

Model-Based Solar Power Generation Project (MATLAB)

This document explains a simple model-based approach to estimate solar power generation considering latitude, longitude, and time. The model is suitable for an undergraduate-level project and can be completed within a day.

Concept Overview

Solar power received by a satellite or ground system depends on the solar irradiance and the solar zenith angle, which is a function of latitude, longitude, day of the year, and time. This model computes the solar elevation angle and estimates effective solar power output.

MATLAB Code

```
%% Satellite Solar Power vs Latitude & Longitude

% Simple one-day undergraduate project

clear; clc; close all;

%% Parameters

G0 = 1361;      % W/m^2, solar constant
eta = 0.25;     % panel efficiency
A = 2;         % m^2, panel area

% Orbit parameters (simple circular LEO model)
incl_deg = 60;      % inclination in degrees
T_orbit = 95 * 60;   % orbital period ~95 minutes (in seconds)
n_orbits = 3;       % simulate 3 orbits
dt = 10;          % time step (seconds)

% Time vector
t = 0:dt:n_orbits*T_orbit; % seconds
t_hours = t/3600;         % hours

%% Choose day of year (1-365)
n_day = 80; % e.g., ~March 21 (near equinox)

%% Solar geometry: declination (deg)
delta_deg = 23.45 * sin( deg2rad(360 * (284 + n_day) / 365) );

%% Satellite sub-satellite latitude & longitude model
% Very simple model:
% - inclination controls latitude amplitude
% - Earth rotates under the orbit -> longitude changes
```

```

Re = 6371e3;          % Earth radius (m) - not heavily used here
omega_E = 360 / (24*3600); % Earth rotation rate (deg/s)

incl_rad = deg2rad(incl_deg);

% Assume argument such that latitude varies sinusoidally with time
% This is a crude but visually nice ground track.
lat_rad = incl_rad * sin(2*pi*t/T_orbit); % lat in radians
lat_deg = rad2deg(lat_rad);

% For longitude, combine orbital motion + Earth rotation
% Assume mean motion of satellite in ground track
% This is approximate and just for visualization
mean_motion_deg_per_s = 360 / T_orbit; % satellite motion
lon_deg = mod( (mean_motion_deg_per_s - omega_E).*t, 360 );
lon_deg(lon_deg > 180) = lon_deg(lon_deg > 180) - 360; % wrap to [-180,180]

%% Solar hour angle based on local solar time
% Approximate UTC time from t (start at UTC = 0)
UTC_hours = t_hours; % start at midnight UTC

% Local solar time (hours), crude approximation
LST = UTC_hours + lon_deg/15;

% Wrap LST into [0, 24)
LST = mod(LST, 24);

% Hour angle H (deg)
H_deg = 15 * (LST - 12); % 0 at local noon

%% Compute solar zenith angle
phi_rad = deg2rad(lat_deg); % latitude
delta_rad = deg2rad(delta_deg);
H_rad = deg2rad(H_deg);

cos_theta_z = sin(phi_rad).*sin(delta_rad) + ...
              cos(phi_rad).*cos(delta_rad).*cos(H_rad);

% When cos_theta_z < 0, Sun is "below horizon"
cos_theta_z(cos_theta_z < 0) = 0;

theta_z_deg = rad2deg(acos(cos_theta_z)); %#ok<NASGU> % in case you want it

%% Irradiance on panel and power

```

```

G_panel = G0 * cos_theta_z;    % W/m^2

P        = eta * A .* G_panel; % W

%% Plots

figure;

subplot(3,1,1);

plot(t/60, lat_deg, 'LineWidth', 1.2);

ylabel('Latitude (deg)');

title('Satellite Ground Track and Solar Power');

grid on;

subplot(3,1,2);

plot(t/60, lon_deg, 'LineWidth', 1.2);

ylabel('Longitude (deg)');

grid on;

subplot(3,1,3);

plot(t/60, P, 'LineWidth', 1.2);

xlabel('Time (minutes)');

ylabel('Power (W)');

grid on;

%% 3D visualization of power vs lat/long

figure;

scatter3(lon_deg, lat_deg, P, 15, P, 'filled');

xlabel('Longitude (deg)');

ylabel('Latitude (deg)');

zlabel('Power (W)');

title('Solar Power as Function of Satellite Latitude & Longitude');

grid on; box on;

view(45, 25);

colorbar;

```

How the Model Works

1. The model starts by taking latitude, longitude, day of the year, and time as inputs.
2. The solar declination angle is calculated based on the day of the year.
3. The hour angle represents the sun's position relative to solar noon.
4. Using these parameters, the solar elevation angle is computed.
5. Effective solar irradiance is calculated using the solar constant.
6. Finally, power output is estimated using panel area and efficiency.

This model can be extended for satellite applications by updating latitude and longitude dynamically using orbital data.