Chapter 1

Embedded Controller Design for Mechatronics System

Mustafa Engin

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/67987

Abstract

Many mechatronic systems have challenging control difficulties due to the high nonlinear structures and time-varying dynamic behaviors. In addition to these, there are external disturbances, which cannot be predicted and change according to uncontrolled working environments. Conventional controllers are insufficient to solve the aforementioned problems and to compensate the environmental disturbances. Therefore, adaptive controllers have been proposed as a solution to these inefficiencies of conventional controllers. Adaptive control is applied for solving the control problem of the mechatronic sun tracker that ensures the movement of the mechanism to harvest maximum energy coming from solar to the PV module surface during the sunshine duration. For this type of control problems, conventional controllers are very limited and they have a lot of deficiencies. The adaptive mechanism governed by an adaptation law is the heart of any adaptive controller. We establish the adaptation law for the plant control system using Lyapunov stability theory. This adaptation law is precise for a generic second-order system; hence, it is obviously applicable for adaptive control of other second-order systems in different realms, such as industrial production system, military, and robotics.

Keywords: embedded system, microcontroller, interfacing, adaptive control, sun tracker

1. Introduction

Time-varying dynamic behaviors of many mechatronics system and the existence of unpredictable external disturbances also present challenging control problems. If conventional controllers such as propotional integral derivative (PID), controllers are used, they must be re-tuned for different operating conditions. This task is costly, time-consuming, and even unfeasible. Unlike conventional controllers, adaptive control systems use special adaptation mechanisms that can self-tune their control parameters to recompense unforeseen deviations in the controller circuit, sensors, and their signal processing sections or in the system workplace environment. Therefore, adaptive control strategy has been progressively applied for solving difficult control problems. Adaptive control system design is much complicated than conventional controller design due to the complexity of adaptive structure and the issue of unknown or time-varying parameters.

In this chapter, we have shown the design-verification flow with various phases: mathematical model; process calculation and modeling; controller design in both continuous-time and discrete spaces; controller simulation; and in-hardware implementation and testing. The mathematical model, which is given in terms of a set of mathematical equations of energy and material balances, provides insightful understanding of the dynamic behaviors of the plant. The process calculation is necessary to obtain steady-state data for control system design in later phases. The reference model is constructed with a mathematical approach and proven to be stable and to ensure the sun tracker's steady-state properties. The discrete-time version must be accomplished prior to implementation of the adaptive design on digital computers. We successfully design and implement the embedded adaptive controller using ARM Cortex-M core processor. We apply the state-of-the-art testing technique in which we took performance data over a data logger. During the system test process, the embedded model reference adaptive controller effectively controlled the attitude of sun tracker and completely fulfilled all controller objectives.

2. Principles of adaptive control

The interest in designing autonomous working machine for such mechatronic systems motivated the development of adaptive control theory, and first steps were taken to automatically adjust the controller parameters to changing mechanism dynamics during the working. There were some stability problems, which arose with the usage of the developed adaptive control theory. Several researchers studied instability of and modifications in the adaptation mechanism of the adaptive control [1]. After these studies, several improvements were added to the theory, and efforts to develop the adaptive control theory have continued to the present day.

As shown in **Figure 1**, an adaptive control system has two loops. The first one is feedback loop same as all traditional closed-loop control systems. Second one is parameter adjustment loop, which is inherently slow feedback loop. Adaptation scheme can be adjusted to controller parameters in two different methods. In the indirect method, at any adjustment step, new controller parameters are obtained by solving the design problem such as self-tuning regulators. Direct adaptive controller technique means that the controller parameters are directly updated each time step by means of an adjustment rule without the characteristics of the process, and its disturbances being determined first. In the literature, it is seen that three different adaptive control schemes are preferred in the control of mechatronic systems. First one is gain-scheduling adaptive control. It needs additional measurement about environment and plant operation parameters. Second one is self-tuning adaptive control. It observes the plant output and adjusts controller gain to maintain the desired output level. Third one is model reference adaptive control. It uses a model that is approximately similar with plant. Every step compares the reference model output and plant output and then adjusts controller gain to get the desired plant output level.

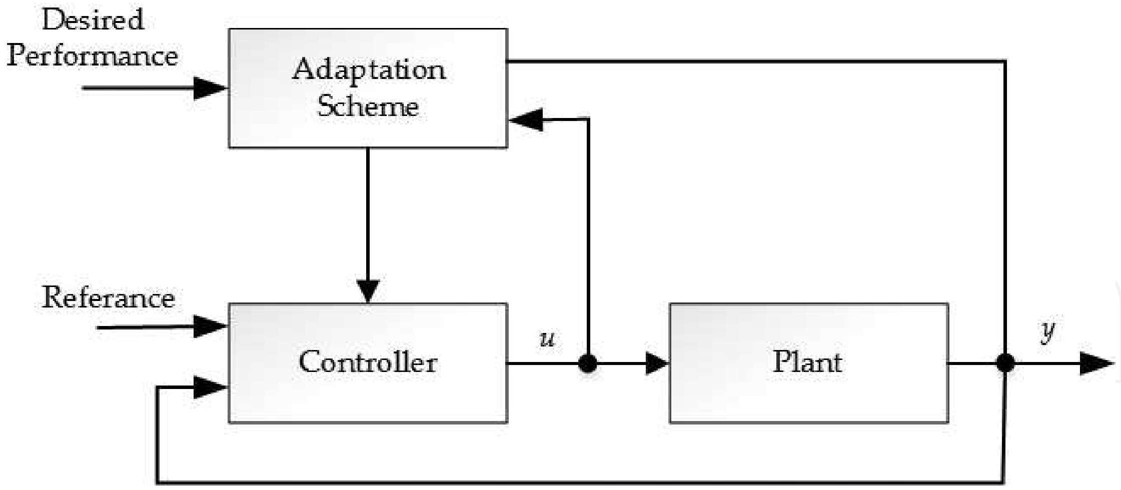

Figure 1. Block diagram of general adaptive controller.

2.1. Gain-scheduling adaptive control scheme

The gain-scheduling approach comes out of the simple idea of changing the gains of a controller according to variations in the operation. This method is very strong in cases where the parameters of the plant change with respect to a subsidiary variable that is available for measurement. Gain-scheduling control in fact is a mapping operation from plant parameters to controller parameters; hence, it can be implemented as a function or a lookup table especially in digital systems. As shown in **Figure 2**, gain-scheduling scheme needs auxiliary measurements to adjust the controller gain.

Despite the many successful industrial applications, traditional gain-scheduling control was considered as an applied approach before 1990 and its theoretical properties were not sufficiently studied [2]. Traditional gain-scheduling linearizes the plant model on predefined operating point and a local controller is set up around this point. Giving probable disturbance variation to the local controller as input and measuring plant output parameter for each input,

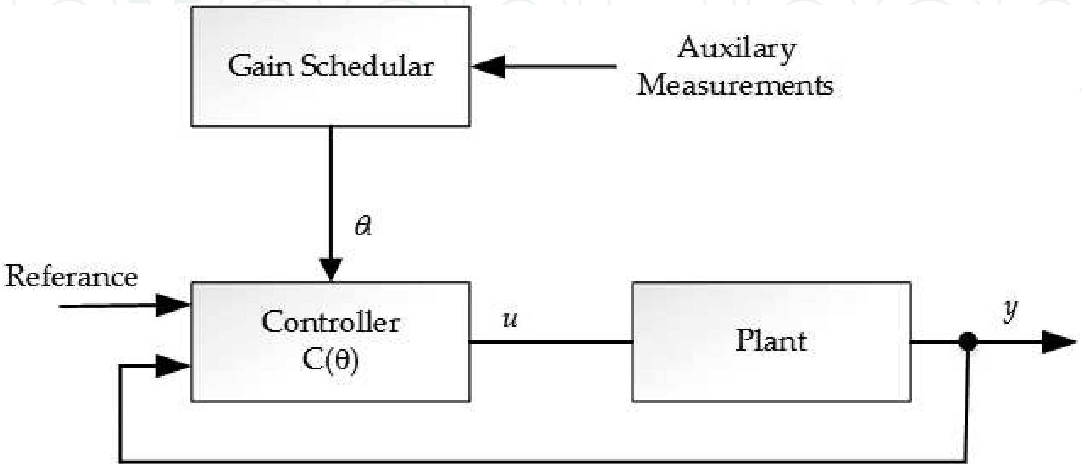

Figure 2. Adaptive controller with gain-scheduling scheme.

the measured output parameter is averaged to define plant output, also controller gain. Controller output unification is such a technique, for example used in Ref. [3], where different controllers are applied at the same time, then outputs are interpolated to determine controller output. Concerning closed-loop behavior, this attitude brings on fine outcomes; nonetheless, the closed-loop stability is only guaranteed for selected fixed operating points. The biggest advantage of traditional gain-scheduling scheme is that gain-scheduling controller always exists when stability conditions are covered.

Modern gain-scheduling control techniques that guarantee global stability were developed in the 1990s [4, 5]. These techniques need online measurements for the varying parameters and also their derivatives. Although a large amount of research has recently been carried out on synthesis and analysis, the existing techniques can still be very conservative for practical applications [6–8]. Before elaborating on the modern gain-scheduling control techniques found in the literature, first some techniques are presented to identify linear parameter varying (LPV) models, since modern gain-scheduling control techniques synthesize controllers directly from LPV of the system. LPV modeling techniques are still under development and a practical and easy algorithm to make an LPV model from experimental system measurements is still missing [9]. In some applications, LPV models are obtained by transforming the underlying physical laws to a parameter-dependent state-space form [10]. This approach remains restricted to academic examples, since it is too time-consuming to reveal the underlying physical laws for industrially relevant large-scale systems. Furthermore, it is difficult to decide which physical effects are applicable and must be involved in the model and which effects can be ignored. Therefore, experimental system identification techniques are more appropriate for industrial applications.

The main problems with gain scheduling are that stability and performance are assured only at the local stationary operating point, but the global stability of the parameter-dependent system is not assured for fast deviations of the fluctuating parameter. Therefore, the applicability of the technique relies on a significant number of simulations and real-time experiments [11]. Even so, it is still one of the powerful adaptive control approaches that is widely used in industrial applications.

2.2. Self-tuning adaptive control scheme

As shown in **Figure 3**, self-tuning adaptive controller includes an estimator section that uses plant output and input parameters as input data to estimate the controller gain tuning.

In the self-tuning adaptive control scheme, an analytical relation between the plant parameters and controller gains is evaluated. During the application, the varying system parameters are identified using a parameter identification technique. These identified system parameters are used to calculate adaptive controller gains through the analytical relation evaluated. Several identification methods such as least squares, maximum likelihood and extended Kalman filtering can be used for parameter identification [12, 13]. Furthermore, various analytical procedures, which use system parameters for control design, can be used for self-tuning control, for example, gain-phase margin design and linear quadratic regulator. This approach was originally proposed in Ref. [14] and explained clearly in Ref. [15]. Various applications of the self-tuning control on spacecrafts were demonstrated in Refs. [16, 17].

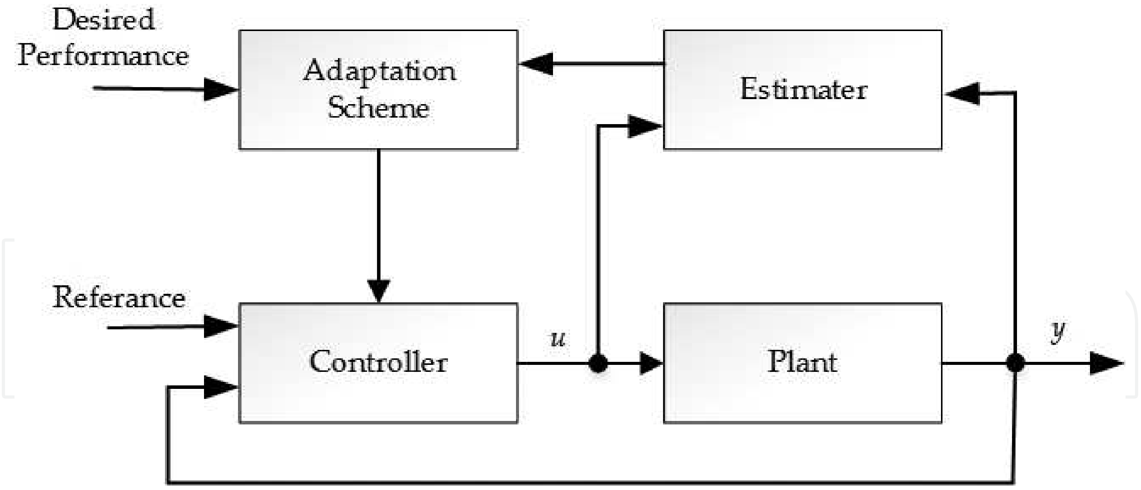

Figure 3. Self-tuning adaptive control system.

Paradoxically, the rather poor performance of the control function leads to unusually striking variations in process variables during control operation. While self-tuning technology has not yet comply perfected, its benefits may be worth the risk. A self-adjusting controller can monitor and adjust the changing gains and time constants of the process to match the adjustment. Even a stationary process may show variable accuracy and time constants if they are not sufficiently linear.

2.3. Model reference adaptive control scheme

MRAC is especially effective when the objective is to track the output with a command signal or trajectory [18, 19]. As shown in **Figure 4**, MRAC scheme uses a second plant model that behaves like an ideal plant to adjust the controller gain. This control scheme can also be used

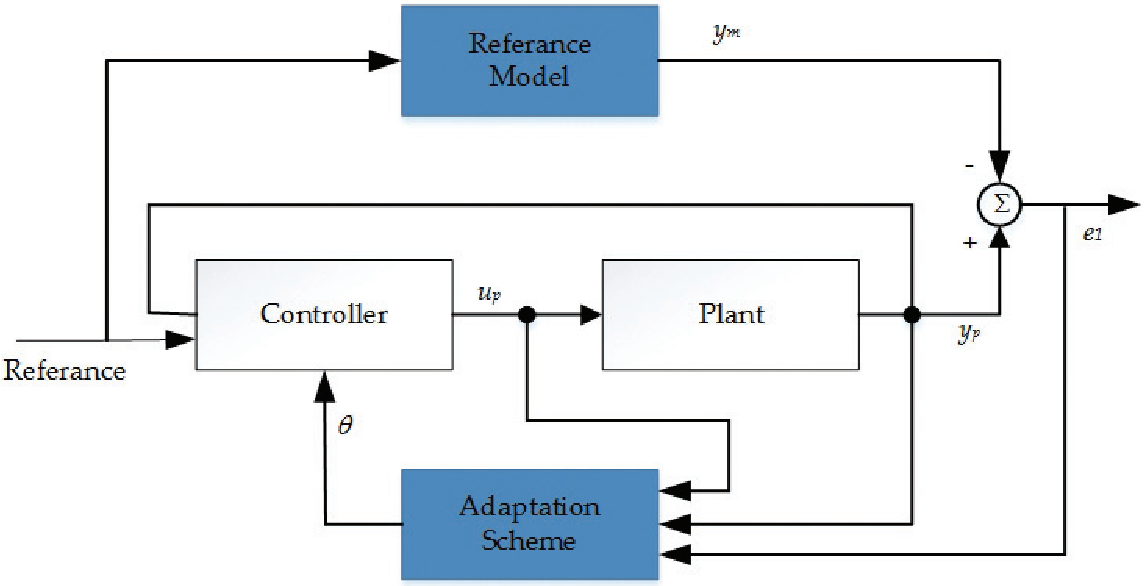

Figure 4. Block diagram of the MRAC.

for controlling unknown LTI plants, both stable and unstable. Two algorithm variations are common in implementing the MRAC scheme. First is the direct model MRAC in which the adaptation is applied using the implicit development of the plant model structure. In the second variation, the model of the plant is developed explicitly. To date the theory behind this technique, and its formal stability proofs have been so matured that there are several important designs using MRAC in the industry with special processors or microcontrollers for their implementation. This technique does not use online identification of the process and the disturbances; the controller also does not exploit the a priori knowledge of a parameter-dependent model, but simply controls the deviations on the plant and output of the reference model. Since the controller is only adjusted when considerable errors occur, this approach is more suited for processes with very slow parameter variations.

Depending on the application, different online parameter estimators can be preferred; this leads to further classification of MRAC. In many mechatronic systems for the defined operating point, system dynamics can be mathematically modeled representing the MRAC general system model.

Objective of the MRAC is to adjust the plant output y_p to the desired level. To describe the plant input u_p, first calculate the reference model output y_m for particular reference input signal $r(t)$. Assuming the system is single input single output (SISO) LTI, its transfer function, $G_p(s)$, is described by Eq. (1).

$$G_p(s) = \frac{y_p}{u_p} = k_p \frac{Z_p(s)}{R_p(s)} \tag{1}$$

k_p is the high frequency gain of plant known sign, $R_p(s)$ and $Z_p(s)$ are polynomials and their degrees are known, and they are n, and m respectively. After polynomial degrees are defined, system relative degree becomes, $n^* = n - m$. Relative degree defines system solution method. For $n^* = 1, 2$ proposed preceding solution methods can be followed for a mechatronic system, which has the same relative degree. But extending the same solution to systems that have $n^* \geq 3$ is still a major obstacle for ongoing researches. According to model reference control scheme, reference model is built similar to plant characteristics. Thus, the reference model transfer function, $W_m(s)$, can be described as follows [1].

$$W_m(s) = \frac{y_m}{r} = k_m \frac{Z_m(s)}{R_m(s)} \tag{2}$$

$Z_m(s)$ and $R_m(s)$ are polynomials same as the plant polynomials, and k_m is high frequency gain of the reference model. Combining control law and known plant parameters, we should choose an adaptive law that allows real-time estimates for the controller parameters. Take in consideration this idea of the control law:

$$u_p = \Theta_1^{*T} \frac{\alpha(s)}{\bigwedge(s)} u_p + \Theta_2^{*T} \frac{\alpha(s)}{\bigwedge(s)} y_p + \Theta_3^* y_p + \Theta_4^* r \tag{3}$$

$\alpha(s) = [s^{n-2}, \ \ldots.., s, 1]^T$ if $n \geq 2$ otherwise $\alpha(s) = 0$, $\Theta_1^*, \ \Theta_2^* \in \mathcal{R}^{n-1}, \Theta_3^*, \ \Theta_4^* \in \mathcal{R}^1$ are constants and they will be defined during system modeling process. $\bigwedge(s)$ is a random Hurwitz polynomial with the degree $n-1$ and $\bigwedge(s) = \bigwedge_0(s) \ Z_m(s)$.

Referring selected control law (3), general plant model can be written in the form

$$y_p = k_p \frac{s + b_0}{s^2 + a_1 s + a_0} u_p \tag{4}$$

Notice that the transfer function (4) has the same form as the plant model. As noted before, reference relative degree should be chosen the same as that of the plant.

$$y_m = W_m(s) r = k_m \frac{s + b_{m0}}{s^2 + a_{m1} s + a_{m0}} r \tag{5}$$

Control law can be rewritten according to the newly defined plant model and reference model.

$$u_p = \Theta_1^* \frac{1}{(s+\lambda)} u_p + \Theta_2^* \frac{1}{(s+\lambda)} y_p + \Theta_3^* y_p + \Theta_4^* r \tag{6}$$

$\alpha(s) = 1$ and $\bigwedge(s) = (s + \lambda)$. The control law (Eq. (7)) can be rewritten in the form

$$u_p = \frac{\Theta_2^* + \Theta_3^*(s+\lambda)}{(s+\lambda) - \Theta_1^*} y_p + \frac{\Theta_4^*(s+\lambda)}{(s+\lambda) - \Theta_1^*} r \tag{7}$$

The closed-loop transfer function is reformulated using new form of the control law (7) and plant model (4), then:

$$\frac{y_p}{r} = G_c(s) = \frac{k_p(s+b_0)\Theta_4^*(s+\lambda)}{(s^2 + a_1 s + a_0)\Big((s+\lambda) - \Theta_1^*\Big) - k_p(s+b_0)(\Theta_2^* + \Theta_3^*(s+\lambda))} \tag{8}$$

By matching $G_c(s) = W_m(s)$ and defining $\theta_4^* = \frac{k_m}{k_p}$ and $\lambda = b_{m0}$ we have

$$(s^2 + a_1 s + a_0)\Big((s+\lambda) - \Theta_1^*\Big) - k_p(s+b_0)\Big(\Theta_2^* + \Theta_3^*(s+\lambda)\Big) = (s+b_0)(s^2 + a_{m1} s + a_{m0}) \cdots \tag{9}$$

By taking the coefficients of s on both sides of Eq. (9), it can be expressed as the matrix algebraic form:

$$\begin{bmatrix} -1 & 0 & -k_p \\ -a_1 & -k_p & -(k_p b_{m0} + k_p b_0) \\ -a_0 & -k_p b_0 & -k_p b_0 b_{m0} \end{bmatrix} \begin{bmatrix} \theta_1^* \\ \theta_2^* \\ \theta_3^* \end{bmatrix} = \begin{bmatrix} a_{m1} + b_0 - b_{m0} - a_1 \\ a_{m0} + b_0 a_{m1} + a_1 b_{m0} - a_0 \\ b_0 a_{m0} + a_0 b_{m0} \end{bmatrix} \tag{10}$$

If we know plant parameters that is in the control law (6), and with parameters $\Theta_4^* = \frac{k_m}{k_p}$ then we can solve the model reference problem using Eq. (10).

In this chapter, Lyapunov's stability theory will be used to calculate adjusting parameters in the MRAC system. First, a differential equation which contains adjusting parameters for error, $e = y_p - y_m$ is derived. Then, Lyapunov function and adaptation mechanism are set according to error.

For a given continuous-time mechatronic system, plant and reference model steady-state space form can be written as follows:

$$\dot{x} = Ax + Bu,\ \dot{x}_m = A_m x_m\ +\ B_m u_c \tag{11}$$

$\boldsymbol{x}$ is state vector of plant and A, and B are constant matrices with suitable dimensions with the system transfer function. Similarly, x_m state vector of plant and A_m and B_m are constant matrices. The control law can be represented in state space form:

$$u(t) =\ Mu_c(t) - Lx(t),\ e(t) =\ x(t) - x_m(t) \tag{12}$$

For given control law, time derivative of error function can be written as

$$\frac{de}{dt} =\ Ax\ +\ Bu_p - A_m x_m -\ B_m u_c \tag{13}$$

$$=\ A_m e\ +\ (A - A_m - BL)x\ +\ (BM - B_m)u_c \tag{14}$$

$$=\ A_m e\ +\ \Psi(x, uc)\cdot(\theta - \theta^0) \tag{15}$$

We can choose following equation as a candidate Lyapunov function.

$$V = \frac{1}{2}\left[e^T Pe + \frac{1}{\gamma}(\theta - \theta^0)^T(\theta - \theta^0)\right] \tag{16}$$

Time derivative of Lyapunov function

$$\dot{V} = \frac{1}{2}e^T[PA_m + A_m^T P]e + (\theta - \theta^0)^T\Psi^T Pe + \frac{1}{\gamma}(\theta - \theta^0)^T\dot{\theta} \tag{17}$$

We can solve Lyapunov function for $P = P^T > 0$

$$PA_m + A_m^T P = -Q,\ Q > 0 \tag{18}$$

And the updated law can be stated as follows.

$$\dot{\theta} = -\gamma\Psi^T Pe = -\gamma\Psi^T(x,\ u_c)\cdot P\cdot(x - x_m) \tag{19}$$

Then, the updated law stated time derivative of Lyapunov function can be rearranged:

$$\dot{V} = -\frac{1}{2}e^T(t)Qe(t),\ e(t) \to \infty \tag{20}$$

3. Case study: design embedded controller for mechatronic sun tracker

Because the gain parameters of the PID controller, which deviate depend on environmental disturbance and working conditions, cannot be tuned during operation, adaptive control methods were offered to eliminate this problem. Photovoltaic (PV) panel only converts solar radiation to electric current when the sun's rays fall perpendicular to the panel surface. Naturally, it is possible only solar noon every day. By adjusting the PV panel to the sun starting from sunrise to sunset, the PV panel output performance may rise up to 46% when compared with fixed slope PV panel [20]. Sun rays' incidence angle changes largely diurnally and slightly seasonally. Sun tracker controller includes two main sections; the first section starts with defining the sun's trajectory. It needs location, time, and date data. Second section is adaptive position control system. Serial tracker architectures consisting of revolute and prismatic joints ensure the rotational movement with two degrees-of-freedom. The tracker is base-fixed on the ground, and an apparatus joins the base to the support of the panels with three legs. This two-part apparatus is designed to revolve the panel-supporting mechanism around a vertical axis. The second part adjusts the slope of the PV panel by prismatic junction. Prismatic junction is controlled by DC motor-driven linear actuators. It is highly precise by design, especially when compared to pneumatic and hydraulic actuators. Screw-based mechanical linear actuators allow to advance or retreat the motive rod by extremely small increments, which is required for the exact positioning of the sun tracker. Electric linear actuator consumes extremely low electric energy available in 12 V DC that can be supplied by the solar panel supported by a battery. Linear actuators can be unusually small, especially when considering the range of motion that is required for moving the sun tracker.

3.1. PV panel attitude model

The robotic manipulator orientation can be defined to different frames, which are reference and sun, and three angles, which are roll, pitch, and yaw. These angles are usually denoted by Ψ, θ, and φ. Sun tracker has two degrees-of-freedom, and to find the sun it has to make two successive rotations. These two rotations can be expressed as compound matrices. It is necessary to get the matrix multiplication of the two preview rotations to obtain the rotation matrix from reference to the sun. Details of the kinematic analysis of sun tracker can be seen in Ref. [21].

The sun tracking algorithm calculates the elevation, α and azimuth, Ψ, angles using latitude, longitude, and date and time data. Then, it uses these parameters to define the slope of the PV panel. α is the solar elevation angle that shows the orientation of the system in the vertical plane and can be defined using Eq. (21):

$$\alpha = \operatorname{asin}(\sin\delta\sin\phi + \cos\delta\cos\phi\cos\omega) \tag{21}$$

where ϕ is the latitude, ω is the hour angle, and δ is the solar declination. It can be defined using Cooper's equation [22].

$$\delta = 23.45\sin\left(\frac{360}{365}(284 + \mathrm{N})\right) \tag{22}$$

N is Julian day and azimuth angle, Ψ, is calculated by using Eq. (23).

$$\Psi = \operatorname{asin}\left(\frac{\cos\delta\sin\omega}{\cos\alpha}\right) \tag{23}$$

3.2. Mathematical model of tracker

As shown in **Figure 5**, the main components of a sun tracker for controller design are two DC motors, sensors to define input reference signal, and position sensors. Mathematical model of the sun tracker mainly deals with actuators for both axes. Both axes have the same type of DC motor except the gear reducer. The major differential equations are given in Eqs. (24) and (25). The Laplace transform of DC motor model equations is given in Eqs. (27)–(30) and the block diagram of this model is shown in **Figure 6**.

$$\frac{di}{dt} = \frac{1}{R}\left(Ri + V_a - K_e\frac{d\theta}{dt}\right) \tag{24}$$

$$\frac{d^2\theta}{d^2t} = \frac{1}{J}\left(K_t i - b\frac{d\theta}{dt}\right) \tag{25}$$

$$v_b(s) = K_r w(s) \tag{26}$$

$$I(s) = \frac{V_a + K_t w(s)}{R + Ls} \tag{27}$$

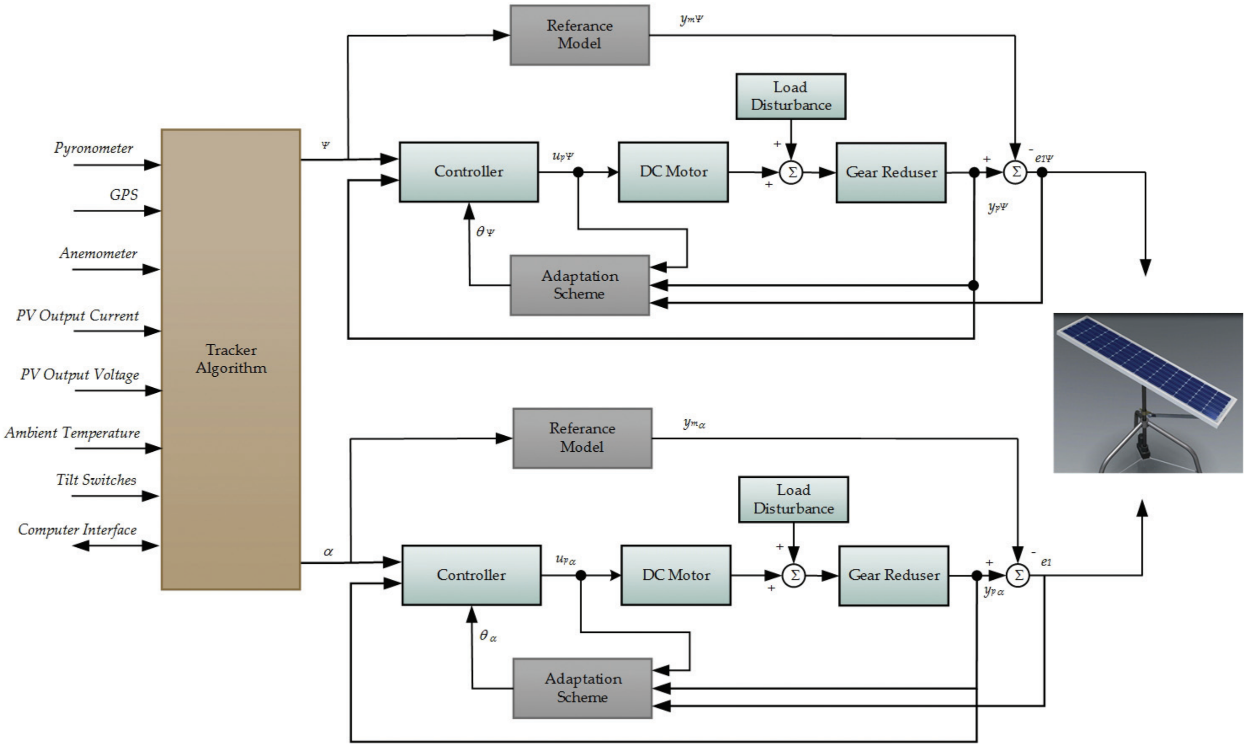

Figure 5. Components of the sun tracker.

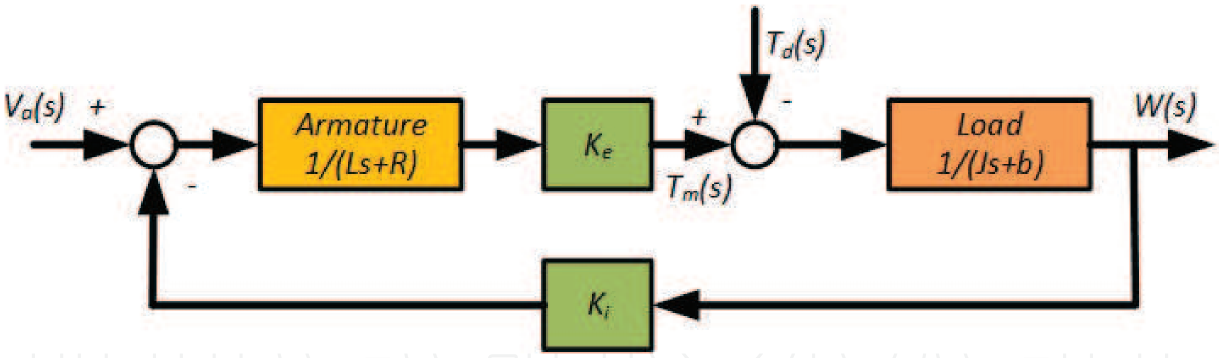

Figure 6. DC motor Simulink model.

$$T_l(s) = Js^2\theta(s) + bs\theta(s) \tag{28}$$

$$T_l(s) = T_m(s) - T_d(s) \tag{29}$$

where I is armature current, V_a is armature voltage, R is the armature resistance, L is armature inductance, V_b is back-emf, K_e is motor coefficient, K_t is back-emf coefficient, J is rotor inertia, b is viscous damping ratio, T_m is motor torque, T_l is load torque, and T_d is the disturbance torque. According to **Figure 7**, the transfer function of DC motor can be represented in Eq. (30).

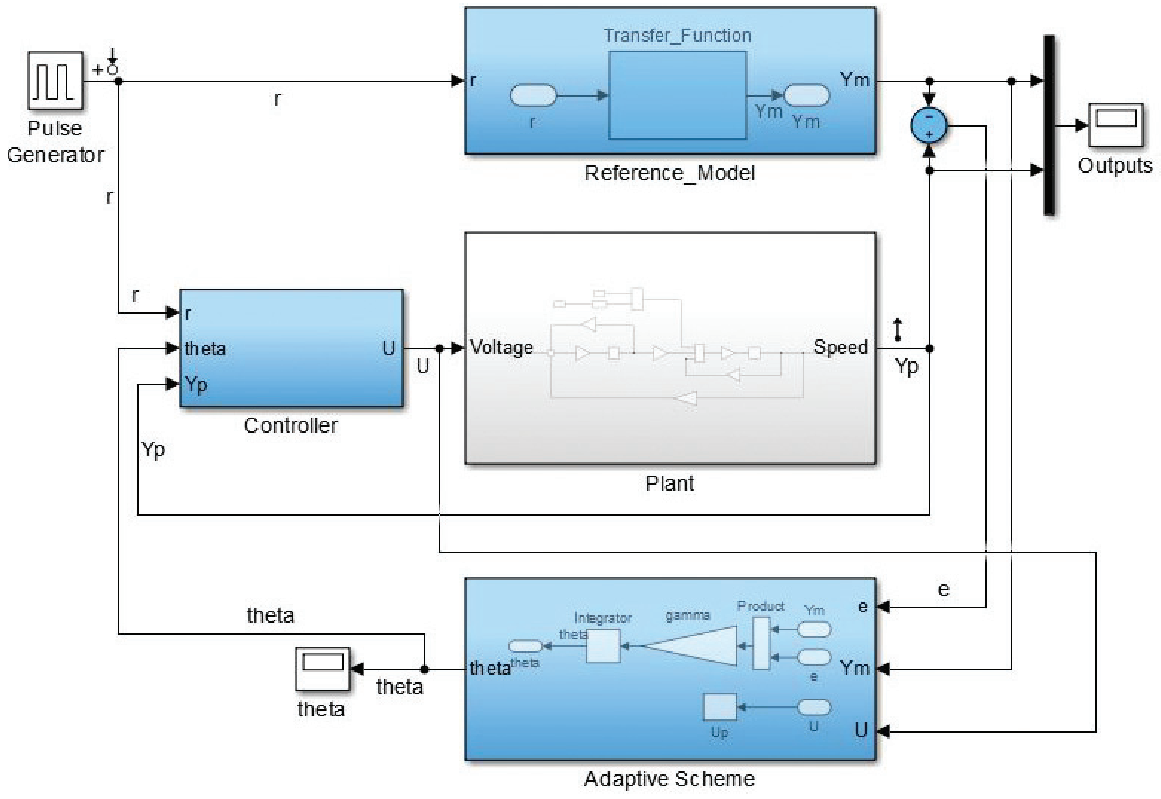

Figure 7. Simulink simulation model of MRAC.

$$\frac{w(s)}{V_a(s)} = \frac{K_e}{(R + Ls)(Js + bs) + K_e K_t} \tag{30}$$

The sun tracker Simulink can be built using DC motor mathematical model, Eqs. (24) and (30), and adaptive model reference controller scheme, Eqs. (6), (9), (24), and (30). As shown in **Figure 8**, Simulink model consists of three main parts: reference model, plant, and adaptive controller. Plant disturbance, T_d, can be changed manually and plant output y_p adapts this variation quickly. Simulation results are shown in **Figure 8** for motor speed and motor voltage.

The hardware design assembles the embedded microcontroller to the solar rotation mechanism consisting of two rotary DC motors driven by PMDC drives, linear actuators controlled by these DC motors, a pyranometer for solar radiation measurements, an anemometer for obtaining wind speed data, a global positioning system (GPS), tilt switches and potentiometers for position sensing. The GPS data in sentences that contain a string of characters is transferred to the microcontroller continuously through serial RS-232 port. The content of these sentences contains the longitude, latitude, altitude, date, and time for the specific location. As a result, the GPS clock signal is used to update the microcontroller's internal time periodically, and thus effects of the long-term errors are eliminated. As part of the effort to improve solar tracker reliability and performance, a pyranometer has been added to the solar tracker. This pyranometer, which is placed on the tracker, allows the data acquisition system to measure precisely the irradiance faced by the PV modules. Besides, monitoring developed algorithm uses pyranometer during optimization process to predict tilt and azimuth angle paths. In solar tracker systems, azimuth angle is usually measured with a potentiometer.

The solar tracker' position is bounded by mechanical limit switches to ensure robust operation. A micro-roller switch mounted on the base of the solar tracker prevents multiple revolution windup of the azimuth tracking stage. Two more limit switches are used for the zenith

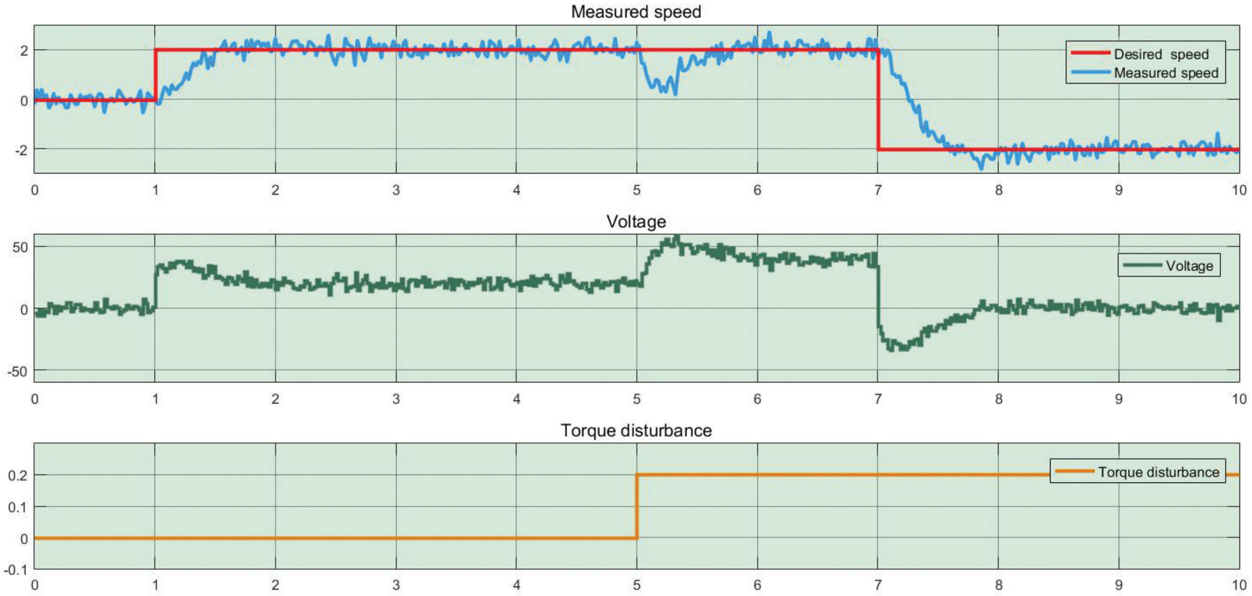

Figure 8. Simulink simulation results for a given input reference signal.

tracking stage to prevent over-travel damage to the linear actuator mechanism. Initial reset balancing is provided by four tilt switches (east, west, south and north). An anemometer measures the local wind speed to protect tracker components from over-wind speed. For more data on hardware design, see Ref. [23].

3.3. Software design

The developed sun tracking algorithm precisely determines sun angles and times for sunrise, solar noon, and sunset all year long. The flowchart of the algorithm is given in **Figure 9**. This

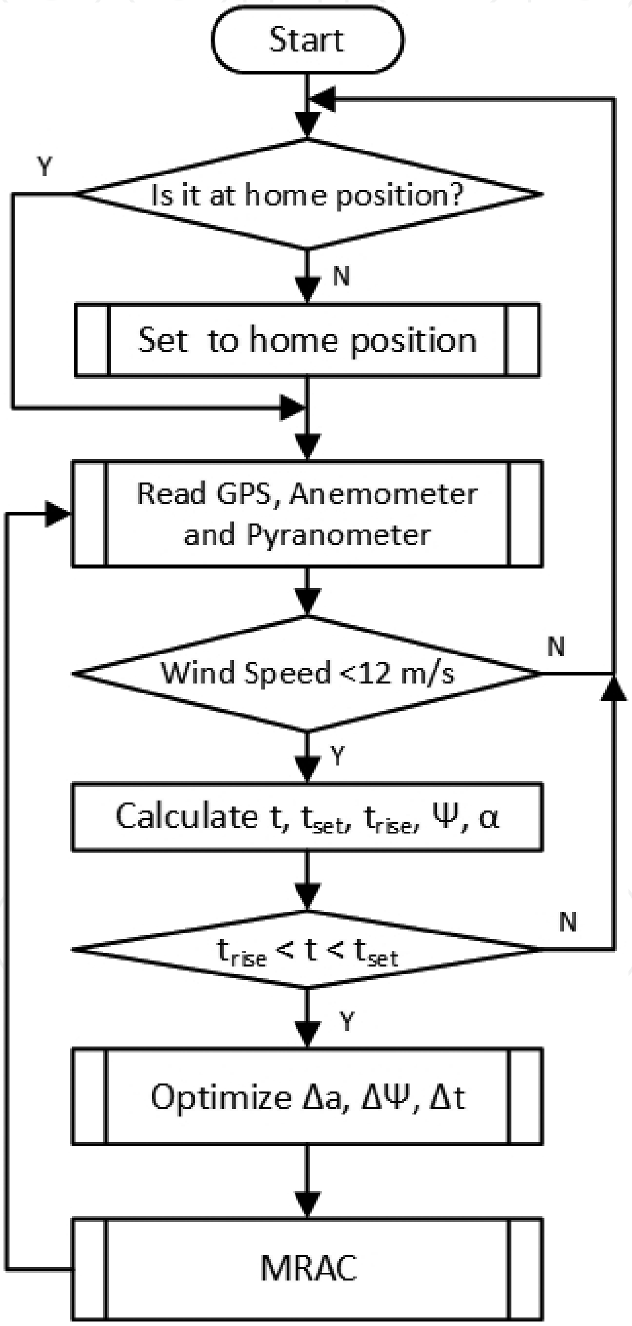

Figure 9. Adaptive controller sun tracker algorithm.

algorithm software calculates the sun angles just by using the date, time and precise longitude, latitude, and elevation of the location through a GPS system.

The microcontroller first sets the tracker to home position and then fetches GPS data to calculate the sunrise and sunset times. For more information, see Ref. [21]. The present solar time is compared with the calculated sunrise and sunset times to make a decision whether the sun tracking procedure will be performed. If the current solar time is between sunrise and sunset times, the microcontroller reads wind speed value from the anemometer to define whether the sun tracker can move safely. At night-time, it waits for the next interval. The sun tracker works in tracking mode if the wind speed is under 12 m/s, else it stands in table position to protect the tracking system. During the tracking phase, the microcontroller reads the pyranometer value to check if there is enough solar radiation to generate power. Otherwise, the sun tracker stays at home position until solar radiation rises at least to its lower limit. When the solar radiation reaches the desired value, the algorithm calls optimization subroutines that calculate the optimum $\Delta\alpha$ and $\Delta\Psi$ angles. Finally, the microcontroller calls the model reference adaptive control subroutine.

MRAC subroutine reads the azimuth, $\Delta\Psi$, and tilt, $\Delta\alpha$, angles, and converts them to a PWM signal to drive the tilt and azimuth motors. The calculated angles α(t) and (t) are then subtracted from the former position values. In accordance with the angle difference and sign, the microcontroller sends PWM and direction signals to the motor drives. Azimuth and tilt motors move the solar panel to a new position to precisely track the sun's trajectory. The two motors make different movements to approach the desired position of the solar panel by comparing the measured angles with the calculated ones. When the sun sets, the system goes back to home position, ready for the next sunrise.

4. Results and discussions

Embedded controller design is inherently a multidisciplinary issue. It involves dynamic systems analysis, modeling, estimation, simulation, control, signal processing, microcontroller architecture programming, power electronics, and motor drives. Embedded adaptive controller design began with the creation of the mathematical model of the sun tracker. The algorithm of the model reference adaptive controller was created in the next stage. In the last stage the program written in the light of this information was embedded in the microcontroller. Embedded controller for sun tracker implied by using Arm cortex M4 core microcontroller TM4C129 produced by TI [24]. Embedded code for adaptive algorithm was produced using Simulink model. The other part of embedded code was written in embedded C using Keil IDE [25], then the two codes were combined. After compiled debug and code optimization was made, it was embedded to the microcontroller.

The designed sun tracker was tested first in laboratory, then it was tested outdoor. System parameters, such as PV panel output current, voltage, solar irradiance, and temperature, were measured and logged during May 12 to November 23, 2016. Analysis of these data revealed that the sun tracker generated 39.2% more energy than the fixed slope system. When compared, the generated energy efficiency is slightly less than [20, 21, 23]. The total generated energy is high than [23] also consumed energy by DC motors higher. The PID controller, which was designed in Ref. [23], was optimized according to minimum energy consumption. It follows the sun's trajectory when it quarantines generated energy at least bigger than the consumed energy. As a result, this criterion increases the yield of generated energy when sun tracking accuracy is lower. Inversely, designed model reference adaptive controller tracks the sun very precisely; this increases both the energy generated and consumed. Meanwhile, MRAC continues tracking the sun in changing environmental conditions such as higher wind speed and higher ambient temperature. It inherently adapts to changing conditions easily, but PID controller usually loses track of the sun under same conditions. Sample day performances of tracker and fixed ones are given in **Figure 10** for sunny and cloudy days. For ideal tracker, the power curve should be like rectangular, but early morning and close to sunset, the sun's trajectory changes rapidly, so the consumed energy increases. The same phenomenon takes place on partly cloudy days; adaptive controller tracks quickly changing sun condition and both consumed and generated energies increase.

The proposed framework for dual-axis sun tracker could be realized in hardware easily and follow-up the hardware the control algorithm can be designed. It will be quite an efficient and cost-effective solution and in the long term it will provide an efficient solution for the users that need high energy from limited area with accurate alignment with sun, such as space stations, sea vehicles, and automatically aligned large telescopes.

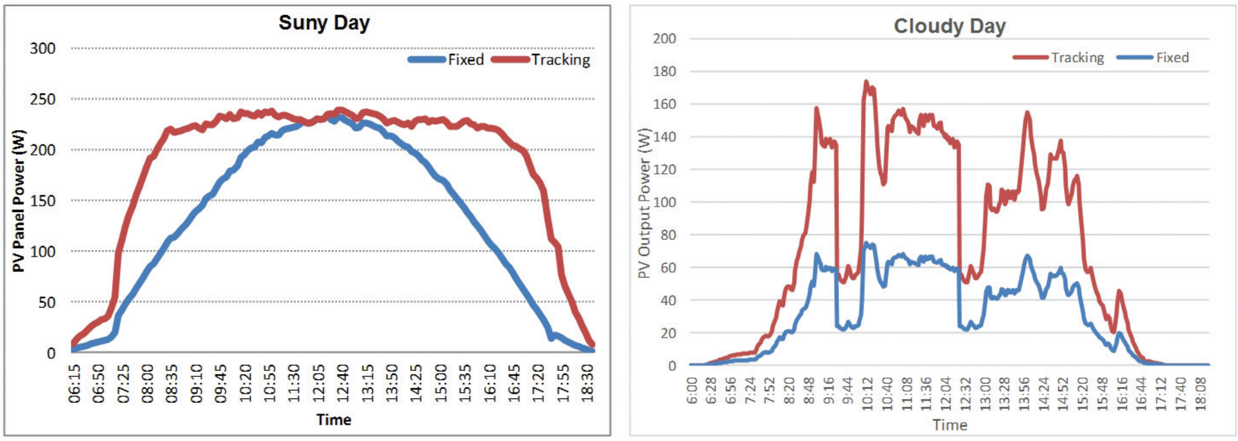

Figure 10. Outdoor test results of sun tracker for sunny and cloudy day.

5. Conclusion

Designed dual-axis sun trackers with embedded adaptive controller and fixed PV system were tested for 6 months. The system performance parameters for both systems were measured and stored using a data logger. The collected data is examined and compared. On a clear sky day, performance data proved that the embedded adaptive controller sun tracker produced 39.2% more energy than the fixed one. The efficiency of the sun tracker decreases to 34% on partly cloudy days. Furthermore, embedded adaptive controller is compared with embedded PID controller in terms of energy production, power consumption, reaction time, and code size.

Author details

Mustafa Engin

Address all correspondence to: mustafa.engin8@gmail.com

Electronic Technology Department, Ege Higher Vocational School, Ege University, Bornova, İzmir, Turkey

References

[1] M. Stefanovic, M.G. Safonov. Safe adaptive control: data-driven stability analysis and robust synthesis (vol. 405). Springer; 2011.

[2] J.S. Shamma, M. Athans. Analysis of gain scheduled control for nonlinear plants. IEEE Transactions on Automatic Control, vol. 35, pp. 898–907, 1990.

[3] N. Aouf, D.G. Bates, I. Postlethwaite, B. Boulet. Scheduling schemes for an integrated flight and propulsion control system. Control Engineering Practice, vol. 10, pp. 685–696, 2002.

[4] P. Apkarian, P. Gahinet, G. Becker. Self-scheduled H8 control of linear parameter-varying systems: a design example. Automatica, vol. 31, pp. 1251–1261, 1995.

[5] G. Becker, A. Packard. Robust performance of linear parametrically varying systems using parametrically-dependent linear feedback. Systems & Control Letters, vol. 23, pp. 205–215, 1994.

[6] M. Pakmehr, N. Fitzgerald, E.M. Feron, J.S. Shamma, A. Behbahani. Gain scheduling control of gas turbine engines: stability by computing a single quadratic Lyapunov function. In ASME Turbo Expo 2013: Turbine Technical Conference and Exposition, San Antonio, Texas, USA, 2013, p. V004T06A027.

[7] R. Márquez, T.M. Guerra, M. Bernal, A. Kruszewski. Asymptotically necessary and sufficient conditions for Takagi–Sugeno models using generalized non-quadratic parameter-dependent controller design. Fuzzy Sets and Systems, vol. 306, pp. 48–62, 2017.

[8] T. Bakka, H.-R. Karimi, S. Christiansen. Linear parameter-varying modelling and control of an offshore wind turbine with constrained information. IET Control Theory & Applications, vol. 8, pp. 22–29, 2014.

[9] C.S. Liu, Q. Ye, S.J. Zhang. Adaptive control for a class of uncertain linear parameter-varying flight aircraft systems. Int. J. Adapt. Control Signal Process. 2017, vol. 31, pp. 210–222.

[10] L. Fernandes, F. Moraes Filho, J. Paulo, J. Oliveira. Gain scheduling adaptive control applied to a particular mixer-settler equipment. Control Engineering Practice, vol. 21, pp. 1121–1127, 2013.

[11] C. Hoffmann, H. Werner. A survey of linear parameter-varying control applications validated by experiments or high-fidelity simulations. IEEE Transactions on Control Systems Technology, vol. 23, pp. 416–433, 2015.

[12] J.L. Crassidis, J.L. Junkins. Optimal estimation of dynamic systems. CRC Press, Boca Raton, FL, USA, 2011.

[13] N. Hovakimyan, C. Cao. L1 adaptive control theory: guaranteed robustness with fast adaptation. Philadelphia, PA: Society for Industrial and Applied Mathematics; 2010.

[14] K.S. Narendra, A.M. Annaswamy. Stable adaptive systems. Courier Corporation; Mineola, New York, USA, 2012.

[15] K.J. Åström, B. Wittenmark. On self tuning regulators. Automatica, vol. 9, pp. 185–199, 1973.

[16] R. Chiang, W. Breckenridge, E. Wong. Self-tuning thruster control for Cassini spacecraft. In Guidance, Navigation, and Control Conference, San Diego, CA, USA, 1996, p. 3822.

[17] L. Shieh, J. Sunkel, Z. Yuan, X. Zhao. Self-tuning control of attitude and momentum management for the space station. Journal of Guidance, Control, and Dynamics, vol. 15, pp. 17–27, 1992.

[18] B. Shackcloth, R. Butchart. Synthesis of model reference adaptive systems by Liapunov's second method. In Theory of self-adaptive control systems. Springer, New York, USA, 1966, pp. 145–152.

[19] A.J. Calise, M. Sharma, J.E. Corban. Adaptive autopilot design for guided munitions. Journal of Guidance, Control, and Dynamics, vol. 23, pp. 837–843, 2000.

[20] M. Engin, D. Engin. Optimization controller for mechatronic sun tracking system to improve performance. Advances in Mechanical Engineering, vol. 5, p. 146352, 2013.

[21] M. Engin, D. Engin. Optimization mechatronic sun tracking system controller's for improving performance. In IEEE International Conference on Mechatronics and Automation (ICMA), Takamatsu, Kagawa, Japan, 2013, pp. 1108–1112.

[22] T. Shivalingaswamy, B. Kagali. Determination of the declination of the sun on a given day. European Journal of Physics Education, vol. 3, 2012, pp. 17–22.

[23] M. Engin, D. Engin. Design of real time embedded PID controller for sun tracking robot manipulator. In IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), 2014, pp. 670–675.

[24] TI. (2015, January 12). Tiva™ C Series LaunchPad Evaluation Kit. Retrieved from http://www.ti.com/tool/ek-tm4c123gxl, 22 November 2016.

[25] KEIL. (2014, March 25). Embedded development tools (UVision 5.01 ed.).