M & I and Control Lab

Project Report

Title:

"Solar Tracking System Using MATLAB"

Submitted To:

Engr. Muhammad Asif

Submitted By:

Jawad Ul Hassan	(21-ME-15)
Muhammad Abdullah Malik	(21-ME-103)
Ali Hassan Khalid	(21-ME-115)
Muhammad Yousaf Raza	(21-ME-163)
Muhammad Talal Azhar	(21-ME-179)

Title: Solar Tracking System Using MATLAB

Abstract:

The use of clean energy (solar, wind etc.) in other words renewable energy is becoming more important for lowering the global warming as the world becomes hotter every day as a result of global warming. Use of solar energy either PV panels or concentrated solar power (CSP) to generate electrical energy is becoming more popular. Most of the solar panels that had been used has a static direction. This report is about a study that developed a "Solar Tracking System" using MATLAB and Solar Orientation based on Location and Time.

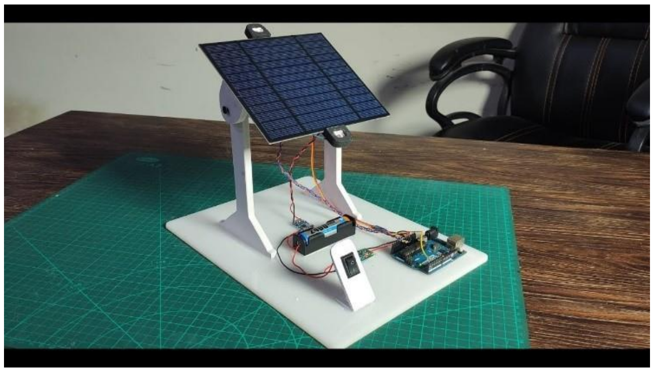


Figure 1: solar tracking system using Arduino

Introduction:

Solar energy is a renewable and sustainable source of energy with immense potential for power generation. However, the efficiency of solar panels is significantly influenced by their orientation relative to the sun. Solar tracking systems are designed to maximize energy capture by dynamically adjusting the position of solar panels to track the sun's movement throughout the day. The use of clean energy (sun, wind, etc.) or renewable energy to slow global warming is becoming more and more important, because the world is getting hotter day by day due to climate warming. Using solar energy to generate electricity, either through solar panels or concentrated solar power (CSP), is becoming increasingly common. Most of the solar panels used have a static orientation. This paper reports on a study in which a solar tracking system was developed using different methods such as conventional, PID controller and position and time based solar orientation..

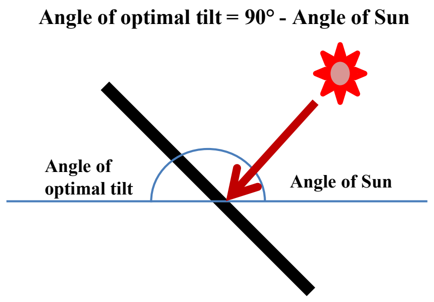


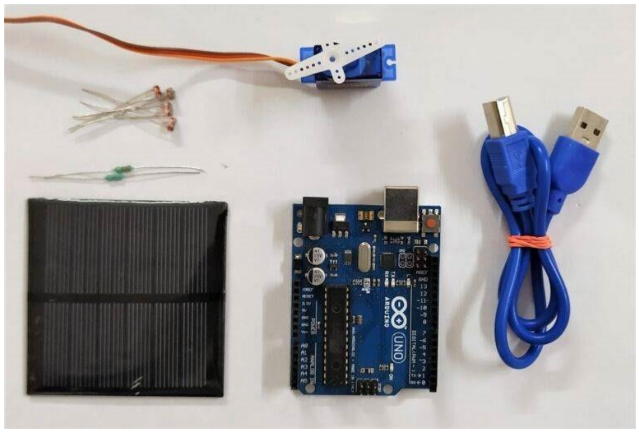
FIGURE 1: SOLAR TRACKER

Principal of Solar tracking system:

Solar tracking systems operate based on the principle of maximizing solar irradiance incident on solar panels or collectors by continuously adjusting their orientation with respect to the position of the sun. There are primarily two types of solar tracking systems: single-axis and dual-axis trackers. Single-axis trackers adjust the orientation of solar panels along one axis (typically eastwest), while dual-axis trackers adjust along both horizontal and vertical axes.

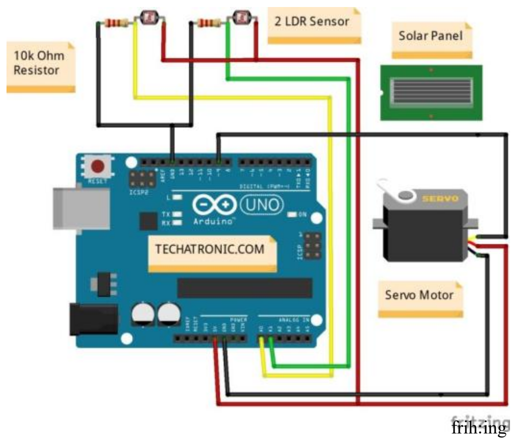
Components:

- Arduino Uno.
- Servo motor.
- Power supply for Arduino.
- LDR.
- connecting wires.
- Solar Panel (use the small size for this prototype)

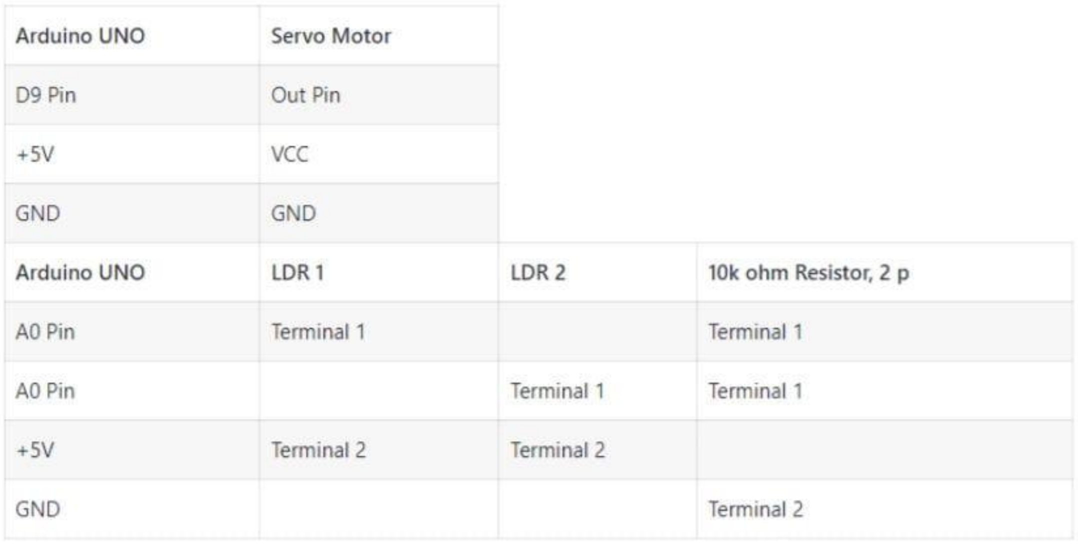


Circuit Diagram for Single Axis Solar System:

The Circuit diagram of a solar tracking system is shown below:



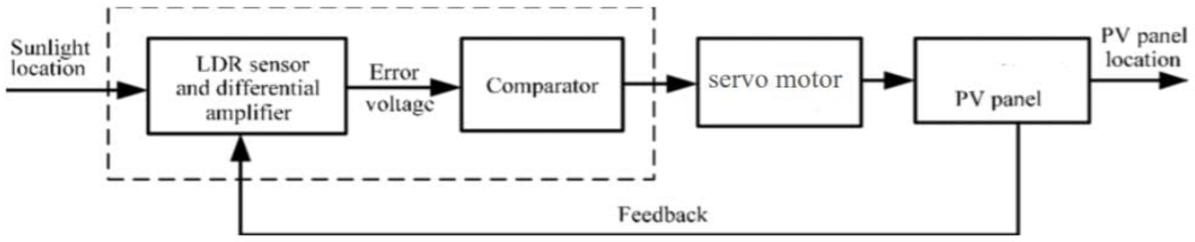
Connection Table of single axis solar tracker:



Arduino UNO	Servo Motor
D9 Pin	Out Pin
+5V	VCC
GND	GND

Arduino UNO	LDR 1	LDR 2	10k ohm Resistor, 2 p
A0 Pin	Terminal 1		Terminal 1
A0 Pin		Terminal 1	Terminal 1
+5V	Terminal 2	Terminal 2	
GND			Terminal 2

Block diagram of solar tracking system:



PID Controller:

PID controller consists of three parts use the algorithm and the system is composed of:

- P: proportional controller
- I: Integral controller
- D: derivatives controller

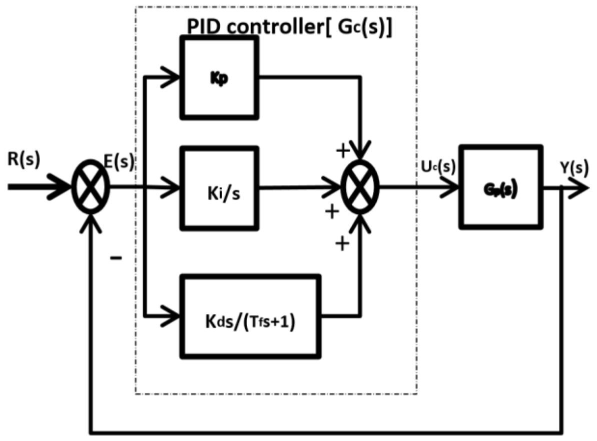


Figure 2: PID structure Transfer
function of PID controller:

The transfer function of a PID controller in a solar tracking system can be represented as:

G(s) = (Kp + Ki/s + Kd_s) / (1 + Kp + Ki/s + Kd_s)

Where:

- G(s) is the transfer function of the PID controller
- Kp is the proportional gain
- Ki is the integral gain
- Kd is the derivative gain
- s is the Laplace variable (1/s represents the derivative operator)

In the context of a solar tracking system, the PID controller is used to control the movement of the solar panel to track the sun's position. The input to the PID controller is the error signal, which is the difference between the desired sun's position and the actual solar panel position. The output of the PID controller is the control signal, which is used to drive the servo motor to move the solar panel.
The transfer function of the PID controller in a solar tracking system can be represented as:

G(s) = θ(s) / e(s)

Where:

- θ(s) is the output angular position of the solar panel
- e(s) is the input error signal (difference between desired and actual sun's position)

The PID controller transfer function can be rewritten as:

G(s) = (Kp + Ki/s + Kd_s) / (1 + Kp + Ki/s + Kd_s) = θ(s) / e(s)

This transfer function represents the relationship between the input error signal and the output angular position of the solar panel. The PID controller adjusts the gains (Kp, Ki, Kd) to minimize the error and track the sun's position accurately.

<u>Note:</u> The values of Kp, Ki, and Kd depend on the specific solar tracking system and may require tuning for optimal performance.

Sensor Transfer Function:

Transfer function => **LDR analog/LDR Voltage =1023/5**

Voltage/analog =5/1023

MATLAB Code:

```
clc
clear
% Define Arduino object
a = arduino('COM4', 'Uno');  % Change 'COM3' to match your Arduino port

% Define pin numbers for LDRs and servo
LDR1_pin = 'A2';
LDR2_pin = 'A0';
servo_pin = 'D9';

% Create servo object
servo = servo(a, servo_pin);

% Define initial servo setpoint
Spoint = 90;

% Error threshold
error = 10;

while true
   % Read LDR values (analog voltage)
   ldr1_voltage = readVoltage(a, LDR1_pin);
   ldr2_voltage = readVoltage(a, LDR2_pin);

   % Convert analog voltage values to analog readings (0 to 1023)
   ldr1 = round((ldr1_voltage / 5) * 1023);
   ldr2 = round((ldr2_voltage / 5) * 1023);
```

```
    % Calculate difference
    value1 = abs(ldr1 - ldr2);
    value2 = abs(ldr2 - ldr1);

    % Determine if adjustment is needed
    if (value1 <= error) || (value2 <= error)
        % No action needed if the difference is within error range
    else
        if ldr1 > ldr2
            if Spoint > 25
                Spoint = Spoint - 1;
            end
        elseif ldr1 < ldr2
            if Spoint < 125
                Spoint = Spoint + 1;
            end
        end
    end

    % Ensure Spoint stays within the range of 35 to 125
    Spoint = max(25, min(125, Spoint));

    % Write new servo position
    writePosition(servo, Spoint/180);  % Scale to range [0, 1]
    disp(['Servo Position: ', num2str(Spoint)]);

    % Pause to allow servo to move
    pause(0.00008);  % Delay matching Arduino sketch
end
```

Practical application:

Solar tracking systems find applications in various sectors, including:

- **Utility-Scale Solar Power Plants:** Enhancing energy production and improving the overall efficiency of solar farms.
- **Residential and Commercial Solar Installations:** Maximizing energy output from rooftop solar panels in residential and commercial buildings.
- **Concentrated Solar Power (CSP) Systems:** Optimizing sunlight concentration for thermal power generation in CSP plants.
- **Portable Solar Devices:** Enabling portable solar chargers and devices to efficiently capture sunlight for charging batteries and powering electronic devices.

Conclusion:

A solar tracking system using Arduino is a cost-effective and efficient way to optimize energy harvesting from solar panels, increasing energy output by up to 40%. The system automatically adjusts the panel's angle, eliminating manual intervention, and can be scaled up or down depending on the application. However, weather conditions, sensor accuracy, and motor control require fine-tuning for optimal performance.

Final Project: