Intusoft Newsletter

Personal Computer Circuit & System Design Tools

intusoft

Copyright © Intusoft, All Rights Reserved

Newsletter Issue #84, November 2010

Solar Power Revisited

With the ups and downs of alternative energy, solar power has maintained an increasing presence in today's alternative energy market. Its cost is becoming more justified in the industrial and consumer markets, as more businesses and people are considering its use within today's popular "green" movement. Appendix 1 reviews the progress over the past 5 years since Intusoft Newsletter 78 described the economics of solar photovoltaic ,PV, power generation. This newsletter goes on to calculate the return on investment for residential PV generation in California.

Now, the technical issue is not maximizing the energy delivered, but to maximize the cost savings. It turns out those periods of maximum solar insolence are correlated with periods of maximum energy usage. When it's hot, there is a greater need for air conditioning and to pump water if you are using a well. That increased energy usage comes at a higher price so that cost savings are maximized during the late summer months. But what can a user do to take advantage of these periodic effects? It turns out that tilting a fixed solar panel installation toward the south will increase power production. There are a simple set of geometrical relationships that lead to prediction of annual cost savings. See http://www.itacanet.org/eng/elec/solar/sun3.pdf for the geometrical relationships shown in Figure 1.
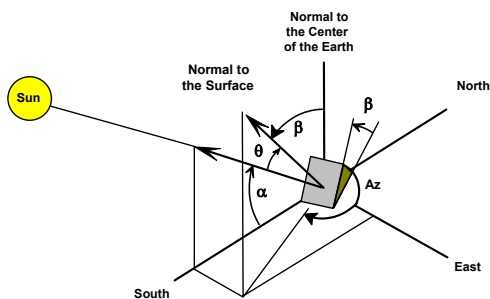

Figure 1, Geometrical relationships

The Greek characters have been replaced in the following scripts by spelling them out because the Greek alphabet isn't available to programming languages. The scripts used here are more efficient if the sine and cosine terms are evaluated ahead of time, so that the trigonometric functions need be evaluated just once. The following definitions apply:

sin(str) = s+str[0]
cos(str) = c+str[0]

For example:

$sin(\delta)$ = sin(delta) = `sd`

The earth is tilted 23.45 degrees in its orbit with respect to the sun. That tilt, measured as delta can be plotted using the following script:

```
set units=rad
day=vector(365)
delta=23.45*pi/180*
sin(2*pi*(285+day)/365.25)
plot delta day
set units=degrees
```

The script is developed from this equation:

$$\delta = 23.45 \frac{\pi}{180} \sin\left[2\pi\left(\frac{285 + day}{365.25}\right)\right] \quad (1.0)$$

Notice that some script lines wrap into the next line. When you copy and paste into a command window or an ised4 page, that won't happen.

January 1 is day[0] in this script and units are set to radians. Notice that units are changed back to their default, degrees after running the script so that other scripts are not trashed. Figure 2 is the resulting IntuScope plot. The symbol, δ, in Figure1 is replaced by `delta` in the script.
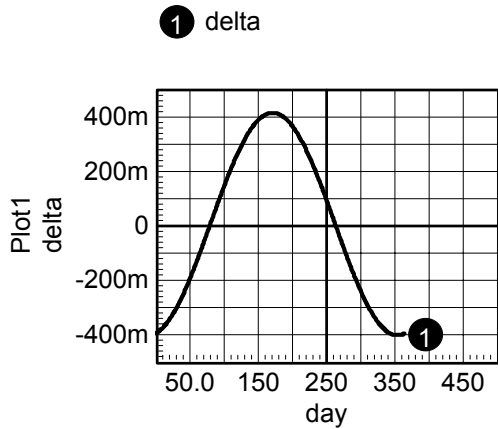
Figure 2, Variation in declination by day of the year.

The declination is the same all over the earth and nearly constant each day of the year. Therefore, delta is considered a constant for hourly calculations along with latitude, longitude, tilt and azimuth.

The hourly constants used in the equations used here are:

```
sd = sin(delta)
cd = cos(delta)
td=sd/cd (tan(delta)
sp = sin(phi), longitude
cp = cos(phi), longitude
tp=sp/cp (tan(phi)
sb = sin(beta), tilt
cb = cos(beta), tilt
tb=sb/cb, (tan(beta)
sa = sin(az), azimuth
ca = cos(az), azimuth
```

The "hourly" constants sd, cd and td are calculated each day and the others are true constants that are defined as initial conditions.

The "hour" clock, w, measures the earth's rotation and is defined to be 0 at noon, local time. Below is the IntuScope script for w:

```
w=vector(240)*2*pi/240-pi
```

The w vector divides time into 0.1-hour increments. The longitude can be used to convert between time zones.

The solar insolence, sol, at a given location, when pointing toward the sun, can be predicted as:

```
sa=sd*sp+cd*cw*cp
sol=AM1*sa
```

The equation for the altitude angle; sa=α, above, is given by the following equation:

$$\sin(\alpha) = \sin(\delta)\sin(\phi) + \cos(\delta)\cos(\phi) \quad (1.1)$$

Where α and δ are illustrated in Figure 1 and ϕ is the latitude.

This does not account for seasonal weather changes. Once the solar insolence is known, then the strength for the offset plane, shown in Figure 1, is sol*cos(θ)=sol*ct, where ct is described by the following script:

```
* script for off axis correction
cd=cos(delta)
sd = sin(delta)
cw = cos(w)
sw = sin(w))
ct = sd*sp*cb + sd*cp*sb*ca +
cd*cp*cb*cw - cd*sp*sb*ca*cw -
cd*sb*sa*sw
```

The angle, theta, in cos(θ), or ct in the above equation, is the angle between a vector pointing at the sun and a vector normal to the solar panel surface.

The equation for cos(θ), ct, is given by:

$$\begin{aligned}
\cos(\theta) = {} & \sin(\delta)\sin(\phi)\cos(\delta) \\
& + \sin(\delta)\cos(\phi)\sin(\beta)\cos(Az) \\
& + \cos(\delta)\cos(\phi)\cos(\beta)\cos(\omega) \\
& - \cos(\delta)\sin(\phi)\sin(\beta)\cos(Az)\cos(\omega) \\
& - \cos(\delta)\sin(\beta)\sin(Az)\sin(\omega) \quad (1.2)
\end{aligned}$$

The actual solar insolence will vary during the year because of local weather conditions. In California, the California Irrigation Management Information System, CIMIS, at <www.cimis.water.ca.gov>. can be used to retrieve this information. Other localities should have similar results that can be applied. In the absence of any information, reducing the AM1 level by 17% is approximately equivalent.

There are hundreds of monitoring locations throughout California that report a variety of parameters essential to crop watering management. Among the parameters is the average solar radiation per day and the average maximum daily air temperature. The temperature can be used to estimate the solar panel operating temperature and the solar radiation accounts for local weather as well as atmospheric attenuation. This is a gold mine of data, some going back to 1982, for global warming research if you are so inclined. The 10-year daily average of solar radiation is shown below in Figure 3.
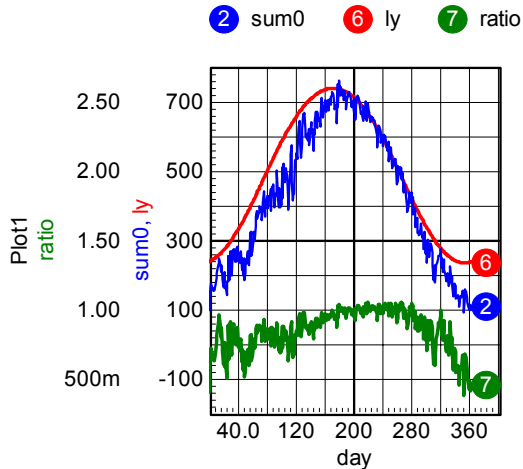
Figure 3, CIMIS 10 year average at Camino, CA in the Sierra foothills: sum0, the ideal cosine shape and the ratio.

The red curve (w6) is the ideal where the solar insolence is the value at the surface of the earth at the equator when the sun is directly overhead. This value, known as AM1, is multiplied by the sine of the elevation angle to calculate the incident radiation accounting for atmospheric attenuation. Notice the attenuation of the measured data during the "rainy" season. The script used to make the ideal curve is in appendix 2. Variation from the ideal is different around the world because seasonal weather affects the data. For this case, the incident solar energy was reduced by 17%.

The CIMIS solar radiation data is averaged over a 24-hour period and is corrected as though the sensor were pointing at the sun. We know, that according to [Sze-2nd edition] pg791 that atmospheric air mass is proportional to the secant (1/cosine) of the angle between the sun and zenith. Therefore, multiplying by the sine of the elevation angle, alpha, accounts for atmospheric attenuation (1/airmass). Then the solar energy received for a solar panel pointing at the sun is described using the following IntuScope script:

```
sa=sd*sp+cd*cw*cp
```

If the atmospheric attenuation caused by weather variations is assumed constant during the day, then the incident radiation can be multiplied by the ratio of the CIMIS data, divided by the ideal AM1 radiation shown in Figure 3. The ratio of sum0 to ly in Figure 3 can be used account for weather caused attenuation:

```
sol = AM1*ratio*sa
```

Next, in order to account for a fixed physical installation, the instantaneous value of sol is multiplied by ct, cos(θ), where θ is angle of incidence. The script for ct is:

```
ct=sd*sp*cb+sd*cp*sb*ca-
cd*sp*sb*ca*cw-cd*sb*sa*sw
```

Integrating `ct*sol` calculates the daily energy. The daily energy can be multiplied by the daily marginal cost in order to calculate the daily return in dollars per square meter for a 100% efficient system. Multiplying this result by the solar panel efficiency and the power conversion efficiency gives the final cost calculation. Repeating for each day of the year, and summing the results, gets the annual marginal cost recovery. The marginal cost is the highest price paid for energy on that day based on the previous years electric bill. The script in appendix 3 performs these calculations and the results for a summer day are shown next in Figure 4.

There are some fine points that may modify these equations. First, the spectral sensitivity of the solar panel may deviate from the CIMIS photo detector. Second, the solar panel may be more or less sensitive to off angle radiation; depending on its construction. Third, the earth's orbit is not circular, its perihelion is off by about 3.5%. Corrections for the first two of these considerations should be available from the solar panel manufacturer or may be measured by the end user. Since the daily average insolence is obtained from nearby data, the third item tends to have little effect on the results. By and large, local weather conditions appear to be considerably more important than these fine points!
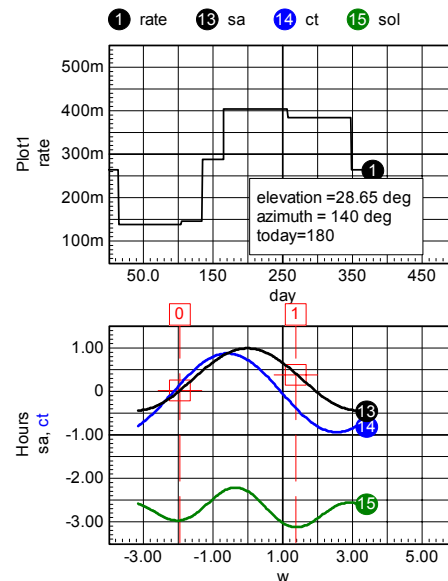

Figure 4, Result for one day.

As a rule of thumb, a root sum square, RSS, of these tolerances provides an overall tolerance on the cost recovery and will result in accuracy on the order of 5%.

Then the annual cost recovery can be computed for a number of tilt angles, β, in order to find the best angle in terms of cost recovery. Intuitively, this moves the optimal angle toward the summer months. The azimuth angle, Az, may be constrained by terrain, aesthetic values or building codes so that the annual cost recovery can be plotted vs. azimuth angle.

While this all seems pretty complex, it should be noted that real property installations have a generally accepted economic lifetime of 50 years. Small improvements become large over a 50-year time span! For example a 50 sq-meter installation can produce $159k over 50 years so savings of $1.59k per percent break even---so a few days to study the problem is worth while.

In the uppermost curve of Figure 5, the azimuth angle is 180 degrees measured from the north, so the panel is south facing. Close examination of Figure 5 reveals that when the azimuth angle is offset 20 deg to the east (curve 2:cost160), the optimum elevation angle changes from 27.28 degrees to 25.9 degrees and there is a 1.1% income loss. It can be concluded that small azimuth offsets are well tolerated and that the optimum elevation is quite a bit smaller than most researchers publish. As a sanity test, when the panel faces east, the elevation angle must be zero to maximize income, and that's the result shown in Figure5.

Remember, the seasonal atmospheric variation and the utility marginal costs are included in these calculations. The results can be reproduced by building a seasonal ratio table and a cost table and using the script in appendix 3.
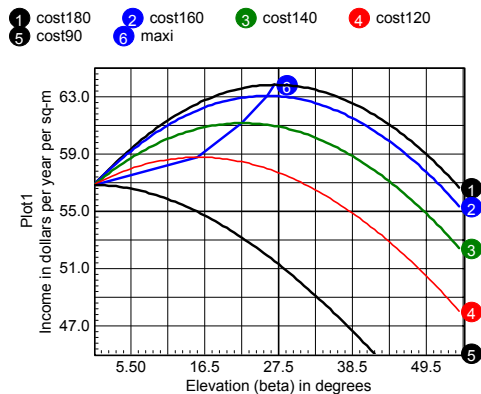
Figure 5, Income vs. elevation angle for different azimuth offsets, maxi, w6, traces the maximum cost.

So the bottom line is for an installed cost of $600 per sq-meter, the annual return on investment is 10.6% without any direct subsidy. We get to $600 watts/sq-meter using a Chinese built 1.2 sq-meter panel producing 175 watts, costing $2.5/watt That's 175*2.5/1.2 = $365.sq-meter. The remaining $235 takes care of installation, batteries and power conversion equipment. Both numbers should come down over time, even as the marginal cost recovery goes up. So, at least in California, solar power is economically viable right now!

Appendix 1, Solar Energy Today:
Back in 2005, Intusoft introduced solar cell modeling coupled with the economics of photovoltaic power generation. Not much has changed on the technology front. The benchmark for energy, that's the price of oil, has gyrated wildly between 20 and 140 dollars per barrel. The average has moved up by nearly 50%, considerably faster than the historical average. The cost of solar panels has been stubbornly constant for most of the past decade; however, in the past several years China has been manufacturing solar panels at about ½ the price of mainstream manufacturers. Much of the problem (price not declining) was caused by subsidies given by Germany and Spain, guaranteeing to purchase solar power for up to $.5 per kilowatt-hour. Evidently, this subsidy gave little incentive to reduce prices. For some reason there was a popular misconception that there was a shortage of Silicon. As we know, Silicon comes from sand, and our beaches are still very sandy! Actually the shortage was in wafer fabrication. Now that we are seeing rough economic times, the PV subsidy in Spain and Germany is going away. The government in California regulates the cost of electricity. Consumers are penalized for using too much, so there is an artificial price structure that is currently decoupled from the price of oil. The net result is a reduction in the PV generation price of about 50% over the past 5 years, which is in line with our previous prediction; although not in accordance with our price-performance model. Unpredictable government policies have played a major role in making predictions about the economics of PV generation being uncertain in the short run. Over the long run, the time-certain march of technology should override government policies.

During this time, Intusoft introduced DSP Designer. DSP Designer is a software suite that brings the power of the IsSpice4 AC, DC and TRAN analysis to the hardware world. Coupled with code generators and template designs, plus the very low price of DSP chips; their incorporation in switch mode power supply, SPMS, designs is increasing rapidly. Intusoft

builds an evaluation board consisting of dual synchronous buck regulators; the target application is for a solar panel battery charger. This belongs to a class of regulators known as micro-inverters. The idea is to provide maximum power point tracking, MPPT, on a per panel basis. Most people use this to tie the panel directly to the grid; hence the micro-inverter. However, we are concentrating on also having a battery backup, so that the grid-tie inverter is a separate component. It turns out the best economics uses one inverter for a number of solar panels. This DSP evaluation board allows the user to explore many modern SMPS problems, including synchronous bi-directional power conversion, virtual current control, MPPT and power factor correction, PFC, technology used in the grid-tie inverter. MPPT provides increased power when uncertainty over the operating environment can't be predicted. The increased efficiency produces about 15% additional power. The Intusoft regulators achieve that with 98% efficiency and at a production cost of about $20 per panel or about 10 cents per watt. That increases cost by 4% for a net energy gain of 9%. That translates into 9% fewer panels to produce the same amount of electrical power.

Other parts of the economic picture include the willingness of the utility and state controlled utility commission to continue various subsidies. If a large number of consumers generate solar power, the utility companies will lose profit and their political lobby will ask the state to revisit its policy. Assuming that occurs, a battery backup allows the user to deliver power to household loads when needed so that the power never flows back into the grid. Batteries become a fairly large cost component with life times that may approach 10 years. Electric vehicle technology will help improve the battery economic picture. Of course, the biggest argument for battery backup is energy independence. Configured properly, a backup system can supply power at a "life-support" level indefinitely. That's a big plus in communities where electrical power approaches third-world reliability.

Appendix 2, Script to predict ideal solar radiation:

```
set units=rad
*unused  here,  used  to  translate
local solar time to something else
longitude=120.00
latitude=39.084
* make the hour angle 0 at noon
w=vector(240)*2*pi/240-pi
*newplot theplot w
plot w w
```

```
AM1=975
*oct 29
today=0
phi= latitude *pi/180
sp=sin(phi)
cp=cos(phi)
day=vector(365)
Ly=vector(365)
cw=cos(w)
delta=23.45*pi/180*sin(2*pi*(285+day
)/365.25)
while today < 365
sd=sin(delta[today])
cd=cos(delta[today])
sa=sd*sp+cd*cw*cp
plot sa w
homecursors
movecursorright 0 sa 0
movecursorleft 1 sa 0
sol=AM1*integrate(sa)
plot sol w
solsum=getcursory1(sol)-
getcursory0(sol)
Ly[today]=2.06*solsum/(2*pi)
today=today+1
end
plot Ly day
set units=degrees
```

Appendix 3, Calculating annual return

```
*load ratio.grf, takes care of day
vec
Ly=vector(365)
newplot days day
plot plot1.ratio
set units=rad
delta=23.45*pi/180*sin(2*pi*(285+day
)/365.25)
plot delta
setplot constants
longitude=120.00
latitude=39.084
newplot hours
w=vector(240)*2*pi/240-pi
sw=sin(w)
cw=cos(w)
plot sw w
setplot constants
askvalues   az "azimuth  in  degrees
from north"
az=pi/180*az
today=0
AM1=975
phi=latitude*pi/180
sp=sin(phi)
cp=cos(phi)
tp=sp/cp
beta=vector(10)/10
cost = vector(10)
energy=0
```

```
nb=0
nc=0

beta[0]=1m

nb=0
saz=sin(az)
caz=cos(az)
taz=saz/caz
while (nb < 10)
        sb=sin(beta[nb])
        cb=cos(beta[nb])
        tb=sb/cb
        constants.today=0
        while today <  365

                setplot days

        constants.argd=delta[today]
                constants.todaysratio =
ratio[today]

                setplot hours
                sd=sin(argd)
                cd=cos(argd)
                td=sd/cd
                sa=sd*sp+cd*cw*cp

        ct=sd*sp*cb+sd*cp*sb*caz+cd*c
p*cb*cw-cd*sp*sb*caz*cw-cd*sb*saz*sw

        sol=AM1*todaysratio*sa*ct
                cwp=-tan(phi)*td
                ws=acos(-tp*td)
                a=cp/(saz*tb)+sp/saz
                b=td*(cp/taz-
sp/(saz*tb))
                cwpp=(a*b+sqrt((a*a-
b*b+1)))/(a*a+1)
                cwnp=(a*b-sqrt((a*a-
b*b+1)))/(a*a+1)
                *sunset
                wpp=acos(cwpp)
                wp=ws
                *sunrise
                wnp=-acos(cwnp)
                wn =-ws
                function smin(x,y)  (x <
y) * x + (x >= y) * y
                function smax(x,y)  (x >
y) * x + (x <= y) * y
                sunstart = smax(wn,wnp)
                sundone = smin(wp,wpp)
                setcursor 0 sunstart
                setcursor 1 sundone
*                 plot sa w
*                 plot ct
                soli=integrate(sol)
*                 plot  sol
                constants.energy      =
2.06*(getcursory1(      soli)      -
getcursory0(soli))/(2*pi)
                plot1.Ly[today]         =
energy
                constants.today=today+1
        end
        setplot plot1
        constants.cost[nc]=average(Ly
*rate/2.06*24m)*365*.17*.9
        setplot constants
        family[nc]=nb
        nc=nc+1
        nb=nb+1
end
setplot plot1
plot Ly
setplot constants
beta = beta*180/pi
setunits beta degrees
setunits cost dollars
plot cost beta
set units=degrees
```