Alt Az Telescope Mount Design

Ashish Vanjani[1], Sagar Atre[2]

[1]*B Tech Student, School of Mechanical Engineering, MIT World Peace University, Pune*
[2]*Professor, School of Mechanical Engineering, MIT World Peace University, Pune*

ABSTRACT: *Astronomy is on the rise of becoming a new booming field of science. With the explorations coming to an end on earth scientist have started showing a deeper interest in what's not on this world. By being able to observe the countless stars in just our galaxy and finding out new horizons in the universe. A Telescope has become one of the basic instrument used by every aspiring scientist. Now while it is very easy to point a telescope on the sky it is equally hectic to do so. After a lot of research and technology advancements we. Were able to make the first GoTo mount that would tell you exactly what you are aiming at in the sky. Now while these mounts are useful are also very costly. This project aims at building an Alt-Az Telescope mount at a cheap cost using Arduino DUE board. The project uses various basic electronic tools and some wood cutting. So proper care must be taken while using these instruments.*

Keywords: Astronomy, Telescope, Electronics, Coding

1) INTRODUCTION:

GOTO refers to a type of telescope mount and related software that can automatically point a telescope at astronomical objects that the user selects. Both axes of a GoTo mount are driven by a motor and controlled by a computer.

GoTo mounts are pre-aligned before use. When it is powered on, it may ask for the user's latitude, longitude, time, and date. It can also get this data from a GPS receiver connected to the telescope or built into the telescope mount itself, and the mount controller can have its own real time clock.

Types of GOTO Telescope Mount:

- Alt Azimuth Telescope Mount
- Equatorial Telescope Mount

Alt-azimuth mounts:

Alt-azimuth GoTo mounts need to be aligned on a known "alignment star", which the user will center in the eyepiece. From the inputted time and location and the star's altitude and azimuth the telescope mount will know its orientation to the entire sky and can then find any object.

For accuracy purposes, a second alignment star, as far away as possible from the first and if possible close to the object to be observed, may be used. This is because the mount might not be level with the ground; this will cause the telescope to accurately point to objects close to the initial alignment star, but less accurately for an object on the other side of the sky.

An additional reason for using two alignment stars is that the time and location information entered by the user may not be accurate. For example, a one-degree inaccuracy in the latitude or a 4-minute inaccuracy in the time may result in the telescope pointing a degree away from the user's target.

When the user selects an object from the mount's database, the object's altitude and azimuth will be computed from its right ascension and declination. Then, the mount will move the telescope to that altitude and azimuth and track the object so it remains in the field of view despite Earth's rotation. Moving to the location is called slewing.

The disconnected hand control of a GoTo telescope mount. The large arrow buttons are used for slewing the telescope. Below these, the number buttons are used for both inputting information and selecting which catalogue to choose objects from.

Equatorial mounts:

For an equatorial GoTo telescope mount, the user must align the mount by hand with either the north celestial pole or the south celestial pole. Assuming the user is accurate in the alignment, the mount points the telescope to a bright star, asking the user to center it in the eyepiece. Since the star's correct right ascension and declination is already known, the distance from what the user considered to be the celestial pole and the actual pole can be roughly deduced. Using another alignment star can further improve the accuracy of the alignment.

After alignment the telescope mount will then know its orientation with respect to the night sky, and can point to any right-ascension and declination coordinates.

When the user selects an object to view, the mount's software looks up the object's right ascension and declination and slews (moves) to those coordinates. To track the object so that it stays in the eyepiece despite Earth's rotation, only the right-ascension axis is moved

2) Reasons for selecting this Project:

A standard conventional GOTO mount is easily available in the stores but they all have 2 major drawbacks:

- cost: Conventional GOTO mounts cost from 50,000 to more than 1,00,000 Rs.
- removability: these telescopes being highly sophisticated will cause high damages if something was to break

Advantages of building a GOTO mount:

- It's a good way to learn a lot of new things like: alt-azimuth and celestial coordinates, local sideral time, earth movement, Arduino coding etc...
- one can control the budget and keep it below 20,000 Rupees
- The telescope will be customized to fit your needs

3) Introduction to the Celestial Co-ordinate System:

The most important thing is understanding the co-ordinate system of the stars.

In astronomy, a celestial coordinate system is a system for specifying positions of satellites, planets, stars, galaxies, and other celestial objects relative to physical reference points available to a situated observer Coordinate systems can specify an object's position in three-dimensional space or plot merely its direction on a celestial sphere, if the object's distance is unknown or trivial.

There are 5 most crucial celestial co-ordinate systems that are:

- Alt-Azimuth system
- Equatorial system
- Ecliptic system
- Galactic system
- Super galactic system

However, two co-ordinate systems we need to study for the project are the Alt-Az co ordinate system and the Eq coordinate system. And the conversion between the two systems:

Alt-Azimuth coordinate system:

The horizontal, or altitude-azimuth, system is based on the position of the observer on Earth, which revolves around its own axis once per sidereal day (23 hours, 56 minutes and 4.091 seconds) in relation to the star background. The positioning of a celestial object by the horizontal system varies with time, but is a useful coordinate system for locating and tracking objects for observers on Earth. It is based on the position of stars relative to an observer's ideal

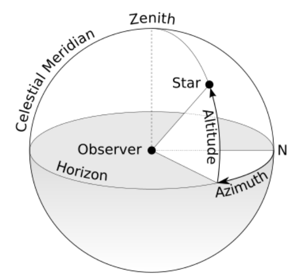

Fig. 1 Alt-Azimuth Co-ordinate System

Equatorial co ordinate system:

The equatorial coordinate system is centered at Earth's center, but fixed relative to the celestial poles and the March equinox. The coordinates are based on the location of stars relative to Earth's equator if it were projected out to an infinite distance. The equatorial describes the sky as seen from the Solar System, and modern star maps almost exclusively use equatorial coordinates

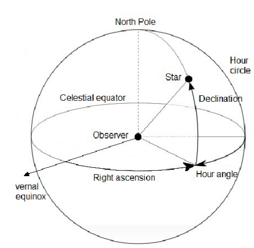

Fig. 2 Equatorial Co-ordinate System

For this project we will be making an Alt-Azimuth Telescope mount. And convert the Readings taken in Alt-Azimuth co-ordinate system to Equatorial co-ordinate system and the output will be shown in Stellarium.

4) The workflow on an Alt-Az Telescope:

The Work flow of an Alt-Az Telescope system isn't complex at all in any way. It has a very simple flow procedure and even an amateur can use the telescope without any help.

1) Decide on an object to point in the sky
2) move the Telescope accordingly
3) feed the rotor and motor angles to the Arduino code using the Arduino board
4) Convert the rotor and motor values to Alt-Az Angles
5) change the Alt-Az values to Equatorial values.(by calculating the Ra and DEC of the respective altitude and azimuth direction)
6) Sync the Arduino to Stellarium to get the resulting output onto the stellarium.

Note: We haven't directly converted the Rotor values to Equatorial values due to the complexity of the Equatorial Co ordinate system. As such the co ordinates keep changing with every passing second it is easier to first convert the mechanical readingsto Alt-Az then to Equatorial system.

The flowchart for the Above same has been shown in the following figure:

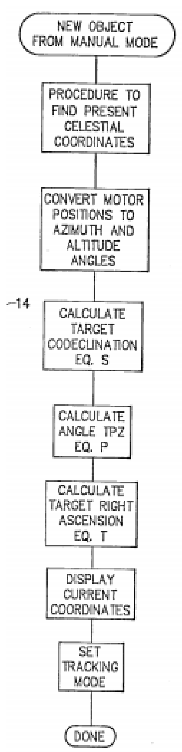

Fig. 3 Flowchart of Telescope Working[1]

5) Designing the Telescope Mount:

Since we already have own a 6in Telescope we just want to build a mount and add the electronics to it . The most important thing about the mount is that it has to be sturdy so that it doesn't fall of when the weight of the telescope is shifted while trying to rotate it along its axis.

Before Designing the Mount we have to note down a few things.

- Mirror size = 6in
- Tube outer diameter= 8.25in
- Wood thickness: 0.5-0.75 in

Designing the Cradle

As shown in the figure a cradle is a simple box with support beams that will hold the telescope. The Dimensions of the cradle are dependent on the size of the mirror and the outer diameter of the tube.

The Cradle consists of 3 components:

- The Box
- Felt pads
- Supporting Triangles
- Supporting Beams

- **The box:**

The Following are the dimensions used for making the box of the cradle.

- Cradle length = 2 times mirror size
- Side Height = Tube OD + ¼" bottom gap + ¾" top gap + (2 × Plywood Thickness)
- Top & Bottom Width = Tube OD + (2 × ¼" side gap)

- **Felt pads:**

In the furniture feet and glide section of a hardware store or home center, you should be able to find self adhesive thick felt pads. Most commonly these come in rounds, but there are often strips or 4 × 6 sheets available also, which is what we used, cutting out 2 inch by ¾ inch pads from the sheet. The package we bought claims the thickness is 0.22 inches; when measured at home we got between 0.20 and 0.18 thickness. For our design, we used a value of 0.20 inches to size our triangles

- **Supporting triangles:**

We need to make eight triangles. We are using ¾ inch N-N plywood as a wider bearing surface is desired to press on the tube. Allowance needs to be made for the felt pad thickness.

The easiest way to size the triangles is to make a scale drawing of the cradle with a ruler and compass. You only need to draw one quarter of the cradle, from the tube center out to one corner. Once you have the tube outside diameter and a side and bottom drawn in, draw a 45° line tangent to the tube. Then draw a parallel line to this 0.20 inches away (the pad thickness) from box side to box bottom - this defines your triangle.(THIS PROCDURE HAS BEEN SHOWN IN THE FOLLOWING FIGURE)

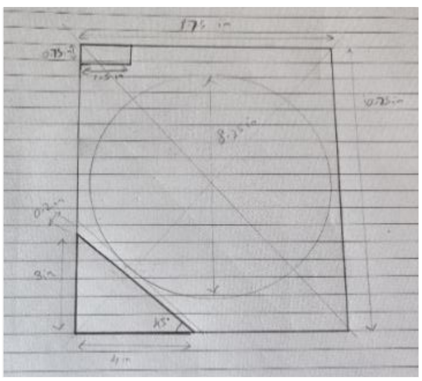

Fig. 4 Procrdure to size the Supporting Triangles[6]

- **Supporting Beams:**

Each pair of upper triangles are connected by a pressure beam made out of ¾ inch plywood. They are the length of the tube cradle. They should be as wide as possible without projecting beyond the triangles and hitting the tube; for our 8¼ inch tube we choose a width of 1½ inches. Adjust to your tube size as necessary.

Assembling the cradle:[6]

- **Hardware required:**

i. [4] ¼-20 3 inch long Pan Head Bolt - fully threaded (retaining bolts for pressure beams)

ii. [2] ¼-20 2 inch long Carriage Bolts - fully threaded (pressure bolts for pressure beams)

iii. [6] ¼-20 Tee Nuts

iv. [4] ¼ inch Washers

v. [2] ¼-20 Hex Nuts

vi. [4] ¼-20 Lock Nuts

vii. [2] Knobs for ¼-20 shafts (Found in "Small Parts Drawers")

viii. [2] Mending Plates (Carriage Bolt head presses on these)

ix. [1] Large metal handle (for carrying cradle and tube.)

- **Assembling the wooden parts:**

Now assemble the cradle box using carpenters glue and a few finishing nails to hold the parts in place while the glue dries. We used three nails on each joint, starting the 6 nails on each side piece before assembly. The first joint is the hardest to manage when assembling, start with a side and top, spread glue on the edge of the top (remember the tee-nuts are on the inside of the box), and have the bottom nearby. Now with the top standing on it's non-glues edge, align the side to the top and set it on the glue. Then with a free hand (you have at least four hands, right?) place the bottom on edge, at right angles to the top, to support the side piece while you do final alignment and nailing of the glued joint. Slightly countersink the nail heads to just below the surface. Wipe off any excess glue with a damp rag as you work. With the first joint done, you can attach the bottom to the installed side, then flip this U-shaped assembly over and install the last side.

Use a square to make sure the box is square - nudge it into square if it is not. Now nail and glue the four lower triangles into the bottom of the box, which will also help keep the box square.

If you have clamps, you might want to clamp the box sides together until the glue sets. If you don't, you could rest the box on a side, and pile some heavy objects on top of it. Do a final check for squareness and then let it dry, usually 24 hours for most glues.

- **Hardware assembly:**

Felt Pads: Cut eight ¾ × 2½ inch self adhesive felt pads from a felt sheet. Attached to the center of each or the eight triangles. Note: After about a week of use in hot summer weather with the tube stored vertically, the adhesive on the pads failed, and the tub dropped about 2 inches into the rocker box. We now recommend you use two small brads in each pad, at the ends where they will not contact the tube, to secure these pads to the cradle.

Mending Plates: Attach a mending plate in the center of each pressure beam, so that the pressure bolt head will press against it. We cut a single square mending plate in half for this. We suggest using a block of wood to work on, lifting the beam off the triangles and providing a flat, firm work surface.

Bolts in the Top: Check your two carriage bolt heads, some have embossed lettering on them that should be filed off leaving a smooth, rounded head (this time, our bolt heads we unblemished by lettering). Then screw them fully into the two middle tee nuts from inside the box, so that they stick up above the box (no need to tighten them). Then thread a hex nut about an inch onto the projecting bolt. Next screw on your threaded knobs, and tighten the hex nut against them to hold them tight to the bolt. If you can't find threaded knobs, try buying some "radio knobs" at Radio Shack or another electronics distributor - ¼ inch shafts are standard, and they will have set screws to hold them tight.

Now place a washer on each of the four pan head pressure beam retaining bolts and screw them down from the top into the box and tighten.

Install Pressure Beams: Now install the pressure beams onto the pressure beam retaining bolts. This is likely were you may have some fit problems - they should slide freely over the bolts. The first beam we installed was perfect - it slipped on easily and moved freely. The second beam was a different story, it would not even fit on. We had to straiten one bolt (the tee nut was not seated fully) and then enlarge one pressure beam hole a lot, and the other just a very little, until we got good free sliding action. It looks like on the one hole that needed a lot of enlargement, we did not drill very accurately. And now you know we do test fits before finishing! Screw lock nuts onto each of bolts to prevent the pressure beams from falling off. Screw them on until the bolt is flush with the top of the nut, firmly held by the plastic locking part.

Insert Tube and Test: With the pressure bolts fully released, You should be able to slide your telescope tube into the cradle, primary end first. This requires a little juggling, as you need to hold the pressure beams up (maybe 4 light springs between the lock nuts and beam would be a nice improvement?). Roughly center the tube in the cradle, and apply gentle pressure by tightening the pressure knobs - you might be surprised at how little it takes to grip the tube, and it is possible to distort the tube with too much pressure - so go easy! Then release the pressure slightly and make sure you can rotate the tube and slide it up and down a bit. If this does not work, check clearances and make adjustments as necessary. Make sure the retaining bolts are not so long as to touch the tube. Congratulations, you have made a very nice adjustable cradle. Just a few simple things left to complete it.

Fig 5. Possible view of final Cradle[3]

Designing Altitude Bearings:[6]

- Sizing the Altitude Bearings

Many small Dobsonians are built with altitude bearings that are too small to work well. This has happened because for many years there was no good information about the need for large diameter bearings, and because many builders looked for objects to make the bearings out of, and what was commonly available, such as plumbing parts, were generally available in small diameters.

With many thousands of Dobsonians built, much research into the best materials and sizes, and plenty of field experience, it has been learned that a key parameter in obtaining the smooth, jerk-free motion desired in a quality Dobsonian is to have the altitude bearings move a high linear velocity for even small motions. This is achieved by having large diameter bearings: the larger the diameter of the bearing, the longer the linear motion will be over the bearing surface for any given angular movement.

The Dobsonian Telescope by David Kriege and Richard Berry recommend an altitude bearing diameter of 1.8 × Tube Outside Diameter.

So after many years of research and trial it was concluded that the altitude bearings can be 1.2 to 1.8 time Outer Diameter for our **8.25 inch tube**, we will choose a **14 inch diameter bearing** (which is approximately 1.7 × tube diameter)

- Cutting the altitude bearing:

We cut a 14in circular wheel of wood and then divide it into two halves. Any technique to cut the wood is fine as long as it is a perfect circle and proper caution is taken while machining. After the wood pieces are acquired finish must me done and lamination too as it prevents any damage in future to the wood or cuts on the hands due to the sharp edges.

After the bearings are ready it is now time to install them into the Cradle.

- Installing the Altitude Bearings:

It is really important that the altitude bearings be well aligned on the cradle, so that your telescope will not wobble. The bearings will be screwed, not glued onto the cradle in case we might need to make a small adjustment afterwards.

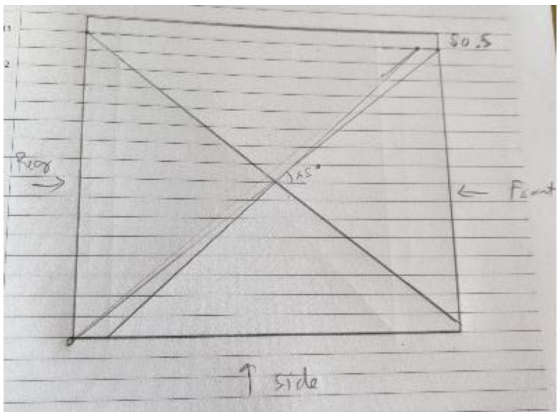

Fig. 6 Finding Bearing Position[6]

Since our cradle is finished, we use masking tape to draw on. If you have a simple box cradle, draw line from opposite corners to find the center; if you have built our adjustable box cradle, measure ½ inch down from the top corners draw your line from there. Where the lines cross is the center of the cradle on the tube. Through this center point, draw a 45° line this is the top edge of the bearing.

Now lay the bearing on the 45° line, with the center of the bearing on the center mark. Trace the curved edge of the bearing onto the cradle in places where the cradle sticks out beyond the bearing - this will help you locate appropriate places for screws, and help you keep the bearing aligned when attaching it.

Since our bearing is ¾ inch thick and the cradle is ½ inch thick, we will mount the bearing by screwing through the cradle from the inside, to get a ½ inch 'bite' into the ¾ inch bearing with the #6 × 1 inch wood screws we are using. It will also look nicer not to have any screw heads visible on the bearings. You can certainly surface screw through the bearing from the outside if you wish. We located 3 screw holes near the edges of the bearing and the corners of the cradle as shown in the photo - adjust to your dimension in approximately the same relative locations. Be sure to avoid the corner triangles. We also had to notch the pressure beams to avoid hitting the upper screw heads. Make notes so you can repeat hole locations on the other side (we will use out photo to do this, which is why we marked the insets on the tape). Drill 1/8 inch clearance holes in the cradle, remove all but the diagonal masking tape, carefully align and center the bearing, clamp it to the cradle, and screw the bearing in. Now remove the clamps, loosen the screws, remove the last piece of tape, and tighten the screws.

Repeat for the other bearing. Now your cradle and is really done with bearings attached!

After following the above Procedure properly the cradle with the bearing is ready and should look something as shown below:

Fig. 7 Possible cradle with Bearing[3]

Finding the Balance Point:[6]

Insert your tube into your cradle, taking a guess at where it will balance. Then put your heaviest eyepiece in the focuser, and make sure all tube mounted accessories like finders are on the tube. We want the top of the tube to be in its heaviest configuration, because that will cause the back of the tube to stick out the farthest to achieve balance.

Now that the cradle/tube assembly is in balance, you need to measure from the back end of the tube to the center mark on the cradle/altitude bearings (the center of rotation, previously the center of the disk that became the altitude bearings). Write this number down, we will use in to make sure our rocker box has adequate height so that your telescope can point at the zenith without colliding with the bottom of the rocker box.

I have assumed a distance of 21¼ inches from the rotation point of the altitude bearings to the back of the tube (this is assumed as it will be used to create the stand of the Mount)

Designing the Rocker box:[6]

The Rocker Box is a major structural component of a Dobsonian Mount. On the top side, it supports the Teflon pads for the Altitude Bearings, and on its bottom side, it has laminate and a pivot bolt hole for the Azimuth Bearing. In between, we try to build a light yet rigid box structure to connect it all together. While there are a number of minor parts to the rocker box assembly, the four major parts are two sides, one front and a bottom

The following procedure and figure give proper guidance to how the box is to be made:

We start by laying out the two sides, which we will make out of ¾ plywood for added stiffness. The width should be approximately the width of the tube cradle (9¾ inches in our case) or approximately 1.2 × the Tube Outside Diameter (9.9 inches in our case). Since this is not a critical dimension, we just rounded up at 10 inches even.

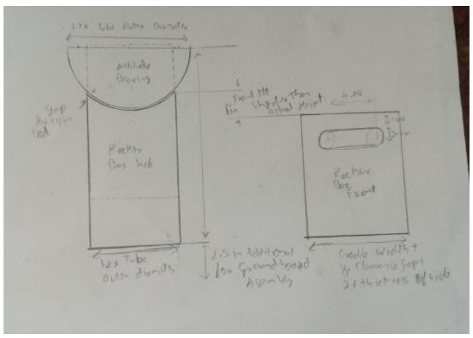

Fig. 8 Layout of Rocker Box[6]

In the previous step we measured the projection of the balanced tube from the center of rotation of the altitude bearings (for us, 21¼ inches). To this we need to add a bit of margin, both to insure clearance of the tube to the bottom of the rocker box, and to allow for additional tube extension should we need balance a heavier eyepiece in the future. You might also want to add a bit to put the scope at a more convenient viewing height; don't go wild here, if you want to add a lot of height it would be better to build a separate, stiff box to set the scope on rather than to make a tall rocker box which could be hard to keep stiff, and with the azimuth bearing at the end of a long rocker box 'lever' make the scope less stable during rotation. We would make the margin no less than 2 inches and no more than 8 inches. We will choose to make ours 2¾ inches bringing our rotational height to an 'even' 24 inches.

The other two numbers you need to layout the sides are the Altitude Bearing radius (known, but add 1/16 inch for laminate thickness which is now attached) and the thickness of the Teflon pads you will use. We are using ¾ inch square by 1/8 inch thick pads. Pad size is not critical, you want a ¾ inch width to fit on top of the side board, the length could be anywhere from ¾ to 1½ inches. Thicknesses of 1/8 to ¼ inch are fine

The front is mainly used to stiffen the sides, we will use ½ inch plywood although you could certainly use ¾ if you wish. The width is critical and needs to be cut with care so your cradle can swing freely:

Rocker Box Front Width = Cradle Width + 1/8 inch Clearance Gap + (2 × Side Thickness)

As shown in the diagram, the front height needs to be about an inch less then the laid out maximum height of the rocker box, to give the cradle plenty of room to swing.

The Rocker Box Bottom is a circle of ¾ inch plywood. It must be sized to fit the entire rocker box within it. Calculate the diagonal length of the rocker box bottom (= squareroot[width^2 + length^2]) remembering to add on the thickness of the front panel to the length of the sides. We get approximately 15½ inches for our rocker box, so we will make the bottom 16 inches in diameter.

Make a good mark in the center of the circle ,you will need it to drill out the hole for the pivot bold later. Also, you should draw two straight line at right angles through the center point, we will use these to align the top and sides to the bottom. Before you cut the bottom, you might want to lay it out, and place the rocker box sides and front on it to make sure it fits

Making the Azimuth Bearing:[6]

We will use a 3/8 inch diameter bolt for the pivot. You might want to put a bushing in the rocker box bottom pivot hole to prevent wear over time. We went looking for a 3/8 I.D., 1/2 O.D., ¾ inch long plastic spacer in the small parts drawers of our local hardware store. We didn't find them, but did find a 3/8 I.D., 1/2 O.D., 1 inch long bronze bushing. We purchased that and cut it down to ¾ inches long. You need to drill a hole of the appropriate size for what you plan to use - 3/8 inch if you are going to just use a bolt, bigger for a bushing. Make it as vertical as possible so the telescope will rotate smoothly. We had a 31/64 inch drill, 1/64th small than our ½ bushing, and used that, making the bushing a 'press fit' when coaxed into position with a wooden block and hammer. If your bushing slides out of the hole, try a little tape around the outside or some glue to hold it in place for now and to prevent any wobble. Once we install the laminate, it won't slip down, and in use there is no vertical pressure on it.

The final step in constructing the azimuth Bearing is to cover the bottom of the rocker box with Ebony Star Laminate. Only the outer 2 inches need to be laminated, but for small dobs it is more work to cut a ring so we just laminate the entire bottom. Use the same procedures we used for the Altitude Bearing laminate - cut the laminate a little bigger than the surface to be covered (we cut our sheet into an octagon 1/8 inch larger then the rocker box bottom on all sides), coat each part with contact cement (2 coats), and when dry press together. It is often helpful to place some thin dowels or sticks between the laminate and rocker box to prevent the two parts from touching while you align them - then slip out the dowels to make contact and get adhesion. Trim (file and sand) the edges flush. Once that is done, put a scrap block of wood under the center hole and drill out the 3/8 inch hole for the pivot bolt to pass through.

Ground Board:

- **Introduction:**

The Ground Board is a simple piece of plywood that hold the feet, azimuth bearing pads and pivot in place. Because the pads are place directly over the feet, there is essentially no vertical load on this part; it only has some small loading in holding the parts in place horizontally. Therefore it can be made ½ inch plywood (which we will use) and have quite a bit of wood removed, as in our triad design without failing. You can choose to make the traditional circular ground board if you wish, and you can use ¾ inch plywood for either design.

The following diagram and procedure give clear instructions on designing the ground board:

Start with a square of plywood that is slightly larger than your rocker box bottom (our Rocker Box has a 16 inch bottom, therefore we will start with a 16¼ inch square of ½ inch plywood). Draw a circle of the same size as your rocker box bottom on the plywood. If you want to make a circular ground board, this is the line you will cut and shape to. If you would rather make mostly straight cuts, you can layout our triad design, shown in the diagram. We used a hole saw to make rounded corners near the center, you could make them straight and acute if your wish. In either design, the holes for other parts are in the same place on the diagram.

Now draw three diagonals through the center, spacing them 120° apart so that you end up with six wedges of 60° each. These lines will be used to space the feet and pads around the circumference.

We recommend using hockey pucks for feet - they are weather-proof and large enough to provide a good stable footing. Hockey pucks are 3 inches in diameter and 1 inch thick. We need to mark the three holes for them 1 inch in from the edge, and therefore the hockey puck will stick out ½ from the ground board and rocker box.

We are using 1 inch square Teflon pads. We want them them inset ¼ inch from the edge, so we mark their location ¾ inch from the edge. If you are using a different pad size, adjust the hole location accordingly.

The center pivot bolt will pass through a 3/8 inch tee nut. Drill out a hole in the center for it; ours was 7/16 inch in outside diameter

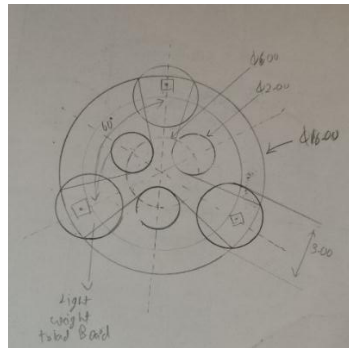

Fig. 9 Ground Board Layout[6]

For the triad design, we need to mark the additional holes and cuts. We used a 2 inch hole saw to make the inner radius on the arms. Use your compass to draw a circle of 6 inch diameter; the 2 inch hole centers are on this circle where they cross the 60° lines that do not have feet or pads on them. We also need to mark one inch on either side of the line that do have feet and pads on them at the outermost circle - this is the width of the triad arms. You can then draw tangent lines to mark the straight arm cuts. This triad design can be adjusted to your taste, and to the size of hole saws you may already own. You can also make this with all straight cuts if you wish. Just keep the central hub about 5 inches in diameter, and the arms about 2 inch wide, or larger, and you will be fine.

Drilling & Cutout

We start by drilling holes. First up is the center pivot hole, which must be drilled out to fit our 3/8 inch threaded Tee-nut. Ours had a 7/16 outside diameter, and we drilled the center hole to this size.

Next, we use an awl to poke starter holes for the pads - these are the marks ¾ of an inch in from the edge. Then we drill out an countersink the three holes for the feet - we are using #8 × 1¼ inch flat head screws, so the clearance hole diameter is 11/64 followed by a countersink deep enough to keep the head below the surface, so it does not interfere with the nearby bearing pad.

- **Ground Board Assembly:[6]**

We are using three hockey pucks for feet - mark the center of each puck and drill a 9/64 inch pilot hole into each one, at leas 1-1/8 deep (it's OK to go all the way through if you wish).

We are using 1 inch square, ¼ inch thick Teflon Pads for the Azimuth Bearing (you can use 1/8 inch thick pads, and sizes slightly smaller than larger than ours). mark the center of each pad, and drill a 9/64 clearance hole for the #6 × ¾ flat head screws we will use to attach the pads. Then countersink the hole so that the screw heads are well below the pad surface.

Once the ground board finish has dried, we can proceed with assembly. First, press in a 3/8 inch Tee nut into the center pivot hole from the top - the may require some 'serious persuasion' with a hammer and wood block, as ours had some giant tangs to sink into the plywood.

Then screw in the hockey puck feet with #8 × 1¼ inch flat head screws - the pilot hole in the puck should let you firmly attach them. Finally we use #6 × ¾ inch flat head screws to attach the three Teflon pads

The Ground board should be ready

Final Assembly:

All that is left to do is assemble the ground board to the rocker box. For this we will use the following hardware:

- [1] 3/8 × 2 inch Hex Head Bolt (use 2¼ inch for a ¾ inch thick ground board, you may need to cut one down from a 2½ inch bolt, a more standard length)
- [2] 3/8 inch Fender Washer, 1½ inch diameter (Exact O.D. not critical)
- [1] 3/8 inch Tee Nut (already installed in Ground Board)
- [1] 3/8 inch Hex Nut

Place one washer on the bolt, pass it through the hole in the rocker box, and screw it into the ground board tee nut. Do not tighten, screw it in until about a 1/16 inch gap exists between the bolt head an washer. Then place the second washer on the bolt that is sticking out the bottom of the ground board, spin on the hex nut, and tighten the hex nut to lock the bolt in place while maintaining the 1/16 inch gap below the bolt head. Check to make sure the rocker box rotates smoothly on the ground board, and if it doesn't figure out what is binding and fix it.

6) Design Calculations:

(Summarizing)

Given:

- Mirror diameter= 6in
- Tube outer diameter= 8.25in
- Wood thickness: 0.5-0.75 in

ADJUSTABLE CRADLE:

- **Cradle Side Height = TOD (8.25”) + (0.25” BG) + (0.75” TG) + 2× PT**

Cradle Side Height = 8.25" + 0.25" + 0.75" + 1" = **10.25"**

- **Cradle Top Bottom Width= TOD(8.25”)+(2×0.25”)**

Cradle Top Bottom Width = 8.25” + 0.5” = **8.75”**

- **Cradle Length = 2× MD**

(2× 6”) = **12”**

Cradle Parts:

- Cradle Sides : **10.25” × 12”**
- Cradle Top/Bottom : **8.75” × 12”**

ALTITUDE BEARINGS:

- **Diameter = 1.2 TO 1.8 × Tube OD**

1.7 X 8.25” = 14” Diameter

Altitude Bearing Parts:

- 14" Diameter Disk cut in half makes two Altitude Bearings

ROCKER BOX & GROUND BOARD:

- **Rocker Box Width = 1.2 × Tube OD**

(1.2 × 8.25” = 9.9”) = **10”**

- **Rocker Box Height = Balance Point + 2.75”**

21” + 2.75” = **24”**

- **Rocker Box Front Width = Cradle Width + Cradle Side Thickness + Clearance + Rocker Box Side Thickness**

Rocker box Front Width = 8.75” + 1” + 0.125” + 1.5” = **11.375”**

- **Rocker Box Front Height = Rocker Box Side Height – 1”**

  24” – 1” = **23”**

- **Rocker Box Bottom Diagonal = Square Root(Rocker Box Width^2 + Rocker Box Front Width^2)**

  Sqrt($10^2 + 11.375^2$) = **15.13"**

- **Rocker Box Bottom Diameter = Round Up (Rocker Box Bottom Diagonal)**

  15.13 => **16”**

- **Ground Board = Rocker Box Bottom Diameter = 16"**

Rocker Box & Ground Board Parts:

- Rocker Box Sides 10” × 24”
- Rocker Box Front 11.375” × 23”
- Rocker Box Bottom Circle of 16" diameter
- Ground Board Circle of 16" diameter

Plywood Layout:

Once you have calculated parts sizes and have a parts list, you need to figure out how much plywood to buy.

The best way to do this is make a scale drawing of the parts, packed as tightly together as possible. (As shown in fig)

A standard sheet of plywood is 4-feet by 8-feet, or 48-inches by 96-inches. Many home centers in the US also sell partial sheets in the plywood aisle. For our sample, we need one half sheet of ¾-inch plywood (4-feet by 4-feet), and a 2-foot by 2-foot sheet of ½-inch plywood.

Now that we have finished designing the mount and our base is ready it is now time to do the electronic part and make this telescope automated so that it can track or tell us what we are looking at in the night sky.

A standard sheet of plywood is 4-feet by 8-feet, or 48-inches by 96-inches. Many home centers in the US also sell partial sheets in the plywood aisle. For our sample, we need one half sheet of ¾-inch plywood (4-feet by 4-feet), and a 2-foot by 2-foot sheet of ½-inch plywood.

Now that we have finished designing the mount and our base is ready it is now time to do the electronic part and make this telescope automated so that it can track or tell us what we are looking at in the night sky.

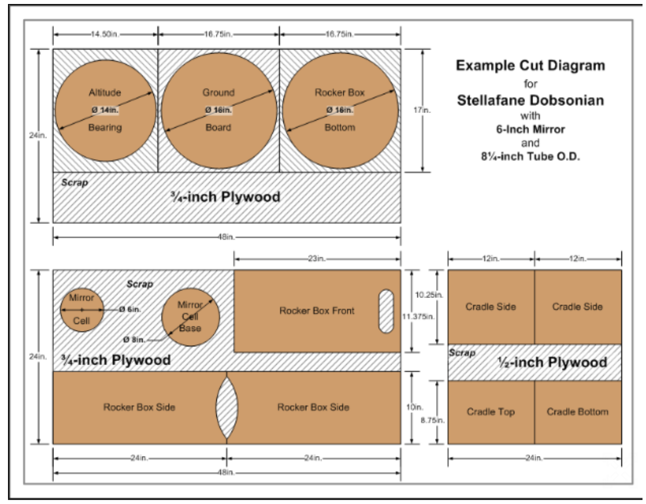

Fig. 10 Reference layout (From The Internet)[6]

7) Working with the Electronics:[7]

Basic Material List:

- 2 pulleys GT2
- 2 Incremental encoders
- 2 timing belts GT2 with 600mm,
- 2 bearings
- Arduino DUE and USB cable
- a small proto PCB to solder the components
- some screws, bolts, nuts, washers
- 2 springs
- metal plate
- plywood and some pieces of wood
- wire and connectors
- solder, shrink tube, cable ties, tape, etc
- some dupont connectors for arduino
- black paint, wood glue, hot glue
- 4 trimmers (10 kOhm)
- 4 resistors (470 Ohm)

Gear material and Gear calculation:

First I thought maybe we could connect directly the encoders to the telescope. Then I realized it would be better if we use a gear-wheel fixed to the mount and telescope to drive the encoders. This gear will allow a smoother mechanical connection, more important, will allow more resolution from the encoder - by multiplying the turns of the encoder by a gear ratio. This configuration give a **gear ratio of 15.** This means, a complete telescope's turn will give 15 complete encoder's turns , so you will multiply the encoder resolution by 15. We will be using gear to count the changes in rotor and encoder values. We have two options to do so. **We could either manufacture a gear to count these rotations or we could use a timing belt.**

- **Gear Material:**

The material for Gear or the base of timing belt can be any of the following depending on how much cost you want save and their availability around you:

  - Wood
  - Teflon
  - Aluminium

- **Gear Calculation:**

A timing belt of 190mm diameter has **300 teeth.**

The encoder pulley has **20 teeth.**

Hence, **gear ratio=(300/20)=15**

Thus we multiply resolution encoder by 15.

Introduction to Encoders:

An encode is a device (sensor) that can read, and transmit a position or an angle.

We have incremental ones. These encoders are reliable, cheap easy to find and, most important, they can be very accurate while having high resolution.

The major problem using this type of encoders is that they never give you their position instead, they give you pulses So, in order to know the angle we use the following steps:

- There are two channels at each encoder (channel A and B);
- Each channel can switch between ON and OFF (0 or 1) as you turn the encoder. That is a pulse.
- The 2 channels are not in phase. There is a 90º lag between them. That is called a quadrature signal;
- By using quadrature signals, we can know the direction of movement;
- It's also possible multiply the resolution by 2 or by 4
- We can calculate the position by counting the number of pulses (CW and CCW);

To avoid loosing pulses we use Interrupts feature of Arduino UNO. So, each time that one of this pins changes his state this special pin activate a special piece of code

This technique makes it possible to define a "time sensitive" section of code, and that was what we will do to count all the pulses, with no exception.

Conducting Electric Interface:

- this part will adapt the votage level from encoder(5V) to Arduino DUE(3.3V)
- Add a pull-up resistor (Its a simple interface, where a 470 ohm resistor is used to pull-up the encoder's output and a 10K ohm trimmer is used just as a voltage divider. The trimmer's center pin is then connected to the Arduino's input pins.)

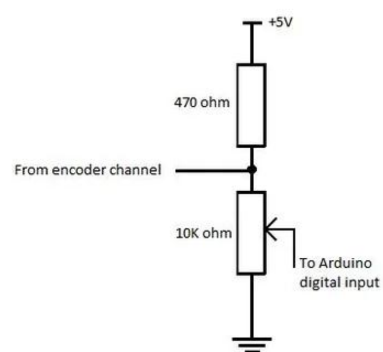

Fig. 11 Diagram of a Pull up Resistor

Arduino Connections:

  - encoder 1, channel A - connected to pin 2
  - encoder 1, channel B - connected to pin 3
  - encoder 2, channel A - connected to pin 5
  - encoder 2, channel B - connected to pin 4

(before we connect the encoders to Arduino, we need to adjust the trimmers to output exactly 3,3V. It's very important that we not exceed 3.3V, otherwise it will damage the Arduino,to do that follow:)

- connect the red wire from the encoders to the +5V Arduino pin
- connect the black wire from the encoder to ground

  (Adjust all the trimmers until you read 3.3V at the middle pin.)

Note:

- To make it more portable, solder the encoder wires to the PCB,
- add USB Solder the female ends to your PCB and the male ends to your encoders.
- make a small box with required holes and ways for passing the wires and fit everything nicely. This is just to make the system more compact and less fussy.

Communication with Stellarium:

The Stellarium plugin works just fine with the LX200 protocol.

While connected to Arduino, periodically Stellarium sends 2 strings:": GR#" - ready to receive RA and ":GD#" - to receive DEC.

(The configuration with stellarium is further explained in the report.)

Converting Alt-Azimuth to Equatorial Co-oridiantes:

- Every Alt-Azimuth mount uses two axis (altitude and azimuth) to create a coordinate system.
- However, this coordinate system is very poor if talking of astronomy. We know the Universe is in constant motion.All the stars describes a circumference centered at the pole star. They rotate 360º in about 24H (15º per hour).
- if at certain moment we are pointing to one object using a certain coordinate, after a few minutes, we need to recalibrate the coordinates because the object has moved, so, altazimuth coordinates are time dependent. They also are location dependent.
- So, to prevent using a coordinate system that changes every single second, the astronomers have found a very ingenious way to determine the sky coordinates - the equatorial coordinate system.
- In Stellarium, through the LX200 protocol, we can read equatorial coordinates, so all now we have to do is find a way to convert the readings from the altazimuth to the equatorial coordinate system.
- Once done we will write a code under the function name "AZ_to EQ()". For the same.(still in research phase)

Formulas for Converting Alt Az to Eq coordinates:

- **sinD=sinAsinL+cosAcosLcosAZ**
- **cosH=(sinA-sinLsinD)/cosLcosD**
- **RA = LST – H**
- D=declination
  H=hour angle
- A=altitude
  AZ-azimuth
  L=latitude

  RA= right ascension

  LST: Local sideral time

8) Coding:

Code Configuration:[7,8]

First we need to configure our telescope and code:

- Define the number of pulses that your encoder (1 and 2) gives by turn
- Define DUE's pins
- enter Pole Star right ascension
- enter Pole Star Hour Angle

Now upload the code in Arduino and compile it.

Once the code is done just connect it to the Arduino board and run it. Make sure you have stellarium installed. In order to sync the Stellarium software with your Arduino board follow the following steps:

- Go to the configuration window and choose the "plugins" tab. Check the box to activate the plugin.
- Click on the "telescope tool" and then select "telescope configuration".

- It will open a new window. At this window, choose "add telescope".

You are now configured and ready to use the telescope. After its done you should see the pointer moving in the Stellarium screen after you move your telescope.

9) References:

[1] Patent Number : 4,764,881
COMPUTER CONTROLLED ALTAZIMUTH TELESCOPE MOUNT
Inventor: Gilbert H. Gagnon, San Marcos, Calif.
Date of Patent: Aug. 16, 1988
[2] The Small Research Telescope Challenge: National Academies Astro2010 Position Paper
[3] Altitude Azimuthal Manuals (available from already existing manufcturers)
[4] https://skyandtelescope.org/observing/stargazers-corner/rduinoscope-boiana-diy-go-to-unit/
[5] http://astro.neutral.org/arduino/arduino-telescope-control.shtml
[6] https://stellafane.org/tm/dob/mount/index.html
[7] https://create.arduino.cc/projecthub/dEskoG/rduinoscope-4d399c
[8] Toptecboy.com Paul McWhorter Lectures of Arduino

10) Appendix:

10.1) Arduino Code:[8]

```
#include "config.h"

unsigned long seg_sideral = 1003;

const double pi = 3.14159265358979324;

volatile int lastEncoded1 = 0;

volatile long encoderValue1 = 0;

volatile int lastEncoded2 = 0;

volatile long encoderValue2 = 0;

char input[20];

char txAR[10];

char txDEC[11];

long TSL;

unsigned long t_ciclo_acumulado = 0, t_ciclo;

long Az_tel_s, Alt_tel_s;

long AR_tel_s, DEC_tel_s;

long AR_stell_s, DEC_stell_s;

double cos_phi, sin_phi;

double alt, azi;

//----------------------------------------------------------------------
--

void setup()

{

 Serial.begin(9600);

 pinMode(enc_1A, INPUT_PULLUP);

 pinMode(enc_1B, INPUT_PULLUP);

 pinMode(enc_2A, INPUT_PULLUP);

 pinMode(enc_2B, INPUT_PULLUP);

 attachInterrupt(digitalPinToInterrupt(enc_1A), Encoder1, CHANGE);

 attachInterrupt(digitalPinToInterrupt(enc_1B), Encoder1, CHANGE);

 attachInterrupt(digitalPinToInterrupt(enc_2A), Encoder2, CHANGE);

 attachInterrupt(digitalPinToInterrupt(enc_2B), Encoder2, CHANGE);

 cos_phi = cos((((latHH * 3600) + (latMM * 60) + latSS) / 3600.0) * pi / 180.0);

 sin_phi = sin((((latHH * 3600) + (latMM * 60) + latSS) / 3600.0) * pi / 180.0);

TSL = poleAR_HH * 3600 + poleAR_MM * 60 + poleAR_SS + poleH_HH * 3600 + poleH_MM * 60 + poleH_SS;
```

```
 while (TSL >= 86400) TSL = TSL - 86400;

}

//----------------------------------------------------------------------
--

void loop()

{

 t_ciclo = millis();

 if (t_ciclo_acumulado >= seg_sideral) {

 TSL++;

 t_ciclo_acumulado = t_ciclo_acumulado - seg_sideral;

 if (TSL >= 86400) {

 TSL = TSL - 86400;

 }

 }

 read_sensors();

 AZ_to_EQ();

 if (Serial.available() > 0) communication();

 t_ciclo = millis() - t_ciclo;

 t_ciclo_acumulado = t_ciclo_acumulado + t_ciclo;

}

//----------------------------------------------------------------------
--

void communication()

{

 int i = 0;

 input[i++] = Serial.read();

 delay(5);

 while ((input[i++] = Serial.read()) != '#') {

 delay(5);

 }

 input[i] = '\0';

 if (input[1] == ':' && input[2] == 'G' && input[3] == 'R' &&
input[4] == '#') {

 Serial.print(txAR);

 }

 if (input[1] == ':' && input[2] == 'G' && input[3] == 'D' &&
input[4] == '#') {

 Serial.print(txDEC);

 }

}

//----------------------------------------------------------------------
--

void read_sensors() {

 long h_deg, h_min, h_seg, A_deg, A_min, A_seg;

 if (encoderValue2 >= pulses_enc2 || encoderValue2 <= -
pulses_enc2) {

 encoderValue2 = 0;

 }

 int enc1 = encoderValue1 / 1500;

 long encoder1_temp = encoderValue1 - (enc1 * 1500);

 long map1 = enc1 * map(1500, 0, pulses_enc1, 0, 324000);

 int enc2 = encoderValue2 / 1500;

 long encoder2_temp = encoderValue2 - (enc2 * 1500);

 long map2 = enc2 * map(1500, 0, pulses_enc2, 0,
1296000);

Alt_tel_s = map1 + map (encoder1_temp, 0, pulses_enc1, 0,
324000);

 Az_tel_s = map2 + map (encoder2_temp, 0, pulses_enc2, 0,
1296000);

 if (Az_tel_s < 0) Az_tel_s = 1296000 + Az_tel_s;

 if (Az_tel_s >= 1296000) Az_tel_s = Az_tel_s - 1296000 ;

}
```

```
//-----------------------------------------------------------------------
--

void Encoder1() {

 int encoded1 = (digitalRead(enc_1A) << 1) | digitalRead(enc_1B);

 int sum = (lastEncoded1 << 2) | encoded1;

 if (sum == 0b1101 || sum == 0b0100 || sum == 0b0010 || sum == 0b1011) encoderValue1 ++;

 if (sum == 0b1110 || sum == 0b0111 || sum == 0b0001 || sum == 0b1000) encoderValue1 --;

 lastEncoded1 = encoded1;

}

//-----------------------------------------------------------------------
--

void Encoder2() {

 int encoded2 = (digitalRead(enc_2A) << 1) | digitalRead(enc_2B);

 int sum = (lastEncoded2 << 2) | encoded2;

 if (sum == 0b1101 || sum == 0b0100 || sum == 0b0010 || sum == 0b1011) encoderValue2 ++;

 if (sum == 0b1110 || sum == 0b0111 || sum == 0b0001 || sum == 0b1000) encoderValue2 --;

 lastEncoded2 = encoded2;

}

//-----------------------------------------------------------------------
--

void AZ_to_EQ()

{

 double delta_tel, sin_h, cos_h, sin_A, cos_A, sin_DEC, cos_DEC;

 double H_telRAD, h_telRAD, A_telRAD;

 long H_tel;

 long arHH, arMM, arSS;

 long decDEG, decMM, decSS;

 char sDEC_tel;

 A_telRAD = (Az_tel_s / 3600.0) * pi / 180.0;

 h_telRAD = (Alt_tel_s / 3600.0) * pi / 180.0;

 sin_h = sin(h_telRAD);

 cos_h = cos(h_telRAD);

 sin_A = sin(A_telRAD);

 cos_A = cos(A_telRAD);

 delta_tel = asin((sin_phi * sin_h) + (cos_phi * cos_h * cos_A));

 sin_DEC = sin(delta_tel);

 cos_DEC = cos(delta_tel);

 DEC_tel_s = long((delta_tel * 180.0 / pi) * 3600.0);

 while (DEC_tel_s >= 324000) {

 DEC_tel_s = DEC_tel_s - 324000;

 }

 while (DEC_tel_s <= -324000) {

 DEC_tel_s = DEC_tel_s + 324000;

 }

 H_telRAD = acos((sin_h - (sin_phi * sin_DEC)) / (cos_phi * cos_DEC));

 H_tel = long((H_telRAD * 180.0 / pi) * 240.0);

 if (sin_A >= 0) {

 H_tel = 86400 - H_tel;

 }

 AR_tel_s = TSL - H_tel;

 while (AR_tel_s >= 86400) {

 AR_tel_s = AR_tel_s - 86400;

 }

 while (AR_tel_s < 0) {

 AR_tel_s = AR_tel_s + 86400;
```

```
}
arHH = AR_tel_s / 3600;
arMM = (AR_tel_s - arHH * 3600) / 60;
arSS = (AR_tel_s - arHH * 3600) - arMM * 60;
decDEG = abs(DEC_tel_s) / 3600;
decMM = (abs(DEC_tel_s) - decDEG * 3600) / 60;
}
decSS = (abs(DEC_tel_s) - decDEG * 3600) - decMM * 60;
(DEC_tel_s < 0) ? sDEC_tel = 45 : sDEC_tel = 43;
sprintf(txAR, "%02d:%02d:%02d#", int(arHH), int(arMM), int(arSS));
sprintf(txDEC, "%c%02d%c%02d:%02d#", sDEC_tel, int(decDEG), 223, int(decMM), int(decSS));
```

10.2) CAD Drawings:

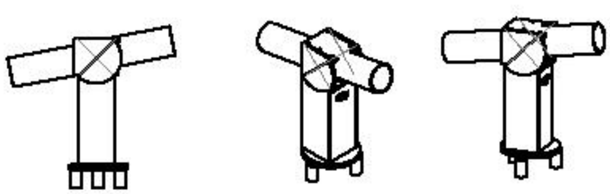

Fig. 12 Telescope Final Assembly

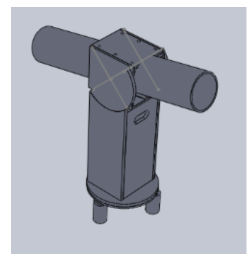

Fig 13. Isometric View of Telescope