

so combining (7.2) and (7.3):

$$c + d = g + p + l * r \qquad (7.4)$$

Take the dot product of (7.4) with the target plane's normal vector $n$. This cancels vector $d$ since the vector $d$ is in the target plane.

$$c \cdot n + 0 = (g + p) \cdot n + l * r \cdot n \qquad (7.5)$$

solving for $l$, the length of the ray from the heliostat center to target intersection is:

$$l = \frac{(c-(g+p))\cdot n}{r\cdot n} \tag{7.6}$$

substitute (7.6) into (7.4), yields d, the “off center” vector on the target as a function of the vectors {**g**,**p**,**c**,**n**,**r**):

$$\begin{aligned} d &= g + p - c + l\, r \\ &= g + p - c + \frac{(c-(g+p))\cdot n}{r\cdot n} * r \\ &= F(g,p,c,n,r) \end{aligned} \tag{7.7}$$

To determine the velocity of the spot **a** on the target, we expand the derivative of equation 4.5 into a sum of partial derivatives, collecting terms in **p** and terms in **r:**

$$v = \frac{\partial F}{\partial p}\frac{dp}{dt} + \frac{\partial F}{\partial r}\frac{dr}{dt} \tag{7.8a}$$

$$= v_p + v_r \tag{7.8b}$$

the components of v from **p** are:

$$v_p = \frac{d}{dt}p + \frac{r}{r\cdot n} * \frac{d}{dt}(-p)\cdot n \tag{7.9a}$$

and noting that the derivative of **p** is equal to the angular velocity cross vector **p** as discussed by Kane and Levinson[4]

$$v_p = \omega \times p - \frac{r}{r\cdot n} * (\omega \times p)\cdot n \tag{7.9b}$$

the components of v from **r** are:

$$v_r = ((c-(g+p))\cdot n) * \frac{d}{dt}\left(\frac{r}{r\cdot n}\right) \tag{7.10a}$$

$$= ((c-(g+p))\cdot n) * \frac{1}{(r\cdot n)^2} * (((r\cdot n)\frac{d}{dt}r) - (\frac{d}{dt}r \cdot n)r) \tag{7.10b}$$

$$= \frac{l}{(r\cdot n)} * ((r\cdot n)\frac{d}{dt}r) - (\frac{d}{dt}r \cdot n)r) \tag{7.10c}$$

where from equation (4.1) dr/dt is:

$$\frac{d}{dt}r = 2\,(s\cdot(\omega\times h))\,h + 2\,(s\cdot h)\,(\omega\times h) \tag{7.11}$$

[4] See Kane and Levinson, Dynamics, Theory and Applications, chapter 2 section 1

By definition, the transformation matrix for the Target coordinate system formulated using its unit vectors (each expressed in field coordinates) is:

$$T = [\ x'_T\ y'_T\ z'_T\ ] \quad (7.12)$$

The point-on-target vector **d** and the velocity-on-target **v** can expressed in the target coordinate system by matrix multiplication with the inverse of the matrix **T**.

$$d_{(target)(x,y)} = T^{-1} \cdot d_{(field)} \quad (7.13)$$

$$v_{(target)(x,y)} = T^{-1} \cdot v_{(field)} \quad (7.14)$$

where we ignore the z component on the target because they are zero.

We now turn to expressing speed of a light spot on the target as a function of the angular velocity of the heliostat reflector. Angular velocity in terms of pitch and roll from (4.5) is:

$$\omega = \frac{d}{dt}\theta \cdot x + \frac{d}{dt}\phi \cdot y_h \quad \text{(ref 4.5)}$$

$$\frac{d}{dt}\theta = \omega \cdot x \quad \text{(ref 4.4a)}$$

$$\frac{d}{dt}\phi = \omega \cdot y_h \quad \text{(ref 4.4b)}$$

So with equations (7.12) and (7.14) and (4.12) we can express the relationship between point-on-target velocity $v$ and heliostat angular rates.

$$[\ v_x\ v_y\ ] = \psi\ [\ \frac{d}{dt}\theta\ \frac{d}{dt}\phi\ ] \quad (7.15)$$

where $\psi$ is constructed in two parts, a target transformation matrix for just x and y on target (2x3), and a map from angular speeds to on-target speeds (3x2):

$$\psi = (T^{-1})_{x,y}\ V_{\omega} \quad (7.16a)$$

$$V_{\omega} = [\ v|'_{\omega=x}\ v|'_{\omega=y_h}\ ] \quad (7.16b)$$

where using equation (7.8), $v|_{\omega=x}$ is the x,y components of on-target velocity computed given the angular velocity equal to the vector $x$, and $v|_{\omega=y_h}$ is the x,y components of on-target velocity computed given the angular velocity equal to the vector $y_h$.

The transformation matrix $\psi$ relates angular speeds (in heliostat pitch, roll) to the resultant motion of the point d on the target plane, expressed as coordinates in the target plane. The components of the matrix $\psi$ are dependent on the vectors $s, h$ and $y_h$, which are a function of sun position and heliostat orientation. These vectors and this matrix need to be recomputed periodically as sun position and heliostat orientation change.

NOTE: By using differentials instead of derivatives, transformation (7.15) can be used to map small position errors on target $[\,\delta x\;\delta y\,]$ into small changes in orientation $[\,\delta\theta\;\delta\phi\,]$. This could be useful for estimating the impact of wind, where we can measure a change in a spot position of $[\,\delta x\;\delta y\,]$, and from that estimate heliostat orientation movement.

### 7.2. Inverse Kinematics: From On-Target Spot Speeds to Linear Actuator Rates

We showed in section 7.1 how heliostat angular rates relate to light spot on target speeds (equation 7.16). In the Heliostat-to-Actuator kinematics (section 5), we also showed how heliostat angular rates relate to our linear actuator motion (equation 5.5):

$$[\,s_1\;s_2\,]' = \chi\;[\,\frac{d}{dt}\theta\;\frac{d}{dt}\phi\,]' \qquad \text{(ref 5.5)}$$

$$[\,v_x\;v_y\,]' = \psi\;[\,\frac{d}{dt}\theta\;\frac{d}{dt}\phi\,]' \qquad \text{(ref 7.15)}$$

These equations (5.5) and (7.15) can be rearranged to translate on-target speeds $[\,v_x, v_y\,]$ into linear actuator rates $[\,s_1, s_2\,]$ and is used in the light spot position feedback control system below:

$$[\,s_1\;s_2\,]' = \chi\,\psi^{-1}\,[\,v_x\,v_y\,]' \qquad (7.17)$$

## 8. On-Target Reflected Light Spot Position Control

Due to the 10 cm pointing precision required for on-target control, our complete on-target control law consists of both feedback and feed-forward components. First, we introduced the error-based feedback control law component, then followed it with feed-forward compensation for the sun’s constant motion across the sky. Finally, we brought them together into a complete on-target light spot position control law.

### 8.1. Spot Position Feedback Control

We selected a feedback control law that produces a first-order error response. This is valid as long as inertia effects (e.g. motor acceleration to speed) are about an order of magnitude faster in the time domain from the control system time constant. The dynamics of the heliostat (in particular, inertia, stiction and flexibility) were not modeled - of these only flexibility proved to have observable effects.

A first-order response means the on-target error will experience an exponential decay with the time constant (1/k). Picking the value of the feedback gain “k” requires some care. As a rule of thumb the time constant (1/k) should be an order of magnitude longer than the sampling interval. Our controller had a sampling interval of 1 second, so we used a gain k of 0.1.

We took the approach of measuring the heliostat on-target error - using a control law to yield a corrective on-target speed - and then transform on-target speeds via heliostat reflector angular rate into desired cable drive actuator rates.

The two equations describing the position error feedback control scheme involve turning an (x,y) position error into a desired on-target spot speed to fix the error using a control gain, and then transforming this spot speed into actuator rates. The equations are basically:

$$\{desired\ x{,}y\ velocity\} = -k_{gain} * \{target\ x{,}y\ position\ error\} \tag{8.1}$$

$$\{actuator\ rates\} = transform(\{desired\ x{,}y\ velocity\}) \tag{8.2}$$

and therefore from (7.17) and (8.1,8.2) the actuator control rates can be computed using this control law, which acts on an error signal$[\, d_x\, d_y \,]$ that describing how far off the desired position on target that spot is:

$$[\, \frac{d}{dt}s_1 \, \frac{d}{dt}s_2 \,]' = -k_{gain} * \chi\, \psi^{-1} [\, d_x\, d_y \,]' \tag{8.3}$$

Clearly, this control law requires that the light spot is perceived by the light spot position sensor.

To provide the operator with more data for a dashboard, we can compute effective desired angle rates (from 7.15 and 8.1,8.2) and also small angle errors corresponding to the measured on-target error:

$$[\, \frac{d}{dt}\theta \, \frac{d}{dt}\phi \,]' = -k_{gain}\, \psi^{-1} [\, d_x\, d_y \,]' \tag{8.4}$$

$$[\, \Delta\theta\ \Delta\phi \,]'_{error} = \psi^{-1} [\, d_x\, d_y \,]' \tag{8.5}$$

As the on-target error decreases, so do the desired on-target rates and linear actuator rates. If the on-target error is exactly zero, no change in the linear actuators is commanded. In practice, error angles converge to near-zero because the sun is always moving in the sky.

### 8.2. Sun Motion Feed-Forward Control

A heliostat needs to move constantly to compensate for the sun's motion in the sky. While a heliostat using the feedback control law (8.1) above tracks close to its intended location on target, it will have a residual error. This residual error provides the actuators with the tracking rate necessary to keep up with the motion of the sun. A consequence is that the spot on target always lags the sun.

How large is this residual error? The sun performs a full rotation relative to the earth once per day. By rearranging equation (8.4) and using our feedback gain $k = 0.1 / sec$, with the furthest heliostat 100 meters from the target:

$$[\, d_x \;\; d_y \,]' = \frac{1}{k}\,\psi\,[\,\frac{d}{dt}\theta_{commanded}\,\frac{d}{dt}\phi_{commanded}\,]' \qquad (8.6)$$

If we reduce this to a scalar approximation then we can compute the approximate magnitude of the holdoff error based on a 100m reflected light path.

$$\|d\| = \frac{1}{k}\,\|\psi\|\;\|rotation\ rate\| \qquad (8.7a)$$

we substitute an approximation for the scalar value of the matrix $\psi$ (see section 9):

$$= \frac{1}{k}\,(2\,l)\;\;\omega_{earth}$$

and then substitute the length 100m and earth’s rotation rate:

$$\begin{aligned} &= 10 * 2 * 100 * \; (2\,\pi \;/\; 60 * 60 * 24) \\ &= 0.15\; m \qquad (8.7b) \end{aligned}$$

This suggests a standoff residual error of 15cm without the use of feed-forward. Our pointing accuracy goal was 10cm, so for distant heliostats, we violate our pointing requirement without compensating for sun motion in the sky.

We will use knowledge of the sun’s motion in the sky as a feed-forward component to aid real-time tracking accuracy. This feed-forward term can also provide guidance to the heliostat in the event of short-term loss of spot location system signal, or major cloud events.

A traditional approach to the feed-forward would be:

$$\frac{d}{dt}\,commanded\ angles = \frac{d}{dt}\,ideal\ sun\ angles \qquad (8.8)$$

where the ideal angle rate is approximated at run time by examining the difference between sequential ideal heliostat angle values:

$$\frac{d}{dt}\theta_{ideal} \approx \Delta\,\theta_{ideal} \;/\; \Delta T \qquad (8.9a)$$

$$\frac{d}{dt}\phi_{ideal} \approx \Delta\,\phi_{ideal} \;/\; \Delta T \qquad (8.9b)$$

and this translates into feed-forward linear actuator rates using equation (5.5):

$$[\; s_1 \; s_2 \,]' = \chi\; [\; \Delta\theta_{ideal} \,/\, \Delta T \;\; \Delta\phi_{ideal} \,/\, \Delta T \;]' \qquad (8.10)$$

In section 6.1 we discussed that basing references off of angles was not that useful for our heliostat design. An alternate way to calculate feedforward angular rates is to determine the difference between successive ideal mirror unit vectors.

Feed-forward can be computed from successive “ideal mirror orientation” **h** vectors, and expressed as a desired feed-forward heliostat reflector angular velocity:

$$\omega_{feedforward} = (\frac{1}{\Delta T})\,(h_{desired(t-1)} \times h_{desired(t)}) \qquad (8.11)$$

then decompose this angular velocity into angular speeds using equations (4.4a,b)

$$\frac{d}{dt}\theta_{feedforward} = \omega_{feedforward} \cdot x \qquad (8.12a)$$

$$\frac{d}{dt}\phi_{feedforward} = \omega_{feedforward} \cdot y_h \qquad (8.12b)$$

Ultimately, these transform into actuator commands using equation (5.5). We show how this comes together in the next section.

Feed-forward control ensures that the actuators are by default commanded at rates very close to those required to track the sun, even with zero error on-target.

## 8.3. Complete Reflected Light Spot Position Control Law

The complete control law is the sum of the feedback (8.3) and feed-forward (8.12) components. The commanded actuator rates are:

$$[\, s_1\ s_2 \,]'_{commanded} = \chi\, [\, \frac{d}{dt}\theta_{feedforward} \frac{d}{dt}\phi_{feedforward} \,]' - k * \chi\, \psi^{-1}\, [\, d_x\, d_y \,]' \qquad (8.13)$$

We used a control gain $k = 0.1$. The feedforward rate was computed using sequential ideal mirror orientation normal vectors, the on-target position error for the given heliostat spot is in meters (provided by a photometry sensor subsystem), the transformation matrix $\psi$ is as computed in (7.16) and transformation matrix $\chi$ is as computed in (5.5) .

## 8.4. Control Strategy

The control strategy is a set of rules that determine which control law and goals to apply to meet the overall goals of the heliostat field. For example, from sunset until sunrise, the coarse orientation controller is used to hold the heliostat in a night parking orientation, at sunrise, rough orientation is used to bring successive heliostats to place their light spots in the area of the receiver target, corresponding to receiver warm-up. Once power generation starts, the receiver is maintained at a desired operating point by bringing on-target (or taking off-target) heliostats, keeping those off-target at some parking orientation relative to the target. As reflected spots come near target, the photometry capture detection is used to trigger the transition from rough orientation control to precision on-target spot position control. For our experiments, we coded this as a finite state machine.

## 9. Computational Optimization

### 9.1. Approximation to $\psi$ Transformation Matrix

Experimental demonstrations convinced us that a less computationally intensive approach was feasible. The first approach was to do something very much like Kribus et al. by substituting a trivial matrix $\psi$. In the case of our pitch-roll heliostat, with a north-facing target, this trivial matrix $\psi$ can be determined by observing the precisely-computed values for $\psi$ and simplifying to:

$$\psi_{trivial} = [\,0\ -2l;\ -2l\ 0\,] \qquad (9.1)$$

The quantity $l$ is the slant distance from the heliostat to the receiver target. The reason for the value $2l$ in the non-zero positions is that angle changes translate into on-target positional changes proportional to the throw distance and twice the angle the reflector is moved.

This simplified $\psi_{trivial}$ matrix appeared to work fairly well, but upon comparison with the ideal in equation (7.16), one can see that where this simplified matrix has zero terms, the ideal transformation matrix can have terms approaching 30-40% cross-coupling. This resulted in a slight spiral in the error-correction response motions, which delays settling a small amount and is a bit energy wasting in excess actuator motion.

The matrix $\psi$ is dependent on the sun and heliostat orientations. Given that this matrix varies very slowly for a typical tracking heliostat, this matrix would only need to be updated about once an hour. By spreading out updates of $\psi$ for the various heliostats, the average computational load for the feedback control system is a matrix multiply (very low), while near ideal control performance can be achieved. A similar argument applies to the $\chi$ matrix.

It also turns out that having an “approximately” correct $\psi$ transformation is very useful during operation. Software assertions that check that instantaneous computations of the $\psi$ matrix are close to the last update, and not hugely different from the trivial matrix rapidly flags bugs. This can prevent unwanted control malfunction.

## 10. Sensitivity Analysis

This proposed control approach allows heliostat cost reduction because heliostats do not need to be precisely situated, nor precisely aligned and calibrated, and have a lighter cheaper structure with some flexibility. We demonstrated this to ourselves with some of our experiments, but this section seeks to justify this prediction. The following two subsections discuss how heliostat installation imprecision is expected to impact on-target control.

### 10.1. Residual Feedback Error due to Imprecise Heliostat Installation

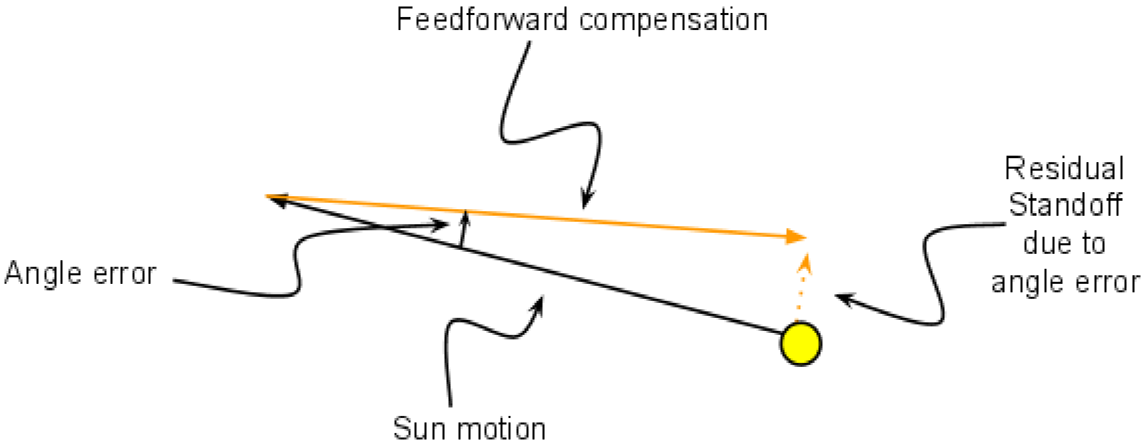
In our control law (equation 8.13) , the feedback terms apply corrective action only if the beam spot is not where it is supposed to be. Imagine the sun is not moving in the sky. If the spot is where it is supposed to be, the feedback terms exert no control effort. This is a subtle but important point: imprecise heliostat installation will *not* result in steady-state residual or standoff errors from the feedback terms of the control law.

If the heliostat spot is off-center (in error), then imprecise installation of a few degrees results in a matrix $\psi$ and matrix $\chi$ that have values that deviate by a few percent from perfect. This impacts the feedback gains and cross-coupling between x and y axes and hence the error response characteristics, but since the matrix $\psi$ has a strong positive (anti-)diagonal, the impact on desired angular rates is that they will be a few percent greater or smaller than they would ordinarily be and there may be several percent cross-coupling.

Practically, what this means is that the control system may be a few percent more sluggish in responding, and the beam spot response to error may result in a barely detectable spiral approach to zero-error, instead of the optimal straight, damped approach to zero-error. However, here will be no persistent standoff error from the feedback controller due to imprecise heliostat installation if the sun is stationary in the sky.

## 10.2. Residual Feed-forward Error due to Imprecise Heliostat Installation

In our control law (equation 8.13) , the feed-forward terms are there to cancel out spot-on-target motion due to the sun moving in the sky. If the heliostat's installation is off by a few degrees in orientation, then the compensating feed-forward terms can act in a manner that does not completely cancel out sun motion in the sky.



**Imprecise Cancellation of Sun Motion with Feed-Forward**

If the imprecision in orientation is a few degrees, then the major component of error in cancellation is:

$$residual\ standoff = standoff \cdot sin(orientation\ error) \quad (10.1a)$$

Combining maximum standoff in equation (8.7) and an orientation error of 5 degrees, one would expect to see a maximum stand-off (i.e. persistent) on-target spot position error of:

$$\begin{aligned} residual\ standoff &= 0.15\ sin(5\ degrees) \\ &= 0.013 \text{ meters} \end{aligned} \qquad (10.1b)$$

This is about an order of magnitude less than our desired precision of 10 cm, and so It is anticipated that the feedforward control system will not introduce any significant residual error due to imprecise heliostat installation.

## 11. Conclusions

In order to be cost effective, heliostat fields need to operate automatically, with little operator intervention. Therefore, our field control system needed to keep a set of heliostats operating automatically - starting up automatically in the morning, putting light on target in distinct locations on a target, recovering from cloud events, moving into or out of standby positions, and stowing at night.

We wanted to achieve precise spot-on-target control of a heliostat with low- precision installation (e.g. off by several degrees, or placement off by a meter). We wanted to avoid calibration, and use minimal sensors on the heliostat. On top of all this, our heliostat was more flexible than the norm, and could bend a bit in the wind.

To demonstrate success we operated several heliostats over a period of time, and transitioning control goals over time. Our experiments and demonstration videos in the Heliostat Control and Targeting document offered proof that this control system could do this. We have provided open source Matlab control software for those interested in pursuing this work further.

In addition, not only did we achieve these goals, but we could do so using a computationally efficient approach that

1. Relied on two quasi-static 2x2 matrices for most of control updates, while
2. Doing periodic matrix updates to achieve precision with low average compute load, and
3. Using vector math extensively, thereby mostly avoiding the use of sines and cosines of angles.

## 12. Further Work

It would have been interesting to gather more data to confirm the expected negligible on-target error in spite of low precision installation, as discussed in sections 10.1 and 10.2.

Some ranges of motion serve no purpose and/or may be destructive to the mirror module. We did not put limit switches or other protection mechanisms into our heliostat design. If one is willing to depend on the 3-axis accelerometer for orientation sensing, a permissible envelope of orientation can be determined and enforced in the control system.

Given that the control work is primarily done using two slowly varying matrices $\chi$ and $\psi$, it may be practical to generate these matrices on-the-fly from observed spot-position-error behavior. Such a form of adaptive control may result in a control system with greater simplicity and even less computation.