Control of Solar Tracking to Collect Maximum Solar Irradiance

Authors: Muhammad Abdullah (393969), Azzan Rauf (373199), Syed Ahmad Shah (369231), Muhammad Saad Faran Malik (367006).
School of Electrical Engineering and Computer Science, National University of Sciences and Technology

***Abstract- T*his project focuses on the design and implementation of a solar power tracker using Light Dependent Resistors (LDRs), an Arduino Uno microcontroller, a servo motor, and a solar panel. The primary objective is to maximize the efficiency of the solar panel by orienting it towards the direction of maximum sunlight throughout the day. This is achieved through the implementation of a Proportional Integral Derivative (PID) controller algorithm in the Arduino Uno, which continuously adjusts the position of the solar panel based on the readings from the LDRs. The project demonstrates the feasibility and effectiveness of using a PID controller for solar power tracking, and by maximizing exposure to sunlight, the project underscores the potential of PID control mechanisms in enhancing solar panel efficiency, thereby advancing renewable energy utilization.**

Introduction

In the pursuit of addressing the contemporary energy challenges and contributing to sustainable solutions, our project revolves around the harnessing of renewable energy, with a specific emphasis on Control Systems. Solar energy is one of the most abundant and sustainable sources of renewable energy available to us. However, the efficiency of solar panels heavily depends on their orientation towards the sun. Fixed solar panels are not able to adapt to changes in the position of the sun throughout the day, resulting in suboptimal energy harvesting. Due to the fixed installation of the panel, the power output decreases as the angle of the sunlight on a solar panel deviates from the perpendicular because on a flat surface area, the light's average intensity decreases as the angle decreases from 90 degrees. To address this issue, solar power trackers have been developed to automatically adjust the position of solar panels to maximize their exposure to sunlight and to allow for them to be placed at certain positions other than a roof if there are space constraints.

In this project, we propose motorized solar power tracker in which a solar panel is free to rotate on its axis according to the direction of incoming sunlight. The solar tracker system utilizes LDRs to detect the intensity of sunlight in 2 different directions. These LDR readings are then processed by an Arduino Uno microcontroller, which implements a PID controller algorithm to calculate the optimal position for the solar panel. A servo motor is used to physically adjust the orientation of the solar panel based on the output of the PID controller.

Project Design Procedure

1. *Simulation of Circuit*:

   Firstly the circuit design was simulated in Proteus using an Arduino Uno block, 2 servo motors for rotation, 2 LDRs and 2 resistors. When light was brought nearer to the LDR in Proteus we saw how the servo motors reacted to the change in Stimulus and how the result was compared to our expected output.

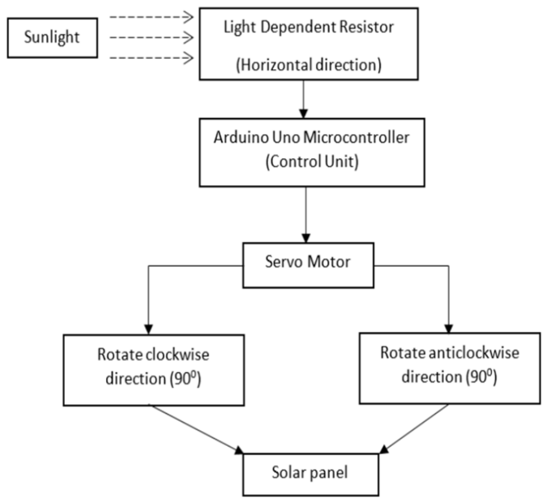

Figure 1. Basic Block Diagram of the Circuit

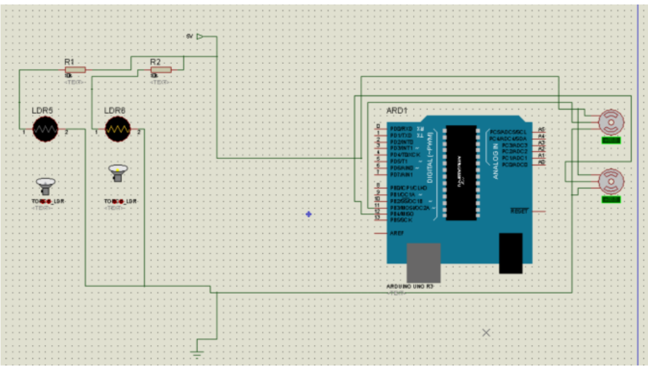

Figure 2. Proteus Simulation

2. *Using MATLAB for PID tuning:*

   Next to make our PID controller we used the MATLAB PID block and obtained the values of Kp, Ki and Kd from the "PID tuner" function in MATLAB. We then plotted curves after tuning our model along with the transfer function of the servo motor we that we used. The curves plotted in MATLAB are shown below along with our Simulink model of the PID controller and servo motor transfer function.

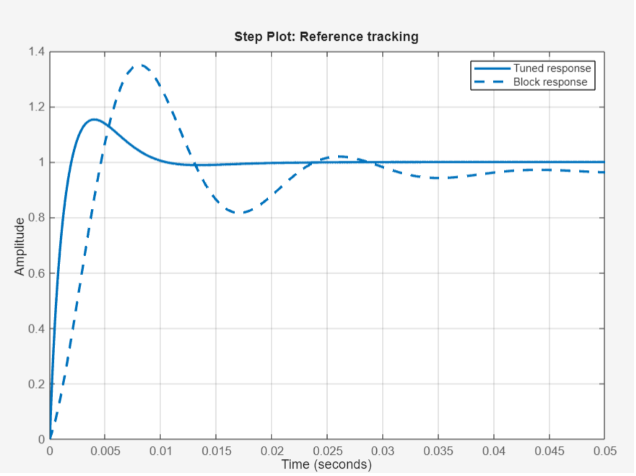

Figure 3. Plot of Tuned response vs Block response

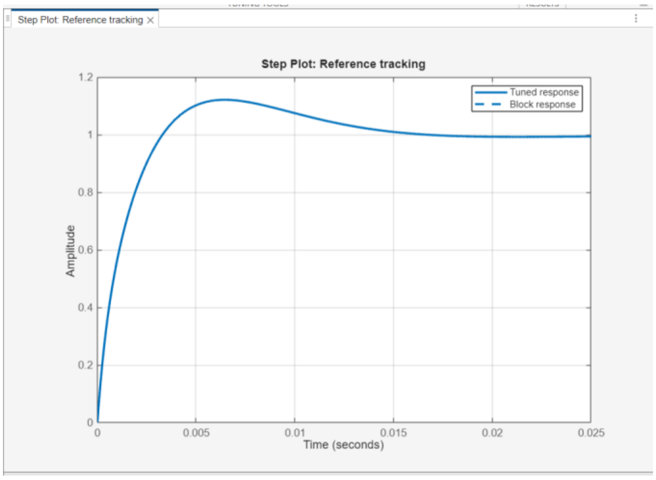

Figure 4. Tuned response

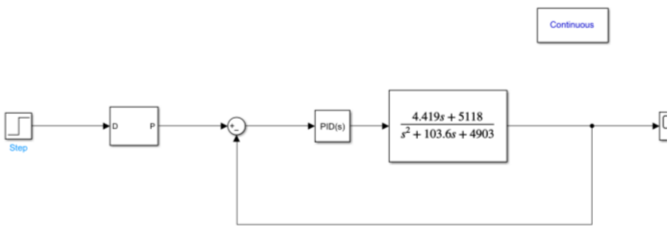

Figure 5. Simulink model of PID Controller

3. *Coding the Arduino:*

In the next step we used the values of Kp, Ki and Kd obtained from MATLAB PID tuner and used them in our Arduino code. The provided code implements a PID controller to regulate the positions of two servo motors in response to readings from two Light Dependent Resistors (LDRs). Initially, the servo motors are initialized and positioned, ensuring they are ready for operation. In the main loop, the code continuously gathers readings from the LDRs and calculates the error between them. Using this error value, the PID controller computes proportional, integral, and derivative terms to determine the appropriate adjustments needed to minimize the error and maintain alignment between the LDRs. These

```
#include <Servo.h>

#define LDR1 A0
#define LDR2 A1
#define error_threshold 10

int Spoint1 = 90;
int Spoint2 = 90;

double Kp = 43.5523;  // Proportional gain
double Ki = 3766.4748;   // Integral gain
double Kd = 0.10286; // Derivative gain

double prevError = 0;
double integral = 0;

Servo servo1;
Servo servo2;

void setup() {
 servo1.attach(11);
 servo2.attach(12);
 servo1.write(Spoint1);
 servo2.write(Spoint2);
 delay(1000);
}

void loop() {
 int ldr1 = analogRead(LDR1);
 int ldr2 = analogRead(LDR2);

 int value1 = abs(ldr1 - ldr2);
 int value2 = abs(ldr2 - ldr1);

 int error = value1 - value2;

 integral = integral + error;
 double derivative = error - prevError;

 double pidOutput = (Kp * error) + (Ki * integral) + (Kd *
derivative);

 Spoint1 = constrain(Spoint1 + pidOutput, 30, 150);  // Constrain
the servo angle between 30 and 150
 Spoint2 = constrain(Spoint2 + pidOutput, 30, 150);  // Constrain
the servo angle between 30 and 150

 //Check these values using a IF condition
 if ((value1 <= error) || (value2 <= error)) {

 } else {
  if (ldr1 > ldr2) {
   Spoint1 = --Spoint1;
  }
  if (ldr1 < ldr2) {
   Spoint1 = ++Spoint1;
  }
 }

 //Check these values using a IF condition
 if ((value1 <= error) || (value2 <= error)) {

 } else {
  if (ldr1 > ldr2) {
   Spoint2 = ++Spoint2;
  }
  if (ldr1 < ldr2) {
   Spoint2 = --Spoint2;
  }
 }
 servo1.write(Spoint1);
 servo2.write(Spoint2);

 delay(20);

 prevError = error;
}
```

Arduino Code

adjustments are applied to the servo positions, ensuring they move in the correct direction to achieve the desired balance. Through the PID control mechanism, the system can effectively and autonomously adjust the servo positions, providing a stable and responsive control system capable of maintaining alignment even in the presence of external disturbances or changes in environmental conditions.

4. *Graphical Results*

Based on the comparison graphs, solar panel with solar tracking mechanism has higher performance than a static solar panel while there is a similar value from 11.00 am until 2.00 pm due to both solar panels being positioned horizontally, facing vertically upward and receive the same amount of radiation from the Sun.
An outdoor experiment had been conducted (by A Mohamad, MTA Rahman, K Phasinam, MS Bin Mohamad "Analysis of an Arduino based solar tracking system" - IOPscience) to measure the solar parameters and the result obtained is validated by comparing it with the previous model. Although the value compared is different, but comparison graph had demonstrated a similar trend which indicates the measured value is obtained in a correct path.

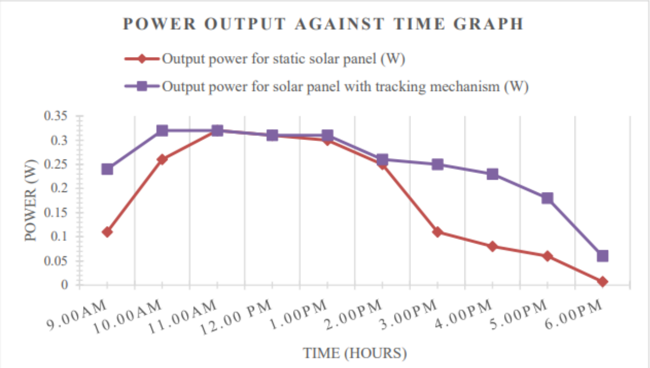

Figure 6. Power Output vs Time

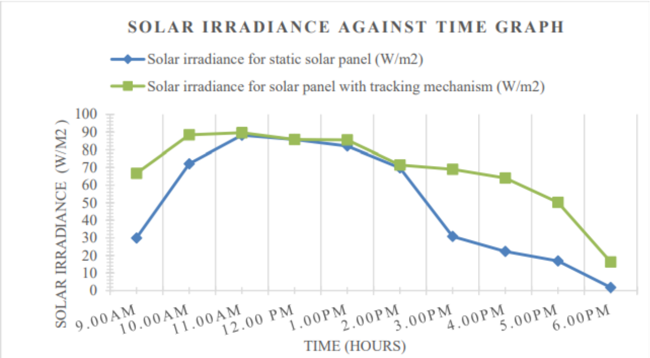

Figure 7. Solar Irradiance vs Time

Hardware Implementation:

After simulation, tuning the PID and coding our Arduino we assembled the motorized solar panel by using 2 LDRs connected to each side of the solar panel that it has to rotate on along with 2 servo motors. We connected the Arduino and assembled our circuit on a breadboard.

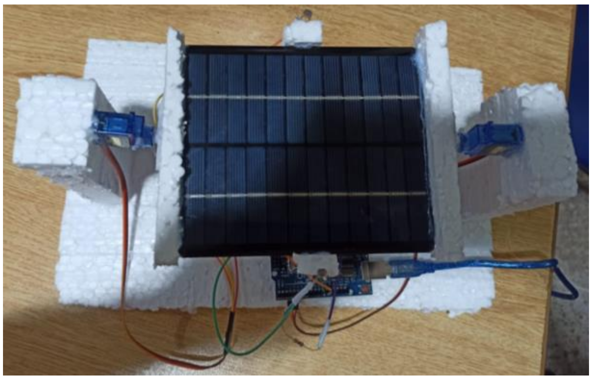

Figure 8. Final Hardware Implementation

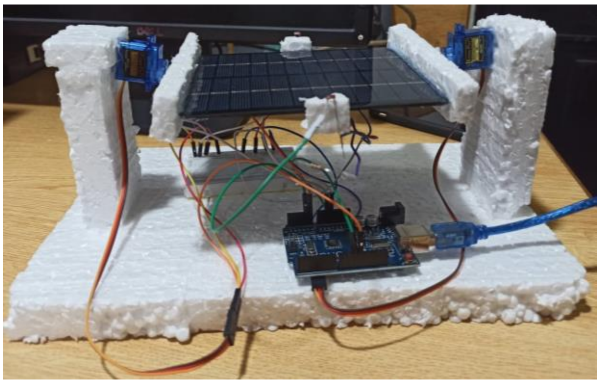

Figure 9. Final Hardware Implementation

Conclusion:

In conclusion, this project demonstrates the design and implementation of a solar power tracker utilizing Light Dependent Resistors (LDRs), an Arduino Uno microcontroller, a servo motor, and a solar panel. By integrating a Proportional Integral Derivative (PID) controller algorithm into the Arduino Uno, the system dynamically adjusts the position of the solar panel throughout the day to maximize exposure to sunlight. Through simulations in Proteus and PID tuning in MATLAB, the performance of the PID controller was optimized for efficient solar tracking. The Arduino code, employing the PID controller, effectively regulates the servo motors based on LDR readings, ensuring the solar panel maintains alignment with the sun for optimal energy harvesting. Graphical results and outdoor experiments validate the efficacy of the solar tracker, showing improved performance compared to static solar panels, especially during varying sunlight angles. Overall, this project highlights the potential of PID control mechanisms in enhancing solar panel efficiency, contributing to the advancement of renewable energy utilization.

References

PID - Arduino Reference

(PDF) Design and Implementation of Sun Tracking Solar Panel Using Microcontroller (researchgate.net)

(PDF) SG90 Servo Characterization (researchgate.net) (for servo motor transfer function)

Analysis of an Arduino based solar tracking system - IOPscience (for graphical analysis of output power and solar irradiance against time)