



ELSEVIER

Contents lists available at ScienceDirect

MethodsX

journal homepage: www.elsevier.com/locate/mex

Method Article

Method to determine the tracking angles of heliostats



Victor Grigoriev*, Kypros Milidonis, Manuel Blanco,
Marios Constantinou

Energy, Environment and Water Research Center, The Cyprus Institute, 20 Konstantinou Kavafi Street, 2121 Nicosia, Cyprus

A B S T R A C T

The heliostats with two tracking axes are considered, and the method is presented to find the tracking angles for reflection of sun light to a given target. An important advantage of the method is that the tracking axes are not required to be orthogonal like in azimuth-elevation, tilt-roll or target-aligned heliostats. All of these configurations are covered in a unified way, and the presented solution is valid even for arbitrary orientation of tracking axes. The ability to have such a general solution is very valuable, because the orthogonality condition may not hold precisely for manufacturing reasons or due to degradation of heliostats. These deviations need to be corrected properly to achieve a high concentration of sun light. The offsets between tracking axes are also taken into account. However, the targeting problem for heliostats in this case becomes considerably different from the inverse kinematic problems for robotic arm manipulators. It is shown that the tracking angles can be found iteratively, and the convergence of results is very fast for a typical set of parameters used in solar thermal plants. To simplify the use of the method, a Python-library Heliok was developed, and it is demonstrated how to work with it in a Jupyter-notebook. To explain the kinematics of heliostats better, a 3D model of heliostat is provided, which was made and animated in an open-source 3D editor Blender.

The main highlights of the method:

- The tracking axes and the facet of heliostat can have an arbitrary orientation, and there can be offsets between them.
- The tracking problem is solved both for targets attached to heliostat (local aiming) and for separated targets (global aiming).
- The single-axis trackers are included as a limiting case.

© 2021 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>)

DOI of original article: [10.1016/j.solener.2020.07.086](https://doi.org/10.1016/j.solener.2020.07.086)

* Corresponding author.

E-mail address: v.grigoriev@cyi.ac.cy (V. Grigoriev).

<https://doi.org/10.1016/j.mex.2021.101244>

2215-0161/© 2021 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>)

ARTICLE INFO

Method name: Method to determine the tracking angles of heliostats

Keywords: Heliostat, Sun tracking, Solar tracker, Two-axis tracking, Dual-axis tracker

Article history: Received 16 December 2020; Accepted 19 January 2021; Available online 23 January 2021

Specifications Table

Subject Area	Energy
More specific subject area	Concentrating Solar Power
Method name	Method to determine the tracking angles of heliostats
Name and reference of original method	Paden–Kahan subproblems for inverse kinematics [2,3]
Resource availability	<p><i>HelioK repository:</i></p> <p>https://scmt.cyi.ac.cy/bitbucket/projects/ART/repos/heliok</p> <p><i>Anaconda (Python distribution):</i></p> <p>https://www.anaconda.com/products/individual</p> <p><i>Blender (3D editor):</i></p> <p>https://www.blender.org/download</p>

Method details

Armature of heliostat

The rotation of heliostat is described by using an armature (Fig. 1). The armature consists of connected bones, which represent different parts of heliostat. The heads and tails of the bones can be linked by a hinge joint, which supports a rotation around given axis, or a more general joint type. The relative positions and orientations of the bones are defined in local heliostat frame when the primary and secondary angles are set to zero (Fig. 1a). The head of the foundation bone is positioned at the origin of heliostat frame. This bone is always fixed in the heliostat frame, and serves as a parent

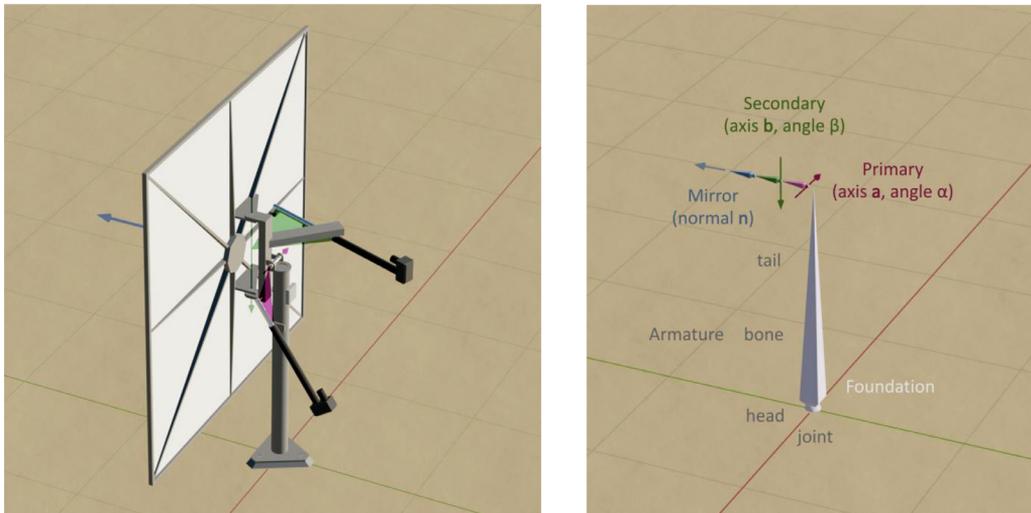


Fig. 1. Heliostat with two tracking drives (a), and its armature (b). The arrows show the directions of primary (purple) and secondary (green) axes as well as the normal of mirror facet (blue). The rotation angles are controlled by changing the expansion length of linear actuators. The triangles show the rotation plane for each axis and the attachment points of linear actuators. The 3D model of heliostat is also provided as a supplementary material. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

for other bones. The primary head has a translation \mathbf{p}_a from the foundation head. The hinge joint in the primary head can rotate around the axis \mathbf{a} in the range $[\alpha_{\min}, \alpha_{\max}]$. The secondary head has a translation \mathbf{p}_b from the primary head. The hinge joint in the secondary head can rotate around the axis \mathbf{b} in the range $[\beta_{\min}, \beta_{\max}]$. The rotation angles around the axes are measured according to the right-hand rule, and the static configuration (Fig. 1a) is used as a reference for zero angles. Notice that there is some ambiguity how to place the armature bones. Only the positions of heads are important where the hinges are, while the positions of tails can be selected to match the geometry of different moving parts. In particular, this was used to make the foundation bone vertical, although the primary axis is slightly shifted from it (Fig. 1b). The default directions of the primary and secondary bones can be used as a reference to define zero angles for the corresponding tracking axes. The mirror bone is fixed rigidly to the secondary bone, and it is necessary just to separate the mirror facet.

Heliostat facets

In general, one or more mirror facets can be attached to the secondary bone. They can be described by appropriate translations \mathbf{p}_m and rotations \mathbf{R}_m . It is necessary to define a vertex (a combination of point \mathbf{r}_{M0} and normal \mathbf{n}_0), which can be used to determine the main direction of reflection. For a single facet, it usually corresponds to the center of the facet. The position \mathbf{r}_M and normal \mathbf{n} of the tracking vertex depend on the primary α and secondary β angles as (the subscript ‘0’ is used when the values are taken for zero tracking angles)

$$\begin{pmatrix} \mathbf{r}_M \\ 1 \end{pmatrix} = \begin{bmatrix} \mathbf{R}_a(\alpha) & \mathbf{p}_a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_b(\beta) & \mathbf{p}_b \\ 0 & 1 \end{bmatrix} \begin{pmatrix} \mathbf{r}_{M0} \\ 1 \end{pmatrix}, \tag{1}$$

$$\mathbf{n} = \mathbf{R}_a(\alpha)\mathbf{R}_b(\beta)\mathbf{n}_0. \tag{2}$$

Heliostat drives

The positive rotation around the axes is determined by the right-hand rule. The rotation matrix around the axis \mathbf{a} for angle α is computed as

$$\mathbf{R}_a(\alpha) = \cos \alpha \mathbf{I} + \sin \alpha \mathbf{a}^\times + (1 - \cos \alpha) \mathbf{a} \otimes \mathbf{a}, \tag{3}$$

where \mathbf{I} is the identity operator, \mathbf{a}^\times is a cross product in matrix form, and $\mathbf{a} \otimes \mathbf{a} = \mathbf{a}\mathbf{a}^T$ is a dyadic product. The rotation can be performed by slew drives or linear actuators. The former control the rotation angle directly, while the latter control the rotation angle via expansion length. If the distances from an axis to two attachment points are p and q , the expansion length λ according to the law of cosines is

$$\lambda^2 = p^2 + q^2 - 2pq \cos \alpha. \tag{4}$$

This relation can be inverted to get the mapping $\alpha(\lambda)$.

Heliostat field

The position and orientation of heliostat in global frame is defined by 4×4 transformation matrix

$$\begin{pmatrix} \mathbf{r} \\ 1 \end{pmatrix}_{\text{global}} = \begin{bmatrix} \mathbf{R}_h & \mathbf{p}_h \\ 0 & 1 \end{bmatrix} \begin{pmatrix} \mathbf{r} \\ 1 \end{pmatrix}_{\text{local}}. \tag{5}$$

The rotation matrix of heliostat can be decomposed into 3 Euler angles as $\mathbf{R}_h = \mathbf{R}_x(\varphi_x)\mathbf{R}_y(\varphi_y)\mathbf{R}_z(\varphi_z)$. In general, heliostats can be used to concentrate light on a target attached to it (local aiming) or a separated target (global aiming). A typical example of the former is a heliostat dish with a CPV (Concentrated Photovoltaics) element in the focus. The latter is mostly applied in CST (Concentrated Solar Thermal) power plants.

Local aiming

The unit vector in the direction of the sun is converted to heliostat frame as $\mathbf{s} \rightarrow \mathbf{R}_h^T \mathbf{s}$. Since the aiming point \mathbf{r}_A is specified in the same frame as the facet vertex \mathbf{r}_{M0} , the target vector can be found as $\mathbf{t}_0 = (\mathbf{r}_A - \mathbf{r}_{M0})/|\mathbf{r}_A - \mathbf{r}_{M0}|$, and it does not depend on the tracking angles. The corresponding sun vector according to the reflection law is $\mathbf{s}_0 = (2\mathbf{n}_0 \otimes \mathbf{n}_0 - 1)\mathbf{t}_0$. The tracking angles are found as a rotation of \mathbf{s}_0 to \mathbf{s} (See Sequential rotation).

Global aiming

The unit vector in the direction of the sun is converted to heliostat frame as $\mathbf{s} \rightarrow \mathbf{R}_h^T \mathbf{s}$. The aiming point \mathbf{r}_A can be converted from global to heliostat frame as $\mathbf{r}_A \rightarrow \mathbf{R}_h^T(\mathbf{r}_A - \mathbf{p}_h)$. The target vector $\mathbf{t} = (\mathbf{r}_A - \mathbf{r}_M)/|\mathbf{r}_A - \mathbf{r}_M|$, where for the first iteration \mathbf{r}_{M0} can be used instead of \mathbf{r}_M . The required position of the normal according to the reflection law is $\mathbf{n} = (\mathbf{s} + \mathbf{t})/|\mathbf{s} + \mathbf{t}|$. The tracking angles are found as a rotation of \mathbf{n}_0 to \mathbf{n} (See Sequential rotation). Once the angles are found, they can be used to update \mathbf{r}_M according to Eq. (1). The accuracy of iteration is checked by substituting the new value of \mathbf{r}_M into $|\mathbf{t} \times (\mathbf{r}_A - \mathbf{r}_M)|$, which corresponds to the lateral displacement of reflected ray from target. Typically, a sub-millimeter accuracy is acceptable. Otherwise, a new iteration starts with updating the value of \mathbf{t} .

Sequential rotation

The primary α and secondary β angles for rotation of vector \mathbf{v}_0 to \mathbf{v} are found by composing an intermediate vector \mathbf{m} , which separates the sequential rotation into two single-axis rotations [1]. The intermediate vector satisfies the equations $\mathbf{a} \cdot \mathbf{v} = \mathbf{a} \cdot \mathbf{m}$ and $\mathbf{b} \cdot \mathbf{m} = \mathbf{b} \cdot \mathbf{v}_0$. It can be searched in the form $\mathbf{m} = m_a \mathbf{a} + m_b \mathbf{b} \pm m_k \mathbf{k}$ where $\mathbf{k} = \mathbf{a} \times \mathbf{b}$, and the solution of the system is

$$\begin{aligned} m_a &= \frac{\mathbf{a} \cdot \mathbf{v} - (\mathbf{a} \cdot \mathbf{b})(\mathbf{b} \cdot \mathbf{v}_0)}{1 - (\mathbf{a} \cdot \mathbf{b})^2}, \\ m_b &= \frac{\mathbf{b} \cdot \mathbf{v}_0 - (\mathbf{a} \cdot \mathbf{b})(\mathbf{a} \cdot \mathbf{v})}{1 - (\mathbf{a} \cdot \mathbf{b})^2}, \\ m_k &= \frac{\text{sqrt}[1 - m_a^2 - m_b^2 - 2m_a m_b (\mathbf{a} \cdot \mathbf{b})]}{|\mathbf{k}|}. \end{aligned} \quad (6)$$

For each of the \mathbf{m}_{\pm} branches, the angles of single-axis rotations are found as

$$\begin{aligned} \alpha &= \text{atan}[\mathbf{a} \cdot \mathbf{m} \times \mathbf{v}, \mathbf{m} \cdot \mathbf{v} - (\mathbf{a} \cdot \mathbf{v})^2], \\ \beta &= \text{atan}[\mathbf{b} \cdot \mathbf{v}_0 \times \mathbf{m}, \mathbf{v}_0 \cdot \mathbf{m} - (\mathbf{b} \cdot \mathbf{v}_0)^2]. \end{aligned} \quad (7)$$

By default, the angles are returned in the range $[-\pi, \pi)$. It is possible to cast them to the range $[\varphi_0, \varphi_0 + 2\pi)$ with the following transformation

$$\varphi \rightarrow \varphi - 2\pi \left\lfloor \frac{\varphi - \varphi_0}{2\pi} \right\rfloor, \quad (8)$$

where $\lfloor \cdot \rfloor$ is the floor function. This simplifies the check whether the angle is within a given range $[\varphi_{\min}, \varphi_{\max}]$. It is also possible to introduce a function $w(\alpha, \beta) = |\alpha| + |\beta|$ to sort the solutions and select the branch which requires less rotation.

Single-axis trackers

The presented method can also be applied for single-axis trackers like troughs or Fresnel collectors (line focusing as opposed to point focusing). Due to the axial symmetry, it can be assumed that $\mathbf{n}_0 \cdot \mathbf{a} = 0$. The vectors \mathbf{s} and \mathbf{t} are projected on the rotation plane by using an operator $1 - \mathbf{a} \otimes \mathbf{a}$ and renormalizing the result. After that the similar steps (See Local aiming and Global aiming) can be applied to find the primary tracking angle α . The main difference is that rotation of vector \mathbf{v}_0 to \mathbf{v} is solved as

$$\alpha = \text{atan}[\mathbf{a} \cdot \mathbf{v}_0 \times \mathbf{v}, \mathbf{v}_0 \cdot \mathbf{v}]. \quad (9)$$

```

1  <?xml version='1.0' encoding='UTF-8'?>
2  <Heliok version="2020.2" unitLength="m" unitAngle="deg">
3    <World>
4      <Location name="Pentakomo" Longitude="33.261" Latitude="34.707"/>
5      <Sun azimuth="120." elevation="45." irradiance="1000.">
6        <SunShapePillbox/>
7      </Sun>
8    </World>
9    <Factory>
10     <Material name="mirror" reflectivity="1." roughness="0.001"/>
11     <Material name="absorber" reflectivity="0." roughness="0.001"/>
12     <HeliostatKit name="AzEl">
13       <Primary translation="0., 0., 1.5" rotationAxis="0., 0., -1." rotationAngles="-90., 90."/>
14       <Secondary translation="0., 0.1, 0." rotationAxis="1., 0., 0." rotationAngles="0., 90."/>
15       <Facet rotation="90., 0., 180.">
16         <ShapeKit>
17           <ShapeParabolic focus="70."/>
18           <ApertureRectangular xWidth="2." yWidth="2."/>
19           <Material ref="mirror"/>
20         </ShapeKit>
21       </Facet>
22       <Angles primary="0." secondary="0."/>
23       <Tracking point="0., 0., 0." normal="0., 1., 0."/>
24       <Aiming point="0., 0., 20."/>
25     </HeliostatKit>
26   </Factory>
27   <Scene>
28     <Node name="Receiver" translation="0., 0., 20." rotation="90., 0., 180.">
29       <ShapeKit>
30         <ShapeFlat/>
31         <ApertureRectangular xWidth="2." yWidth="2."/>
32         <Material ref="absorber"/>
33       </ShapeKit>
34     </Node>
35     <Node name="Heliostats">
36       <Node name="HeliostatA" translation="30., 50., 0." rotation="0., 0., 149.036">
37         <HeliostatKit ref="AzEl"/>
38       </Node>
39       <Node name="HeliostatB" translation="-30., 50., 0." rotation="0., 0., -149.036">
40         <HeliostatKit ref="AzEl"/>
41       </Node>
42     </Node>
43   </Scene>
44 </Heliok>

```

Fig. 2. Definition of heliostat field in xml-format.

Implementation of method

Input data

The method was implemented in Python and can be used as a library (Heliok). The parameters of heliostat field are stored in xml format (Fig. 2). The tag Heliok is used as a root and has the attributes to set the units of lengths and angles. The subtags World, Factory and provide the following details. The tag World defines the location of the plant and the position of the sun. The tag Factory stores the templates of materials and heliostats. The tag Scene describes the layout of the objects in the scene.

The parameters of heliostat are stored in the tag HeliostatKit and are grouped in the following way. The tag Primary with the attributes *translation* (\mathbf{p}_a), *rotationAxis* (\mathbf{a}) and *rotationAngles* (α_{\min} , α_{\max}) is used for the primary drive. The tag Secondary with the attributes *translation* (\mathbf{p}_b), *rotationAxis* (\mathbf{b}) and *rotationAngles* (β_{\min} , β_{\max}) is used for the secondary drive. The tag Facet with attributes *translation* (\mathbf{p}_m , zero by default) and *rotation* (\mathbf{R}_m , stored as Euler angles φ_x , φ_y , φ_z) defines a heliostat facet and

```

In [1]: # import library
import Heliok as hk

In [2]: # open project
app = hk.Application.file("scene.xml")

In [3]: # find heliostat
heliostatA = app.scene.findNode("HeliostatA").getKit(hk.HeliostatKit)

In [4]: # set tracking angles
heliostatA.setTrackingAngles(20.*hk.degree, 50.*hk.degree)

Out[4]: True

In [5]: # move sun
sun = app.world.sun
sun.motion.setAE(120.*hk.degree, 45.*hk.degree)

In [6]: # update tracking angles
heliostatA.setTrackingSun(sun.motion.vector, debug=True)

Solution 0:
 $\alpha = -36.9152^\circ$ ,  $\beta = 40.6306^\circ$ , trackable = True, accuracy = 0.0604 m
 $\alpha = -36.8761^\circ$ ,  $\beta = 40.6415^\circ$ , trackable = True, accuracy = 0.0001 m
Solution 1:
 $\alpha = 143.0848^\circ$ ,  $\beta = 139.3694^\circ$ , trackable = False, accuracy = 0.0811 m
 $\alpha = 143.0545^\circ$ ,  $\beta = 139.4126^\circ$ , trackable = False, accuracy = 0.0000 m
Selected:
 $\alpha = -36.8761^\circ$ ,  $\beta = 40.6415^\circ$ , trackable = True

Out[6]: True

```

Fig. 3. Example of using Heliok in Jupyter notebook.

can be repeated several times. The tag Angles with the attributes *primary* (α) and *secondary* (β) stores the orientation of heliostat. The tag Tracking with the attributes *point* (\mathbf{r}_{M0}) and *normal* (\mathbf{n}_0) defines a vertex on the facet which will be used for tracking. The tag Aiming with the attribute *point* (\mathbf{r}_A) sets the aiming point.

The actual placement of heliostats is described in the tag Scene. It contains a tree of Node tags, which define geometrical transformations. Each node has the attributes *translation* (\mathbf{p}) and *rotation* (\mathbf{R} , stored as Euler angles φ_x , φ_y , φ_z). The nodes can be nested, which combines the transformations. The heliostats are attached to the nodes by using the tag HeliostatKit with the attribute *ref*, which refers to a specific template with the attribute *name*.

Example of usage

The Heliok library can be used in Jupyter notebook (Fig. 3). The cell 1 imports the Heliok library with a short name *hk*. The cell 2 opens a heliostat field in xml-format and assigns it to a variable *app*. The cell 3 shows how to find a heliostat by name. The cell 4 sets the primary and secondary angles of the selected heliostat. The output is True, when the angles are within the tracking range. The cell 5 creates a reference to the sun object in the scene and changes its azimuth and elevation. The cell 6 updates the tracking angles of heliostat according to the position of the sun. The optional parameter *debug* enables a debug output. It shows two possible solutions and how the accuracy changes with iterations. If there is a solution within the tracking range, the tracking angles of heliostat are updated, and the output is True.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the European Union's Horizon 2020 research and innovation program within the context of the CySTEM ERA Chair project, under grant agreement No. 667942.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:[10.1016/j.mex.2021.101244](https://doi.org/10.1016/j.mex.2021.101244).

References

- [1] V. Grigoriev, K. Milidonis, M. Blanco, Sun tracking by heliostats with arbitrary orientation of primary and secondary axes, *Solar Energy* 207 (2020) 1384–1389.
- [2] B. Paden, Kinematics and Control of Robot Manipulators, PhD thesis, 1985.
- [3] R. Murray, Z. Li, S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, 1994.