International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 24 Issue 8 – MAY 2017.

SENSORLESS TEMPERATURE ESTIMATION AND FAULT REMOVAL DUE TO SWITCH FAULT

M.MUTHUKUMAR[#1] and Prof.N.KARTHICK[*2]

[#]*Final year, M.E, Power Electronics and Drives, P.S.V College of Engineering and Technology, Krishnagiri (D.T), India*

[*]*Associate Professor, Electrical and Electronics Engineering, P.S.V College of Engineering and Technology, Krishnagiri (D.T), India*

***Abstract*— This project presents a technique to estimate the temperature of each power electronic device in a thermally coupled, multiple device system subject to dynamic cooling. Using a demonstrator system, the thermal transfer impedance between pairs of devices is determined in the frequency domain for a quantized range of active cooling levels using a technique based on pseudorandom binary sequences. The technique is illustrated by application to the case temperatures of power devices. For each cooling level and pair of devices, a sixth order digital IIR filter is produced which can be used to directly estimate temperature from device input power. When the cooling level changes, the filters in use are substituted and the internal states of the old filters are converted for use in the new filter. Two methods for filter state conversion are developed—a computationally efficient method which is suited to infrequent changes in power dissipation and cooling, and a more accurate method which requires increased memory and processing capacity. Results show that the temperature can be estimated with low error using a system \which is suitable for integration on an embedded processor. In this work, we compare the performance of the system with both h-infinity controller and fuzzy based control and it is identified that, fuzzy control systems performs better than other controllers. Hardware implementation has been done with Arduino uno board. A load current of about 3A has been allowed to pass through the load. This makes the rise in temperature of the switch and the same is estimated and controlled through fuzzy rule based stored in Arduino.**

***Index Terms*— pseudorandom binary sequences, IIR filter, filter state conversion, h-infinity controller**

I. INTRODUCTION

Inverters, like all semiconductor-based equipment, are sensitive to overheating and, in general, operate best at cooler temperatures, while suffering power losses and damage at higher internal temperatures. It's well understood that heat affects all PV modules – they are tested and rated at 25 degrees Celsius and every degree above that causes power output to drop by up to .5% per degree. The temperature of the module is directly affecting voltage and the two critical things to consider are the highest voltage at the lowest local temperature and the lowest voltage at the highest possible local temperature. At the lowest temperature, string voltage cannot exceed the maximum input voltage of the inverter (usually 600Vdc) and at the highest temperature, string voltage needs to be above the minimum startup voltage of the inverter's MPPT algorithm (usually around 200Vdc).What is not as well understood is that heat also affects solar inverters. The reasons are not the same – although the solar inverter has semiconductor parts in it which loose efficiency as they heat up, the semiconductors themselves are pretty sturdy and can tolerate high heat without breaking down (to a point). As the inverter works to convert DC power to AC power, it generates heat. This heat is added to the ambient temperature of the inverter enclosure, and the inverter dissipates the heat through fans and / or heat sinks. The heat needs to stay below a certain level at which the materials in the inverter will start to degrade. Insulation will become brittle, solder can expand and crack and metal components in capacitors can fatigue. In order to keep the heat low, the inverter will stop generating power or reduce the amount of power it generates by "derating" as it passes programmed temperature milestones. . At about 45 degrees C. it starts to ramp down power. This ramp-down of power can be prevented with a couple of key system design considerations Install inverters in cool locations, e.g. in the cellar rather than on the roof. Choose locations with sufficient air exchange. Ensure additional ventilation, when necessary. Do not expose inverters to direct sunlight. For outdoor installations, use existing shadow or roof over the inverters. Maintain the minimum clearance to neighboring inverters or other objects given in the installation guide. Increase the clearance when it is foreseeable that higher temperatures could occur at the installation location. Arrange multiple inverters so that they do not draw in the warm air of other inverters. Offset passively cooled inverters to allow the heat from the heat sinks to escape upward. Even putting a shade over an inverter may not be enough. In figure 3, an attempt is being made to shade an SMA inverter – but it's been installed in a south-facing corner of a masonry structure. The ambient temperature is going to be very high in this corner and the inverter will be trying to cool itself with hot air – not a good situation. The shade won't have much of an effect. Figure 2 shows an inverter running normally with some components at close to 55 degrees C. It wouldn't take much to raise internal temperatures to more than 70 C. Most inverters

will derate at around 45 – 50 Degrees C. In the inhabited places of Planet Earth, temperature will rarely climb above 45 degrees C (113 Degrees F).

So, simply putting the inverter in a shaded area with good airflow will almost always result in an inverter that doesn't derate. In modern industrial production and people's daily life, technology of heating treatment has been widely used, with the continuous development of science and technology, the accuracy and reliability requirements on the heating treatment are also rising. As an important kind of tools on detecting and controlling of the temperature, the electric furnace is becoming more and more important. The current system for multi-temperature gathering and controlling is based on microcontroller core, although it can meet the basic need of data acquisition and controlling, it has less terminal functions, poorer sampling accuracy, lower speed but more complicated design, its compatibility and readability will be poor, and it will be difficult to upgrade or store for along time which is very useful for future. This paper designs a kind of high-precision temperature gathering and controlling system, which bases on embedded microcontroller 89s52, also makes a detailed description on hardware designs, the driver and the software of this system, including the design of fuzzy controlling and so on. The system based on embedded microprocessor and embedded system can gather the data and control the system based on the feedback of the data.

The data from external sensor is processed by a series of treatment, such as filtering and amplification and so on□it is converted from analog quantity to digital quantity by the 8-bit converter in ADC 0809, the maximum switching frequency of which is 2.5MHZ. Then the data displays in the LCD monitor, it will display in a variety of ways, such as the curve or digital display and so on. After that the data will be processed through the fuzzy algorithm, comparing the result with the setting value to control the system by the controller; this system uses two kinds of control operations on temperature-automatic control and manual control methods. The operating system used is powerful, which can gather and visual display the data in real-time, also it can store the data for the using of later queries; the operation is very easy and convenient as it has a very good man-machine interface and interaction.

II. Strategic Modeling

A. Closed Loop Control System

Most control systems utilize feedback in some manner. Here's a look at several fundamental feedback mechanisms, culminating in a description of a basic PID controller. Many real-time embedded systems make control decisions. These decisions are usually made by software and based on feedback from the hardware under its control (termed the "plant"). Such feedback commonly takes the form of an analog sensor that can be read via an A/D converter. A sample from the sensor may represent position, voltage, temperature, or any other appropriate parameter. Each sample provides the software with additional information upon which to base its control decisions.

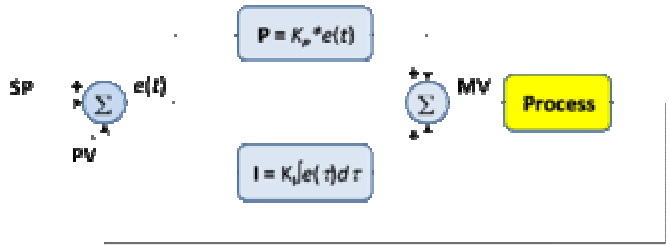


Fig:1.1 Basic block of a PI controller.

A **PI Controller** (proportional-integral controller) is a special case of the PID controller in which the derivative (D) of the error is not used.

The controller output is given by

$$K_P\Delta + K_I \int \Delta \, dt$$

whereΔ is the error or deviation of actual measured value (***PV***) from the set-point (***SP***).

$\Delta = SP - PV$.

A PI controller can be modeled easily in software such as Simulink using a "flow chart" box involving Laplace operators:

$$C = \frac{G(1 + \tau s)}{\tau s}$$ where,

$G = K_P$ = proportional gain

$G / \tau = K_I$ = integral gain

Setting a value for G is often a tradeoff between decreasing overshoot and increasing settling time. The lack of derivative action may make the system more steady in the steady state in the case of noisy data. This is because derivative action is more sensitive to higher-frequency terms in the inputs. Without derivative action, a PI-controlled system is less responsive to real (non-noise) and relatively fast alterations in state and so the system will be slower to reach setpoint and slower to respond to perturbations than a well-tuned PID system may be.

B. H-infinity methods in control theory

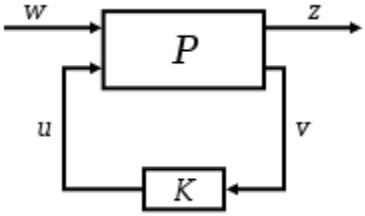
$\boldsymbol{H_\infty}$ (i.e. "***H*-infinity**") **methods** are used in control theory to synthesize controllers achieving robust performance or stabilization. To use H_∞ methods, a control designer expresses the control problem as a mathematical optimization problem and then finds the controller that solves this. H_∞ techniques have the advantage over classical control techniques in that they are readily applicable to problems involving multivariable systems with cross-coupling between channels; disadvantages of H_∞ techniques include the level of mathematical understanding needed to apply them successfully and the need for a reasonably good model of the system to be controlled. Problem formulation is important, since any controller synthesized will only be 'optimal' in the formulated sense: optimizing the wrong thing often makes things worse rather than better. Also, non-linear constraints such as saturation are generally not well-handled.

The term $\boldsymbol{H_\infty}$ comes from the name of the mathematical space over which the optimization takes place: H_∞ is the space of matrix-valued functions that are analytic and bounded in the open right-half of the complex plane defined by Re(s) > 0; the H_∞ norm is the maximum singular value of the function over that space. (This can be interpreted as a maximum gain in any direction and at any frequency; for SISO systems, this is

effectively the maximum magnitude of the frequency response.) H_∞ techniques can be used to minimize the closed loop impact of a perturbation: depending on the problem formulation, the impact will either be measured in terms of stabilization or performance. Simultaneously optimizing robust performance and robust stabilization is difficult. One method that comes close to achieving this is H_∞ loop-shaping, which allows the control designer to apply classical loop-shaping concepts to the multivariable frequency response to get good robust performance, and then optimizes the response near the system bandwidth to achieve good robust stabilization. Commercial software is available to support H_∞ controller synthesis.

C. Problem Formulation

First, the process has to be represented according to the following standard configuration:



Plant *P* has two inputs, the exogenous input *w*, that includes reference signal and disturbances, and the manipulated variables *u*. There are two outputs, the error signals *z* that we want to minimize, and the measured variables *v*, that we use to control the system. *v* is used in *K* to calculate the manipulated variable *u*. Notice that all these are generally vectors, whereas **P** and **K** are matrices.

In formulae, the system is:

$$\begin{bmatrix} z \\ v \end{bmatrix} = \mathbf{P}(s) \begin{bmatrix} w \\ u \end{bmatrix} = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix}$$

$$u = \mathbf{K}(s)\, v$$

It is therefore possible to express the dependency of *z* on *w* as:

$$z = F_\ell(\mathbf{P}, \mathbf{K})\, w$$

Called the *lower linear fractional transformation*, F_ℓ is defined (the subscript comes from *lower*):

$$F_\ell(\mathbf{P}, \mathbf{K}) - P_{11} + P_{12}\mathbf{K}(I - P_{22}\mathbf{K})^{-1} P_{21}$$

Therefore, the objective of $\mathcal{H}_\infty$ control design is to find a controller $\mathbf{K}$ such that $F_\ell(\mathbf{P}, \mathbf{K})$ is minimized according to the $\mathcal{H}_\infty$ norm. The same definition applies to $\mathcal{H}_2$ control design. The infinity norm of the transfer function matrix $F_\ell(\mathbf{P}, \mathbf{K})$ is defined as:

$$\|F_\ell(\mathbf{P}, \mathbf{K})\|_\infty = \sup_\omega \bar{\sigma}(F_\ell(\mathbf{P}, \mathbf{K})(j\omega))$$

where $\bar{\sigma}$ is the maximum singular value of the matrix $F_\ell(\mathbf{P}, \mathbf{K})(j\omega)$. The achievable H_∞ norm of the closed loop system is mainly given through the matrix D_{11} (when the system *P* is given in the form (A, B_1, B_2, C_1, C_2, D_{11}, D_{12}, D_{22}, D_{21})). There are several ways to come to an H_∞ controller:

- A Youla parameterization of the closed loop often leads to very high-order controller.
- Riccati-based approaches solve 2 Riccati equations to find the controller, but require several simplifying assumptions.
- An optimization-based reformulation of the Riccati equation uses Linear matrix inequalities and requires fewer assumptions.

D. H-Infinity Optimization

The standard task of H-Infinity optimization is formulated in the following way: given a plant model P, find controller model K satisfying the following conditions:

(Inf1) feedback interconnection of P and K is well-posed;
(Inf2) the closed loop state space model G is stable;
(Inf3) H-Infinity norm of G is smaller than $\gamma = (1 + \varrho)\gamma\min$, where $\gamma\min$ is the maximal lower bound for the H-Infinity norm of G achievable subject to constraints (Inf1),
(Inf2), and $\varrho > 0$ is a pre-specified tolerance;
(Inf4) the γ - entropy of G, defined by

$$E_{CT}(G) = -\frac{1}{2\pi}\int_{-\infty}^{\infty} \log\det(I - \gamma^{-2}G(j\omega)'G(j\omega))d\omega$$

in the CT case, and by

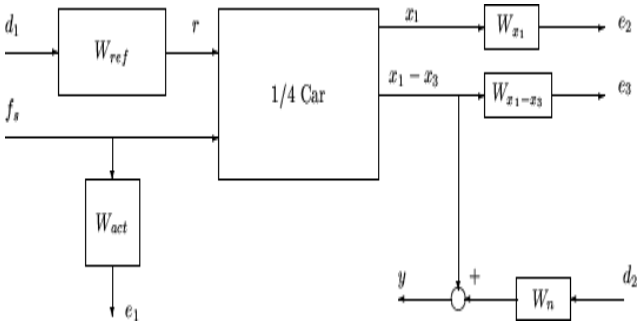
$$E_{DT}(G) = -\frac{1}{2\pi}\int_{-\pi}^{\pi} \log\det(I - \gamma^{-2}G(e^{j\omega})'G(e^{j\omega}))d\omega$$

in the DT case, is minimized subject to constraints (Inf1)-(Inf3).

Practically, the MATLAB's algorithm of H-Infinity optimization implemented in function hinfsyn.m applies only when conditions (CT1) - (CT7) are satisfied in the CT case, and conditions (DT1) - (DT7) are satisfied in the DT case. Just as h2syn.m (even to a higher degree), hinfsyn.m can return an incorrect answer for a poorly conditioned setting.

E. H-Infinity Controller Design in MATLAB:

The design of linear suspension controllers that emphasize either passenger comfort or suspension deflection. The controllers in this section are designed using linear H_∞ synthesis. As is standard in the H_∞ framework, the performance objectives are achieved via minimizing weighted transfer function norms. Weighting functions serve two purposes in the H_∞ framework: They allow the direct comparison of different performance objectives with the same norm, and they allow for frequency information to be incorporated into the analysis. For more details on H_∞ control design. A block diagram of the H_∞ control design interconnection for the active suspension problem is shown below.



The measured output or feedback signal *y* is the suspension deflection x_1–x_3. The controller acts on this signal to produce the control input, the hydraulic actuator force f_s. The block W_n serves to model sensor noise in the measurement channel. W_n

is set to a sensor noise value of 0.01 m.

```
Wn = 0.01;
```

In a more realistic design, W_n would be frequency dependent and would serve to model the noise associated with the displacement sensor. The weight W_{ref} is used to scale the magnitude of the road disturbances. Assume that the maximum road disturbance is 7 cm and hence choose Wref = 0.07.

```
Wref = 0.07;
```

The magnitude and frequency content of the control force f_s are limited by the weighting function W_{act}. Choose

$$W_{act} = \frac{100}{13}\frac{s+50}{s+500}.$$

The magnitude of the weight increases above 50 rad/s in order to limit the closed-loop bandwidth.

```
Wact = (100/13)*tf([1 50],[1 500]);
```

F. H-Infinity Control Design 1

The purpose of the weighting functions W_{x_1} and $W_{x_1-x_3}$ is to keep the car deflection and the suspension deflection small over the desired frequency ranges. In the first design, you are designing the controller for passenger comfort, and hence the car body deflection x_1 is penalized.

```
Wx1 = 8*tf(2*pi*5,[1 2*pi*5]);
```

The weight magnitude rolls off above 5×2π rad/s to respect a well-known H_∞ design rule of thumb that requires the performance weights to roll off before an open-loop zero (56.7 rad/s in this case). The suspension deflection weight $W_{x_1-x_3}$ is not included in this control problem formulation. You can construct the weighted H_∞ plant model for control design, denoted qcaric1, using the sysic command. The control signal corresponds to the last input to qcaric1, fs. The car body acceleration, which is noisy, is the measured signal and corresponds to the last output of qcaric1.

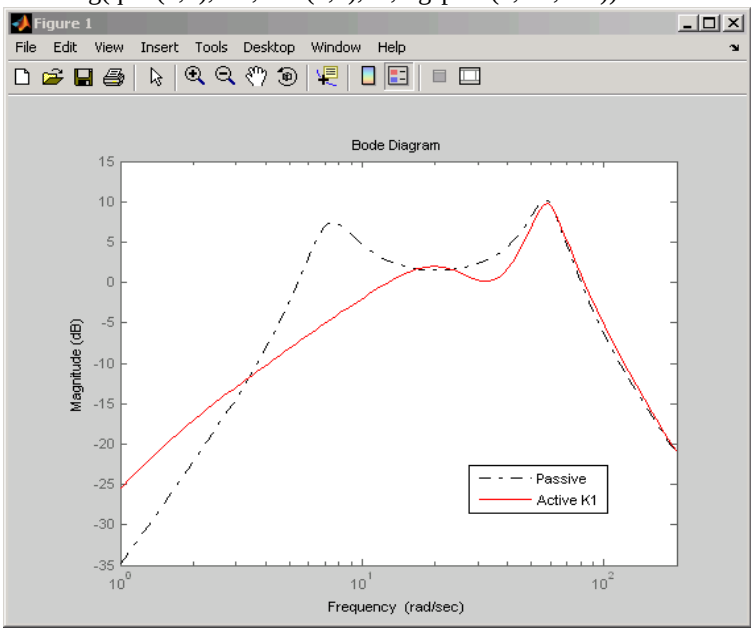
```
systemnames = 'qcarWnWrefWact Wx1';
inputvar = '[ d1; d2; fs ]';
outputvar = '[ Wact; Wx1; qcar(3)+Wn ]';
input_to_qcar = '[ Wref; fs]';
input_to_Wn = '[ d2 ]';
input_to_Wref = '[ d1 ]';
input_to_Wact = '[ fs ]';
input_to_Wx1 = '[ qcar(1) ]';
qcaric1 = sysic;
```

An H_∞ controller is synthesized with the hinfsyn command. There is one control input, the hydraulic actuator force, and one measurement signal, the car body acceleration.

```
ncont = 1;
nmeas = 1;
[K1,Scl1,gam1] = hinfsyn(qcaric1,nmeas,ncont);
CL1 = lft(qcar([1:4 3],1:2),K1);
sprintf('H-infinity controller K1 achieved a norm of %2.5g',gam1)
ans =
H-infinity controller K1 achieved a norm of 0.61043
```

You can analyze the H_∞ controller by constructing the closed-loop feedback system CL1. Bode magnitude plots of the passive suspension and active suspension are shown in the following figure.

```
bodemag(qcar(3,1),'k-.',CL1(3,1),'r-',logspace(0,2.3,140))
```



III. Operational Analysis

Sandia National Laboratories (Sandia) and the National Renewable Energy Laboratory have a long history evaluating the reliability of photovoltaic (PV) systems. Inverters are an integral part of a PV system and must function properly for the system output to be optimized. The lifecycle reliability of power electronic devices is highly dependent on operating temperature, which is related to loads and ambient conditions. Fans and heat sinks are employed to mitigate heating of components in an attempt to improve long -term reliability. There are many existing publications focusing on temperature assessment of PV modules and solar heat collectors, but fewer references discussing the temperature and reliability evaluation for the PV inverter and related components. Knowledge of the thermal history of individual components (capacitors, IGBTs, transformers, circuit boards, heat sinks, etc.) may be useful in assessing system reliability. This paper includes three primary parts: documentation of measured inverter temperatures, analysis of this data, and the application of these to reliability modeling.

A. Temperature Measurement

Inverters may operate at a wide range of temperatures. A comprehensive model for predicting the temperatures as a function of conditions for all types of inverters is beyond the scope of this paper, but we take an initial step in that direction by developing a model to predict the inverter heat sink temperature as a function of open -rack conditions based on observed heat -sink temperatures. This model must be extended to be able to predict component temperature data; for the reliability calculations we use temperatures measured directly for the

components of interest. The inverter heat-sink temperatures were measured for inverters connected to three grid -connected PV test systems in Golden, Colorado, US. The inverters were installed in the open under each latitude –tilted PV array, and the temperature sensors were fixed in the heat sink of each inverter.

To verify a model of inverter temperature rise and calculate wind speed factor and heat sink factor of the inverter, more than one year of inverter DC /AC power, irradiance, wind speed, and heat sink temperature rise data (5 min averaged per data point) were collected. For the collection of the inverter component temperature history, six inverters (three manufacturers with similar inverters going into both locations) located in Florida and New Mexico were instrumented with thermocouples to monitor the temperature of individual inverter components. Four -channel data loggers were used to record the temperature of three components and the internal ambient for each of the inverters. Data were collected at 30 second intervals, and then filtered to provide 10 minute measurements. The data were downloaded from the data loggers on a monthly basis for analysis.

B. Inverter Temperature Calculation

We expect that the temperatures of components in inverters may be modeled by understanding 1) a function defining the difference between the ambient temperature and heat-sink temperature and 2) the dissipation of heat in each component and the associated temperature drop between the component and the heat sink. The first of these we expect to be dependent on the total heat dissipation, the transfer (both convective and radiative) of heat from the heat sink depending on the area of the heat sink and other aspects of the heat sink design, and the wind speed, which provides forced cooling. We propose to model the difference between the ambient and heat sink temperatures ΔT , as

$$\Delta T = \frac{k}{(1+c \times Vw)} \times \frac{(Pdc - Pac)}{Pr} \quad (1)$$

Where Pdc, Pac, Pr, Vw, c and k represent the input DC power, the output AC power, the rating power of the inverter, the wind speed, the wind speed factor and heat sink factor of the inverter, respectively. The temperature difference between the inverter components and the heat sink can be approximated by:

$$\Delta T' = k' \times Pc \quad (2)$$

where ΔT’ is the temperature difference between each inverter component and heat sink, Pc is the consumed power of the inverter components and k’ is the heat transfer coefficient of each inverter component. Each inverter components’ temperature, Tc, can be calculated from eqs 1 and 2 by:

$$Tc = Ta + \Delta T + \Delta T' \quad (3)$$

where Ta is the ambient temperature. In general, each component may have a different level of heat dissipation and thermal connection, so equation 3 may be written for each component. Equation 1 was used to fit the heat sink data measured for one of the three inverters installed in Colorado. For this inverter, k was found to be 387 (°C) and c to be 0.29 (s/m). Similar analysis for other inverters found k to range from 350 to 650 (°C) and c from 0.20 to 0.30 (s/m). Instantaneous measurements of the inverter temperature are correlated with the model (R2 = 0.71). The fit is much better when the data are averaged (R2 = 0.98). In this case, the average temperature points were obtained by averaging 50 different temperature values for the inverter with the same inverter consumed power ratio and equivalent wind speed. For the purpose of evaluating reliability of inverters, knowledge of the average temperature may be adequate, alleviating the need to develop a transient model, though degradation processes that have high activation energies may be dominated by the short times spent at high temperatures.
Data from the six inverters in Florida and New Mexico were analyzed to provide insight into thermal management issues in general and for specific inverter/location combinations. Figure 3 presents data for inverter SB1. Several observations can be made from the data. There is a significant daily swing in temperature that can be seen for all of the components. Additionally, the temperature varies among individual components. In this case the torroid operated at a significantly higher temperature than did the fuse. A seasonal variation can also be seen. The temperature range (max to min) does not appear to be season-dependent, but the mean temperature correlates with the time of the year (hotter in the summer, colder in the winter). Most of the present systems are equipped and available as given below. Fuzzy based systems are available. Apart from this, PI, PID control systems are used. Sliding mode controls are used.

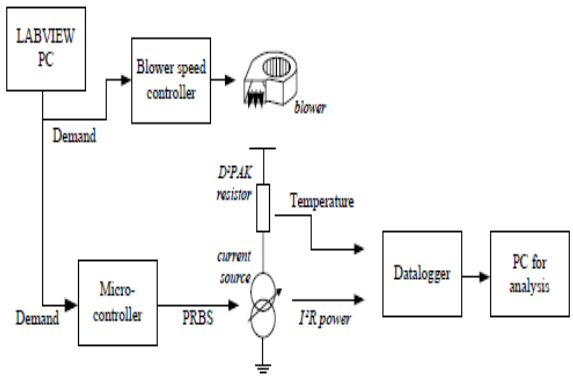


Fig.2.1 Block Diagram

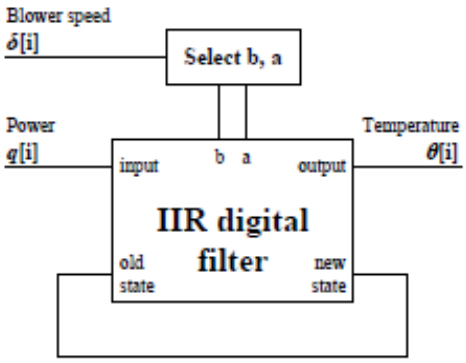


Fig.2.2 Temperature estimation block

C. Computational Requirements and Problems

COMPUTATIONAL REQUIREMENTS PER DEVICE PAIR FOR THE PROPOSED METHODS

Method	Steady -state	Scaled input	Unit
Programme memory required (for code)	5262	5724	byte
Flash data memory required (for filter coefficients)	5808	5808	byte
Read-write memory required (for static variables)	30	4230	byte
Floating point operations per sample			
(multiplications)	13	1855	
(divisions)	2	139	
(faster operations, e.g. addition)	16	1804	
Processing time per sample (at 8 MHz clock)			
(average)	1.0	87	ms
(maximum)	1.7	225	ms
Maximum sample rate (worst case)	590	4.4	S/s
Root-mean-square error compared to MATLAB code	3.4	64	mK

Conditions: 4.6 s sampling period; 132 quantised cooling levels for both methods

The system and cooling must be sized for the worst-case condition which only occurs when the system is under full load (or fault conditions) in high ambient temperature conditions for prolonged operating periods. This is an unusual situation, however, and, under practical conditions, the system operates significantly below the worst case. The design for worst case leads to increased cost due to excess heat sink requirements and increased energy usage in the active cooling system. Secondly, where active cooling is not controlled, temperatures rapidly increase and decrease with system load which can lead to significant thermal cycling.

D. Proposed System

Because the span of temperature variations is high at higher temperatures, it is preferable to use H infinity controller. An Arduino-Uno built based on ATMEGA 328 micro controller based system is proposed to calculate the temperature without any sensor and fuzzy logic based control of temperature has been proposed.

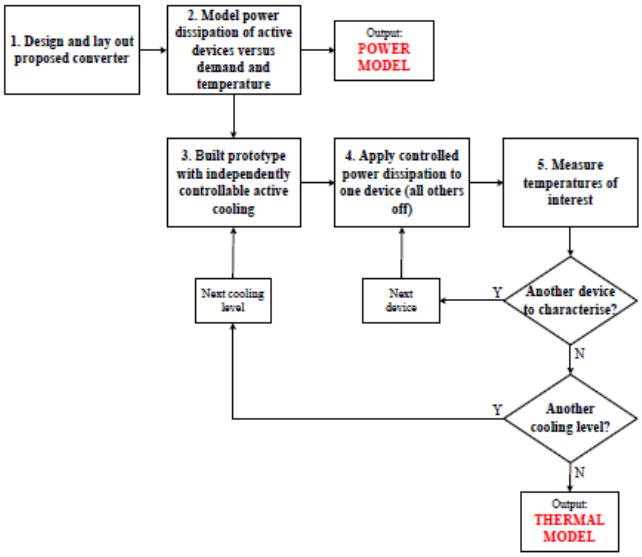


Fig.2.3 Proposed System

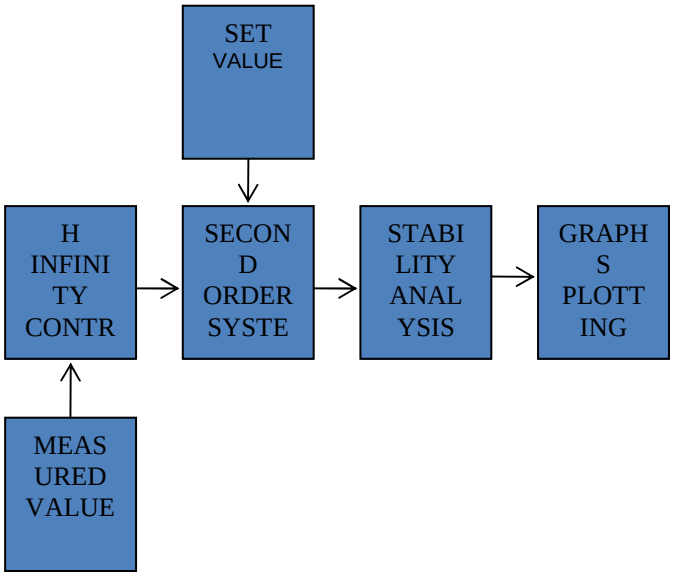


Fig.2.4 Simulation

E. Design of temperature gathering circuit

Data acquisition circuit uses thermocouple to measure the temperature and the graduating relationship of thermocouple is detected in the case of zero centigrade, but in reality, the reference temperature cannot be kept at zero centigrade, so that the system need the cold junction compensation. Bridge method is used to compensate the temperature. Bridge method of compensation is the use of unbalanced bridge which produces a corresponding potential to compensate for thermocouple reference junction temperature caused by changes in thermoelectric power. When the reference is T0, according to the formula above, we can calculate the corresponding potential.

$$E(T, 0) = E(T, T0) + E(T0, 0)$$

The electric potential is introduced to the internal analog digital converters of S3C2440A through the amplification and filter. Filter cutoff frequency can be calculated by the formula above

$f=1/(2\pi cR)$

where c is the data of capacitance and the R is the data of resistance AD converted value, within the system, after a look-up table can be obtained through the temperature after temperature compensation, which is handled by the software part.

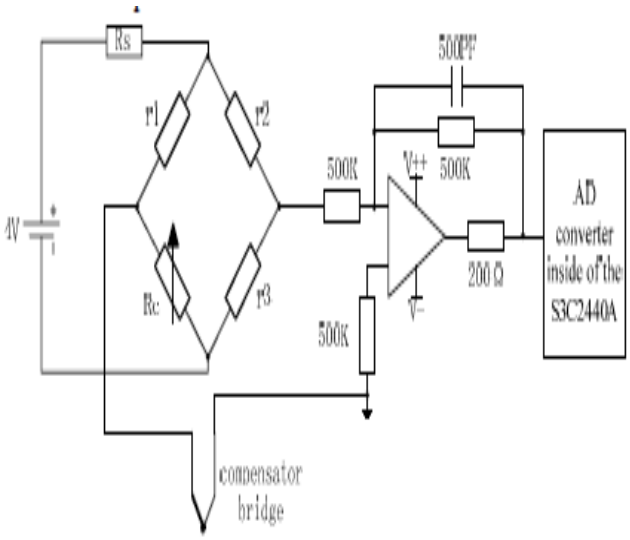
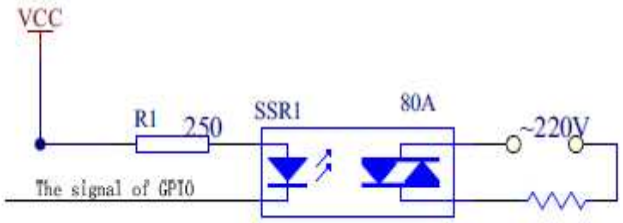


Fig.2.5 The Circuit of the Gathering Part

F. Design of the temperature controlling circuit

The temperature controlling circuit of the system is controlled by the Solid State Relays which control the heating cord to heat. The signal of the solid state relays is provided by the general purpose input and output. The signal, which is handled by the software of the system, can control the duty cycle to heat more accurately.



IV. Software Design

In this system, the software design is relate to the accuracy and stability of the data acquisition,, And there are more requirements for software design in high-speed data gathering and controlling system. Software design includes the design of internal AD converter drive, controlling of the temperature and how data stores.

A. Realization of the Internal AD Converter's Driver

The internal AD conversion of ARM is operated by the work of the AD converting register inside of the chip, the registers included the ADCCON (ADC Controlling Register), ADCDAT (ADC converting data register), ADCDLY (Delay Register of ADC Start), and so on. ADCCON controlling registers are mainly used to set AD converter-related parameters. The setting of control variables such as the pre scaler value, analog input channels and the enable of AD converter, are finished in the initialization; and the variables of delaying are set in the ADCDLY. The AD converted data is stored in the zero-nine bits of the AD converting data register, by the use of bottom-driven mapping, the data of sensor can be read. Applications access the driver in the form of accessing to the files through the file system, and the low-level driver communicates with the hardware through the Device Manager which is a connection between hardware and software.

B. The Realization of Temperature Controlled

When the system is measuring the temperature, it is also necessary for it to control the temperature, to meet certain requirements. Therefore, the realization of temperature controlled in this system is very important. The temperature is controlled by the electric heating wire which is also controlled by a solid-state relay, and the relay's controlling signals come from the PWM of the ARM, Similar to AD conversion, PWM signals are also controlled by the appropriate registers, where the application calls the flow driven interface functions. The temperature control modes are divided into automatic mode and manual mode. In automatic mode, according to the detected data from sensor and the process which has been preconfigured, the relays are controlled by the system to change the temperature. But in the manual mode, the system will close the process and operate the relay according to users' requirement completely. In automatic mode, when the temperature is below a certain set value, which is called the critical temperature, the system starts the corresponding relay for heating.

The electric oven uses different heating cord, the lower the temperature is, the heating cord with larger power will start, so that the temperature can reach target temperature quickly; as the heating cord still has heat after the heating has stopped, so stop heating before it reach the required temperature.. At that time, the temperature is called stop heating temperature. The heat left will make the temperature continues to rise. When the temperature dropped to a certain value, then the system starts the relay, according to the different temperature drop, start different powerful heating cord, and take different number. In the manual mode, the automatic program is closed, and adjustment is only done by the operation of the user. The manual mode can be used to short-term control as circumstances need to be closed

C. Design of Fuzzy control system

The environment that electric furnaces be used tends to be complex and with so much interference, which has a serious non-linear and time-variant. The conventional control method is very difficult to achieve precise control of temperature. The traditional fuzzy control method is to control object model for uncertain variables described by fuzzy conditional statements, and to determine the parameters of the model and control rules based on experience; adaptive control method is through adaptive learning, thinking and analogy to adjust the control parameters, reduce the error of a control method. It has a good fault-tolerance and robustness. The system combines the fuzzy control with the adaptive control method, having the advantages of two kinds of control methods in order to achieve precise control of temperature.

D. Adaptive Fuzzy Logic Controller

Traditional FLC requires the expert knowledge of the process operation for the FLC parameter setting, and the controller can be only as good as the expertise involved in the design. FLC with a fixed parameter is inadequate in applications when the operating conditions change in a wide range and the available expert knowledge is not reliable. To make the controller less dependent on the expert knowledge, AFLC could be introduced. However, the computation cost is much higher than conventional FLC. AFLC is composed of two parts: fuzzy knowledge base controller and a learning mechanism.

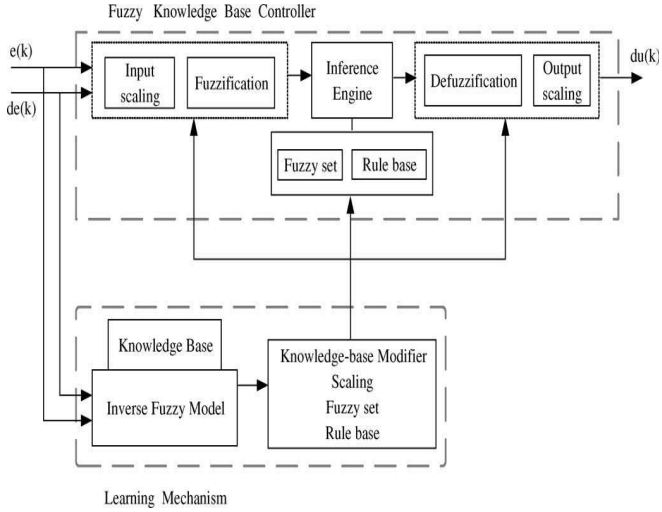


Fig: 3.1 Structure of adaptive fuzzy logic controller.

i. Fuzzy knowledge-base controller

The fuzzy knowledge-base controller is one part of FLC which is composed of three main parts: fuzzification, inference engine and defuzzification.

ii. Fuzzification

Membership function values are assigned to the linguistic variables, using seven fuzzy subsets: NB (negative big), NM (negative medium), NS (negative small), ZE (zero), PS (positive small), PM (positive medium), and PB (positive big). The partition of fuzzy subsets and the shape of membership function, which can adapt shape up to appropriate system, are shown in Fig. The value of input error (e) and change of error (de) are normalized by an input scaling factor. In this system the input scaling factor has been designed such that input values are between K1 and 1. The triangular shape of the membership function of this arrangement presumes that for any particular input there is only one dominant fuzzy subset. The input error (e) for the fuzzy logic controller can be calculated from the maximum power point as follows

$$E(k) = \frac{\Delta I}{\Delta V} + \frac{I}{V} = \frac{\Delta P}{\Delta V} = \frac{\Delta P}{\Delta I}$$

where I is the output current fromPVarray,DI is the change of output current, I(k)KI(kK1), V is output voltage from PV array, DV is change of output voltage, V(k)KV(kK1).

iii. Inference method

The composition operation is the method by which a control output is generated. Several composition methods such as Max–Min and Max-Dot have been proposed in the literature. The commonly used method, Max–Min, is used in this paper. The output membership function of each rule is given by the Min (minimum) operator and Max (maximum) operator. Table 1 shows rule base of the FLC.

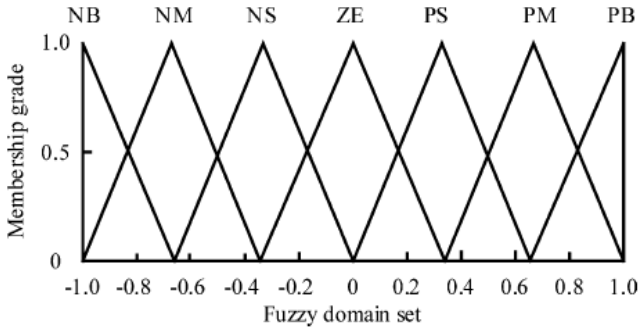


Fig:3.2 Fuzzy logic control membership function for input and output.

iv. Defuzzification

As a plant usually requires a non-fuzzy value of control, a defuzzification stage is needed. Defuzzificaion for this system is the height method which is both simple and fast, and is in a system of m rules given by

$$du = \left(\frac{\sum_{k=1}^{m} c(k) * w_k}{\sum_{k=1}^{n} w_k} \right)$$

Where du is the change of control output, c(k) is the peak value of each output and wk is height of rule k. The output of FLC is used to modify control output. Then, control output is compared with the saw tooth waveform to generate a pulse for controllable switch (SB) of the buck converter.

v. Learning mechanism

The purpose of the learning mechanism is to learn the environmental parameters and to modify the FLC accordingly so that the response of the overall system is close to the optimum operation point. The learning mechanism is composed of an inverse fuzzy model and a knowledge base modifier.

vi. Inverse fuzzy model

The error (e) or the change of error (de) of the system and the knowledge base modifier are used to modify the fuzzy parameter to optimize the system operation. The fuzzy parameter can be adapted by using the following condition: If error!3 (limit value) then knowledge base modifier will be chosen.

vii. Knowledge base modifier

In this part fuzzy parameter will be modified as follows.

Scaling factor: Simple schemes for altering the scaling factor to meet various performance criteria can be devised. When a scaling factor of a fuzzy variable is changed, the definition of each membership function will be changed by the same ratio. Hence, changing of any scaling factor can change the meaning of one part of any rule. The relation between the error, change of error, and output of FLC is similar to relation of conventional proportional and integral controller.

Fuzzy set membership function: Tuning peak values,

such as error in Fig, can improve both responsiveness and stability. A large error, NM and PM, can improve responsiveness. While a small error, NS and PS, can improve stability. Changing of width of membership affects the interpolation between two peak values. The modification can be performed by shifting the membership functions of both input and output.

Tuning rule base: Modifying rule base can affect the control system such as overshoot, setting time, stability, and responsiveness. When the fuzzy set membership function is modified, it may affect some rule bases. However, when a rule is changed, only this rule is involved. The modification is performed by adjusting the rule such that the rule firing trajectory always moves toward the stable point. The input for the invert fuzzy model is error and change of error. While the output of the knowledge base modifier are the change of peak of membership and scaling factor. In this paper, 3Z 0.1, d1Z0.1, d2Z0.2, and d3Z0.1 are used to modify the peak of membership and scaling factor for AFLC.

Predicted current control: From the predicted current control described, line current can defined as

$$\Delta I = I(t_n + T_s) - I(t_n) = [V_s(t_n) - V_{inv}(t_n)]T_s/L$$

where I is the inverter line current, Vs is utility voltage and Vinv the output voltage of inverter.

The output voltage of inverter (Vinv) can calculated from as follows:

$$V_{inv(t_n)} = V_s(t_n) - (L/T_s)[I(t_n + T_s) - I(t_n)]$$

The current of a single phase inverter can be controlled by controlling switches S1–S4. The switches S1 and S2 are used to shape the waveform to follow the reference current. While the switches S3 and S4 are used to correct the polarity of the waveform. Hence, the Vinv can be described as follows

$$V_{inv} = d_k V_{dc}$$

Where dk is the duty ratio for switch S1 and S2 over one switching period and Vdc is the DC bus voltage from boost converter. The change in line current over one period can defined as:

$$\Delta I = I(t_n + T_s) - I(t_n) = I(t_n) - I(t_n - t_s)$$

From, the duty ratio for single phase inverter can be defined as a function of source voltage (Vs) and the change in line current (DI) as follows:

$$d_k = f(V_s, \Delta I) = \frac{1}{V_{dc}}\left(V_s - \frac{L\Delta I}{T_s}\right)$$

We will use to control the duty ratio of switch S1 and S2 for the single phase inverter.

V. Simulation Results

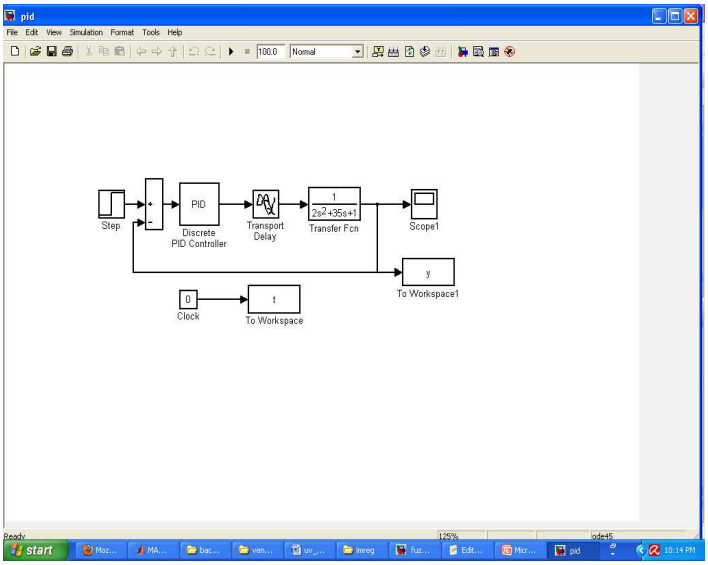


Fig.4.1 PID Controller based Temperature Controller

The diagram shown above is a PID controller to control the temperature of plant where the cooling system is assumed to be dynamic and cools instantaneously. However, this cannot be used in systems such as chemical systems and electrical systems where the temperature rise and fall needs enormous time to settle down with the target values. The transfer function of the heating element is given as 1 / ($2s^2$+35s+1). There is now arrangement for cooling element in the above diagram.

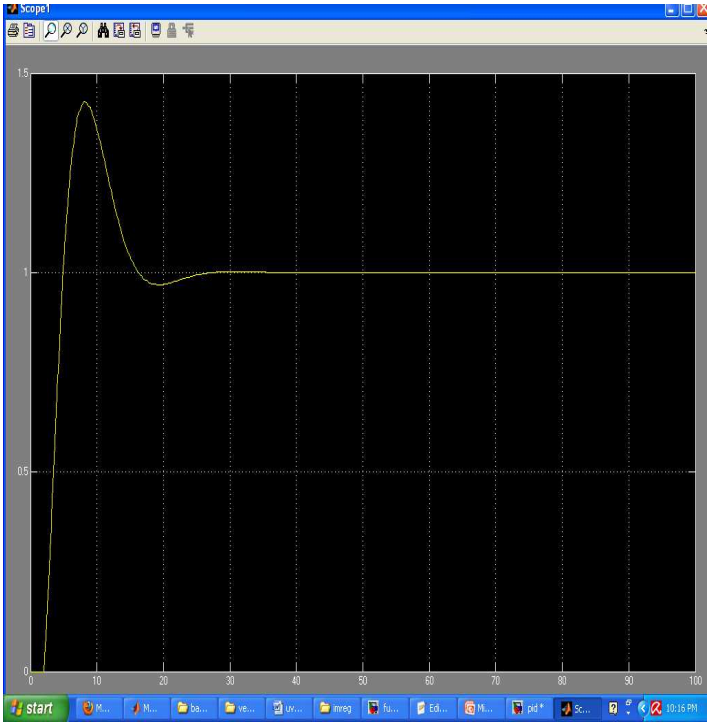


Fig.4.2 Temperature Variation

The figure shows the temperature variation and its settling time based on the plant model with transfer function 1 / ($2s^2$+35s+1). There exists an over shoot of almost 50% since the system is a second order control system. Though the settling time around 30 s, which is found to be less, the controller is not preferred due to its overshoot.

X axis – time in sec

Y axis – temperature in ‘C

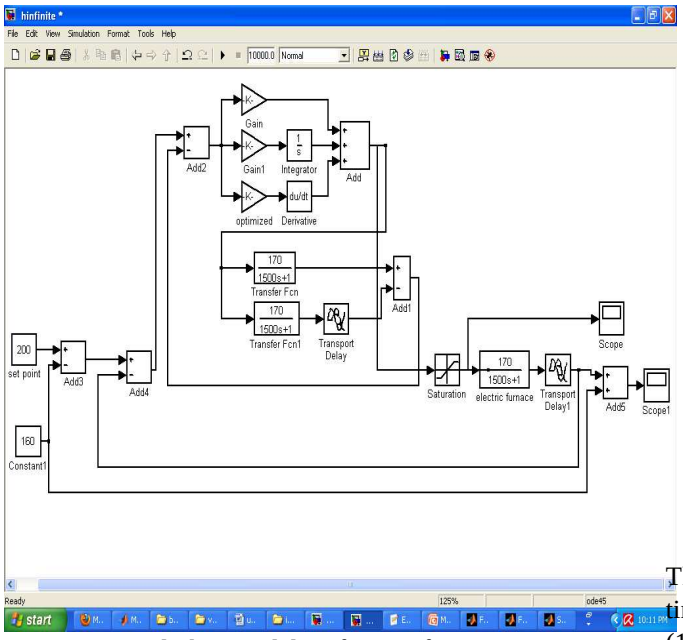


Fig.4.3 Simulink Model of H-Infinite Temprature Controller

The plant model has been done with a transfer function of 170/ (1500s+1). Similarly, a heating source and cooling source has been used to control the temperature both in positive and negative directions so as to attain immediate control. The set point in the figure is fixed as 200'C,. The system has a dynamic range of control over 150'C to 250'C, beyond which the control effect becomes unstable. The final temperature waveforms are seen in scope1. The models have been simulated using Matlab version 2013.

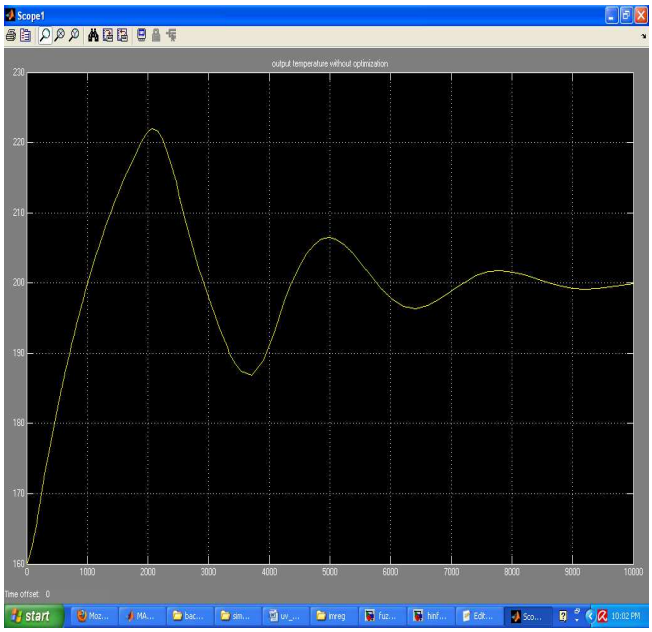


Fig 4.4Output Without H Optimization

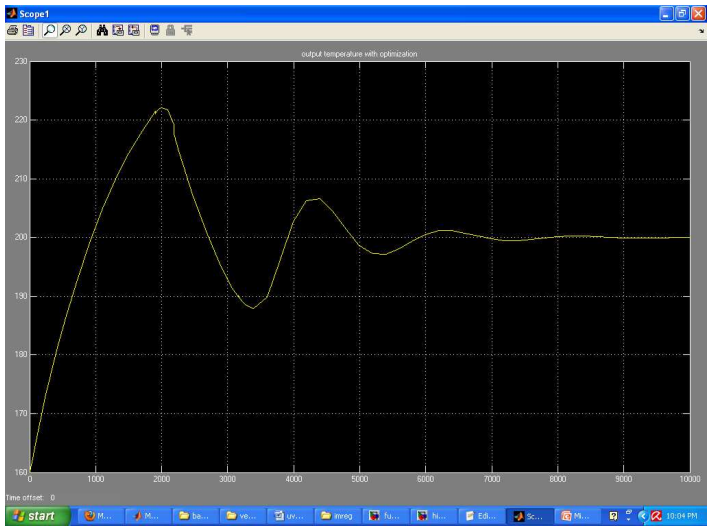


Fig.4.5 Output With H Optimization

The figure shows the temperature variation and its settling time based on the plant model with transfer function 170 / (1500s+1). There exists an over shoot of almost 10% since the system is a second order control system. The settling is more and it is around 1000s, which is found to be very high, the controller is not preferred due to its overshoot. Even with h-infinity optimization, the time is reduced only upto 700s. The only advantage is less overshoot.

X axis – time in sec

Y axis – temperature in 'C

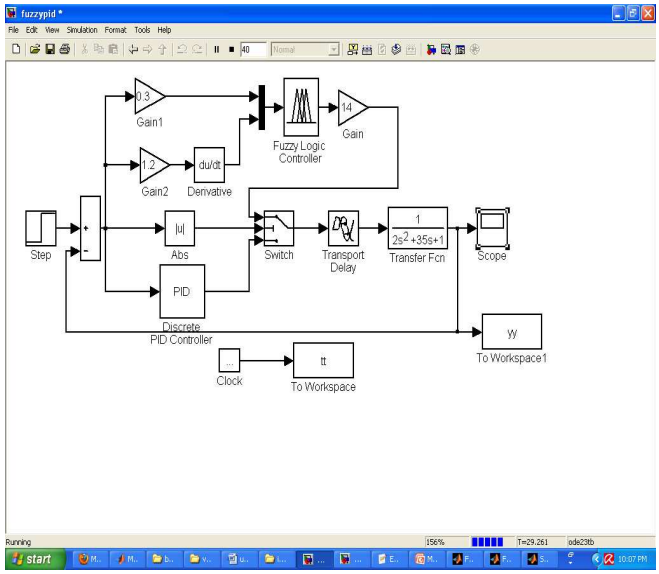


Fig.4.6: Simulink model of temperature control system

The diagram shown above is a fuzzy and PID controller to control the temperature of plant in a switched manner. The system takes the control of PID when the threshold is les, when it crosses a limit, fuzzy control is activated through a selection switch. Here the cooling system is assumed to be dynamic and cools instantaneously. However, this cannot be used in systems such as chemical systems and electrical systems where the temperature rise and fall needs enormous time to settle down with the target values. The transfer function of the heating element is given as 1 / ($2s^2$+35s+1). There is now arrangement for cooling element in the above diagram. Fuzzy rules are framed in a .fis file and invoked when the simulation is started.

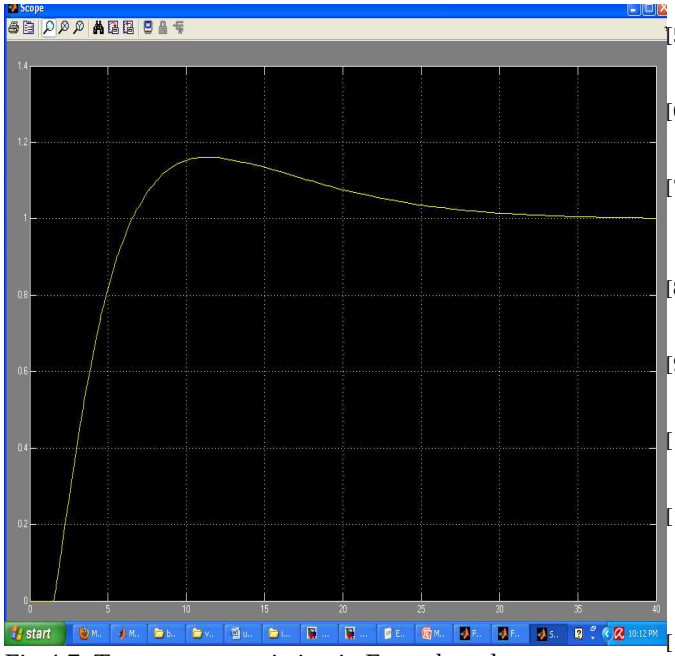


Fig.4.7: Temperature variation in Fuzzy based system

The figure shows the temperature variation and its settling time based on the plant model with transfer function 1 / ($2s^2$+35s+1). There exists an over shoot of almost 20% since the system is a second order fuzzy non-linear control system. The settling is very less and it is around 35s, which is found to be very low. Hence this controller highly preferred due to its least possible settling time.

X axis – time in sec

Y axis – temperature in 'C

VI. CONCLUSION & FUTURE WORK

This Project presents a temperature gathering and control system. The system is controlled based on the fuzzy adaptive algorithm, which can make the system more accurate and quickly. The embedded system combine with the embedded processor makes the system process even better. The proposed method is verified with various temperature ranges and corresponding cooling through the blowers has been implemented. The truthiness of the proposed method is verified with a temperature measurement using commonly used thermistor and the sensitivity is calculated.

Power factor control may be added to the same control circuit. Harmonic elimination may be added to this module. Hardware based control system is to be implemented along with the circuit protection circuit and fault tolerant capability.

REFERENCES

[1] M. Gerber, J. A. Ferreira, N. Seliger and I. W. Hofsajer, 'Integral 3-D thermal, electrical and mechanical design of an automotive DC/DC converter', Power Electronics, IEEE Transactions on, vol.20, no.3, pp.566-575, May 2005

[2] V. Smet, F. Forest, J.-J.Huselstein, F. Richardeau, Z. Khatir, S. Lefebvre and M. Berkani, 'Ageing and Failure Modes of IGBT Modules in High- Temperature Power Cycling', Industrial Electronics, IEEE Transactions on, vol.58, no.10, pp.4931-4941, Oct. 2011

[3] H. Lu, C. Bailey and C. Yin, 'Design for reliability of power electronics modules', Microelectronics Reliability, Volume 49, Issues 9–11, September–November 2009, pp. 1250-1255

[4] P. Ning, G. Lei; F. Wang and K. D. T. Ngo, "Selection of heatsink and fan for high-temperature power modules under weight constraint", in Proc. Applied Power Electronics Conference and Exposition, 23rdAnnual IEEE conference on, pp.192-198, February 2008

[5] L. Meysenc, M. Jylhakallio and P. Barbosa, 'Power electronics cooling effectiveness versus thermal inertia', Power Electronics, IEEE Transactions on, vol.20, no.3, pp.687,693, May 2005

[6] L. Fried, 'Prediction of Temperatures in Forced-Convection Cooled Electronic Equipment', Component Parts, IRE Transactions on, 5, (2), pp.102-107, Jun 1958

[7] C. Lundquist and V. P. Carey, 'Microprocessor-based adaptive thermal control for an air-cooled computer CPU module', Semiconductor Thermal Measurement and Management, Seventeenth Annual IEEE Symposium on, pp.168-173, 2001.

[8] M. Musallam and C. M. Johnson, 'Real-Time Compact Thermal Models for Health Management of Power Electronics', IEEE Transactions on Power Electronics, 25, (6), pp. 1416-1425, 2010.

[9] G. C. James, V. Pickert and M. Cade, 'A thermal model for a multichip device with changing cooling conditions', in Proc Power Electronics, Machines and Drives (PEMD), 2008, pp. 310-314.

[10] G. Xiong, M. Lu, C.-L. Chen, B. P. Wang and D. Kehl, 'Numerical optimization of a power electronics cooling assembly', Applied Power Electronics Conference and Exposition, vol.2, pp.1068-1073, 2001

[11] P. J. Rodgers, V. R. C. Eveloy, and M. R. D. Davies, 'An Experimental Assessment of Numerical Predictive Accuracy for Electronic Component Heat Transfer in Forced Convection—Part I: Experimental Methods and Numerical Modeling', Journal of Electronic Packaging, vol. 125, pp. 67- 75, 2003.

[12] P. J. Rodgers, V. R. C. Eveloy, and M. R. Davies 'An Experimental Assessment of Numerical Predictive Accuracy for Electronic Component Heat Transfer in Forced Convection—Part II: Results and Discussion', Journal of Electronic Packaging, vol. 125, pp. 76-83, 2003.

[13] J. N. Davidson, D. A. Stone, and M. P. Foster, 'Real-time prediction of power electronic device temperatures using PRBS-generated frequencydomain thermal cross-coupling characteristics', IEEE Transactions on Power Electronics, 30, (6), June 2015

[14] J. N. Davidson, D. A. Stone, M. P. Foster and D. T. Gladwin, 'Improved bandwidth and noise resilience in thermal impedance spectroscopy by mixing PRBS signals', IEEE Transactions on Power Electronics, 29, (9), September 2014, pp. 4817-4828

[15] B. Miao, R. Zane and D. Maksimovic, 'System identification of power converters with digital control through cross-correlation methods', Power Electronics, IEEE Transactions on, vol.20, no.5, pp.1093-1099, Sept. 2005

[16] MathWorks (8 July 2013). MATLAB 2013a documentation: invfreqz [Online]. Available: http://www.mathworks.co.uk/help/signal/ref/invfreqz.html

[17] A.V. Oppenheim and R.W. Schafer, 'Discrete-time signal processing', New Jersey: Prentice-Hall, 1999

[18] R. Hanus, M. Kinnaert and J.-L.Henrotte, 'Conditioning technique, a general anti-windup and bumpless transfer method', Automatica, 23, (6), November 1987, pp. 729-739

[19] J. D. Bendtsen, J. Stoustrup and K. Trangbaek, 'Bumpless transfer between observer-based gain scheduled controller', International journal of control, 78, (7), 2005

[20] E. W. Kamen, 'Introduction to signals and systems', New York: Macmillan, 1990, pp. 333-334

[21] European Community, 'Directive 90/C81/01', EEC Journal Officiel No. C81, 30 March 1990 p. 110

[22] G. Di Capua and N. Femia, 'A Versatile Method for MOSFET Commutation Analysis in Switching Power Converter Design', IEEE Transactions on Power Electronics, 29, (3), pp.920-935, February 2014

[23] M. Orabi and A. Shawky, 'Proposed Switching Losses Model for Integrated Point-of-Load Synchronous Buck Converters', IEEE Transactions on Power Electronics, 30, (9), pp. 5136-5150, September 2015

[24] J.N. Davidson; D.A. Stone and M.P. Foster, 'Arbitrary waveform power controller for thermal measurements in semiconductor devices', Electronics Letters 48, (7), 2012, pp. 400-402

[25] C. E. Bash, C. D. Patel and R. K. Sharma, 'Dynamic thermal management of air cooled data centers', Thermal and Thermomechanical Phenomena in Electronics Systems, 2006

[26] M. März and P. Nance, 'Thermal modeling of power electronic systems', Munich: Infineon Technologies AG, 2000.