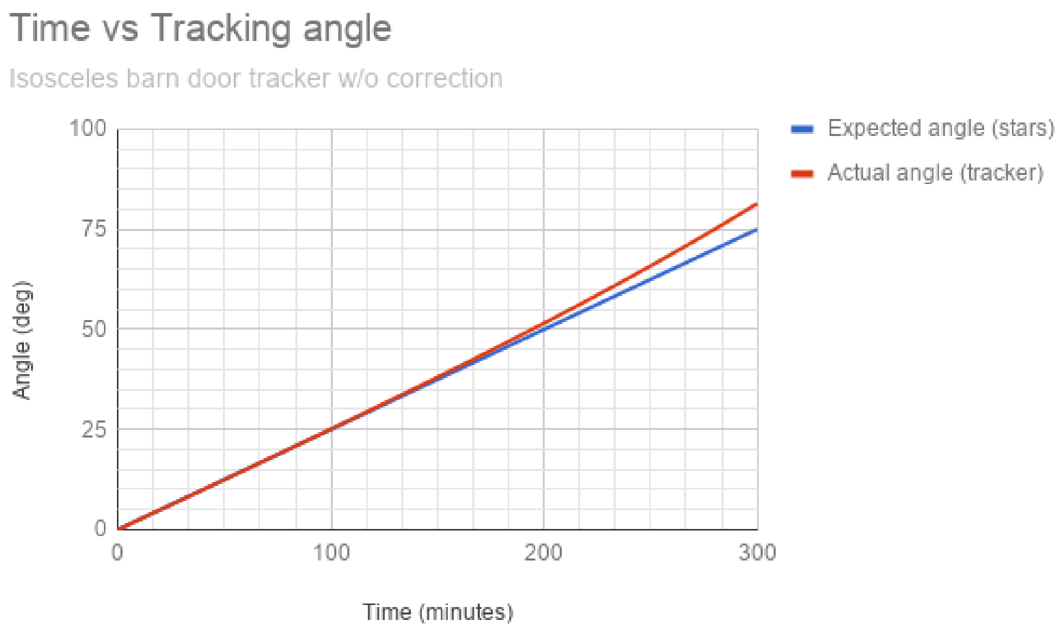
Posted on April 11, 2018 by Arun Venkataswamy — 14 Comments

The tangent error in a barn door tracker

The barn door tracker's geometry has an inherent error which creates a slow drift while tracking the sky. In the most basic of barn door trackers, this drift starts becoming apparent in about 10 minutes. In the "isosceles" barn door tracker described in this website, the drift starts to become apparent after 20 minutes. Mathematically the drift or error is because of the non linear relationship between the included angle of the triangle and the length of the base (threaded rod/nut displacement). A linear (or constant speed) increase in the length of the base of a triangle does not produce a linear increase in the included angle. This is apparent from the math below:

> *$S = 2 \cdot L \cdot \sin(\theta/2)$*
> *'L' is the length of the two the arms*
> *'θ' is the included angle of the arms*
> *'S' is the base or displacement of the nut/threaded rod.*
>
> $\Rightarrow \theta = 2 \cdot \sin^{-1}(S/(2 \cdot L))$
>
> *When we solve for dθ/dt (rate of increase if angle) given a constant dS/dt, we get a non linear solution. In simpler terms, the angular velocity of the barn door tracker will not be constant when the threaded rot is rotated at a constant rate.*

The following chart shows how the actual angle starts drifting away from the expected angle for a barn door tracker which is not compensated for the tangent error. This is for an isosceles barn door mount with an arm length of 300mm



The following chart shows the actual drift error, when the barn door tracker is not compensated for the tangent error. This is for an isosceles barn door mount with an arm length of 300mm

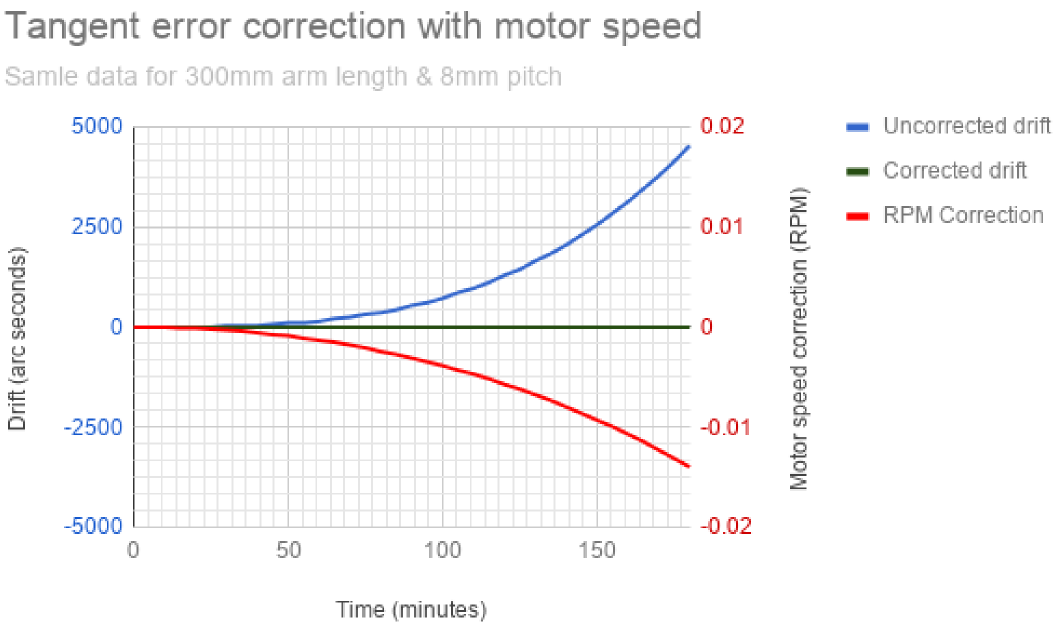
Traditional solutions

The traditional way to solve this problem was by means of mechanical improvements. Typically a second arm was introduced in the camera side. By the introduction of a second arm to drive the camera arm, tracking accuracy was greatly increased. This increase in accuracy allows exposure times of up to one hour. Another traditional method was by introducing a specially designed guide profile between the the camera arm and the actuator. This profile has a negative version of the tracking error thus canceling out the tracking error.

Modern “smart” solution

The traditional solutions substantially increased the fabrication difficulty of the barn door tracker. Also, these solutions were from an time when electronics and micro controllers were the toys of specialized engineers and technologists, out of reach of the average DIY enthusiasts. The solution to the tangent error through software is a simple and elegant one.

The tangent error is created because a constant speed applied on the threaded rod of the barn door tracker produces a non linear response in the angle of the tracker. The solution for this is to adjust the speed applied to the threaded rod continuously so as to maintain a constant linear response in the angle of the tracker. This can be achieved by software running in a small micro controller like the Arduino. This concept is visualized in the chart below:



The software source code to perform this correction is provided free and open source here.

```
/*

Copyright 2018 Arun Venkataswamy

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files
(the "Software"), to deal in the Software without restriction,
including without limitation the rights to use, copy, modify, merge,
publish, distribute, sublicense, and/or sell copies of the Software,
and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
```

```
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS
BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

*/

var reqRisePerMinute_deg = 0.25068448;
var reqRisePerSec_deg = reqRisePerMinute_deg / 60.0;

//Modify these parameters to suit your design
var armLength = 300;
var pitch = 8;
var microSteps = 32;
var fullStepsPerRotation = 200;

var microStepsPerRotation = microSteps * fullStepsPerRotation;
var deviceReqRisePerMinute = 2 * armLength * Math.sin(reqRisePerMinute_deg / 2 * Math.PI
var deviceReqRisePerSecond = deviceReqRisePerMinute / 60.0;
var deviceReqRPM = deviceReqRisePerMinute / pitch;
var devicePulseFreq = deviceReqRPM * microStepsPerRotation / 60.0;

//console.log(deviceReqRisePerMinute,deviceReqRPM,devicePulseFreq);

var oDeviationInDistanceTravelled = 0;
var ot = t = 0;
var correctedDistanceAccumulator = 0;

var arrayString = '';

/*
This is the "bias" time in seconds. The barn door at
it's home position will have an offset (around 20-30mm).
That is, it would already be open. It would not be at 0°

x = offset in mm
L = arm length in mm
offset time = (x * 60) / rise per minute
                        = (x * 60) / (2 * L * sin(angular rise per min / 2))
                        = (x * 60) / (2 * L * sin(0.00437/2)) Note:0.00437 is in radians
*/

var offsetTime = 1371;

for (t = offsetTime; t <= offsetTime+(3 * 60 * 60); ot = t, t += 5) {

        oDeviationInDistanceTravelled = deviationInDistanceTravelled;

```

```
        var expectedAngleDeg = t * reqRisePerMinute_deg/60.0;
        var distanceTravelled = devicePulseFreq * t / microStepsPerRotation * pitch;
        var actualAngleDeg = 2 * Math.asin(distanceTravelled / 2 / armLength) * 180 / Ma
        var deviationDeg = actualAngleDeg - expectedAngleDeg;
        var deviationInDistanceTravelled = 2 * armLength * Math.sin(deviationDeg / 2 * M
        var correctedDistanceToTravel = distanceTravelled - deviationInDistanceTravelled
        var projectedAngleDeg = 2 * Math.asin(correctedDistanceToTravel / 2 / armLength)
        var projectedDeviationDeg = projectedAngleDeg - expectedAngleDeg;

        var deltaDeviationInDistanceTravelled = deviationInDistanceTravelled - oDeviatio
        var pulsesForDelta = deltaDeviationInDistanceTravelled / pitch * microStepsPerRo
        var pulsesInTimeSegment = ( t - ot ) * devicePulseFreq;
        var correctedPulsesInTimeSegment = pulsesInTimeSegment - pulsesForDelta;
        var correctedDevicePulseFreq = correctedPulsesInTimeSegment / ( t - ot );

        if (t > 0)
                correctedDistanceAccumulator += correctedDevicePulseFreq * ( t - ot ) /

        var correctedDevicePulseTimePeriodMicros = 1000000 / correctedDevicePulseFreq;
        correctedDevicePulseTimePeriodMicros /= 2; //Square wave inversions require doub

        // console.log(
        //      t,
        //      distanceTravelled.toFixed(2),
        //      expectedAngleDeg.toFixed(2),
        //      actualAngleDeg.toFixed(2),
        //      deviationDeg.toFixed(2),
        //      deviationInDistanceTravelled.toFixed(2),
        //      correctedDistanceToTravel.toFixed(2),
        //      correctedDistanceAccumulator.toFixed(2),
        //      deltaDeviationInDistanceTravelled.toFixed(4),'*',
        //      pulsesForDelta.toFixed(2),
        //      correctedDevicePulseFreq.toFixed(2),
        //      correctedDevicePulseTimePeriodMicros.toFixed(0),'*',
        //      projectedAngleDeg.toFixed(2),
        //      projectedDeviationDeg.toFixed(4)
        // );

        if (t != 0) arrayString += ',';
        arrayString += correctedDevicePulseTimePeriodMicros.toFixed(0);
}

console.log(arrayString);
```

tangent_error.js hosted with ❤ by GitHub view raw

Category: Tracker

Mathematics behind the barn door tracker An automatic barn door tracker calculation tool

14 thoughts on “The tangent error in a barn door tracker”

Cory

April 30, 2018

Great write up. And nice code.
I am building up a tracker based on your design and need to get the tangent error correction values base on my measurements.

At the end of the Tantent error section you mention “The software source code to perform this correction is provided free and open source here”
I cannot find the link to the code, is it still available?

Thanks.

Arun Venkataswamy

May 1, 2018

Thanks Cory. I have node-js code which is available in a repo. I am not able to make it “live” on this wordpress site. If you can run node-js code I am happy to share it with you.

Cory

May 2, 2018

That would be great. Nodejs code is fine.

Cory

May 2, 2018

BTW I had to modify your code slightly due to constraints with the “const PROGMEM uint16_t timerLookup[]” array and “unsigned int newTimerValue”;.
My stepper driver board can only do 1/8 microsteps. So when you use your calculator to determine the stepper timings the value is beyond 64k and thus beyond the scope of uint16_t.
It unfortunately too me way too long to work out why my stepper timings were so far out, the code was truncating timing values 🙁
Anyway I just changed that array and variable to ‘uint32_t’

Arun Venkataswamy

May 2, 2018

Thanks for pointing out the change required to handle 1/8 microstepping. I have updated the gist to reflect this change.

Arun Venkataswamy

May 3, 2018

Find the code for the tangent error correction here : https://gist.github.com/indstronomy/ecea64a8d838d29312092903324f8acd

Jose

October 3, 2018

How do you add this code (The tangent error) to the main program

Arun Venkataswamy

October 9, 2018

You need to copy the array string which is printed in the console and use it in the main program. Let me know if you need more info

Barn Door Tracker – Part 1 – Vincent Gledhill

January 6, 2019

[…] found this website by Arun Venkataswamy who offers a different approach. The tangent error is not corrected with mechanical means, it is […]

Arran Townsend

January 16, 2019

Hi, I'm slightly confused with this im new to coding, what do you mean by a 8mm pitch thread the one in your photos is certainly not 8mm the thread I have is 1.5mm? And where do i copy this tangent error code to in my code? To the setup or loop or do i need to create a new function all together? Thanks

Arun Venkataswamy

January 27, 2019

The pitch in my version is 8mm. You can use the code provided and modify it for your requirement of 1.5mm. Let me know if you need help I can make a copy for you

Arran Townsend

January 28, 2019

Hi, thanks for your reply I am new at coding I have a basic understanding of the tangent error code, but I can't see where it's used in the main code of yours

Arun Venkataswamy

February 6, 2019

You will have to run it separately. It is not part of the Arduino code. Give me the details of your build, I will generate the correction data for you.

Arran Townsend

February 12, 2019

Hi Arun, thanks for your help. I have tried to run the tangent error code on my own but the ide keeps saying var does not name a type, I don't know weather I should change the var to int on each line of code. anyway my specifications for my tracker are as follows:

arm length between hinge and rod : 300mm
motor: 28byj-48 stepper motor with uln2003 driver
ratio between the motor shaft and rod: 32 to 38 (toothed gears) ps I had to reduce the load of the motor my doing this reduction

Comments are closed.

Search …

Posts and articles

Categories