Centre Nacional de
Microelectrònica

CSIC

Universitat Autònoma de Barcelona

Escola Tècnica Superior d'Enginyeria

Thermal Conductivity and Specific Heat Measurements for Power Electronics Packaging Materials

Effective Thermal Conductivity Steady State
and
Transient Thermal Parameter Identification Methods

PhD Thesis report by
Francesc Madrid Lozano

Supervisor
Dr. Xavier Jordà Sanuy

Cerdanyola del Vallès, June 2005

Acknowledgments

After all those years, I've quite a long list of people who contributed in some way to this thesis, for which I would like to express my grattitude.

First of all, special thanks to Prof. Francesc Serra, who let me use the C.N.M. facilities for my research. I would like to acknowledge the supervision of Dr. Xavier Jordà during the development of this work. I'm also grateful with Prof. José Millán, Dr. José Rebollo, Dr. Salvador Hidalgo, Dr. Miquel Vellvehí, Dr. David Flores and Dr. Philippe Godignon for their advise and encouragement throughout this investigation.

I would like to thank all my colleagues for their help, comradeship and endless patience, Amador Perez, Dr. Jaume Roig, Jesus Urresti, Xavier Perpinyà, Jose Luis Galvez, Dr. Riccardo Rurali, Dr. Servane Blanqué and Dr. Dominique Tournier.

I am grateful for the wholehearted support that Gisela, my parents, my brother and the rest of my family have given to me throughout. A warm embrace to the members of *Pirata Roberts*, *Nyèbit*, *R.N.R.* and friends.

This work was partially supported by the Comisión Interministerial de Ciencia y Tecnología (CICYT) project TIC1999-1222 and by the EC Growth Project ATHIS (G1RD-CT-2002-00729).

Contents

Chapter 1

Introduction

The work exposed in this document has been developed within the Power Devices and Systems Group in the *Centre Nacional de Microelectrònica*[1] of Barcelona (IMB-CSIC).

Its main motivation is the fact that in the field of Thermal Management of Power Electronics packages there is a lack of knowledge about the thermal properties of the packaging materials. This work proposes two measurement methods with the objective to obtain reliable thermal parameter values for such materials. First method derives the thermal conductivity value from a steady state thermal resistance measurement on a sample. It is also proposed an alternative method based on a parameter identification procedure, which allows to determine both the specific heat and thermal conductivity of a sample material from a transient temperature record.

Next, the Power Electronics and Thermal Management disciplines, where the present work has been developed, are briefly defined. Thus, the motivation and objectives are explained in detail.

1.1 Power Electronics and Thermal Management

The task of Power Electronics is to convert and control the flow of electric energy from a power source to a user load. A power converter, based on power semiconductor devices, converts electrical power with high efficiency minimizing energy looses. The power input (voltage, current, frequency and number of phases), usually from the electric utility, is converted in order to provide the load with an optimally suited voltage and current. Normally, a

[1] website: http://www.cnm.es/imb

feedback controller compares the output with a reference desired value and drives the converter to minimize the error. This controller circuit is made of digital signal processors or linear circuits. Silicon power devices operate as a switch, i.e. fully on or fully off, this minimizes the power loss.

The use of Power Electronics converters saves energy and contributes to the environmental pollution control [1]. It has become an essential tool in industrial process, transportation, aerospace, telecommunications, residential and commercial applications [2]. The field of Power Electronics has experienced a large growth in recent years. Converter and controller technologies, following the evolution of power semiconductor devices, microelectronics, drives and control methods are advancing rapidly. At the same time, the market for Power Electronics has significantly expanded. Within electrical engineering, Power Electronics encompasses power systems, solid-state electronics, electrical machines, analog/digital control and signal processing and electromagnetic field calculations. It is truly a multidisciplinary technology.

In any power conversion process a small amount of power is always lost and heat is produced as an unavoidable by-product. Power semiconductors dissipate power internally both during the on-state and during the transitions between the on and the off states. This loss is important to be minimized because of the cost of the wasted energy and the difficulty in removing the generated heat. In addition, the produced heat and the eventual excessive increment of temperature has a detrimental effect on the performance and reliability of an electronic circuit. Chemical reactions rate, the same way as leakage currents, double with every 10 °C temperature rise above 50 °C. This accelerates failure mechanisms (e.g. junction failure, metalization failure, corrosion and electromigration diffusion in power devices) and the circuit performance becomes poor. This effects may ultimately result in circuit failure.

The term Thermal Management encompasses both the study of the heat generation in electronics circuits and techniques to control the temperature rise [3]. One of the more important tasks of the packaging engineer is to apply the principles of Thermal Management in order to keep the temperature of an electronic device below the level that will cause a premature circuit fail or performance degradation. The prediction of the device temperature during performance is the basis of this task. The generated heat must be carried away from the device and dissipated safely into the ambient. This requires the design and construction of an efficient heat path from the device through the mounting surface to the outside world, i.e. with a minimum thermal resistance. Device manufacturers typically guarantee the optimum device parameters at a specified maximum temperature, which varies from one type of device to another and is often in the $125\,°C - 175\,°C$ range [2]. In

a design process, one of the design inputs is the worst-case junction temperature which is commonly 125℃ or less, depending on the application. This discipline requires an extensive knowledge of thermal properties of materials and thermodynamics.

All heat sources in an electronic circuit must be considered for a certain temperature prediction. The device temperature is a result of operational losses of the semiconductor itself, the heat generated by devices in the proximity, the ambient temperature and even solar radiation. Moreover, not only active components generate heat, resistors, conductors, and even small wire bonds can generate a significant amount of heat. The location of the heat generation is just as important as the generated amount, temperature of a given device will increase above self heating due to other dissipating components in the proximity. Each particular circuit must be analyzed on an individual basis as it will behave differently depending on the components, the layout, the materials and external conditions. Most of the parameters of every circuit are temperature dependent in several degrees. Therefore, in order to establish the maximum temperature for the circuit all the cumulative effects must be known.

The trend in packaging electronics has been to reduce size and increase performance, i.e. more functions and more power in a smaller package. Higher levels of integration in semiconductors have been attained and the usage of hybrids and multichip modules (MCM) has been increased. This contribute to increase heat generation and concentration resulting in high heat generation densities. All this mandate the Thermal Management to be given a high priority in the design process in order to maintain or even increase system performance and reliability.

To achieve the degree of accuracy in temperature prediction needed for optimal Thermal Management, thermal simulation with CAD tools has become an essential tool. These methods use finite element analysis (FEA) or finite volume analysis (FVA) and require a certain mathematical expertise from the user.

1.2 Motivation and Objective

It has been stated that in the field of Power Electronics packaging, the increasing required thermal performance of power modules makes indispensable the use of thermal simulation in order to achieve the strict design aims [3]. Therefore, if realistic predictions have to be obtained from a thermal simulation of a given design, the thermal properties of the used materials have to be well known.

Values for thermal parameters of materials are, in most cases, easily available from handbooks, physical property tables, technical papers or manufacturer data sheets. Nevertheless, the value of the thermal conductivity of most ceramic materials used in power packaging greatly depends on the raw material characteristics (purity, particle size distribution, fabrication process), the processing route and its parameters (temperature, pressure, atmosphere, ...) and the final component properties such as the density or grain size in the case of sintered ceramics. In addition, manufacturers provide values for their materials in a wide range. Therefore, values from literature may not fit the actual value of a particular sample, resulting in low reliable predictions of a given design performance. As an example it can be found, depending on the source, a thermal conductivity for Alumina (Al_2O_3) between 16 W/ m K and 36 W/ m K, for Aluminum Nitride (AlN) between 80 W/ m K and 260 W/ m K, for Beryllia (BeO) between 184 W/ m K and 300 W/ m K and similar ranges for all ceramics used in power packaging [4].

Therefore, it is almost impossible to obtain reliable parameter values for thermal simulation of a given material only from the literature although this knowledge is getting more critical every time. In conclusion, in order to obtain accurate thermal values of a particular sample material it is necessary to do *in-situ* measurements [4].

With this objective, two measurement methods have been designed and implemented. First, an effective thermal conductivity extraction method based on direct thermal resistance measurement for the parameter extraction. The implementation includes the design of an innovative heat power generation controller, and its advantages are to be versatile as well as simpler and cheaper than the existing sophisticated commercial systems. Most of those systems allow to measure only thermal diffusivity (e.g. Laser Flash Diffusivity), needing to know *a priori*, from other sources or measuring, the density and either specific heat (e.g. by Modulated Differential Scanning Calorimetry) in order to deduce thermal conductivity. Therefore, the deduced values are sometimes not more reliable than those found in literature. The proposed method is thought to perform direct steady state measurements of the effective thermal conductivity parameter of materials, such as ceramics for Power Electronics substrates fabrication. Measurements over a wide range of values, from less than 1 W/ m K to 170 W/ m K can be performed with good accuracy. This range embraces those of some different, much more expensive systems [5]. Its measuring principle is based on the Guarded Hot Plate method as it is also the Dual Cold Plate [6]. As the proposed method is aimed at getting results not only reliable but also accurate, reduction and characterization of the experimental error has been a major issue.

Secondly, an alternative simultaneous thermal conductivity and specific heat measurement method has been conceived and implemented using essentially the same thermal setup and equipment for the steady state measurements. Thermal diffusivity can also be deduced as a derived parameter. Neither specific heat nor thermal diffusivity can be measured with a steady state measurement, thus a transient method is needed. Efforts have been made to develop accurate transient thermal parameter measurement methods such as contact transient measurement methods (CTM) [7], and several procedures have been developed and standardized for different applications and needs [8]. Illustrative examples of CTM are the Transient Plane Source technique proposed by Gustafsson [9] and the Step-Wise Method of Kubičár [10]. Both methods yield a transient temperature record from a thermal system where the heat source is a plane and heat conducts across the material under test. Thus, comparing the measured response with an analytical expression, the thermal parameters are deduced. The CTM proposed in this work obtains the transient temperature measurement in a similar way but the thermal parameters are deduced in an innovative way. The transient measure is compared with a 3D thermal simulation, instead of an analytical method, and thermal properties of the materials are deduced by parameter identification algorithms. The procedure takes advantage of the growing calculation power of modern computers. The main advantage of this method is that the experimental constraints can be relaxed, it is no more necessary to adapt the experimental setup to have an analytical solution, e.g. a plane or linear heat source. The measuring capabilities in this sense are wider due to the fact that any thermal system that can be properly simulated can be used.

Organization

The effective thermal conductivity measurement method is described entirely in Chapter 2, while the transient measurement method is divided into two chapters. Chapter 3 proposes the principle of operation of the transient measurement and the data analysis procedure, based on a parameter identification software specifically developed for this application. Chapter 4 describes the proposed transient data acquisition procedure, which includes the design and implementation of a thermal test chip devoted to this transient tests, together with some parameter identification results for two example substrates.

Chapter 2

Thermal Conductivity Measurement

The work presented in this chapter describes the design and construction of a thermal conductivity measurement system devoted to get realistic thermal parameters for CAD thermal simulation and material characterization. The basic concepts used to design this measurement system as thermal conductivity, heat conduction, thermal resistance and all details concerning the operation and implementation of the thermal conductivity measurement system are exposed. Finally, essential considerations about the measurement capabilities and experimental issues are presented together with some experimental results. Measurement results for several ceramic materials using two measurement modes and a thermal interface material comparative study are also given.

2.1 Measuring Principle

The proposed method is based on the heat conduction phenomena through solids [3, 11, 12, 13]. Other heat transfer modes; i.e., convection and radiation, have been avoided in the system implementation as explained further on in section 2.2.

2.1.1 Concept

In a steady temperature scalar field inside a given material media, thermal resistance is defined as the difference in temperature between two closed isothermal surfaces divided by the total heat flow between them [14],

$$R_{th} = \frac{\Delta T}{P} \tag{2.1}$$

defining ΔT as the temperature difference between two surfaces (in °C) and P as the total heat flow (in W). It is supposed that all the heat that flows through one surface also flows through the other one and no heat generation occurs between surfaces. It should be noted that these surfaces are not physical, solid surfaces but rather imaginary surfaces of constant temperature.

By straightforward analogy with the Ohm's law of the electrical resistance, an analogue thermal schematic circuit can be used similarly the electrical one [3, 11]. The thermal resistance takes the role of the electrical resistance, the heat power that of the electrical current and the temperature rise between both sides of the thermal resistance that of the voltage drop. Every node in such thermal circuit represents in fact a whole isothermal surface.

The relationship between the heat flow and the temperature gradient in heat conduction is given by the Fourier law [12]. For an homogeneous and isotropic solid, the Fourier law is given in the following form:

$$\vec{q}(\vec{r}, t) = -k\vec{\nabla}T(\vec{r}, t) \tag{2.2}$$

where $\vec{q}$ represents heat flow per unit time and per area in the decreasing temperature direction, its units being $\mathrm{W/\,m^2}$. Because of the temperature gradient $\vec{\nabla}T$ factor, the heat flow density vector $\vec{q}$ is normal to the isothermal surfaces. The thermal conductivity of the media is noted henceforth with the symbol k. It is a scalar physical property of the material defined by the equation (2.2), its units being $\mathrm{W/\,m\,K}$.

An example with an easy analytical solution of the Fourier's law is the case of a sample material of area A, thickness L and thermal conductivity k being crossed by a steady, homogeneous heat flow density normal to its faces. Figure 2.1 represents all elements for the solution. Flat isothermal surfaces are generated parallel to the sample face, two of them will coincide with the up and down faces of the sample with temperatures T_{hot} and T_{cold} respectively. Temperature difference is $\Delta T = T_{hot} - T_{cold}$. The thermal resistance, defined in equation (2.1), of that simple geometry is obtained to be [3, 12]:

$$R_{th} = \frac{L}{kA} \tag{2.3}$$

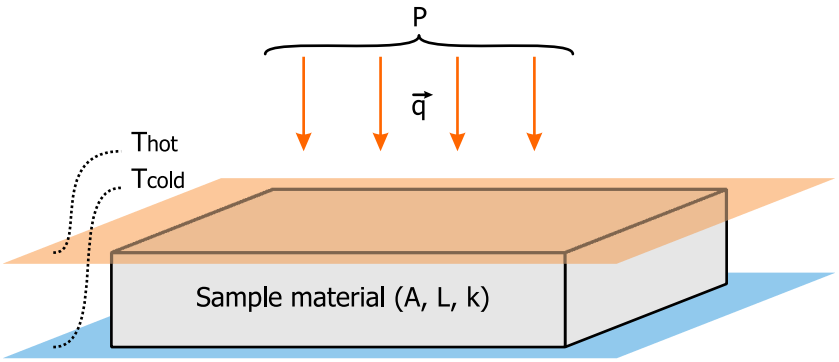


Figure 2.1: *Material of area A, thickness L and thermal conductivity k crossed by a steady, homogeneous heat flow density $\vec{q}$ normal to its surfaces (P being the total heat flow). It is also the ideal configuration for the thermal conductivity measurement.*

Therefore, if sample dimensions are known, the thermal conductivity of the sample can be directly calculated from a thermal resistance measurement, according to equation (2.3), simply re-arranging the expression:

$$k = \frac{L}{R_{th}A} \tag{2.4}$$

The measurement of the material thermal resistance must be carried out by injecting a continuous amount of heat flow through a sample of unknown thermal conductivity and measuring the temperature drop between both faces, thus its thermal resistance is determined through equation (2.1). The heat flow density must be uniform along the sample and perpendicular to its faces generating isothermal surfaces parallel to both sides. Thus, equation (2.4) provides the thermal conductivity of the material.

In the case of an anisotropic material; i.e., its thermal conductivity depends on the direction of conduction, k is a tensorial magnitude represented by a square 3×3 matrix, positive and symmetric:

$$k = (k_{i,i}) = \begin{pmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_z \end{pmatrix} \tag{2.5}$$

An example of such a material can be seen in section 2.5.1 on page 41. Its thermal resistance depends on the thermal conductivity component along the direction of heat flow k_i. For a heat conduction along the direction i:

$$R_{th\,i} = \frac{L}{k_i A} \tag{2.6}$$

Therefore, the thermal conductivity component measured with the explained procedure is k_i, the component on the direction of $\vec{q}$.

$$k_i = \frac{L}{R_{th\,i} A} \tag{2.7}$$

2.1.2 Setup

Figure 2.2 shows how the desired measuring conditions described above are met. A thick and very conductive heat spreader homogenize the heat flow going through the sample generated by a heating power MOSFET device. The total heat flow P is kept steady by a power controller described in section 2.3. This makes the isothermal surfaces to be flat and parallel in the bulk sample material, as needed in order to validate the expression (2.4). Thus, the conditions of homogeneity and direction of heat are satisfied.

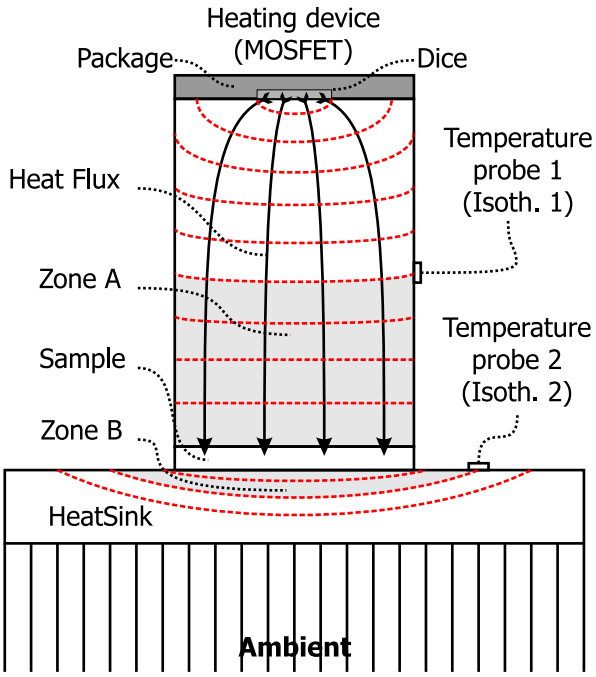


Figure 2.2: *Experimental setup for thermal resistance measurement.*

With this configuration, it is impossible to measure the temperature directly on the surfaces of the sample in order to get ΔT. Nevertheless, a measurement can be carried out placing two temperature probes as shown in figure 2.2. The measured thermal resistance of such system is:

$$R_{th\ system} = \frac{T_{isoth.1} - T_{isoth.2}}{P} \tag{2.8}$$

where $T_{isoth.1}$ and $T_{isoth.2}$ are the temperatures of isothermals 1 and 2, respectively.

The thermal system of figure 2.2 is represented in figure 2.3 using the Ohm's law analogy. This electrical analogy, keeping in mind that every node in the electric-like circuit represents an isothermal surface, allows to symbolize all serial thermal resistances of the setup. Thermal interface materials (often abbreviated as TIM) used to put in thermal contact the sample with the spreader and the heat-sink, also contribute to the measured resistance. Thus, thermal resistance between isothermal 1 and 2 include part of the heat spreader $R_{th\ ZoneA}$, the sample $R_{th\ sample}$, thermal interface materials on both sides of the sample, $R_{th\ TIM1}$ and $R_{th\ TIM2}$, and part of the heat sink $R_{th\ ZoneB}$:

$$\begin{aligned} R_{th\ system} &= R_{th\ ZoneA} + R_{th\ TIM1} + R_{th\ sample} + \\ &\quad + R_{th\ TIM2} + R_{th\ ZoneB} \end{aligned} \tag{2.9}$$

Two measurements of thermal resistance between temperature probes are performed in order to get a value by comparison. One measurement is done on a thick sample and another one on a thinner one of the same material. Due to the fact that heat distribution is homogeneous within the sample, it is assumed that a change on the thickness of the sample will only remove a given range of horizontal isothermals without having any effect on the shape of the rest of the temperature field, including isothermals 1 and 2. Being $R_{th\ system\ thick}$ and $R_{th\ system\ thin}$ the measured thermal resistances with the thick and the thin samples, respectively; its subtraction can also be expressed as:

$$R_{th\ system\ thick} - R_{th\ system\ thin} = R_{th\ thick} - R_{th\ thin} \tag{2.10}$$

where $R_{th\ thick}$ and $R_{th\ thin}$ are the thermal resistances of the thick and thin samples, respectively.

Both measured system thermal resistances have the same serial resistances contribution except that of the sample. Thus, every factor is canceled

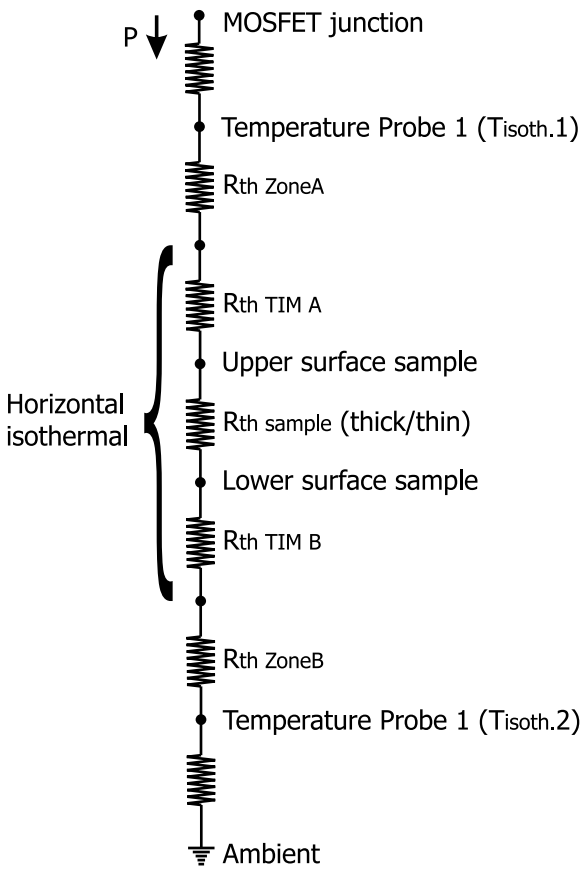


Figure 2.3: *Representation of thermal path of figure 2.2 using the Ohm's analogy.*

except that of the samples shown in equation (2.10). This difference is noted as $R_{th\ virtual}$ henceforth on and is assumed to be the thermal resistance of a virtual sample of material of thickness $L_{virtual}$.

$$R_{th\ virtual} = R_{th\ system\ thick} - R_{th\ system\ thin} \tag{2.11}$$

$$L_{virtual} = L_{thick} - L_{thin} \tag{2.12}$$

$R_{th\ virtual}$ and $L_{virtual}$ are used in equation (2.4) to compute the thermal conductivity of the material under test as follows.

$$k = \frac{L_{virtual}}{R_{th\ virtual} A} \tag{2.13}$$

2.2 Thermal Design

The measuring procedure described above requires to reproduce all the contributions to the $R_{th\ system}$ for various sample thickness measurements in order to be canceled by the subtraction of equation (2.11). Applying the same pressure, the TIM material and the surface roughness of samples will provide the same thermal resistance contribution. Figure 2.4 shows a diagram of the proposed setup to fulfil the above explained requirements. Figures 2.5 and 2.6 show different views of the implemented thermal measurement setup.

2.2.1 Thermal Interface Materials

The thermal grease RS HeatSink Compound from RS Components is used to thermally connect the MOSFET to the spreader, the spreader to the sample and the sample to the heatsink. The thermal conductivity nominal value of the used grease was $0.9\,\mathrm{W/m\,K}$.

It has been also tested the use of laminate thermal interface materials as Cho-Therm® T500 or Sil-Pad® A2000. Although, the RS HeatSink Compound has been considered more appropriate due to its lower thermal resistance in comparison with other options. Thus, higher heat flow can be injected with better accuracy results. In section 2.5.3 an experimental study of several thermal interface materials is discussed.

The sample is packed tight with a long piece of steel and wood that applies pressure to all thermal contacts as shown in figures 2.4 and 2.5. The piece is attached through two screws to the heatsink and the mounting torque is controlled by a torque wrench (R.304DA 1-5Nm 4% accuracy from FACOM).

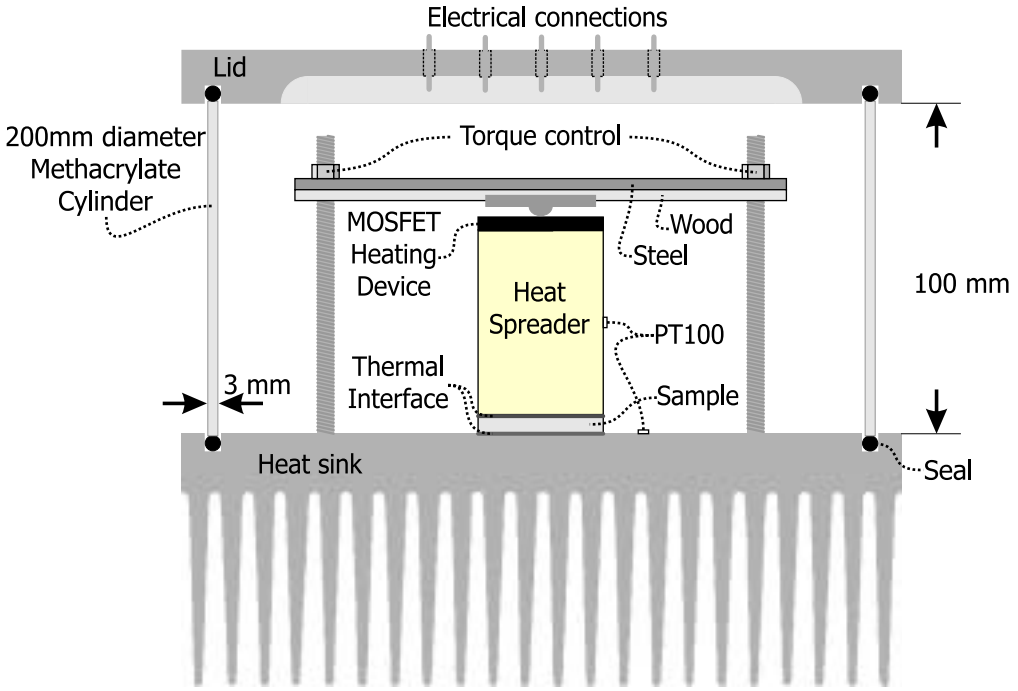


Figure 2.4: *Implementation of the thermal conditions of the measurement principle.*

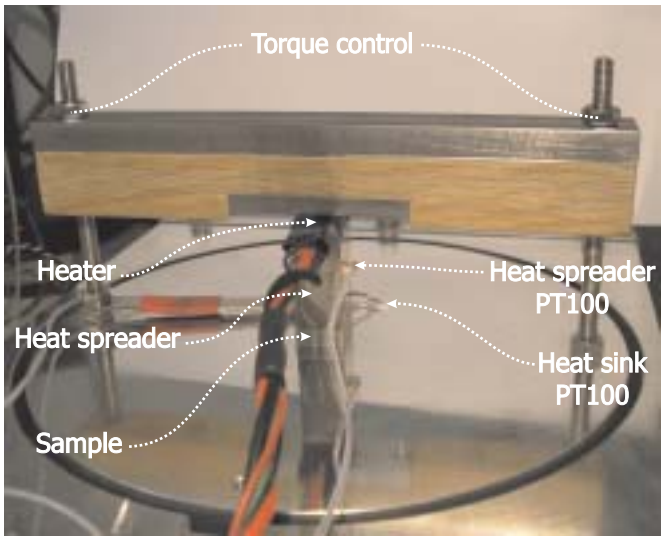


Figure 2.5: *Detail view of the inside vacuum cavity.*

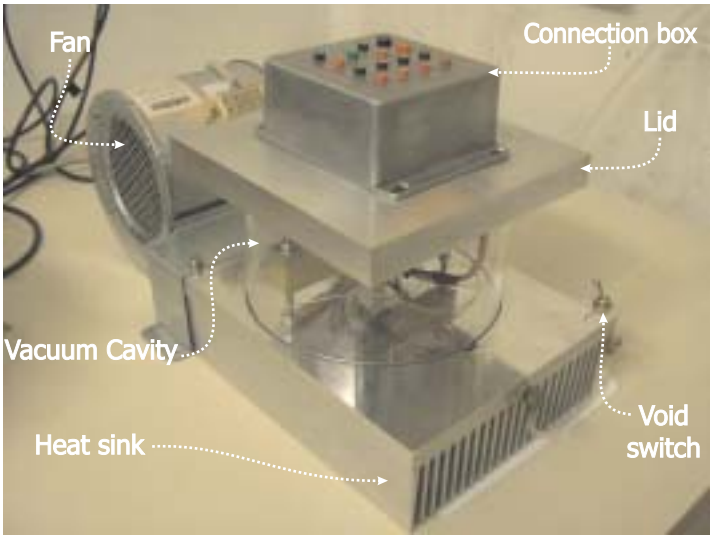


Figure 2.6: *General view of the setup.*

The pressure piece has been designed to minimize heat leakage by conduction through it, that would constitute a parallel thermal resistance $R_{th\,P}$ between the MOSFET and the ambient. A study of the effect of this parallel thermal resistance is exposed in section 2.4.3.

The effective thermal resistance of a thermal interface material placed between two surfaces to be thermally connected has two components [15]: the bulk resistance $R_{th\,bulk}$ arising from its finite thermal conductivity and the contact resistance $R_{th\,C}$ between the TIM and the adjoining solid. $R_{th\,TIM}$ may be expressed as:

$$R_{th\,TIM} = R_{th\,C1} + R_{th\,bulk} + R_{th\,C2} \tag{2.14}$$

Mahajan [15] establishes that contact resistance depends on the pressure, the surface roughness, the thermal conductivity of the TIM and its capillary force. Due to these facts, both samples to be compared have to be polished in the same way in order to get identical surface quality and, therefore, repetitive thermal resistances of the interface materials $R_{th\,TIM}$. This is an important issue: surface roughness of both samples must be equal.

It has been weight up the possibility of performing a measurement using only one sample. Two measurements would be done with and without sample. Nevertheless, since the method is based on a comparison, thermal contact resistance is not repeated and results are only reliable under special conditions. In section 2.5.2 such measurements will be exposed together with a discussion about the particular conditions that make them reliable.

2.2.2 Heatsink, Fan and Vacuum

In spite of the fact that air is a good thermal insulator, heat transport by natural convection around the heating device and the spreader could be a heat leakage mechanism that would lead to wrong results. It is essential that the generated heat power cross the sample without any leakage, because a correction of this effect would involve a quite complex and inaccurate model. This phenomenon is avoided keeping the heater, the spreader and the sample in a vacuum cavity placed on a big air cooled heat sink as shown in figures 2.4 and 2.6. The connections to the temperature probes and to the heating device are located through the lid of the cavity. The chosen heatsink is a P16/300 from Semikron, its dimensions are $215 \times 300\,\mathrm{mm}$ and the manufacturer assures a thermal resistance of $0.24\,^{\circ}\mathrm{C}/\,\mathrm{W}$ when air cooled. A big tangential fan SKF 16B-230-01 from Semikron is attached to the heatsink, its maximum flow is $610\,\mathrm{m}^3/\,\mathrm{h}$.

Heat transport by irradiation has not been considered because it is totally negligible at usual working temperatures in Power Electronics (up to 175°C).

2.2.3 Heat Spreader

The heat spreader is the piece that allows the heat arriving the sample homogeneously distributed. The measurement principle lays on this assumption, therefore special attention has been payed to this part of the design. Copper is chosen due to its high thermal conductivity, around $386\,\mathrm{W}/\,\mathrm{m\,K}$ according to the literature.

Thermal simulations have provided the optimal dimensions for the heat spreader, assuring the required conditions with a variety of sample materials. Thermal simulations of the thermal system to be implemented as shown in figure 2.7 have been carried out with FLOTHERM™ 4.2 software [16, 17]. Its cross section area is $15 \times 20\,\mathrm{mm}^2$ because of the MOSFET TO247 package dimensions. Then the thickness has to be chosen. It is worth setting the spreader and sample area as small as possible, thus the heat flow density and temperature difference will be higher and thermal resistance more accurately determined. Some issues of this subject are also analyzed in section 2.4.1.

Figure 2.8 shows the simulated temperature distribution with the chosen spreader thickness of 30 mm. The heating device is generating a heat power of 20 W. The spreader spreads the concentrated heat flux injected by the MOSFET leading to parallel isothermal surfaces at its bottom; i.e., homogeneous heat flux density flowing through the sample.

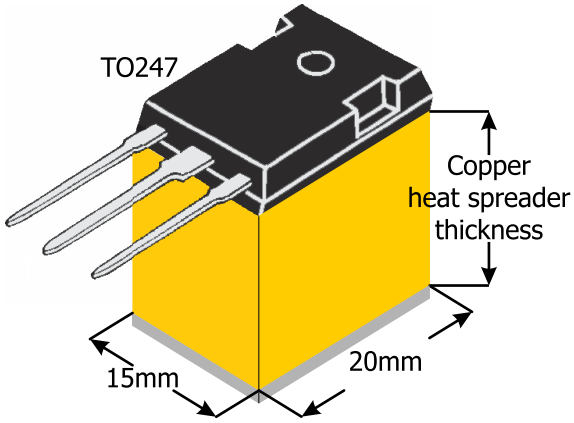


Figure 2.7: *Heat spreader model with the MOSFET heating device on top.*

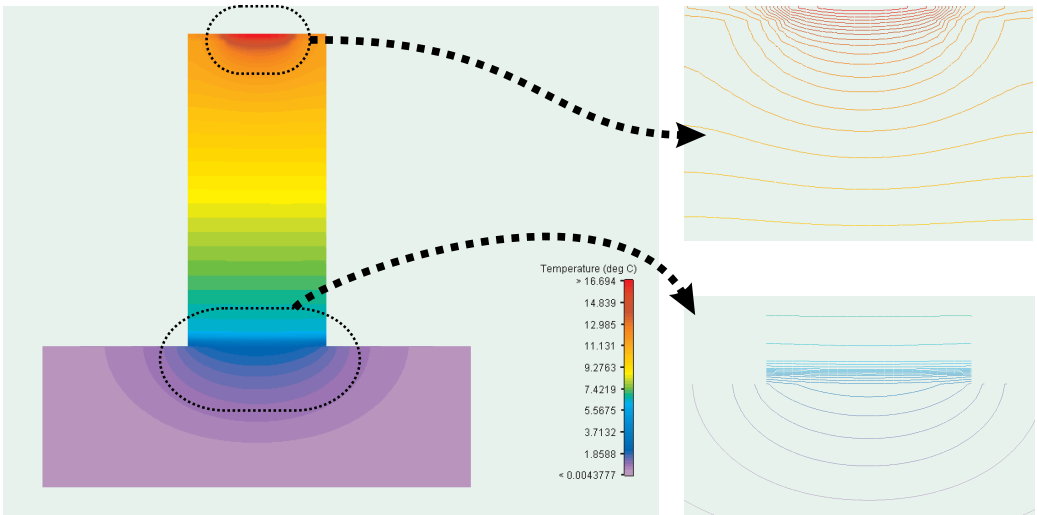


Figure 2.8: *Temperature field from a 3D thermal simulation. Heat spreader thickness is* $30\,\mathrm{mm}$ *with the MOSFET producing* $20\,\mathrm{W}$ *of heat power.*

2.2.4 Temperature Measurement

Temperature acquisition is one of the most critical issues of the proposed measurement system. Temperature probes must be as small as possible to ensure measurement on an isothermal zone and to avoid heat flow disturbances. Measurement has to be accurate as well due to the fact that materials under test show high thermal conductivity resulting on low thermal resistances and temperature differences to be measured.

Platinum Resistance Temperature Detectors (PRTD) are chosen. The probes are the thin film elements 2105 PT100 1/10 DIN from Sensycon. This is the tiniest and most precise thermal probe found. It has also an excellent long term stability. The measurement in this application is carried out with a Keithley 2700 multimeter.

The platinum is deposed on an $2 \times 2.3\,\mathrm{mm}^2$ and 0.5 mm thick Alumina substrate and etched by laser trimming. Thanks to this technique it is possible to deal with very small probes. The probe has four copper wires covered with Teflon for the four wire measurement of its electrical resistance. Figure 2.9 shows the probe structure.

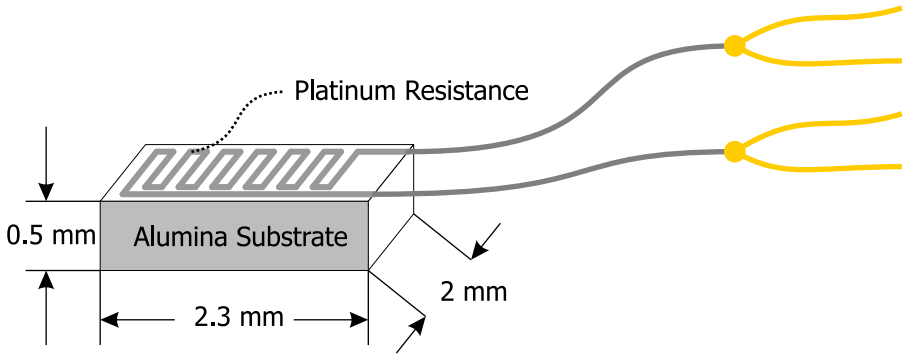


Figure 2.9: *Structure of the 2105 PT100 temperature probe.*

The IEC 751 standard establishes the temperature to platinum resistance relationship by means of a Kelvin measurement of the probe resistance [18, 19]. Standardized tolerances of the probes are also considered. The relationship is given for two ranges, first from $-200\,^{\circ}\mathrm{C}$ to $0\,^{\circ}\mathrm{C}$ and second from $0\,^{\circ}\mathrm{C}$ up to $850\,^{\circ}\mathrm{C}$. Nevertheless, our interest is only in the positive temperature range $[0\,^{\circ}\mathrm{C} - 850\,^{\circ}\mathrm{C}]$ due to the fact that a negative temperature will be never measured.

The electrical resistance of the platinum element may be expressed as follows:

$$R_T(T) = R_0(1 + AT + BT^2) \tag{2.15}$$

where T is the temperature of the platinum element in ℃, $R_T(T)$ is the temperature dependent electrical resistance and R_0 its nominal value; i.e., its electrical resistance at 0℃. The coefficients are the following:

$$A = 3.9083 \times 10^{-3}\,℃^{-1}$$
$$B = -5.775 \times 10^{-7}\,℃^{-2}$$

Thus, one can deduce the temperature vs resistance function, considering $R_0 = 100\,\Omega$:

$$T = 3383.809524 - \sqrt{11450166.89 + \frac{R_T - 100\,\Omega}{-5.775 \times 10^{-5}\,℃}} \tag{2.16}$$

This is the equation that the multimeter uses for temperature measurements.

According to the standard, a DIN 1 (or B class) platinum resistance PT100 tolerance is:

$$\Delta T(℃) = \pm(0.3 + 0.005T) \tag{2.17}$$

ΔT being the error of a measured temperature. The thin film element 2105 available for this application is 1/10 DIN; i.e., their accuracy is ten times higher:

$$\Delta T(℃) = \pm(0.03 + 0.0005T) \tag{2.18}$$

In other terms, the accuracy is between 0.03℃ and 0.08℃ for the 0℃ to 100℃ temperature range.

The two temperature probes used in the proposed setup are attached to the heat spreader and the heatsink by means of two springs. The spring applies pressure in order to keep the probe fixed and to assure a thermal contact. The thermal grease mentioned in section 2.2.1 is also used to make thermal contacts between the substrate of the probe and the surface to be measured. Figure 2.10 shows a picture of the system.

It is also remarkable that self-heating is an inherent effect in resistance sensors. An amount of current is injected in order to measure the electrical resistance, heat is dissipated and a temperature increase may happen. A study of such effect has been carried out by three means: analytical analysis, thermal simulation and direct measurement. The most unfavorable conditions will be considered in analytical and simulated estimations in order to assure that the self-heating temperature rise is imperceptible.

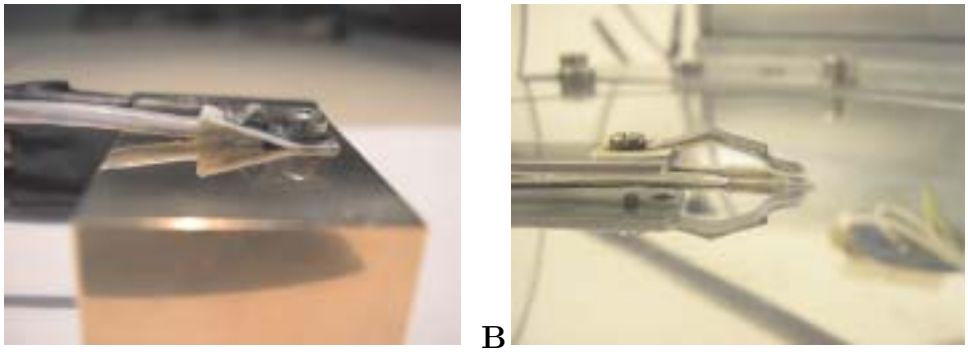


Figure 2.10: *View of the mounting system for the PT100 probes. A. to the heat spreader and B. to the heatsink.*

The analytical estimation is carried out using the equation (2.3). The thermal resistance of the Alumina substrate can be estimated considering that the platinum is an homogeneous and wide source of heat. The calculated thermal resistance is:

$$R_{th} = \frac{L}{kA} = 6.8\frac{\mathrm{K}}{\mathrm{W}} \tag{2.19}$$

where $A = 4.6\,\mathrm{mm}^2$, $L = 0.5\,\mathrm{mm}$ and $k = 16\,\mathrm{W/\,m\,K}$. The thermal conductivity of Alumina is within the $16 - 35\,\mathrm{W/\,m\,K}$ range and the lowest value is used. The sense current of the 2700 Keithley multimeter is 1 mA, and considering $R_T(130\,℃) \approx 150\,\Omega$ the generated power is:

$$P = 150\,\Omega(1\,\mathrm{mA})^2 = 0.00015\,\mathrm{W} \tag{2.20}$$

Once again it is considered the worst case, a high temperature of 130 ℃ instead of ambient temperature where $R_T(25\,℃) = 110\,\Omega$. Hence, the temperature rise, using the definition of thermal resistance in equation (2.1) is:

$$\Delta T = 6.8\frac{\mathrm{K}}{\mathrm{W}}0.00015\,\mathrm{W} = 0.00102\,℃ \tag{2.21}$$

This self-heating, even under the worst conditions, is almost 30 times lower than the maximum accuracy of the probe. Accordingly, this effect can be neglected.

Regarding the estimation by thermal simulation using FLOTHERM™ 4.2, the same conditions have been assumed, with similar results. The estimated model provides a temperature rise of 0.001122℃.

Finally, the self-heating has been measured experimentally. A probe is mounted on copper block and a series of temperature measurements have been carried out at increasing sense currents. The thermal equilibrium is reached before each temperature measurement. The voltage drop at the probe is also measured in order to get the electrical resistance and the dissipated heat power for every current $P = I_{sense}V$. The equation (2.16) provides the temperature values from the resistance readings. Results have been represented as temperature vs heat power $T(P)$. According to the definition of thermal resistance a closed-form of this function can be written as:

$$T(P) = PR_{th\,platinum-copper} + T_{copper} \tag{2.22}$$

where $R_{th\,platinum-copper}$ is the thermal resistance between the platinum resistance and the copper block and T_{copper} the block temperature. A lineal fit is performed in order to get the values of these parameters:

$$T_{copper} = T(0\,\mathrm{W}) = 22.000\,^{\circ}\mathrm{C}$$
$$R_{th\,platinum-copper} = 29.4\frac{\mathrm{K}}{\mathrm{W}}$$

Then, the operation of equation (2.21) is repeated considering the experimentally deduced thermal resistance and the resulting self-heating temperature increase is deduced:

$$\Delta T = 29.4\frac{\mathrm{K}}{\mathrm{W}}0.00015\,\mathrm{W} = 0.004\,^{\circ}\mathrm{C} \tag{2.23}$$

This value is higher than the estimated and simulated ones. The Alumina substrate may have lower thermal conductivity and the thermal grease may introduce a high thermal resistance contribution. Nevertheless, this self-heating temperature rise is still ten times lower than the maximum probe accuracy.

In conclusion, the self-heating effect can be definitively neglected as it is imperceptible compared with the current probe accuracy.

2.3 The Power Generation Controller

It has been explained in the previous sections how heat flow is achieved to be homogeneous and perpendicularly crossing the sample. This section exposes how the total heat flow P can be chosen and maintained steady.

A power MOSFET in TO247 package (IXFH 76 N07-11 from IXYS) has been chosen as a heat source. This power device is robust, versatile and can operate at relatively high temperatures (up to 175ºC). Moreover, it is easy to set a given amount of dissipated heat power simply connecting a voltage source between drain and source and driving its gate by means of a power controller circuit. The equipment must be versatile and any MOSFET device (packaged or not) or voltage controlled gated device (such as an IGBT) may be used. Apart from the steady heat power generation operation, the controller includes the possibility to generate arbitrary heat power waveforms.

The total heat power production of the heating system must be immune against electrical or thermal changes during operation. Nevertheless, an increase of the MOSFET junction temperature due to self-heating shifts its $I_D - V_{DS}$ characteristics [20]. It has been observed experimentally that drain current I_D decreases if device is heated. Thus, a continuously adaptation of the gate signal is necessary. A proportional controller based on an operational amplifier in inverter configuration has been first considered. This control adapts the gate signal, but the resulting drain current, and thus the power dissipation level, present a steady state error inherent to this control method that is temperature dependent resulting in non-steady power generation while the device is self-heating.

The integral automatic controller [21] described in this section matches the required dissipation conditions. It has been specifically optimized for the application exposed in this work, although other applications may take advantage of its versatility.

2.3.1 Operation

The MOSFET works within its saturation zone and the gate is driven by the integral automatic controller in order to produce an amount of dissipated heat power according to the needs described in section 2.1. The heating device and the control circuit make up the heating system. Its operation can be deduced from figure 2.11.

The power dissipated by the MOSFET is supplied by a high power voltage source V_{CC}. The control circuit is supplied by two auxiliary voltage sources of +15V and -15V integrated together with the circuit into the same box. The ground of the controller circuit and its +15V and -15V sources is chosen to be connected to the source of the MOSFET. Thus, the gate-to-source voltage V_G applied by the circuit is the V_{GS} parameter that controls the power device.

Drain current I_D is continuously measured by means of a shunt resistor, which its voltage drop, V_{shunt}, is proportional to the current.

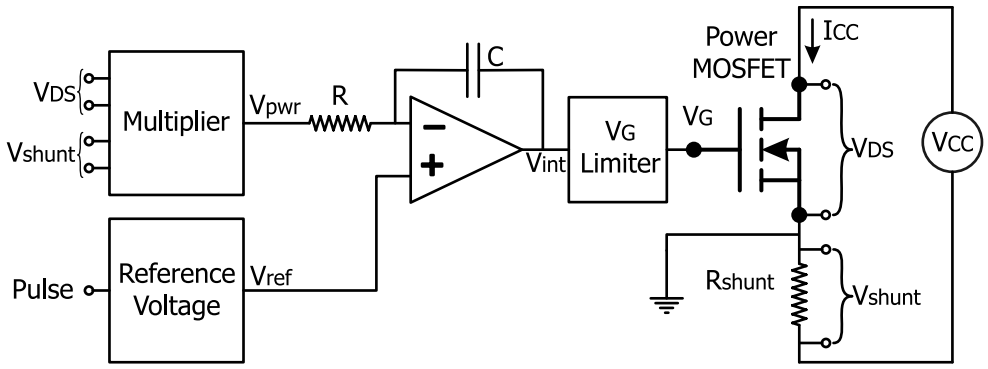


Figure 2.11: *The power controller main components.*

$$V_{shunt} = R_{shunt} I_D \tag{2.24}$$

The resistor must have a small and stable value even if high current causes eventually a temperature rise. Due to the fact that R_{shunt} is connected in series with the heating device, the drain-source voltage V_{DS} will vary depending on the current, equation (2.25). Thus, the integral controller has to compensate continuously the V_{DS} drop increasing I_D in order to reach the desired power dissipation level.

$$V_{DS} = V_{CC} - R_{shunt} I_D \tag{2.25}$$

The values of V_{DS} and V_{shunt} are multiplied by the multiplier block in order to obtain a signal proportional to the power dissipated by the device:

$$V_{pwr} = K V_{DS} V_{shunt} \tag{2.26}$$

where K is a constant resulting from the proposed multiplier block of the circuit, being its dimensions V^{-1}. The heat power dissipated in the MOSFET device is:

$$P = V_{DS} I_D \tag{2.27}$$

The V_{pwr} signal is proportional to P, and it is used as the feedback signal for the automatic control:

$$V_{pwr} = K R_{shunt} P \tag{2.28}$$

The V_{pwr} signal is compared with the reference signal V_{ref}, stable and fixed by the user in the Reference Voltage block. Then, an integrator circuit made up of the operational amplifier and an RC pair generates a gate signal V_{int}:

$$V_{int} = \int \frac{V_{ref} - V_{pwr}}{RC} \mathrm{d}t \tag{2.29}$$

or more conveniently expressed as:

$$\frac{\mathrm{d}V_{int}}{\mathrm{d}t} = \frac{V_{ref} - V_{pwr}}{RC} \tag{2.30}$$

The V_{int} signal is connected to the MOSFET gate through a voltage limiter that will be described further on. The power dissipated by the device closes the feedback loop. The presence of the V_G limiter overcomes some problems due to the compliance of the power source that would prevent a good control operation. The V_G signal level limitation does not affect the amount of controlled power nor the quality of the control. Then, it is not essential for the understanding of the control operation.

If the heating device is dissipating an amount of power that produces a V_{pwr} signal higher than V_{ref}, the gate signal is lowered reducing the current I_D and V_{pwr} towards the equilibrium state expressed by the equation (2.31). The opposite happens when V_{pwr} signal is lower than V_{ref}. Therefore, the controller circuit manages the drain current I_D correcting the gate voltage continuously as expressed in equation (2.30), driving the system to the equilibrium state in which the steady state error is eliminated:

$$V_{pwr} = V_{ref} \tag{2.31}$$

Therefore, the user can choose the power dissipated by the heating device modifying the reference voltage. From equations (2.28) and (2.31) the heat power dissipated by the MOSFET device is deduced.

$$P = \frac{V_{ref}}{K R_{shunt}} \tag{2.32}$$

Figure 2.12 represents the $I_D - V_{DS}$ characteristic of a controlled MOSFET device if the voltage limiter not implemented. Several power parametric curves are represented in red. Every curve represents a different constant

power level $P = V_{DS}I_D$. While the power is disabled; i.e., V_{ref}=0V, V_G=-15V, I_D=0A and $V_{DS} = V_{CC}$. When the control is enabled the gate voltage rises carrying the MOSFET state along the represented line of slope $-R_{shunt}^{-1}$ crossing increasing power values up to the desired power level. The slight changes on the $I_D - V_{DS}$ characteristics of the MOSFET caused by the device heating will be immediately corrected by the controller in order to keep the heat power at a fixed level.

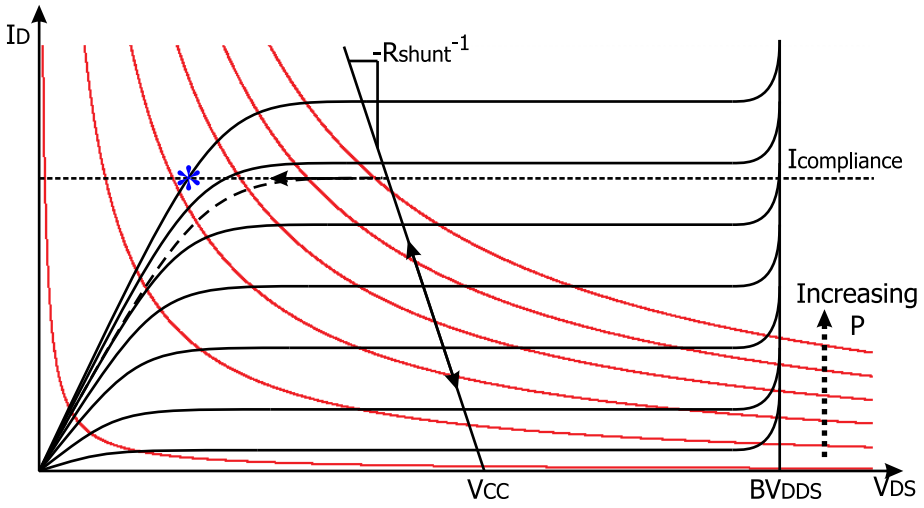


Figure 2.12: *Working MOSFET $I_D - V_{DS}$ scheme without voltage limiter.*

Nevertheless, the power cannot be raised indefinitely since the power source has a compliance current $I_{compliance}$ that cannot be exceeded. If due to the reference demand the compliance current is achieved, the power source will decrease the V_{CC} voltage keeping the current level and bringing the device to a state represented in figure 2.12 with the symbol $*$. The controller will rise the gate voltage up to its limit (+15V) since power dissipation is below the reference demand. Therefore, there would be no way of leaving this trap state. Even fluctuations during normal operation or during the transient time until the power level below the compliance current is stabilized, can also bring the system into the trap state with the same result. As closer to the maximum current more likely is to get a fail. Once the control fails this way, the power supply must be turned off.

The described problem can be solved solved with a voltage divider placed at the output of the integrator. This limiter prevents the gate voltage from exceeding a maximum $V_{GS\,max}$ voltage level selected by the user. A suitable value avoiding to reach the compliance current must be selected as shown in figure 2.13. The most convenient is to set this limit as low as possible in

accordance with the needed power level. Under this conditions the trap state is unreachable and the control operation works correctly, as described above.

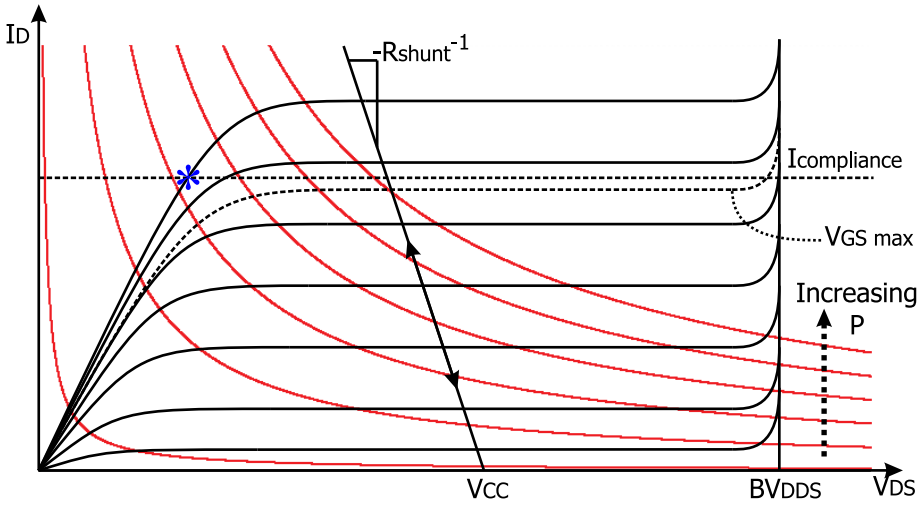


Figure 2.13: *Working MOSFET I-V scheme, the voltage limiter prevents the device from falling into the state ∗.*

2.3.2 Implementation

This section describes the implementation of the controlling circuit, the figure 2.14 shows the schematic. The circuit is mounted in a box and the shunt resistance and the MOSFET are external, being connected through the $V_{shunt}+$ and $V_{shunt}-$, V_D, V_S and V_G connections (shunt, drain, source and gate respectively). There is also a *Pulse* connection for transient reference input. The user can adjust the circuit performance by means of the *Pulse/Ref* and *Enable/Disable* switches, and also with the *Range*, *Control* and *Limiter* potentiometers. All these connections and controls can be seen on pictures of figure 2.15. All connections have been replicated at the rear side of the box to facilitate the connections.

The power supply part is mounted together in the same box and it is not shown in figure 2.14. It consist of two voltage sources connected to *Supply Input* +15V and -15V to the controller circuit.

As shunt resistance it has been chosen a RTO 20 manufactured by Vishay, 1Ω of nominal value, 1% precision and packaged in TO220, the manufacturer ensures a stability of 150ppm/°C.

The reference part of the circuit is in fact two references selected alternatively by the *Pulse/Ref* switch. The DC part of the reference block is based on the REF-01 IC of Analog Devices. It provides a 10V DC signal very

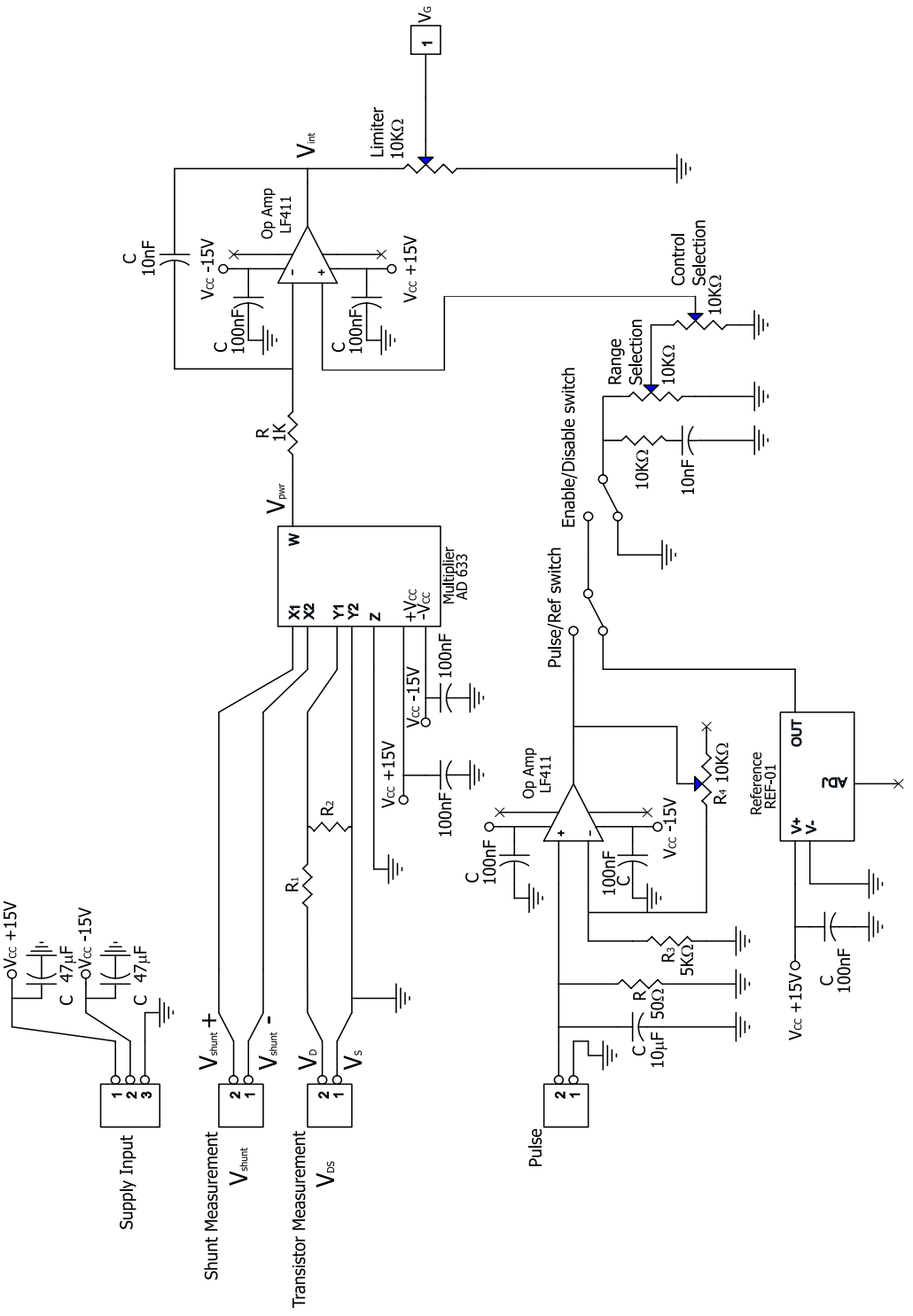


Figure 2.14: *Schematic of the implemented circuit.*

Figure 2.15: *The control circuit.*

stable, the final V_{ref} signal is chosen by the user via a voltage divider implemented with the *Range* and *Control* potentiometers. Both potentiometers are multi-turn and very stable. The power level is maintained very stable during longtime operation. The pulse part has been implemented in order to provide the system with the capability to dissipate arbitrary shaped heat power waveforms. An external signal from a waveform generator is adapted through a non-inverser amplifier and an RC load. The input has been tested with a Hewlett Packard 33120A *arbitrary waveform generator*, it generates 5 V pulses, and has to be in series with a 50 Ω load. Such resistance has been implemented together with a 10 µF capacitance to avoid noise. Different generators may need other RC load, thus this part should be modified accordingly. The non-inverser amplifier is implemented with a variable resistance R_4. Its gain must be adapted to obtain a V_{ref} signal of amplitude 10 V.

The *Enable/Disable* switch connects the V_{ref} between the 10V and the ground 0V forcing the controller to decrease the gate signal to prevent the electrical conduction. A RC pair has been implemented in order to avoid switch bounce. This switch is useful to prevent power dissipation between measurements.

The *Range* potentiometer sets the V_{ref} level controlled by the *Control* potentiometer. The user will proceed in the following way to set a particular power dissipation level:

1. Set *Control* and *Range* potentiometers to its maximum, then increase *Limiter* potentiometer up to a value slightly above the maximum desired current level.

2. Shrink *Range* until the maximum current value.

3. Choose the current level within the available power range via the *Control* potentiometer.

The multiplier part of the circuit is implemented with the AD633 analog multiplier from Analog Devices. Its output signal V_{pwr} is proportional to the product of two independent differential signals X and Y when Z is connected to the ground (0V).

$$V_{pwr} = \frac{(X_1 - X_2)(Y_1 - Y_2)}{10V} + Z \tag{2.33}$$

X signal is connected to the V_{shunt} signal and Y to V_{DS}.

The maximum voltage of the X or Y input signals is 18V. Nevertheless, V_{DS} can be equal to V_{CC}; i.e., up to 63V for the measurements exposed in this work. The problem is solved with a voltage divisor implemented by resistances R_1 and R_2 in figure 2.14. Resistances are chosen to get a maximum Y signal lower than $15V$ even for $V_{CC} = 70\,\mathrm{V}$.

$$Y = \frac{R_2}{R_1 + R_2} V_{DS} \;\Rightarrow\; Y = 0.2136 V_{DS} \tag{2.34}$$

Therefore, as the output signal of the multiplier is:

$$V_{pwr} = 0.02136\,\mathrm{V}^{-1} V_{shunt} V_{DS} \tag{2.35}$$

Thus, the equation (2.28) provides the value of the K constant.

$$K = 0.02136\,\mathrm{V}^{-1} \tag{2.36}$$

Therefore an approximation of the aimed power level is deduced from equation (2.32) considering $R_{shunt} = 1\,\Omega$, yielding:

$$P = 46.82\,\mathrm{A}\,V_{ref} \tag{2.37}$$

The integrator circuit is made of a LF411 general purpose operational amplifier and an RC pair. The velocity of the response depends on the value of $1/RC$, derived from equation (2.30). The response must be fast enough to seek a V_{ref} impulse without endanger the system stability. It has been chosen $R = 1\,\mathrm{k}\Omega$ and $C = 10\,\mathrm{nF}$ then the V_{int} stabilization after a V_{ref} step signal is reached in approximately 1 ms.

The generated heat power must be calculated from the voltage and current applied by the power source as:

$$P = V_{DS} I_D = V_{CC} I_{CC} - I_{CC}^2 R_{shunt} - \frac{(V_{CC} - I_{CC} R_{shunt})^2}{R_Y} \tag{2.38}$$

where R_Y is the input impedance of the V_{DS} measurement connection of the power controller circuit. A voltage divisor is implemented in the Y input of the multiplier, as seen in figure 2.14 and explained on page 29 (section 2.3.2). Therefore, this resistance is the sum of R_1 and R_2. Nevertheless, for better accuracy its value is directly measured:

$$R_Y = (10.2004 \pm 0.0001)\,\mathrm{k\Omega} \tag{2.39}$$

The equation (2.38) is deduced taking into account that the MOSFET device is connected to the power source having a parallel R_Y and the shunt resistance in series.

The resulting circuit, implemented inside a box as described, with the $V_{shunt}+$, $V_{shunt}-$, V_D, V_S and V_G external connections is shown in figure 2.16.

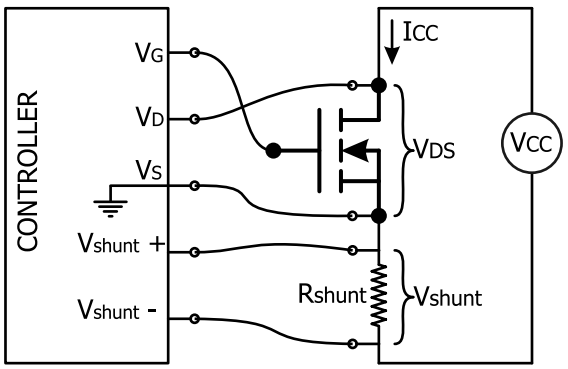


Figure 2.16: *Connection of the controlling circuit.*

The connections are set through the vacuum cavity lid as indicated in figure 2.17. Note that the connection box holds the shunt resistance inside.

2.4 Measuring Considerations

The system design has minimized the undesired parasitic effects such as the heat flow leak through air or the pressure piece. It has been also taken into

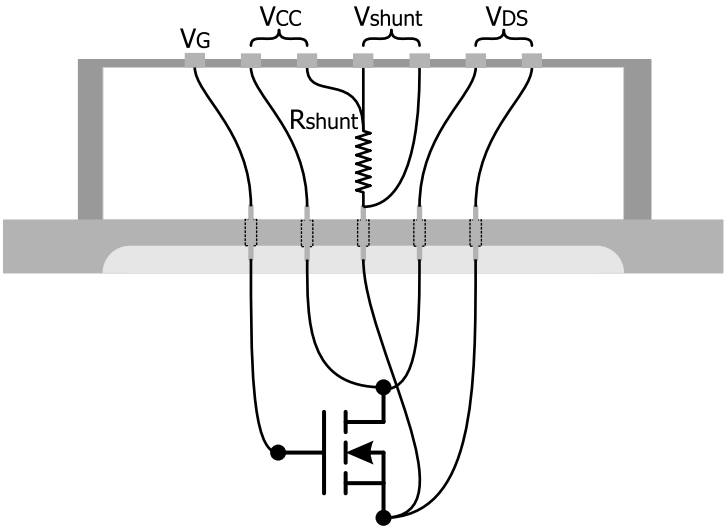


Figure 2.17: *Connection of the controlling circuit through the lid and the connection box.*

account factors that optimize the measurement accuracy and reliability, such as the temperature probe accuracy and size, the thermal interface material selection and the repeatability of the measuring conditions through both differential measurements.

2.4.1 Error Analysis

In this section, a summary of the measuring procedure is addressed together with a study of the error analysis. Given a n variable function $f(x_1, \ldots, x_n)$ and the uncertainties of its variables being Δx_1, ..., Δx_n, the propagation of the variable uncertainties to the result calculation is the following:

$$\Delta f(x_1, \ldots, x_n) = \sum_{i=1}^{n} \left| \frac{\partial f(x_1, \ldots, x_n)}{\partial x_i} \right| \Delta x_i \tag{2.40}$$

First, two samples of the same material must be mechanized in $15 \times 20\,\mathrm{mm}^2$ cross-section and two different thicknesses L_{thick} and L_{thin}. It is important to be sure that the two samples correspond exactly to the same material and their surfaces have the same roughness. The sample is placed under the heat spreader and a pressure is applied with the torque wrench. Identical pressure should be applied to the second sample. The cavity is

closed, the vacuum connected and electrical connections set. The fan must be switched on and the controller adjusted as explained in the previous sections. When thermal equilibrium is reached the temperature lectures, $T_{isoth.1}$ and $T_{isoth.2}$ are recorded together with the voltage and current applied by the power source (V_{CC} and I_{CC}).

The thermal resistance of the system with the sample is determined with the equation (2.8). Using the expression (2.40) its uncertainty is deduced:

$$\Delta R_{th\ system} = \frac{1}{P}\Delta T_{isoth.1} + \frac{1}{P}\Delta T_{isoth.2} + \frac{T_{isoth.1} - T_{isoth.2}}{P^2}\Delta P \qquad (2.41)$$

The uncertainty of the temperature measurement with the PT100 1/10 DIN probes is provided by equation (2.18). Thus, every temperature measurement has the following uncertainty:

$$\Delta T(\,^{\circ}\mathrm{C}) = \pm(0.03 + 0.0005T) \qquad (2.42)$$

The heat power generation is calculated with the equation (2.38), and uncertainty of P is deduced from:

$$\begin{aligned}
\Delta P &= \left| I_{CC} - \frac{2(V_{CC} - I_{CC}R_{shunt})}{R_Y} \right| \Delta V_{CC} + \\
&+ \left| V_{CC} - 2I_{CC}R_{shunt} + \frac{2R_{shunt}(V_{CC} - I_{CC}R_{shunt})}{R_Y} \right| \Delta I_{CC} + \\
&+ \left| -I_{CC}^2 + \frac{2I_{CC}(V_{CC} - I_{CC}R_{shunt})}{R_Y} \right| \Delta R_{shunt} + \\
&+ \left| \frac{(V_{CC} - I_{CC}R_{shunt})^2}{R_Y^2} \right| \Delta R_Y \qquad (2.43)
\end{aligned}$$

The measured voltage and current in the power voltage source (SourceMeter 2420 from Keithey) have also their respective uncertainties that are propagated. The manufacturer assures an accuracy of 0.02% of the readed voltage and 0.035% of the readed current. Thus, as in the case of the temperature measurement, the uncertainty depends on the measured values:

$$\Delta V_{CC} = 0.0002 V_{CC} \qquad (2.44)$$

$$\Delta I_{CC} = 0.00035 I_{CC} \qquad (2.45)$$

Two thermal resistance measurements are performed with two samples, which differ only different in thickness, and the virtual sample thermal resistance is calculated with equation (2.11). Its uncertainty is:

$$\Delta R_{th\,virtual} = \Delta R_{th\,system\,thick} + \Delta R_{th\,system\,thin} \tag{2.46}$$

The final calculation of the material thermal conductivity k is provided by equation (2.13). Its uncertainty being.

$$\begin{aligned} \Delta k &= \frac{1}{R_{th\,virtual}A}\Delta L_{virtual} + \frac{L_{virtual}}{R^2_{th\,virtual}A}\Delta R_{th\,virtual} + \\ &+ \frac{L_{virtual}}{R_{th\,virtual}A^2}\Delta A \end{aligned} \tag{2.47}$$

The sample thickness is measured with a micrometer with the following precision:

$$\Delta L = 0.00001\,\mathrm{m} \tag{2.48}$$

The thickness of the virtual sample is deduced from equation (2.12) and therefore its uncertainty is:

$$\Delta L_{virtual} = \Delta L_{thick} + \Delta L_{thin} = 0.00002\,\mathrm{m} \tag{2.49}$$

The area of the samples is $A = 0.0003\,\mathrm{m}^2$. When the sample is cut, its dimensions are checked with a vernier calliper gauge with a precision of $5 \times 10^{-5}\,\mathrm{m}$. Then, the uncertainty of the sample area is:

$$\Delta A = (0.015\,\mathrm{m} + 0.020\,\mathrm{m})5 \times 10^{-5}\,\mathrm{m} = 1.75 \times 10^{-6}\,\mathrm{m}^2 \tag{2.50}$$

2.4.2 Mesurement Optimization and Limits

The uncertainty of the thermal conductivity has been described in section 2.4.1. Nevertheless, the uncertainty can be also expressed in terms of relative uncertainty. From equations (2.47) and (2.3) one gets:

$$\frac{\Delta k}{k} = \frac{\Delta L_{virtual}}{L_{virtual}} + \frac{\Delta R_{th\,virtual}}{R_{th\,virtual}} + \frac{\Delta A}{A} \tag{2.51}$$

The relative uncertainty of the thermal conductivity is the sum of relative uncertainties of all its variables. This expression is more appropriate for separately analyzing the influence of every variable on the final uncertainty of the measured k.

The first term of the equation (2.51) is the relative uncertainty of the sample thickness. The value of L is measured with a $\pm 10^{-5}\,\mathrm{m}$ precision (equation (2.48)). Therefore, the relative uncertainty of a 2 mm thick sample is:

$$\frac{\Delta L_{virtual}}{L_{virtual}} = \frac{0.02}{L_{virtual}} = 1\% \tag{2.52}$$

which results lower if a thicker sample is measured. Thus, the contribution of the L measurement to the final relative uncertainty of the thermal conductivity is less than 1% for all measured samples. This is an acceptable value, such accuracy is considered excellent.

The third term of equation (2.51) is the relative uncertainty of the sample area. Its value is:

$$\frac{\Delta A}{A} = \frac{1.75 \times 10^{-6}\,\mathrm{m}^2}{0.0003\,\mathrm{m}^2} = 0.005833 = 0.5833\% \tag{2.53}$$

This contribution to the final relative uncertainty of k is even better and it is also acceptable.

The critical contribution factor is the second term of equation (2.51). Its value may be much higher than other factors depending on the samples under test or even other conditions. The following expression is deduced from equations (2.11) and (2.46).

$$\frac{\Delta R_{th\,virtual}}{R_{th\,virtual}} = \frac{\Delta R_{th\,system\,thick} + \Delta R_{th\,system\,thin}}{R_{th\,system\,thick} - R_{th\,system\,thin}} \tag{2.54}$$

It is desirable to have the lower contribution from this factor. It is deduced that minimization is achieved if the following condition is accomplished:

$$\Delta R_{th\,system\,thick} + \Delta R_{th\,system\,thin} \ll R_{th\,system\,thick} - R_{th\,system\,thin} \tag{2.55}$$

Then the $(\Delta R_{th\,system\,thick} + \Delta R_{th\,system\,thin})$ factor must be minimized while maximizing the $(R_{th\,system\,thick} - R_{th\,system\,thin})$ factor.

Regarding the minimization of $\Delta R_{th\,system}$ for all the samples, and according to equation (2.41), the temperature measurement must be very precise.

Thus, high precision probes are used. $\Delta R_{th\,system}$ is also minimized if the total heat flow P value is high. The maximum available P value is limited by the maximum thermal capabilities of the heating MOSFET. In addition, it has to be taken into account that if $R_{th\,system}$ is high, the MOSFET temperature will be also high; its maximum permitted value (175 ℃) must not be reached by reducing P.

The maximization of the $R_{th\,system\,thick} - R_{th\,system\,thin}$ factor is achieved using very different thicknesses of both compared samples. A high $R_{th\,virtual}$ must be achieved either measuring thick virtual samples or low thermal conductivity materials.

The typical experimental uncertainty of $R_{th\,system}$ with the available temperature probes, thermal configuration and even with relatively high thermal conductivity samples is about 0.002 ℃/ W or less, as it will be discussed in section 2.5.1. Therefore, from expression (2.54) it is estimated that this contribution is under 10% if the virtual thermal resistance is higher than 0.04 ℃/ W:

$$R_{th\,virtual} > 0.04\,℃/\,\mathrm{W} \implies \frac{\Delta R_{th\,virtual}}{R_{th\,virtual}} < 10\% \tag{2.56}$$

This is the minimum virtual thermal resistance that can be measured yielding a 10% accuracy on thermal conductivity measurements.

2.4.3 Parallel Thermal Resistance

As explained before, the pressure piece could constitute a heat leakage path. Its thermal resistance is noted as $R_{th\,P}$. For calculating the effect of such leakage it is considered the thermal circuit represented in figure 2.18 using the Ohm's analogy. The figure represents the same thermal system of figure 2.3 adding the parallel thermal resistance of the leakage path.

According to the described measurement procedures and equation (2.8), the thermal resistance reading can be expressed as:

$$\frac{T_{isoth.1} - T_{isoth.2}}{P} = R_{th\,system}\left(1 - \frac{R_{th\,system} + R_{th\,1} + R_{th\,2}}{R_{th\,P} + R_{th\,system} + R_{th\,1} + R_{th\,2}}\right) \tag{2.57}$$

The measured value, supposed to be $R_{th\,system}$, is modified by a multiplicative factor, resulting in lower readings of thermal resistance. This is no longer valid if the value of the thermal resistance $R_{th\,P}$ is high enough, this factor being approximately one. On the contrary, high values of $R_{th\,system}$ will increase the measurement distortion.

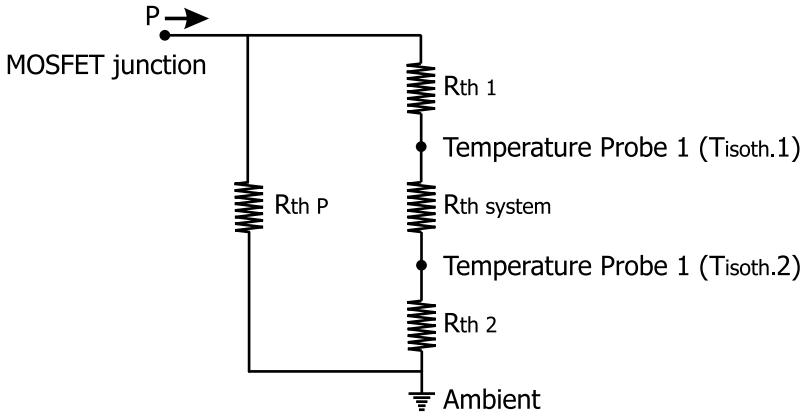


Figure 2.18: *Electric-like thermal circuit (Ohm's analogy) with a heat leakage through a parallel path described by* $R_{th\,P}$.

A worst case estimation of the $R_{th\,system}$ modifying factor, can be estimated as follows. $R_{th\,system}$ is experimentally found to be near $0.5\,℃/\,\mathrm{W}$ when measuring a high thermal resistance sample. The sum of $R_{th\,1}$ and $R_{th\,2}$ is much lower but considering a pessimistic value ($0.5\,℃/\,\mathrm{W}$).

$$\frac{T_{isoth.1} - T_{isoth.2}}{P} = R_{th\,system}(1 - \frac{1\,℃/\,\mathrm{W}}{R_{th\,P}1\,℃/\,\mathrm{W}}) \tag{2.58}$$

A graphical representation of this expression is shown in figure 2.19. It is inferred that values of $R_{th\,P}$ above $500\,℃/\,\mathrm{W}$ make the factor to be practically one. Considering equation (2.3) as an approximation of the thermal resistance of the pressure piece, $R_{th\,P}$ is estimated above $10000\,℃/\,\mathrm{W}$. Therefore, the parallel heat leakage can be neglected in the measurements.

If $R_{th\,P}$ is low enough to have a significant effect on the measurement, the following equation would replace equation (2.8).

$$R_{th\,system} = \frac{(T_{isoth.1} - T_{isoth.2})(R_{th\,P} + R_{th\,1} + R_{th\,2})}{PR_{th\,P} - (T_{isoth.1} - T_{isoth.2})} \tag{2.59}$$

Assuming that values of $R_{th\,P}$, $R_{th\,1}$ and $R_{th\,2}$ are known.

2.4.4 Sample Misalignment

The sample area must match with that of the heat spreader and during the measurement the sample under test must be accurately aligned with it. If the sample and the spreader are not properly aligned, as shown in figure 2.20, the measurement will yield a false thermal conductivity value.

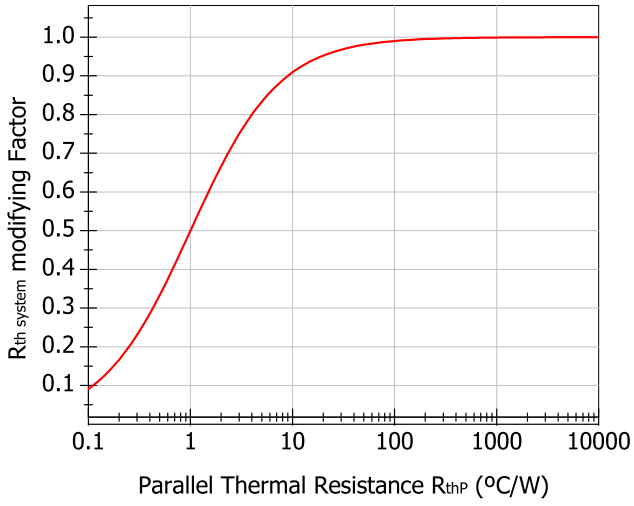


Figure 2.19: *Graphical representation of the expression* (2.58).

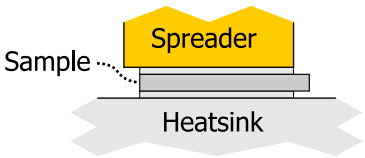


Figure 2.20: *Misaligned sample under the heat spreader.*

Assuming that both samples (area A) of a given measurement are misaligned in such a way that only a certain part of its area is in contact with the spreader, $A_{m\,thin}$ and $A_{m\,thick}$; the relative deviation of the measured thermal conductivity k_m of the material in comparison with the "true" k parameter is approximately:

$$\frac{k_m}{k} = \frac{A_{m\,thin}A_{m\,thick}}{A(A_{m\,thin} - A_{m\,thick})} \tag{2.60}$$

This result is approximate because it is supposed that the rest of measurement conditions are still valid in spite of the sample position. Nevertheless, this estimation gives a rough idea of how a misalignment may affect the measured values. The conclusion from this misalignment estimation is that if no appreciable misalignment is detected by the operator during the measurement (around 100 µm), it can be ignored due to the fact that its relative deviations are insignificant in comparison with other error sources.

2.5 Experimental Results

The procedure and calculations for the measurement have been established in the previous sections of this chapter. Next, thermal conductivity measurements of several ceramics have been performed with the measurement system in order to validate it. It has been also tested with non-differential measurements and for a comparative study of several thermal interface materials.

In order to guarantee temperature stability, the system is placed in a closed room whilst the operator remains outside. Data acquisition for temperature readings is performed remotely via GPIB bus with the Xlinx2700 startup software, specially devoted to work with the Keithley 2700 multimeter. Thermal equilibrium of the thermal system is reached before proceeding with any measurement. Every $R_{th\,system}$ thermal resistance measurement (equation (2.1)) is in fact the average value of 15 different temperature readings performed at regular intervals during one hour. A view of the experimental setup is shown in figure 2.21

A MS Excel spreadsheet has been used to perform the calculations for every particular measurement [1].

[1] An example view can be found on page 182 in appendix C

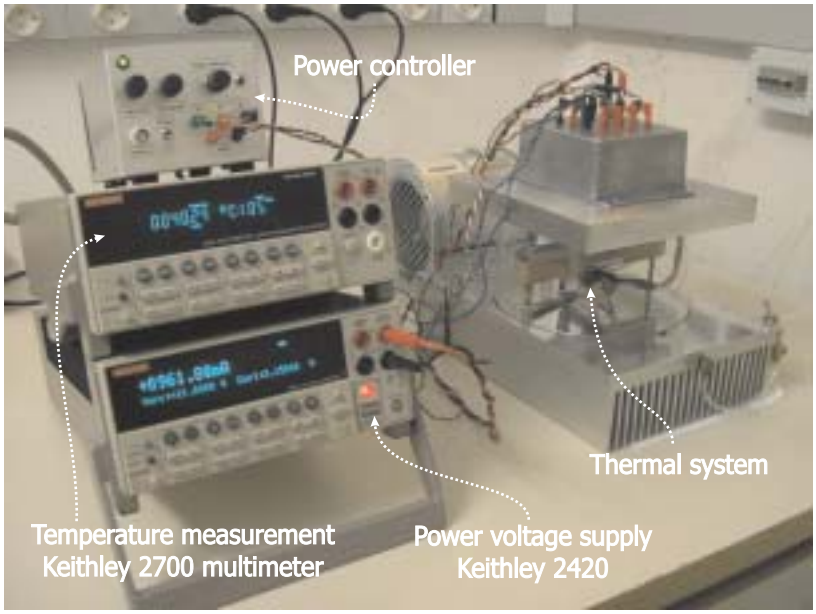


Figure 2.21: *Total measurement system view.*

2.5.1 Thermal Conductivity Measurements of Ceramics

The validation works of the proposed thermal conductivity measurement system have been carried out on three experimental materials provided by INASMET [2].

The first material is an anisotropic BN based ceramic composite measured on two perpendicular directions manufactured by hot pressing. A perpendicular direction to the pressing, referenced as X Direction, shows a nominal thermal conductivity 60% higher than the value along the pressing direction, referenced as Z Direction. Secondly, an AlSiC composite manufactured by squeeze casting. And thirdly, an AlN manufactured by uniaxial pressing and pressureless sintering under nitrogen atmosphere.

The validation methodology consisted basically in comparing the thermal conductivity values obtained using standard equipment with the values obtained using the proposed system. The previous characterization measurements where provided by INASMET.

Characterization with Standard Equipment

Data obtained from the measurement of the specific heat, thermal diffusivity and room-temperature bulk density were used to compute the thermal

[2] website: http://www.inasmet.es

conductivity according to equation (2.61) [12].

$$k(T) = \rho c_p(T) a(T) \tag{2.61}$$

k being the thermal conductivity, ρ the bulk density, c_p the specific heat, a the thermal diffusivity and T the temperature.

The thermal diffusivity was measured using a Netzsch model 427 laser flash diffusivity apparatus (LFA). The used thermal diffusivity results are the average values of five individual tests. The measured standard deviation was less than 1%. The samples were disks with a diameter of 12.5 mm and thickness of approximately 3 mm.

The specific heat measurements were conducted using a Netzsch model DSC 404 C Pegasus® differential scanning calorimeter (DSC). The sample was then run and the specific heat calculated by the standard ratio method. The samples were tested with the DSC had masses of $120 - 130$ mg. The system was equipped with a temperature-calibrated DSC c_p type S sensor. Platinum crucibles with lids were employed for the tests.

The uncertainty of both instruments for standard thermal conductivity measurement (DSC and LFA) is better than 3%. This value does not consider the influence of sample inhomogeneities. Therefore, an uncertainty of 5% for the test results should be considered.

Figure 2.22 shows the thermal conductivity on both directions of the ceramic material up to 200 °C, data are summarized in table 2.1. As it can be observed, around 50 °C the thermal conductivity following X Direction is 43 W/ m K, whilst the value for Z Direction is 25.5 W/ m K. Due to the increase of the specific heat with temperature, the thermal conductivity shows a lower temperature dependence than that of the thermal diffusivity. The thermal conductivity determination was carried out using a room-temperature bulk density of 2.913 g/ cm^3.

Temperature (°C)	Z direction k (W/ m K)	X direction k (W/ m K)
26	43.382	25.74
48	42.774	25.505
98	41.482	24.832
199	38.484	23.198

Table 2.1: *Characterization results for the BN ceramic.*

Table 2.2 contains the thermal conductivity results for the second studied material, the AlSiC composite, from 20 °C up to 200 °C. A graphical representation is shown in figure 2.23. Thermal conductivity is obtained to be around 150 W/ m K at room temperature, decreasing as temperature rises.

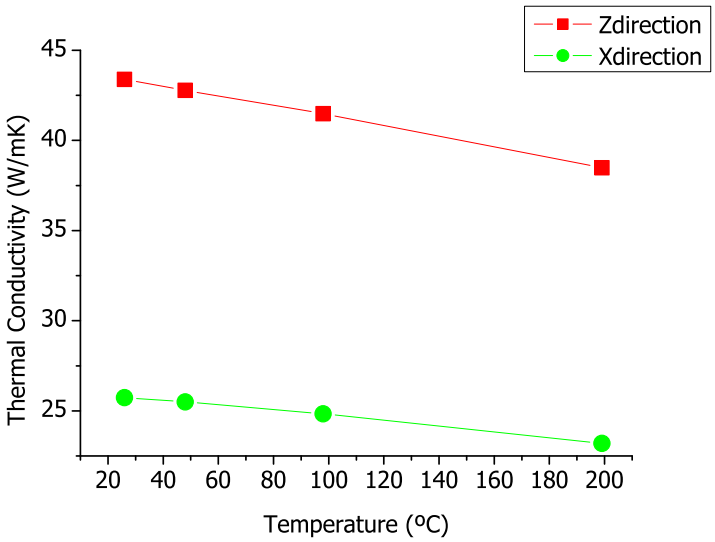


Figure 2.22: *Characterization results for the BN ceramic.*

There is no previous characterization data available for the third tested material, the AlN ceramic.

Temperature (°C)	k (W/m K)
20	150.43
50	148.40
100	141.80
150	134.60
200	129.58

Table 2.2: *Characterization results for the AlSiC ceramic.*

BN Anisotropic Ceramic Measurement

For the X Direction of the first material, there are three samples available of thicknesses 6.01 mm, 14.97 mm and 20.03 mm, and area 380 mm^2. Thus, three differential measurements can be performed getting three different virtual samples. Table 2.3 shows the obtained results.

There are available three samples of thicknesses 4.06 mm, 7.94 mm and 15.02 mm for Z Direction. The results for the three possible differential measurements are indicated in table 2.4.

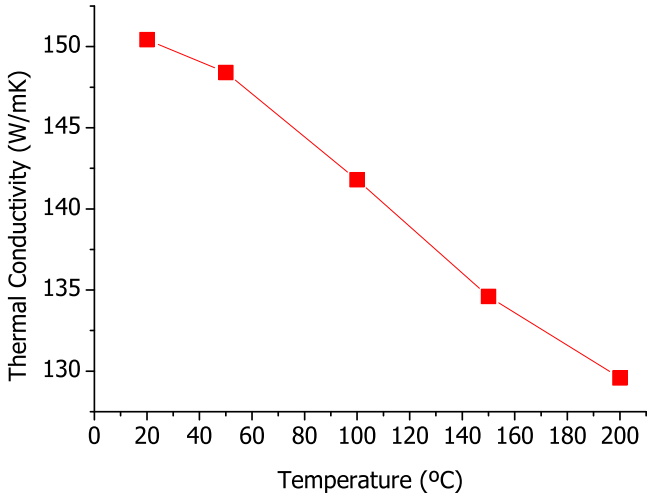


Figure 2.23: *Characterization results for the AlSiC Metal Matrix composite.*

Sample combination	$L_{virtual}$ (mm)	k ($\mathrm{W/mK}$)
20.03 mm - 14.97 mm	5.06	36.2 ± 1.2
14.97 mm - 6.01 mm	8.96	37.7 ± 0.7
20.03 mm - 6.01 mm	14.02	37.1 ± 0.5

Table 2.3: *Results for the BN anisotropic ceramic X direction measurements, the average value being* $k = 37.0 \pm 0.8\,\mathrm{W/mK}$.

Sample combination	$L_{virtual}$ (mm)	k ($\mathrm{W/mK}$)
7.94 mm - 4.06 mm	3.88	24.8 ± 0.6
15.02 mm - 7.94 mm	7.08	24.4 ± 0.5
15.02 mm - 4.06 mm	10.96	24.5 ± 0.3

Table 2.4: *Results for the BN anisotropic ceramic Z direction measurements, the average value being* $k = 24.6 \pm 0.5\,\mathrm{W/mK}$.

The effective thermal conductivity values presented in tables 2.3 and 2.4, are for a sample mean temperature of 50 °C. This variable can only be adjusted indirectly through the injected heat flux. This fact constitutes the main drawback of the proposed system, because the thermal conductivity cannot be easily evaluated at any given temperature. Nevertheless, the final experimental setup is simpler and cheaper than for standard solutions and provides direct thermal conductivity readings under usual working conditions in Power Electronics packages.

The results of tables 2.3 and 2.4 show good repeatability among sample combinations; i.e., for the different virtual samples. In addition, good repeatability has also been observed along successive measurements and the variations are within the experimental uncertainty that is always lower than 10% of the measured value. Although measurement on Direction Z matches the nominal value obtained from standard measuring methods (it is only 1 W/ m K below), the measured value for Direction X is much lower, about 6 W/ m K, than the nominal one. As explained above, thermal properties of ceramics can show great variations depending on manufacturing procedures. This is one of the reasons why thermal properties of two samples manufactured separately may show different values. In addition in the case of anisotropic samples, mechanization of samples is very critical. The discrepancy between the measurements performed with standard equipment and the proposed one (done on different particular samples) can be understood as caused by one or various of these factors.

AlSiC Composite Measurement

For the second material, the AlSiC composite, there are two samples available. Their thicknesses are 7.99 mm and 3.07 mm, therefore only one comparative measurement can be performed between them, the virtual sample thickness being $L_{virtual} = 4.92\,\text{mm}$. Two measurements have been carried out, the resulting thermal conductivity calculate from the measured virtual thermal resistance is,

$$\begin{aligned} k &= 170.0 \pm 8.5\,\text{W/ m K} \\ k &= 167.8 \pm 8.3\,\text{W/ m K} \end{aligned} \tag{2.62}$$

With an average value,

$$k = 168.9 \pm 8.4\,\text{W/ m K} \tag{2.63}$$

Both measurements agree each other and also with previous characterization.

AlN Ceramic Measurement

Two samples are available for the third material, the AlN, with thicknesses 6.13 mm and 3.90 mm. Again, only one comparative measurement is performed. The thickness of the virtual sample is $L_{virtual} = 2.23\,\mathrm{mm}$. Also two measurements are performed with the following results:

$$\begin{aligned} k &= 91.7 \pm 5.3\,\mathrm{W/\,m\,K} \\ k &= 91.5 \pm 5.3\,\mathrm{W/\,m\,K} \end{aligned} \tag{2.64}$$

And its average value is,

$$k = 91.6 \pm 5.3\,\mathrm{W/\,m\,K} \tag{2.65}$$

This values agree with the expected results reported by the manufacturer and with the values reported on the literature for AlN ceramics.

2.5.2 Non - Differential Measurement

The measurement procedure is designed to be differential; i.e., to perform two measurements and derive a result from a sample comparison. However, a single sample measurement may be also performed as follows.

First, a sample is measured according to equation (2.8) obtaining a reading for $R_{th\ system\ sample}$, with contributions expressed in equation (2.9). A second measurement placing a thin aluminum sheet from Goodfellow[3] instead a thinner sample is performed.

Aluminum purity is 99% and its thickness 0.015 mm. Thus, the thermal resistance of a $15 \times 20\,\mathrm{mm}$ sample is found to be,

$$R_{th\ AlSheet} = \frac{0.000015\,\mathrm{m}}{237\,\mathrm{W/\,m\,K} \times 0.0003\,\mathrm{m^2}} = 0.00021\,\mathrm{K/\,W} \tag{2.66}$$

Its value is below any measurable thermal resistance with the proposed system. It is in fact ten times lower that the accuracy achieved in thermal resistance measurements. Thus, its contribution to the system thermal resistance can be ignored. Such material is used in order to have two thermal interfaces instead of only one if none interface is used.

The surface roughness of the sample may differ from that of the aluminum sheet. Therefore, if both thermal resistance lectures are differentiated as in equation (2.10) different results are obtained:

[3] http://www.goodfellow.com

$$\begin{aligned} &R_{th\ system\ sample} - R_{th\ system\ Alsheet} = \\ &\quad R_{th\,TIM\ spreader-sample} + R_{th\,TIM\ sample-heatsink} + \\ &\quad R_{th\ sample} - R_{th\,TIM\ spreader-Al} - R_{th\,TIM\ Al-heatsink} \end{aligned} \tag{2.67}$$

The sample must be well polished in order to obtain similar surface roughness to the aluminum sheet if trustable results are wanted.

$$\begin{aligned} R_{th\,TIM\ spreader-sample} &\approx R_{th\,TIM\ spreader-Al} \\ R_{th\,TIM\ sample-heatsink} &\approx R_{th\,TIM\ Al-heatsink} \end{aligned}$$

Thus, the measurement can be carried out with the following thermal resistance to compute the thermal conductivity according to equation (2.4).

$$R_{th\ sample} = R_{th\ system\ sample} - R_{th\ system\ Alsheet} \tag{2.68}$$

The AlN ceramic measured in section 2.5.1 may be measured with this single sample procedure since its surfaces are well polished. Both available samples have been measured two times.
From the 4.96 mm thick sample is obtained:

$$\begin{aligned} k &= 92.0 \pm 2.6\,\mathrm{W/\,m\,K} \\ k &= 93.6 \pm 2.7\,\mathrm{W/\,m\,K} \end{aligned} \tag{2.69}$$

with an average value $k = 92.8 \pm 2.7\,\mathrm{W/\,m\,K}$.
From the 2.59 mm thick sample is obtained:

$$\begin{aligned} k &= 92.6 \pm 4.4\,\mathrm{W/\,m\,K} \\ k &= 95.9 \pm 4.7\,\mathrm{W/\,m\,K} \end{aligned} \tag{2.70}$$

with an average value $k = 94.3 \pm 4.6\,\mathrm{W/\,m\,K}$.

The measured values are consistent with those obtained from the differential method ($k = 91.6 \pm 5.3\,\mathrm{W/\,m\,K}$), therefore the single sample measurement has provided good results for the AlN material.

In general, the roughness of sample surface is higher than that of the aluminum sheet surface. Thus, the normal situation is:

$$\begin{aligned} R_{th\,TIM\ spreader-sample} &> R_{th\,TIM\ spreader-Al} \\ R_{th\,TIM\ sample-heatsink} &> R_{th\,TIM\ Al-heatsink} \end{aligned}$$

If the single sample method is used anyway, higher thermal resistance is obtained in comparison with the real value, thus a lower thermal conductivity is derived. Nevertheless, when higher thermal resistance values are used, less distortion in the final k value is obtained. For extremely high thermal resistance samples, the contact resistance does not affect the k value.

As an illustrative example, single sample measurements have been carried out on the AlSiC composite. Its surfaces are relatively rough, therefore no valid results are expected.
With the 7.99 mm thick sample is obtained:

$$\begin{aligned} k &= 121.5 \pm 3.1\,\mathrm{W/\,m\,K} \\ k &= 120.0 \pm 2.9\,\mathrm{W/\,m\,K} \end{aligned} \tag{2.71}$$

t he average value being $k = 120.7 \pm 3.0\,\mathrm{W/\,m\,K}$.
With the 3.07 mm thick sample is obtained:

$$\begin{aligned} k &= 83.5 \pm 3.5\,\mathrm{W/\,m\,K} \\ k &= 82.5 \pm 3.2\,\mathrm{W/\,m\,K} \end{aligned}$$

the average value being $k = 83.0 \pm 3.4\,\mathrm{W/\,m\,K}$.

These two measurements do not agree with the previous thermal conductivity measured for the AlSiC composite. Moreover, two incongruous results are obtained for the same material. These results evidence a thin sample (low thermal resistance) measurement present major distortion effect due to its surface roughness than a thicker one (high thermal resistance), yielding a higher k.

It is always preferable to perform comparative measurements since results are more reliable. Nevertheless, if only one sample is available, the measurement can be carried out anyway getting an acceptable estimation. However special attention to the surface polishing must be payed.

2.5.3 Thermal Interface Materials Study

Quality of thermal contacts is an important issue in electronics and specially in Power Electronics cooling, in which increasing dissipated power levels require an accurate thermal interface management. Air between two solid surfaces acts as a thermal insulator due to the roughness of any surface. There are actually very few contact points even between two well polished surfaces. Thermal interface materials fill the voids and eliminate air. Therefore, heat is conducted through solids (or also through viscous fluids) reducing drastically the thermal resistance of the contact. Applied pressure acts to reduce even more the amount of remaining air at the interface and reduce the material

thickness and trapped air in soft materials. Reviews of the characteristics of thermal interface materials can be found in [15, 22]. The effect of the applied pressure is studied in [23].

An interesting application for the measurement system is the comparative study of the quality of thermal contact using thermal interface materials. The study can be done at different applied pressures (mounting torque). It has been carried out a series of measurements using eight different materials, including thermal grease, pads and phase change materials from various manufacturers. The measurement consists in measuring the thermal resistance of the system without any power substrate as a sample. Three different pressures are applied with the torque wrench.

Next, the material selection under study is listed together with some relevant information provided by the manufacturers:[4]

Grease compounds

HTSP: Silicone Heat Transfer Compound Plus, manufactured by ELECTROLUBE. With metal oxide (ceramic) powders. Thermal conductivity = 3.0 W/mK.

TTG-S104: From TITAN Computer CO., LTD.Thermal conductivity superior to 7.5 W/mK. Carbon, metal oxide and silver oxide compounds.

RS HeatSink Compound: From RS Components. Thermal conductivity = 0.9 W/mK. With metal oxide.

Phase change

Hi-Flow® 625: From The Bergquist Company. Thermal conductivity not provided. Thickness = 0.127 mm.

Pad

Cho-Therm® 1680: From Chomerics. Adhesive on one side. Thermal conductivity = 0.65 W/mK. Thickness = 0.18 mm.

Cho-Therm® T500: From Chomerics. Thermal conductivity = 2.07 W/mK. Thickness = 0.25 mm.

Sil-Pad® 800: From The Bergquist Company. Thermal conductivity = 1.6 W/mK. Thickness = 0.127 mm.

[4]Data-sheets are collected in the appendix C, from page 184 to 194.

Sil-Pad® A2000: From The Bergquist Company. Thermal conductivity = 3.5 W/mK. Thickness 0.279 mm.

Greases, also named thermal compounds, are often silicone oils containing various fillers such as ceramics or metal oxides in order to increase its thermal conductivity. Pads are polymerized silicone rubbers provided in solid sheets to improve handling. Phase change materials are waxes that melt at a given temperature. They are also provided in sheets and, therefore are more convenient for manufacturing and show thermal performances similar to greases.

Measurements are done at three different mounting torques that result in the following interface pressures:

Torque		Interface pressure
0.5 Nm	$\Rightarrow$	20.9×10^6 Pa
1 Nm	$\Rightarrow$	41.8×10^6 Pa
1.5 Nm	$\Rightarrow$	62.7×10^6 Pa

The results of the study are on tables 2.5 and 2.6 and represented on graphs of figures 2.24 and 2.25. Every value of the measured thermal resistance and its error are obtained as described above. The heat spreader and the heatsink are in contact only through the thermal interface material under test.

Material	R_{th} (°C/W) @ 0.5Nm	@ 1Nm	@ 1.5Nm
HTSP	0.377 ± 0.004	0.356 ± 0.004	0.350 ± 0.004
TTG-S104	0.318 ± 0.003	0.312 ± 0.003	0.309 ± 0.003
RS HeatSink Compound	0.304 ± 0.003	0.299 ± 0.003	0.296 ± 0.003

Table 2.5: *Measured thermal resistance using grease thermal interface material.*

First, it is noticed that greases are better than any sheet interface if optimum heat conduction is aimed. Nevertheless, sheet may be preferable due to other considerations such as reliability (no pump out or migration concerns) or manufacturing advisability.

As explained in section 2.2.1 on page 13, the effective thermal resistance of a thermal interface material is the sum of a bulk resistance $R_{th\,bulk}$ and a contact resistance $R_{th\,C}$ as shown in equation (2.14).

It is derived from the results that the contact resistance component of the effective thermal resistance dominates, being the main difference between the tested greases. It can be observed that HTSP or TTG-S104, which

Material	R_{th} (°C/W) @ 0.5Nm	@ 1Nm	@ 1.5Nm
Cho-Therm® 1680	1.297 ± 0.004	1.281 ± 0.004	1.273 ± 0.004
Cho-Therm® T500	0.672 ± 0.002	0.641 ± 0.002	0.620 ± 0.002
Hi-Flow® 625	1.200 ± 0.003	1.157 ± 0.003	1.139 ± 0.003
Sil-Pad® 800	0.970 ± 0.003	0.901 ± 0.003	0.842 ± 0.003
Sil-Pad® A2000	0.633 ± 0.002	0.632 ± 0.002	0.621 ± 0.002

Table 2.6: *Measured thermal resistance using sheet thermal interface material.*

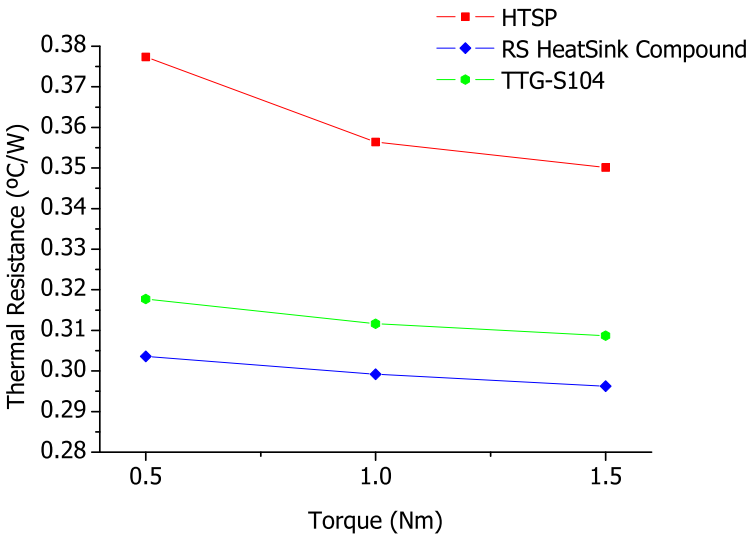


Figure 2.24: *Measured thermal resistance using grease thermal interface material.*

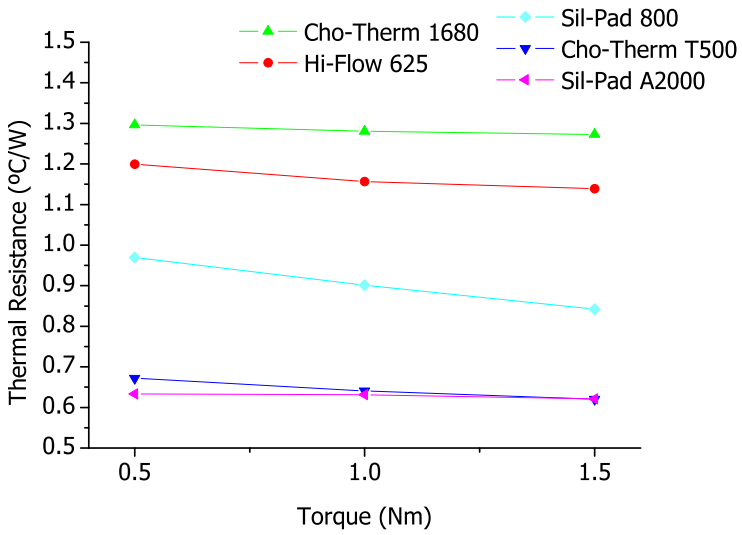


Figure 2.25: *Measured thermal resistance using sheet thermal interface material.*

have higher thermal conductivity than RS HeatSink Compound, show higher thermal resistances. It is due to the fact that the contact resistance of RS HeatSink Compound is lower, yielding its effective thermal resistance the lowest one. Its fluid texture let it to fill better the surface irregularities increasing the effective contact surface. Differences on bulk thickness of the grease layer can also have minor effects.

Bulk thermal resistance of sheet interface materials is higher than in grease compounds due to their higher thicknesses values. Effective thermal resistance variations are due to differences in thicknesses, thermal conductivity and the capability of filling surface irregularities. Although the thermal conductivity of Sil-Pad® A2000 (3.5 W/ m K) is higher than that of Cho-Therm® T500 (2.07 W/ m K) having almost the same thickness, similar results are found. This is due to the fact that Cho-Therm® T500 can fill surface irregularities more effectively than Sil-Pad® A2000.

$R_{th\,bulk}$, $R_{th\,C1}$ and $R_{th\,C2}$ cannot be distinguished from these experimental data. Nevertheless, it can provide valuable comparison information establishing the optimum material and pressure for a given application.

Several improvements can be suggested to optimize the proposed system for this kind of measurements. First, the available torque wrench cannot apply less than 0.5 Nm. Consequently, it may be increased the contact area in order to extend the applied pressure range through lower values. A more

in-deep work may be carried in order to found measurement and analysis techniques for measuring separately the bulk and contact thermal resistances or their thermal impedance.

2.6 Conclusion

An effective thermal conductivity extraction system has been designed and implemented. All details concerning the principle of operation and its implementation have been addressed. The proposed system is simpler, more versatile and cheaper than existing sophisticated equipments. With the proposed system the effective thermal conductivity of a material is deduced from thermal resistance differential measurement of two samples of different thicknesses.

An integral power controller has been designed, as a part of the measurement system, which is optimized to drive a power MOSFET as it is found to be the most suitable heating device. However, other voltage controlled or gated devices may be driven by the proposed system with little adaptation. The proposed configuration enables the circuit to work with up to 70V voltage power source but, with little adaptation, this value can be raised. This part of the system may be suitable for other experimental work needing a heat power source.

Characterization and minimization of the experimental error is a major issue to obtain accurate values. In this sense, practical aspects such as the reduction of heat flow leakage mechanisms or high accuracy temperature measurements have been discussed. It has also been deduced from the uncertainty analysis that the relative experimental error can be kept under 10% if the thermal resistance differential measurement (virtual thermal resistance) is higher than $0.04\,\mathrm{K/W}$. This allows to analyze materials with relatively high thermal conductivities.

The differential measurements performed on three test materials evidence the capability of the proposed measurement system for determining the thermal conductivity of materials in a wide range of values (up to $170\,\mathrm{W/mK}$) with a good accuracy. The extracted thermal conductivity values agree with previous thermal diffusivity and specific heat measurements and the results have shown very good repeatability, with variations within the experimental error. The special conditions that make possible to perform alternative single sample measurements have been stated and illustrative measurements have been also described. A comparative study of a selection of thermal interface materials has been also carried out.

The works described in this chapter have been published in international

conference papers [24, 25] and in a scientific journal [26, 27].

Chapter 3

Thermal Conductivity and Specific Heat Measurements by Parameter Identification (Software Part)

This chapter describes the first part of the parameter identification method developed in this work. The method consists in a computer-based data analysis procedure, described in this chapter, and an experimental setup similar to the one used for the steady state measurements that will be described in detail in the next chapter.

First, the principle of the method and the identification algorithm with equations and concepts are described. However, the source code of the program is not exposed in this chapter, a complete transcription can be found in appendix A. The thermal conduction simulation engine specially developed for this application is also described, and its complete source code transcribed in appendix A. Next, a brief guide of all the source code organization, together with the use of the set of programs that constitute the method can be found after the algorithms explanations. Thus, several validation tests are described. Finally, some conclusions are drawn from the work exposed in this chapter.

3.1 Principle of the Identification Method

The method is devoted to extract the thermal properties of the materials that constitute a thermal system by comparing 3D thermal simulations with experimental data. Based on a parameter identification algorithm, the pro-

posed method determines both the specific heat and the thermal conductivity, even of several materials simultaneously. A well designed 3D thermal simulation is the best approximation to describe the heat conduction physical phenomena if realistic thermal properties of the materials are used.

Generally, the materials of a power package are placed in series constituting a heat path from a heat source to the ambient (see figure 3.1). If heat is produced in the form of a step-wise function and a transient temperature measurement is performed somewhere near the heat source, a heating temperature curve is obtained. This experimental record can be analyzed to extract the thermal parameter values of the constituting materials.

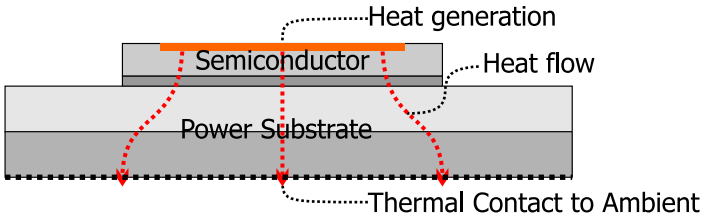


Figure 3.1: *Typical thermal system of a power package.*

The iterative algorithm block diagram of the thermal identification is shown in figure 3.2. The designed computer-based parameter identification algorithm to do this work, performs a thermal simulation of the same system with a given set of initial thermal parameter values. Thus, the error curve can be deduced, comparing the measured curve $T_{meas}(t)$ with the simulated one, $T_{sim}(t)$. The error minimization part performs a set of thermal simulations with the initial parameter values slightly altered in order to obtain the sensitivity of the heating curve to each parameter variation. Based on these sensitivities, the algorithm calculates a convenient combination of thermal parameter modifications in order to get a simulated curve in the next iteration as approximate as possible to the experimental data. The procedure is iterative since a definitive result cannot be reached at the first attempt.

The thermal simulations must be automatically launched by the identification program, introducing the convenient thermal parameter values in each case. It has been considered the use of the FLOTHERM™ 4.2 a thermal simulation software from Flomerics™ Limited. It is possible to launch a thermal simulation from a command line but no arguments modifying the thermal parameter values can be introduced. Hence, this prevents this software to be used and makes necessary the development of a thermal simulation engine specially designed for this application to give thermal conduction simulated results.

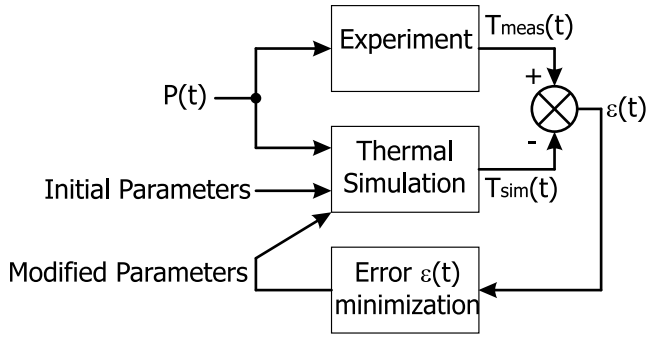


Figure 3.2: *Block diagram of the parameter identification algorithm.*

The set of programs that constitute the computer-based analysis system for the parameter extraction is named *CescThParId 4.1* (Fran**Cesc Th**ermal **Par**ameter **Id**entification). From now on, it will be referred this way or simply as the *identification software*. The system is composed of three programs that perform complementary functions. The core command is the identification program (`identification.exe`) that carries out a whole identification process. Second, a program that establishes the location of the current identification project (`project.exe`). And finally, the simulation program (`simulation.exe`) that performs auxiliary thermal simulations for testing new simulation settings without starting a whole identification process.

The simulation engine is in fact the function contained in the source code file `simule.m`, compiled both inside the identification and the simulation programs. This function is described in detail in section 3.3 and it is available to the parameter identification program. Further details about the source code organization and the folder tree of the *CescThParId 4.1* are exposed further on.

Since during its development and systematic checking the operation of the identification algorithm has been modified in multiple occasions, it is given a version number; e.g. the version described here is the fourth one of the identification algorithm. Within this algorithm, this is the second version (first was 4.0) that has changed fundamental organization mechanisms; i.e., configuration file organization and management that make a difference to the user. As future work will give rise to further improvements, the version number has not been avoided in this work in order to facilitate to keeping track of further changes.

The analysis part of the exposed parameter identification method can analyze data from any experimental setup. Next chapter proposes an exper-

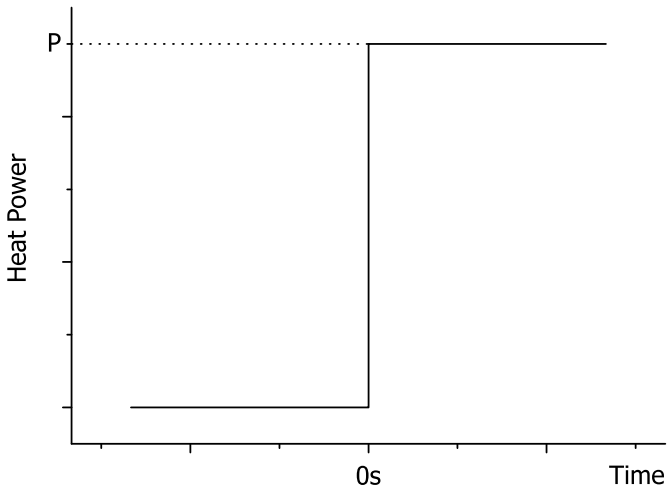


Figure 3.3: *Step-Wise heat power function dissipated in the heat source for the experimental and simulated experiences.*

imental setup but others could be used as experimental data sources for the *CescThParId 4.1* software identification of its parameter values.

3.2 Parameter Identification Program

The starting point of this parameter identification procedure is an experimentally extracted transient temperature record $T_{meas}(t_l)$. This is an array of temperature measurements at a series of time instants t_l, where $l = 1, \ldots, L$ and $t_0 = 0\,\mathrm{s}$. The program is developed with MATLAB® technical language, which is optimized to work with matrices and vectors. Therefore the temperature $T_{meas}(t_l)$ is better expressed in matrix notation as $T_{meas\,l}$, the index l representing the time evolution. The heat power is dissipated as a step-wise function represented in figure 3.3.

Then it is assumed that a 3D simulation engine is available that performs a thermal simulation of the same thermal system provided a given set of thermal properties (thermal conductivity and specific heat) of the different materials. It is obtained then a simulated transient temperature record $T_{sim}(t_m)$ where $m = 1, \ldots, M$ and $t_0 = 0\,\mathrm{s}$. It is noted as $T_{sim\,m}$ from now on.

As the experimental and the simulated temperature curves must be compared, they must be expressed in a common set of time instants. The

sampling period of the measured temperature may not be in general convenient for the simulation. Available sampling periods in experimental measurements, are usually too wide for simulation. Moreover, synchronizing time instants of both measured and simulated heating curves (t_l and t_m) may arise in a needless complex experimental problem. For this reason, the discretization of simulation is independent of the temperature measurement instants having to interpolate curves. The thermal simulation will be carried out with the most convenient sampling period. On the other hand, the experimentally extracted heating curve will be measured with an oscilloscope or any other piece of equipment with a given sampling period. Thus, all simulated temperature curves are interpolated into the time instants of the measured temperature; i.e, $T_{sim}(t_m)$ is interpolated into $T_{sim}(t_l)$, making the time instants t_l to coincide with those of the experimental temperature array. This will be the simulation array used from now on for the heating curve comparison. The Matlab function `interp1` do this work, introducing the condition that the total simulation time T_M must be higher or equal to the total measuring time T_L.

It is assumed that both transient temperature curves, the experimental and the simulated, do not coincide due uniquely to the fact that the thermal properties of materials in the simulation are not appropriate. It is considered that the geometry of the thermal system, the mesh, the heat generation location and the temperature monitoring are correctly introduced into the simulation engine. Thus, the objective is to modify the simulation thermal parameter values in order to reproduce the experimental heating curve as precisely as possible. Once the parameter values optimized, it is supposed that the obtained values are appropriate, yielding further accurate thermal simulations results. This is how the main objective of this work, to find thermal parameter values suitable for realistic simulations, is attained in a direct way. Next, the error minimization procedure making this possible (represented in the block diagram of figure 3.2) is described. It consists of two stages: first, the parameter variations are performed; and second, the optimal parameter modification is found.

Parameter Variations

The first step in modifying the initial set of thermal parameter values from an initial or previous iteration set is to establish the influence of each parameter on the simulated curve. The *CescThParId 4.1* allows the user to select the single or multiple parameters to be simultaneously treated and their values identified. The rest of thermal properties will not be considered in this process but taken as constant. From this point on, the selection of

multiple thermal conductivities and specific heats of different materials are enumerated by the index p and collected in the Par_p array. The thermal conductivity and the specific heat are treated just in the same way in the identification program.

The influence of each thermal parameter on the simulated heating curve form is derived from additional thermal simulations introducing a small variation of its value. A set of thermal simulations introducing a modified thermal conductivity or specific heat value 1% above the original one are carried out, one for each parameter maintaining the rest constant. The two index array $T_{var\,p,m}$ collect the resulting simulated temperature heating curve of the variated parameter p. These curves are simulated with the same time settings than $T_{sim\,m}$, and they are interpolated to the time instants t_l. Thus, it is considered the array $T_{var\,p,l}$ from now on.

The variation curve $DT_{p,l}$ of the thermal parameter p is defined as the subtraction of the temperature resulting from the simulation with the slightly altered parameter p and the initial simulation with unchanged parameter values.

$$DT_{p,l} = T_{var\,p,l} - T_{sim\,l} \tag{3.1}$$

The magnitude $DT_{p,l}$ represents the sensitivity of the thermal simulation for every parameter variation.

It is interesting to know the shape of this sensitivity curve. Typical shapes of the $DT_{p,l}$ curves are represented in figure 3.4, which are extracted from a four material thermal system used as example (a DCB substrate, see figure 3.12). The material one corresponds to the silicon dice, the second to the die attach, the third to the copper and the fourth to the alumina ceramic.

The rise of any specific heat value has its greatest influence during the temperature rising, and disappears as the thermal equilibrium is attained. Higher specific heat values of any material delays the temperature rise since additional energy is needed to heat it, but it does not make any difference in thermal equilibrium conditions. The specific heat is then a dynamical parameter that cannot be identified without a dynamical temperature evolution, this is one of the main reasons for using a dynamical method for extracting its value. Let us have a closer view on the influence of two materials, the third and the fourth (the copper and the alumina), which $DT_{p,l}$ curves are represented in figure 3.5. The graphical representation shows that the curve corresponding to the third material begins the delay in the temperature heating, having a maximum peak before the corresponding to the fourth material. This is due to the fact that the material three is placed closer to the heat source than the number four; i.e., the heat reaches it before the next one.

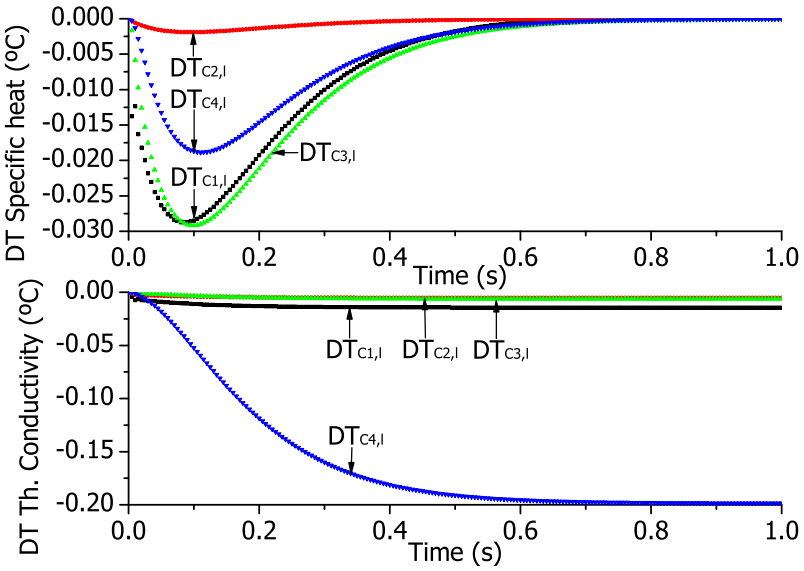


Figure 3.4: *Typical $DT_{p,l}$ curves of a thermal system example (DCB substrate).*

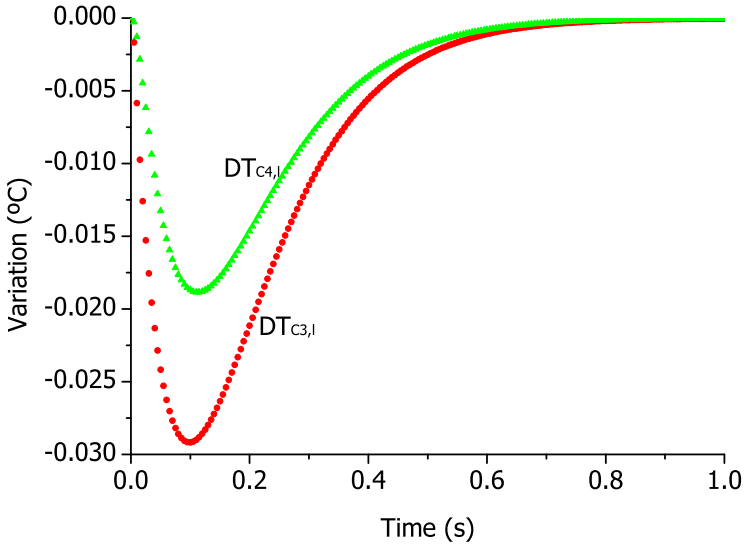


Figure 3.5: *Typical DT curves for specific heat variations. C3 and C4 represent here the index p corresponding to the specific heat of materials three and four, respectively.*

Regarding the thermal conductivity variation, its influence on the temperature appears gradually in time, becoming constant as the thermal equilibrium is reached. The thermal conductivity of the materials is a parameter that influences the thermal resistance of the system. Thus, higher thermal conductivity of any material gives rise to lower equilibrium temperatures. Figure 3.6 shows a more detailed view of the influences of materials three and four. The influence of the third material starts in advance due to the fact that the heat reaches it before than the fourth one. The higher thermal resistance of the material, the higher amplitude of its corresponding $DT_{p,l}$ curve. The third material has a smaller thermal resistance than the fourth one, hence it shows a lower amplitude of its corresponding $DT_{K3,l}$.

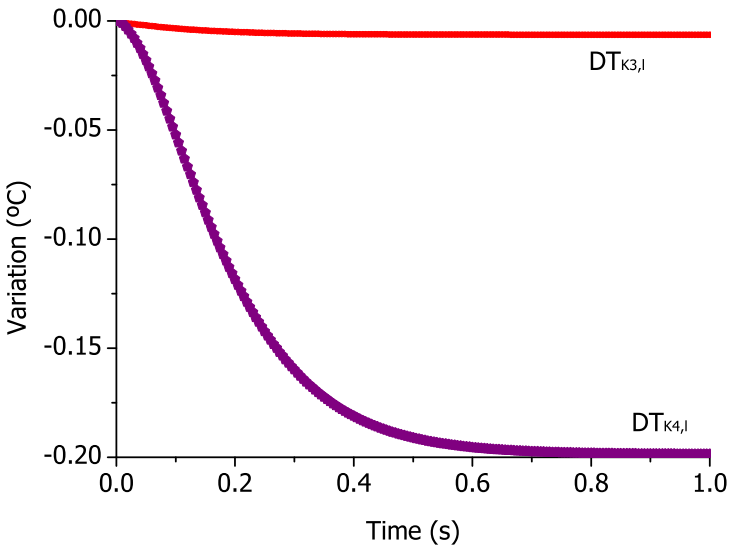


Figure 3.6: *Typical DT curves for thermal conductivity variations. K3 and K4 represent here the index p of the thermal conductivity of materials three and four, respectively.*

Parameter Modification

It has been established that each material has a differentiated contribution to the curve variation represented by $DT_{p,l}$. The objective now is to set a given combination of parameter modifications that give rise to a further thermal simulation result $T'_{sim\,l}$ that might reproduce the measured record:

$$T'_{sim\,l} = T_{meas\,l} \tag{3.2}$$

Let us consider that C_p is an array of index p containing components for each parameter modification as follows:

$$Par'_p = Par_p + \frac{Par_p}{100} C_p \tag{3.3}$$

Now, it is assumed that a linear combination of parameter variations gives rise to a curve variation that is the same linear combination of the sensitivities of every variated parameter; i.e.,

$$T'_{sim\,l} = T_{sim\,l} + DT_{p,l} C_p \tag{3.4}$$

The expression in equation (3.4) is an approximate assumption, valid only for small modifications of the parameter values.

Thus, a given combination of parameter value variations that makes the new simulated temperature to be exactly the measured record must be deduced. It is assumed that there is a unique and optimal C_p accomplishing the following expression:

$$DT_{p,l} C_p = T_{meas\,l} - T_{sim\,l} \tag{3.5}$$

This is a linear equation system, $DT_{p,l}$ being a $P \times L$ matrix, C_p a vector of index p (parameter) and $(T_{meas\,l} - T_{sim\,l})$ is a vector of index l (time). Note that the right-hand side of the equation (3.5) is in fact the error curve since it is the subtraction of the measured and the simulated temperature curves.

MATLAB® solves directly this simultaneous linear equation system with the operator "\", given the following linear system:

```
A*X=Y
```
(3.6)

where `A` is a matrix and `X` and `Y` are vectors. The solution is obtained within the code according to [28]:

```
X=A\Y
```
(3.7)

Therefore, the equation (3.5) is solved like in (3.7) to obtain the best parameter modification combination that approximates the simulated signal to the experimental one.

$$C_p = DT_{p,l} \backslash (T_{meas\,l} - T_{sim\,l}) \tag{3.8}$$

As stated before, the assumption in equation (3.4) is not exact and the resulting variation deduced this way might not be definitive accomplishing (3.2). Nevertheless, the variations performed will get a parameter variation that approximates the simulated curve to the experimental one. By repeating this procedure in an iterative process from the beginning a closer simulated curve with more convenient parameter values is obtained each time. The ideal parameter values will be reached eventually, generating a simulated curve that will be the best fit which reproduces the experimental one.

Other considerations

An automatic control of the process has been also implemented, the identification program leaves the iteration loop when the following is accomplished:

$$\left|Par'_p - Par_p\right| < Tol \quad \text{For all } p \tag{3.9}$$

The tolerance has been implemented to be $Tol = 5 \times 10^{-7}$. This is in general a too restrictive escape condition, the user may need only few decimal positions for the resulting parameter values. Nevertheless, as the result file is always available during the identification process, the user can keep track of the progress of the process and stop it when an acceptable result is attained.

To solve some of the instability problems found during the identification program validation, a restriction on the value of C_p has been implemented. If parameter presents an inappreciable sensitivity to the performed variations, the solution of equation (3.8) could give rise to a component out of the constraints of (3.10).

$$LowLimit < C_p < TopLimit \tag{3.10}$$

The program prevents any parameter to be modified into a value below its 10% or above its 1000% by setting $LowLimit = -90$ and $TopLimit = 1000$.

3.3 Thermal Simulation Engine

This section describes the 3D thermal simulation engine, written into the function `simule.m`, transcribed in the appendix A. The thermal system defined by the user into a geometry file is simulated yielding an array where the temperature vs time values are collected.

First, the heat conduction equation together with all assumptions for the numerical solution are briefly presented. The program only performs simulations of heat conduction in solids, no convection or radiation is considered.

Thus, the discrete numerical solution procedure including the boundary conditions discussion and an introduction to the matrix formulation for its solution with MATLAB® utilities is described.

The integration of this engine within the software part of the thermal parameter identification method is explained in section 3.4.

3.3.1 Heat Conduction Equation and Assumptions

The general differential equation of heat conduction within a solid is the following [12]:

$$\rho c \frac{\partial T(\vec{r},t)}{\partial t} = \nabla[k \nabla T(\vec{r},t)] + q_{gen}(\vec{r},t) \tag{3.11}$$

Where the variables and their dimensions are:

T	the temperature in Kelvin or Celsius degrees.
t	the time (s).
k	the thermal conductivity (W/ m K).
c	the specific heat (J/ kg K).
ρ	the density (kg/ m^3).
$\vec{r}$	the position vector within the body (m).
q_{gen}	the heat generation rate in the medium (W/ m^3).

The equation (3.11) is intended for temperature and space dependent physical properties of the material, k, c and ρ. In case of an anisotropic thermal conductivity of a material, k is a tensorial magnitude as represented in equation (2.5). In this section, the material physical properties are assumed to be isotropic and temperature independent as well. Nevertheless, they are not considered homogeneous as the thermal system to be simulated may be composed of several different materials, each one with different properties. Thus the variable k must be treated as a function of position in the partial derivatives of equation (3.11). For the computer numerical solution, the differential equation of heat conduction is noted as follows:

$$\frac{\partial T}{\partial t} = \frac{1}{\rho c}\left(\frac{\partial k}{\partial x}\frac{\partial T}{\partial x} + k\frac{\partial^2 T}{\partial^2 x} + \frac{\partial k}{\partial y}\frac{\partial T}{\partial y} + k\frac{\partial^2 T}{\partial^2 y} + \frac{\partial k}{\partial z}\frac{\partial T}{\partial z} + k\frac{\partial^2 T}{\partial^2 z}\right) + \frac{q_{gen}}{\rho c} \tag{3.12}$$

Equation (3.12) is then the essential expression for the thermal conduction simulation engine programming, described next.

3.3.2 Discrete Numerical Solution

The differential equation of heat conduction (3.12) arises an initial value diffusive problem. It is chosen to use the *Cranck Nicholson* scheme [29] as an initial setting up of the problem.

For a numerical solution, the equation (3.12) must be discretized both in space and time. Its variables are represented by their values at a discrete set of points equally spaced along the x, y, z and t axis.

$$\begin{array}{ll} x_i = x_0 + i\Delta x & i{=}1,\ \ldots\ , I \\ y_j = y_0 + j\Delta y & j{=}1,\ \ldots\ , J \\ z_k = z_0 + k\Delta z & k{=}1,\ \ldots\ , K \\ t_n = t_0 + n\Delta t & n{=}1,\ \ldots\ , N \end{array} \tag{3.13}$$

where Δx, Δy, Δz and Δt are the grid spacing along x, y, z and t, respectively.

The index n corresponds to time, and indexes i, j and k to the space directions x, y and z respectively. From hereafter, the temperature field $T(t_n, x_i, y_j, z_k)$ is written as $T^n_{i,j,k}$. The other variables of the equation (ρ, k, c and q_{gen}) are considered time independent parameters, thus no time index is used. The discrete form of every variable of the heat conduction equation is indicated in table 3.1.

Variable		Discrete variable
$T(t_n, x_i, y_j, z_k)$	$\Rightarrow$	$T^n_{i,j,k}$
$k(x_i, y_j, z_k)$	$\Rightarrow$	$k_{i,j,k}$
$c(x_i, y_j, z_k)$	$\Rightarrow$	$c_{i,j,k}$
$\rho(x_i, y_j, z_k)$	$\Rightarrow$	$\rho_{i,j,k}$
$q_{gen}(x_i, y_j, z_k)$	$\Rightarrow$	$q_{gen\,i,j,k}$

Table 3.1: *Discrete representation of the equation* (3.12) *variables for the finite-difference representation.*

The equation (3.12) is solved by the *finite-difference* method, which substitutes the derivative terms by a representation in the discretized axis. The partial differences in the x axis of the function $T(\vec{r}, t)$ and of the thermal conductivity k are represented as:

$$\left.\frac{\partial T}{\partial x}\right|_{y,z,t} \Rightarrow \frac{T^n_{i+1,j,k} - T^n_{i-1,j,k}}{2\Delta x} \tag{3.14}$$

$$\left.\frac{\partial k}{\partial x}\right|_{y,z} \Rightarrow \frac{k_{i+1,j,k} - k_{i-1,j,k}}{2\Delta x} \tag{3.15}$$

The second partial difference in the x axis of the function $T(\vec{r}, t)$ is represented as:

$$\left.\frac{\partial^2 T}{\partial^2 x}\right|_{y,z,t} \Rightarrow \frac{T^n_{i+1,j,k} - 2T^n_{i,j,k} + T^n_{i-1,j,k}}{(\Delta x)^2} \tag{3.16}$$

And similarly for other derivative terms of equation (3.12), $\partial k/\partial y$, $\partial k/\partial z$, $\partial T/\partial y$, $\partial T/\partial z$, $\partial^2 T/\partial^2 y$ and $\partial^2 T/\partial^2 z$. This finite-difference approximation is second order accurate in space.

An obvious choice for representing the time derivative term is called the *Forward Euler Differencing.*

$$\left.\frac{\partial T}{\partial t}\right|_{x,y,z} \Rightarrow \frac{T^{n+1}_{i,j,k} - T^n_{i,j,k}}{\Delta t} \tag{3.17}$$

This finite-differencing approximation is only first-order accurate in time. However, it has the advantage that temperature, at time-step $n+1$, can be calculated in terms of known quantities at time-step n. The resulting finite-difference approximation to equation (3.12) is called the *Forward Time Centered Space Scheme* (FTCS).

$$\begin{aligned}\frac{T^{n+1}_{i,j,k} - T^n_{i,j,k}}{\Delta t} = \frac{1}{\rho_{i,j,k} c_{i,j,k}} \Bigg[& \frac{k_{i+1,j,k} - k_{i-1,j,k}}{2\Delta x} \frac{T^n_{i+1,j,k} - T^n_{i-1,j,k}}{2\Delta x} + \\ & + k_{i,j,k} \frac{T^n_{i+1,j,k} - 2T^n_{i,j,k} + T^n_{i-1,j,k}}{(\Delta x)^2} + \frac{k_{i,j+1,k} - k_{i,j-1,k}}{2\Delta y} \frac{T^n_{i,j+1,k} - T^n_{i,j-1,k}}{2\Delta y} + \\ & + k_{i,j,k} \frac{T^n_{i,j+1,k} - 2T^n_{i,j,k} + T^n_{i,j-1,k}}{(\Delta y)^2} + \frac{k_{i,j,k+1} - k_{i,j,k-1}}{2\Delta z} \frac{T^n_{i,j,k+1} - T^n_{i,j,k-1}}{2\Delta z} + \\ & + k_{i,j,k} \frac{T^n_{i,j,k+1} - 2T^n_{i,j,k} + T^n_{i,j,k-1}}{(\Delta z)^2} \Bigg] + \frac{q_{gen\, i,j,k}}{\rho_{i,j,k} c_{i,j,k}}\end{aligned} \tag{3.18}$$

Equation (3.18) can easily be rearranged to express $T^{n+1}_{i,j,k}$ in terms of previous time-step temperature $T^n_{i,j,k}$. The algorithm is easy to derive, and takes little computer storage and executes quickly. The FTCS representation is a *fully explicit* scheme; i.e., $T^{n+1}_{i,j,k}$ can be calculated explicitly from the quantities that are already known at time-step n. It is also an example of a *single-level scheme* since only values at time-step n have to be stored in memory to find values at time-step $n+1$. Nevertheless, the stability

analysis of this algorithm [29] leads to a criterion that usually gives rise to a prohibitive required number of time-steps in most diffusive problems.

The alternative is the *fully implicit* (or backward time) scheme. It is exactly like the FTCS but the spatial derivatives are evaluated at time-step $n+1$.

$$\begin{aligned}\frac{T_{i,j,k}^{n+1}-T_{i,j,k}^{n}}{\Delta t} = \frac{1}{\rho_{i,j,k}c_{i,j,k}}\Bigg[&\frac{k_{i+1,j,k}-k_{i-1,j,k}}{2\Delta x}\frac{T_{i+1,j,k}^{n+1}-T_{i-1,j,k}^{n+1}}{2\Delta x}+\\ &+k_{i,j,k}\frac{T_{i+1,j,k}^{n+1}-2T_{i,j,k}^{n+1}+T_{i-1,j,k}^{n+1}}{(\Delta x)^2}+\frac{k_{i,j+1,k}-k_{i,j-1,k}}{2\Delta y}\frac{T_{i,j+1,k}^{n+1}-T_{i,j-1,k}^{n+1}}{2\Delta y}+\\ &+k_{i,j,k}\frac{T_{i,j+1,k}^{n+1}-2T_{i,j,k}^{n+1}+T_{i,j-1,k}^{n+1}}{(\Delta y)^2}+\frac{k_{i,j,k+1}-k_{i,j,k-1}}{2\Delta z}\frac{T_{i,j,k+1}^{n+1}-T_{i,j,k-1}^{n+1}}{2\Delta z}+\\ &+k_{i,j,k}\frac{T_{i,j,k+1}^{n+1}-2T_{i,j,k}^{n+1}+T_{i,j,k-1}^{n+1}}{(\Delta z)^2}\Bigg]+\frac{q_{gen\,i,j,k}}{\rho_{i,j,k}c_{i,j,k}}\end{aligned} \tag{3.19}$$

To solve equation (3.19), a set of simultaneous linear equations for the temperature at time-step $n+1$ have to be solved at each time-step. The fully implicit method is more adequate for a diffusive problem as it is unconditionally stable for any step-size Δt [29]. However, it is still only first order accurate in time.

The fully explicit and fully implicit schemes have been exposed so far to better understand the definitive solution scheme. The Cranck-Nicholson method combines the stability of an explicit scheme and it is second order accurate both in space and in time. The algorithm is constructed simply from the average of the explicit and implicit schemes. Thus, the finite-difference version of equation (3.12) implementing the Cranck-Nicholson scheme is the following:

$$\frac{T^{n+1}_{i,j,k} - T^{n}_{i,j,k}}{\Delta t} = \frac{1}{\rho_{i,j,k} c_{i,j,k}} \frac{1}{2} \left[\frac{k_{i+1,j,k} - k_{i-1,j,k}}{2\Delta x} \frac{T^{n}_{i+1,j,k} - T^{n}_{i-1,j,k}}{2\Delta x} + \right.$$
$$+ k_{i,j,k} \frac{T^{n}_{i+1,j,k} - 2T^{n}_{i,j,k} + T^{n}_{i-1,j,k}}{(\Delta x)^2} + \frac{k_{i,j+1,k} - k_{i,j-1,k}}{2\Delta y} \frac{T^{n}_{i,j+1,k} - T^{n}_{i,j-1,k}}{2\Delta y} +$$
$$+ k_{i,j,k} \frac{T^{n}_{i,j+1,k} - 2T^{n}_{i,j,k} + T^{n}_{i,j-1,k}}{(\Delta y)^2} + \frac{k_{i,j,k+1} - k_{i,j,k-1}}{2\Delta z} \frac{T^{n}_{i,j,k+1} - T^{n}_{i,j,k-1}}{2\Delta z} +$$
$$+ k_{i,j,k} \frac{T^{n}_{i,j,k+1} - 2T^{n}_{i,j,k} + T^{n}_{i,j,k-1}}{(\Delta z)^2} + \frac{k_{i+1,j,k} - k_{i-1,j,k}}{2\Delta x} \frac{T^{n+1}_{i+1,j,k} - T^{n+1}_{i-1,j,k}}{2\Delta x} +$$
$$+ k_{i,j,k} \frac{T^{n+1}_{i+1,j,k} - 2T^{n+1}_{i,j,k} + T^{n+1}_{i-1,j,k}}{(\Delta x)^2} + \frac{k_{i,j+1,k} - k_{i,j-1,k}}{2\Delta y} \frac{T^{n+1}_{i,j+1,k} - T^{n+1}_{i,j-1,k}}{2\Delta y} +$$
$$+ k_{i,j,k} \frac{T^{n+1}_{i,j+1,k} - 2T^{n+1}_{i,j,k} + T^{n+1}_{i,j-1,k}}{(\Delta y)^2} + \frac{k_{i,j,k+1} - k_{i,j,k-1}}{2\Delta z} \frac{T^{n+1}_{i,j,k+1} - T^{n+1}_{i,j,k-1}}{2\Delta z} +$$
$$\left. + k_{i,j,k} \frac{T^{n+1}_{i,j,k+1} - 2T^{n+1}_{i,j,k} + T^{n+1}_{i,j,k-1}}{(\Delta z)^2} \right] + \frac{q_{gen\,i,j,k}}{\rho_{i,j,k} c_{i,j,k}} \tag{3.20}$$

Left and right-hand sides of the equation are centered at time-step $n + 1/2$, so the method is second order accurate in time as well as second order accurate in space.

The problem arises in solving the coupled linear equation as set in equation (3.20). The solution would involve enormous and very sparse matrices that may exhaust the computer resources and would be excessive time-consuming. Therefore, the equations will be easier to solve applying the *Alternating-Direction Implicit* method (ADI) [29]. The idea is to divide the solution of the problem into three sub-steps. In each sub-step one dimension solution is treated implicitly with the Cranck-Nicholson scheme.

The equation (3.20) is divided into the following three finite-difference expressions. First the one dimension Cranck-Nicholson scheme of the heat conduction along the z direction [1]:

[1]Heat conduction along the z direction: $\frac{\partial T}{\partial t} = \frac{1}{\rho c} \left(\frac{\partial k}{\partial z} \frac{\partial T}{\partial z} + k \frac{\partial^2 T}{\partial^2 z} \right) + \frac{q_{gen}}{\rho c}$

$$\begin{aligned}\frac{T_{i,j,k}^{n+1}-T_{i,j,k}^{n}}{\Delta t} &= \frac{1}{2\rho_{i,j,k}c_{i,j,k}}\left[\frac{k_{i,j,k+1}-k_{i,j,k-1}}{2\Delta z}\frac{T_{i,j,k+1}^{n+1}-T_{i,j,k-1}^{n+1}}{2\Delta z}+\right.\\ &+k_{i,j,k}\frac{T_{i,j,k+1}^{n+1}-2T_{i,j,k}^{n+1}+T_{i,j,k-1}^{n+1}}{(\Delta z)^2}+\\ &+\frac{k_{i,j,k+1}-k_{i,j,k-1}}{2\Delta z}\frac{T_{i,j,k+1}^{n}-T_{i,j,k-1}^{n}}{2\Delta z}+\\ &\left.+k_{i,j,k}\frac{T_{i,j,k+1}^{n}-2T_{i,j,k}^{n}+T_{i,j,k-1}^{n}}{(\Delta z)^2}\right]+\frac{q_{gen\,i,j,k}}{\rho_{i,j,k}c_{i,j,k}} \qquad (3.21)\end{aligned}$$

Thus, the one dimension Cranck-Nicholson scheme along the y direction, as the finite difference version of the heat conduction along the y axis [2] is:

$$\begin{aligned}\frac{T_{i,j,k}^{n+1}-T_{i,j,k}^{n}}{\Delta t} &= \frac{1}{2\rho_{i,j,k}c_{i,j,k}}\left[\frac{k_{i,j+1,k}-k_{i,j-1,k}}{2\Delta y}\frac{T_{i,j+1,k}^{n+1}-T_{i,j-1,k}^{n+1}}{2\Delta y}+\right.\\ &+k_{i,j,k}\frac{T_{i,j+1,k}^{n+1}-2T_{i,j,k}^{n+1}+T_{i,j-1,k}^{n+1}}{(\Delta y)^2}+\\ &+\frac{k_{i,j+1,k}-k_{i,j-1,k}}{2\Delta y}\frac{T_{i,j+1,k}^{n}-T_{i,j-1,k}^{n}}{2\Delta y}+\\ &\left.+k_{i,j,k}\frac{T_{i,j+1,k}^{n}-2T_{i,j,k}^{n}+T_{i,j-1,k}^{n}}{(\Delta y)^2}\right] \qquad (3.22)\end{aligned}$$

And finally the one dimension Cranck-Nicholson scheme along the x direction [3] is:

$$\begin{aligned}\frac{T_{i,j,k}^{n+1}-T_{i,j,k}^{n}}{\Delta t} &= \frac{1}{2\rho_{i,j,k}c_{i,j,k}}\left[\frac{k_{i+1,j,k}-k_{i-1,j,k}}{2\Delta x}\frac{T_{i+1,j,k}^{n+1}-T_{i-1,j,k}^{n+1}}{2\Delta x}+\right.\\ &+k_{i,j,k}\frac{T_{i+1,j,k}^{n+1}-2T_{i,j,k}^{n+1}+T_{i-1,j,k}^{n+1}}{(\Delta x)^2}+\\ &+\frac{k_{i+1,j,k}-k_{i-1,j,k}}{2\Delta x}\frac{T_{i+1,j,k}^{n}-T_{i-1,j,k}^{n}}{2\Delta x}+\\ &\left.+k_{i,j,k}\frac{T_{i+1,j,k}^{n}-2T_{i,j,k}^{n}+T_{i-1,j,k}^{n}}{(\Delta x)^2}\right] \qquad (3.23)\end{aligned}$$

[2] Heat conduction along the y direction: $\frac{\partial T}{\partial t}=\frac{1}{\rho c}\left(\frac{\partial k}{\partial y}\frac{\partial T}{\partial y}+k\frac{\partial^2 T}{\partial^2 y}\right)$

[3] Heat conduction along the x direction: $\frac{\partial T}{\partial t}=\frac{1}{\rho c}\left(\frac{\partial k}{\partial x}\frac{\partial T}{\partial x}+k\frac{\partial^2 T}{\partial^2 x}\right)$

These expressions are solved sequentially, each of them taking the result of the previous one as its initial value. The heat generation term must be threated once and it has been chosen to be included only along the z direction sub-step. The thermal simulation engine solves the heat conduction equation at each time step with the ADI method in the order presented here; i.e., first the z direction, followed by y, and finally x. In the geometry file it is also included the initial temperature of the whole system $T_{initial}$.

$$T_{i,j,k}^{n=1} = T_{initial} \quad \text{for all } i,j \text{ and } k \tag{3.24}$$

From now on, the solution process is focused on single dimension solutions.

3.3.3 Boundaries

The solution of the heat conduction consists in the solution of simultaneous linear equations stated in equations (3.21), (3.22) and (3.23). Nevertheless, each direction needs two extra equations defining the boundary conditions at the beginning and at the end of each axis solution domain. The simulation engine implements automatically the boundary conditions, since it has been specially conceived for the simulation of a heat source mounted in a material or a stack of materials and attached to a heat-sink. This typical model is represented in figure 3.7. Heat pass through the system and it is finally evacuated through a heatsink represented as a fixed temperature boundary.

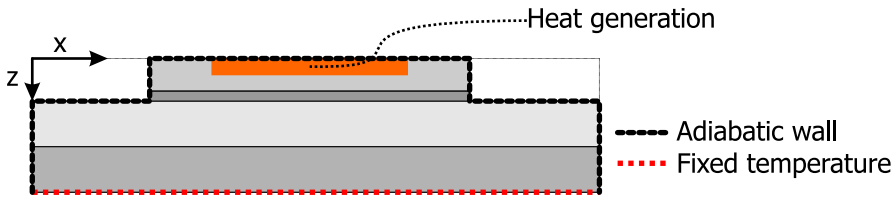


Figure 3.7: *Typical 3D thermal system to be simulated with the simulation engine. An adiabatic wall is implemented at the end of the material medium, except at the end of the z domain that implements a constant temperature boundary.*

Adiabatic surfaces

The Fourier's law in equation (2.2) describes the dependence between the temperature gradient and the heat flow within a thermal conductivity k medium. The z component of the heat flow is imposed to be zero, implementing

an adiabatic boundary:

$$q_z = -k\frac{\partial T}{\partial z} = 0 \tag{3.25}$$

Therefore:

$$\frac{\partial T}{\partial z} = 0 \tag{3.26}$$

Thus, in a finite-difference representation from (3.14) it is derived that:

$$T^n_{i,j,k+1} - T^n_{i,j,k-1} = 0 \tag{3.27}$$

at any time-step n. Applying this to equation (3.21) at point $k = 1$ and rearranging the temperature terms:

$$\begin{aligned}\left(\frac{2\rho_{i,j,1}c_{i,j,1}(\Delta z)^2}{\Delta t} + 2k_{i,j,1}\right) T^{n+1}_{i,j,1} - 2k_{i,j,1}T^{n+1}_{i,j,2} = \\ = \left(\frac{2\rho_{i,j,1}c_{i,j,1}(\Delta z)^2}{\Delta t} - 2k_{i,j,1}\right) T^{n}_{i,j,1} + 2k_{i,j,1}T^{n}_{i,j,2} + \\ +2(\Delta z)^2 q_{gen\,i,j,1}\end{aligned} \tag{3.28}$$

The adiabatic surface in the y axis is treated in the same way. The y component of heat flow is null, and it is deduced a condition similar to (3.27):

$$T_{i,j+1,k} - T_{i,j-1,k} = 0 \tag{3.29}$$

Then, applying this to equation (3.22) at point $j = 1$ and rearranging it is obtained:

$$\begin{aligned}\left(\frac{2\rho_{i,1,k}c_{i,1,k}(\Delta y)^2}{\Delta t} + 2k_{i,1,k}\right) T^{n+1}_{i,1,k} - 2k_{i,1,k}T^{n+1}_{i,2,k} = \\ = \left(\frac{2\rho_{i,1,k}c_{i,1,k}(\Delta y)^2}{\Delta t} - 2k_{i,1,k}\right) T^{n}_{i,1,k} + 2k_{i,1,k}T^{n}_{i,2,k}\end{aligned} \tag{3.30}$$

An adiabatic boundary is also applied to the end of the y axis where $j = J$. At this point:

$$\begin{aligned}\left(\frac{2\rho_{i,J,k}c_{i,J,k}(\Delta y)^2}{\Delta t} + 2k_{i,J,k}\right) T^{n+1}_{i,J,k} - 2k_{i,J,k}T^{n+1}_{i,J-1,k} = \\ = \left(\frac{2\rho_{i,J,k}c_{i,J,k}(\Delta y)^2}{\Delta t} - 2k_{i,J,k}\right) T^{n}_{i,J,k} + 2k_{i,J,k}T^{n}_{i,J-1,k}\end{aligned} \tag{3.31}$$

Similarly, along the x direction expressions similar to equations (3.30) and (3.31) are obtained simply replacing index j by i.

Constant temperature

Constant temperature boundary is implemented at the boundary of the z axis. As it is assumed that the thermal system simulation begins in thermal equilibrium, the imposed temperature at this boundary is the initial one $T_{initial}$. The heat generation term has to be null at the points where this boundary condition is imposed. However, the user can implement a heat source at the bottom of the simulation domain in the z direction; i.e., at node $k = K$. To avoid this eventual problem, an extra node is added by the simulation engine, the $k = K + 1$ node, where the constant temperature boundary is imposed:

$$T^{n+1}_{i,j,K+1} = T^{n}_{i,j,K+1} \tag{3.32}$$

3.3.4 Matrix treatment Implementation

The simulation engine is written in MATLAB®, a technical computing language that works optimally with matrices [28]. As the problem is an initial value problem, temperature at a given time-step T^n are known and the solution process consists in finding the temperature in the next time-step T^{n+1}.

Z Direction

Once i and j indexes are chosen, the problem is one-dimensional considering only the k index. Therefore, indexes i and j can be obviated in the following equations. Hereafter, the magnitudes expressed with this single index in the time sub-step and in the particular choice of i an j, are considered as a vector. The magnitudes T^n_k, ρ_k, c_k, k_k and $q_{gen\,k}$ are vectors with index k.

A re-arrangement of the Cranck-Nicholson scheme along the direction z of (3.21) yields the following:

$$\begin{aligned}
&\left(\frac{2\rho_k c_k(\Delta z)^2}{\Delta t} + 2k_k\right) T^{n+1}_k + \left(-\frac{k_{k+1} - k_{k-1}}{4} - k_k\right) T^{n+1}_{k+1} + \\
&\quad + \left(\frac{k_{k+1} - k_{k-1}}{4} - k_k\right) T^{n+1}_{k-1} = \left(\frac{2\rho_k c_k(\Delta z)^2}{\Delta t} - 2k_k\right) T^{n}_k + \\
&\quad + \left(\frac{k_{k+1} - k_{k-1}}{4} + k_k\right) T^{n}_{k+1} + \left(-\frac{k_{k+1} - k_{k-1}}{4} + k_k\right) T^{n}_{k-1} + \\
&\quad + 2(\Delta z)^2 q_{gen\,k}
\end{aligned} \tag{3.33}$$

Then, equation (3.33) is in fact:

$$A_{k,l}T_l^{n+1} = B_{k,m}T_m^n + Q_k \tag{3.34}$$

where A and B are matrices. Temperatures T_l^{n+1} and T_m^n are the temperature vectors corresponding to the temperature along $l, m = 1, \ldots, K+1$ at time steps $n+1$ and n, respectively.

The elements of the matrices A and B for $k = 2, \ldots, K$ are defined as:

$$\begin{array}{ll} A_{k,k} = \frac{2\rho_k c_k (\Delta z)^2}{\Delta t} + 2k_k & B_{k,k} = \frac{2\rho_k c_k (\Delta z)^2}{\Delta t} - 2k_k \\ A_{k,k+1} = -\frac{k_{k+1}-k_{k-1}}{4} - k_k & B_{k,k+1} = \frac{k_{k+1}-k_{k-1}}{4} + k_k \\ A_{k,k-1} = \frac{k_{k+1}-k_{k-1}}{4} - k_k & B_{k,k-1} = -\frac{k_{k+1}-k_{k-1}}{4} + k_k \end{array} \tag{3.35}$$

According to the expressions (3.28) and (3.32) obtained from the boundary conditions along the direction z (adiabatic at $k = 1$ and constant temperature at $k = K+1$), the rest of the elements of the matrices A and B are:

$$\begin{array}{ll} A_{1,1} = \frac{2\rho_1 c_1 (\Delta z)^2}{\Delta t} + 2k_1 & B_{1,1} = \frac{2\rho_1 c_1 (\Delta z)^2}{\Delta t} - 2k_1 \\ A_{1,2} = -2k_1 & B_{1,2} = 2k_1 \\ A_{K+1,K+1} = 1 & B_{K+1,K+1} = 1 \end{array} \tag{3.36}$$

All other elements of matrices A and B are zero.

The Q vector of equation (3.34) is:

$$Q_k = 2(\Delta z)^2 q_{gen\,k} \tag{3.37}$$

for $k = 1 \ldots K$, and

$$Q_{K+1} = 0 \tag{3.38}$$

at the added node.

Each sub-step heat conduction equation can be reduced to a set of simultaneous linear equations expressed with the following matrix notation.

$$A_{k,l}T_l^{n+1} = V_k \tag{3.39}$$

This is a linear system like in expression (3.6) and it can be solved using the MATLAB® operator "\" as in (3.7). Consequently, finding the values $A_{k,l}$ and V_k is the main concern of this section.

The right-hand side of expression (3.34) can be expressed in terms of already known values (thermal properties, grid, and temperature at previous time-step T^n). It can be evaluated obtaining the V_k vector of equation (3.39).

$$V_k = B_{k,m} T_m^n + Q_k \tag{3.40}$$

All necessary elements to solve the heat conduction along the direction z have been described so far. The software code collected in appendix A contains the implementation of this procedure in detail.

Y and X directions

A re-arrangement of the Cranck-Nicholson scheme along the y direction of (3.22) results in the following:

$$\begin{aligned}\left(\frac{2\rho_j c_j (\Delta y)^2}{\Delta t} + 2k_j\right) T_j^{n+1} + \left(-\frac{k_{j+1} - k_{j-1}}{4} - k_j\right) T_{j+1}^{n+1} + \\ \left(\frac{k_{j+1} - k_{j-1}}{4} - k_j\right) T_{j-1}^{n+1} = \left(\frac{2\rho_j c_j (\Delta y)^2}{\Delta t} - 2k_j\right) T_j^n + \\ \left(\frac{k_{j+1} - k_{j-1}}{4} + k_j\right) T_{j+1}^n + \left(-\frac{k_{j+1} - k_{j-1}}{4} + k_j\right) T_{j-1}^n\end{aligned} \tag{3.41}$$

Then, the equation (3.41) can be considered in fact as:

$$A_{j,l} T_l^{n+1} = B_{j,m} T_m^n \tag{3.42}$$

The elements of the particular matrices A and B solving the y direction for $j = 2, \ldots, J-1$ being:

$$\begin{array}{ll} A_{j,j} = \frac{2\rho_j c_j (\Delta y)^2}{\Delta t} + 2k_j & B_{j,j} = \frac{2\rho_j c_j (\Delta y)^2}{\Delta t} - 2k_j \\ A_{j,j+1} = -\frac{k_{j+1} - k_{j-1}}{4} - k_j & B_{j,j+1} = \frac{k_{j+1} - k_{j-1}}{4} + k_j \\ A_{j,j-1} = \frac{k_{j+1} - k_{j-1}}{4} - k_j & B_{j,j-1} = -\frac{k_{j+1} - k_{j-1}}{4} + k_j \end{array} \tag{3.43}$$

According to expressions (3.30) and (3.31) obtained from the boundary conditions along the direction y (adiabatic conditions at $j = 1$ and $j = J$) the rest of the A and B matrix elements are:

$$\begin{array}{ll} A_{1,1} = \frac{2\rho_1 c_1 (\Delta y)^2}{\Delta t} + 2k_J & B_{1,1} = \frac{2\rho_1 c_1 (\Delta y)^2}{\Delta t} - 2k_1 \\ A_{1,2} = -2k_1 & B_{1,2} = 2k_1 \\ A_{J,J} = \frac{2\rho_J c_J (\Delta y)^2}{\Delta t} + 2k_J & B_{J,J} = \frac{2\rho_J c_J (\Delta y)^2}{\Delta t} - 2k_J \\ A_{J,J-1} = -2k_J & B_{J,J-1} = -2k_J \end{array} \tag{3.44}$$

Note that all elements in expression (3.43) have the same form than those along the direction z. Solution along directions y and x is different from the z at the boundary conditions. Moreover, simulation along y and x does not involve the heat generation term.

Similarly to equation (3.39), the solution along the y direction in matrix notation is reached just solving the following equation with the operator "\":

$$A_{j,l}T_l^{n+1} = V_j \tag{3.45}$$

The right-hand side of equation (3.42) is deduced in terms of already known quantities. Therefore, the vector of expression (3.45) for the extraction of temperature at next time sub-step is:

$$V_j = B_{j,m}T_m^n \tag{3.46}$$

The procedure for the solution of the heat conduction along the x axis is similar to that along y. The index to be treated in this case is i, all the equations being of the same form.

3.4 Application implementation

Three MATLAB® functions have been compiled into three MS-DOS executable commands available to the user: `project`, `identification` and `simulation`. Each of the three programs takes the name of the file containing the main function. Some routines of large functions have been written into sub-functions contained in separate source code files. All the source code is contained in nine files that are transcribed in the appendix A.

Inside the application root folder, the executables and the user files are not mixed, see figure 3.8. The folder `binaries` contains the executables (`.exe`), a collection of required MATLAB® dynamic link libraries (`.dll`) and the `info` file. The folder `projects` contains all the user projects, each in a separate folder, e.g. `Project1` or `Project2`. The project files are the geometry file, the identification configuration file and the experimental data file created by the user as well as the result and temporary files created by the programs `identification` and `simulation`.

Next, a brief introduction to the work of the three programs is given. All details of their implementation can be found in the code transcription in the appendix A. The following descriptions only give a brief guide to the use of the programs and to the contents of the software code.

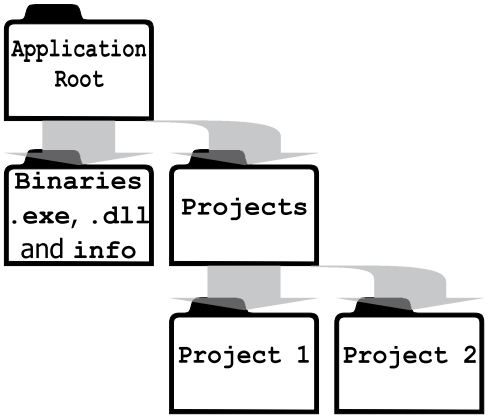


Figure 3.8: *Folder tree of the thermal parameter identification software.*

3.4.1 `project.exe`

The command line:

```
project ProjectPath
```

establishes the path of the current project folder. It writes the path into a file named `info` inside the `binaries` folder, where the parameter identification and simulation programs will find it when necessary. This is the first command the user must use in order to set first the location to find the project files. The argument `ProjectPath` is the path to the current project folder relative to the application root, e.g. '`Projects/Project1`'.

The source code of this program is entirely contained in the `project.m` file (page 145).

3.4.2 `identification.exe`

The command line:

```
identification IdConfigFile LogFile
```

performs a complete parameter identification of a given project previously defined by the user. All needed files must be placed inside the same project folder. The argument `IdConfigFile` is the name of the identification configuration file containing all necessary information to perform the identification process. The optional argument `LogFile` is the name of a log file, where selected information about the process evolution and eventual error messages are written. If not provided, the messages are printed on the screen.

The main function of this program is written in the `identification.m` file (page 136) and makes use of the following sub-functions:

`idfileread`, inside the file `idfileread.m` (page 166). Reads the identification configuration file and performs some interpretations to detect errors. Returns all the collected information to the main function.

`simfileread`, inside the file `simfileread.m` (page 156). Reads the geometry file and performs some interpretations to detect errors. Returns all the readed information to the main file.

`expfileread`, inside the file `expfileread.m` (page 170). Reads the experimental data file, provides the data in a proper chronological order and returns an array.

`simule`, inside the file `simule.m` (page 146). Performs the 3D thermal simulation. In its turn, it calls also the function `simfileread` to read the particular geometry and parameter values to be used. This is the simulation engine described in section 3.3.

`varsimfilewrite`, inside the file `varsimfilewrite.m` (page 171). Writes a copy of the current geometry file of the identification iteration with an altered thermal property of a given material.

`resultwrite`, inside the file `resultwrite.m` (page 173). Writes the resulting parameter values in the resulting file. Also writes a geometry file with current parameter values that will be used in the next identification iteration.

3.4.3 `simulation.exe`

The command line:

```
simulation GeometryFile ResultFile LogFile
```

performs a 3D thermal simulation of a given thermal system defined in a geometry file. The argument `GeometryFile` is the name of the geometry file containing all specifications, which define the thermal simulation, `ResultFile` is the name of the file where the temperature result is written and the optional argument `LogFile` is the name of a log file where several informations about the simulation process and eventual error messages are written. All these files are searched and written in the project folder defined previously.

This program makes use of the simulation engine function `simule`, the core function of the thermal simulation program, which allows performing a thermal simulation independent from any parameter identification process.

Any valid geometry file can be simulated in this way, an example of the use of this auxiliary tool are the validation identifications carried out in section 3.5.2.

3.5 Software Validation Tests

In order to test the self-consistence and the accuracy of the computer-based part of the thermal parameter identification method, various tests have been carried out. Next, the thermal simulation engine is compared with a powerful and reliable commercial software. Thus, the self-consistence of the identification process is also checked.

3.5.1 Simulation Engine vs FLOTHERM™ 4.2

The whole identification process is hold on the thermal simulations performed by the simulation engine. Therefore, its correspondence with real heat conduction phenomena has been a primary aim in its development. The results of the simulation program (`simulation.exe`, implementing the simulation engine) are compared here with results from FLOTHERM™ 4.2. This is a sophisticated, reliable and powerful thermal simulation tool from Flomerics™ Limited[4], a renowned company by its virtual prototyping software.

A simulation of an identical geometry, using the same thermal parameter values and conditions has been carried out with both programs and their results have been compared. The FLOTHERM™ 4.2 application is one of the best thermal simulation tools in the market, thus it has been taken as a reference in this work. Special attention has been put in implementing the same geometry configuration, the same spatial and time grid and the same thermal properties of the materials in both simulations.

Diffusion within a single material: aluminum bar example

The first example is the thermal simulation of the heat diffusion within an aluminum bar. Its dimensions are $10\,\mathrm{mm} \times 10\,\mathrm{mm} \times 100\,\mathrm{mm}$, and a $10\,\mathrm{W}$ of heat generation in the form of an step-wise function is dissipated in the zone shown in figure 3.9.
The used thermal properties of the aluminum are:

Thermal conductivity: $k_{Al} = 201\,\mathrm{W/\,m\,K}$
Density: $\rho_{Al} = 2710\,\mathrm{kg/\,m^3}$
Specific heat: $c_{Al} = 913\,\mathrm{J/\,kg\,K}$

[4]Website: http://www.flomerics.com

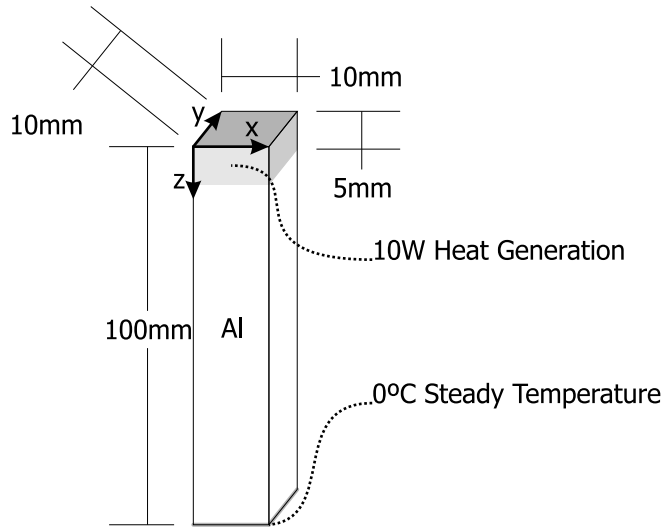


Figure 3.9: *Geometry configuration of the single material thermal simulation.*

In both simulations the temperature has been monitored at two points. First monitor point is located at $(5\,\text{mm}, 5\,\text{mm}, 0\,\text{mm})$ and the second one at $(5\,\text{mm}, 5\,\text{mm}, 50\,\text{mm})$. The obtained results from FLOTHERM™ 4.2 and the simulation engine are shown in figure 3.10. Figure 3.11 represents the subtraction of the obtained temperatures with both programs; i.e., the curve $(T_{\text{SimEngine}}(t) - T_{\text{FLOTHERM™ 4.2}}(t))$, giving a representation of the coincidence of both simulations. The average deviation between temperatures at the same instant has been, $0.004\,℃$ in the monitor point 1 and $-0.0007\,℃$ in the monitor point 2. And the maximum discrepancy at the same time instants between the simulated temperatures is $0.010\,℃$ in the monitor point 1 and $0.001\,℃$ in the monitor point 2. Thus, both results can be considered identical as relative differences are extremely low compared, for instance, with the experimental accuracy of the temperature probes.

The fact that the thermal system is composed of a single material cancels the k derivative terms of equation (3.12). Thus, a thermal simulation involving conduction through different medium interfaces is taken into consideration next.

Multiple material stack: DCB example

A power substrate DCB[5], with a dissipating semiconductor chip on it has been simulated with both thermal simulation tools. Figure 3.12 shows the

[5]This kind of substrate is described on section 4.3 at page 119.

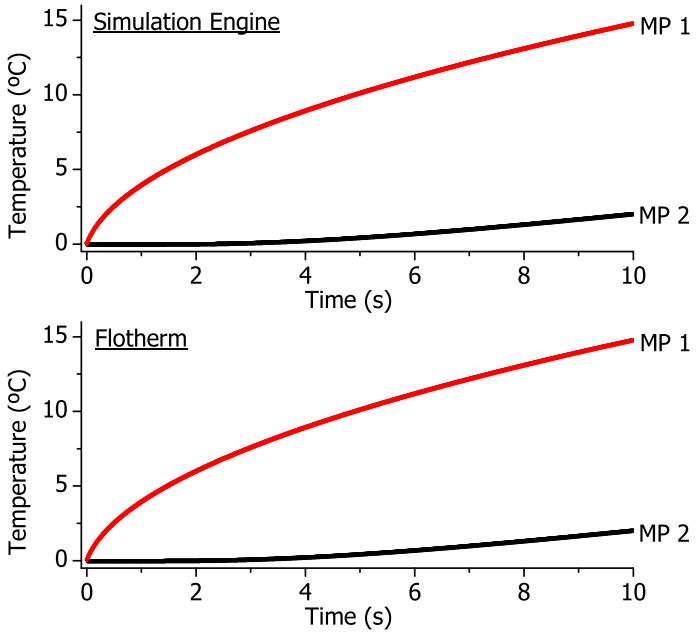


Figure 3.10: *Transient temperature curve obtained by thermal simulation with FLOTHERM™ 4.2 and the simulation engine described in this work.*

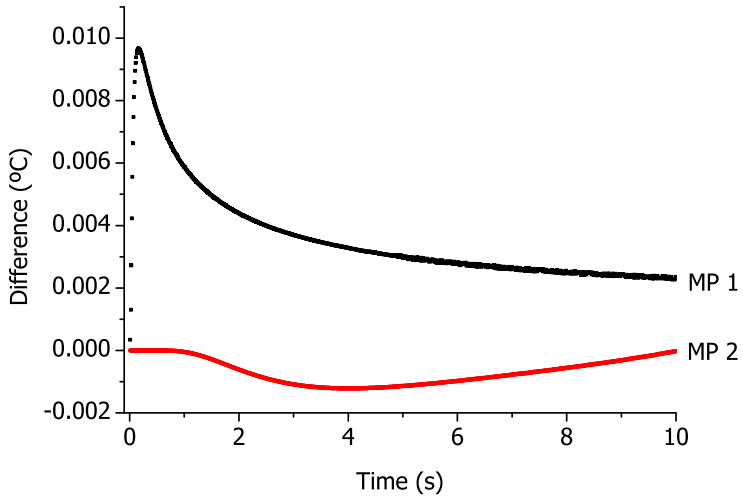


Figure 3.11: *Difference between simulated temperatures from FLOTHERM™ 4.2 and the simulation engine.*

simulated geometry configuration. Within the silicon, 20W of heat in a step-wise function shape is dissipated. On the bottom of the substrate a steady temperature of 0 ℃ is fixed as a boundary condition.

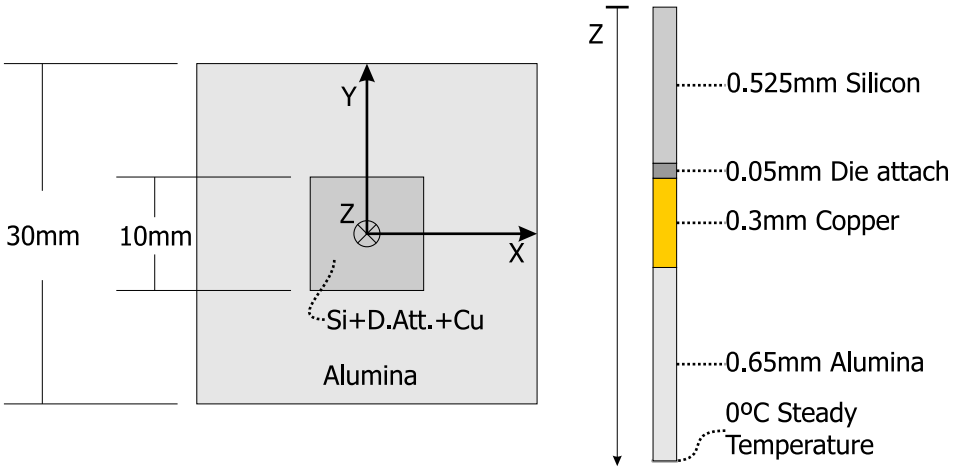


Figure 3.12: *DCB substrate geometry for the simulation engine and FLOTHERM™ 4.2 comparison.*

The thermal conductivity k, the density ρ and the specific heat c of the four materials used in both simulations are collected in table 3.2.

	Alumina	Copper	Die Attach	Silicon
k (W/ m K)	16	385	50.9	117.5
ρ (kg/ m^3)	3970	8930	8400	2330
c (J/ kg K)	765	385	150	700

Table 3.2: *Thermal properties of the materials constituting the DCB substrate used in both simulations.*

A monitor point is placed at coordinates (0 mm, 0 mm, 0 mm) where the temperature readings are performed. The transient temperature curves obtained from both simulation softwares are represented in figure 3.13. The subtraction of both results $((T_{\text{SimEngine}}(t) - T_{\text{FLOTHERM™ 4.2}}(t)))$ is shown in figure 3.14.

Again, the obtained results from FLOTHERM™ 4.2 and the simulation engine are almost identical, no significant difference can be appreciated in the representation of figure 3.13. The average difference between temperatures at the same instant is -0.014 ℃, having a maximum of 0.046 ℃.

From the exposed results, it can be concluded that the simulation engine provides good results. Discrepancies are acceptable since they are smaller

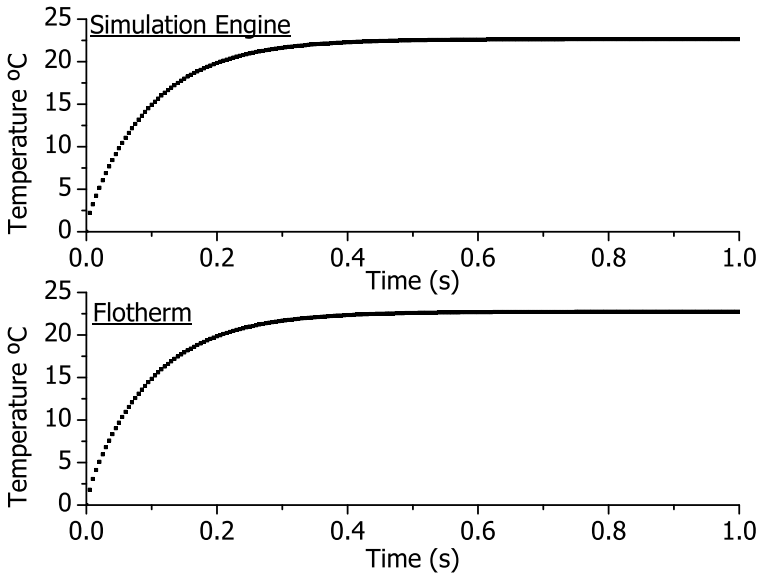


Figure 3.13: *Transient temperature curve obtained from thermal simulations with the simulation engine and FLOTHERM™ 4.2.*

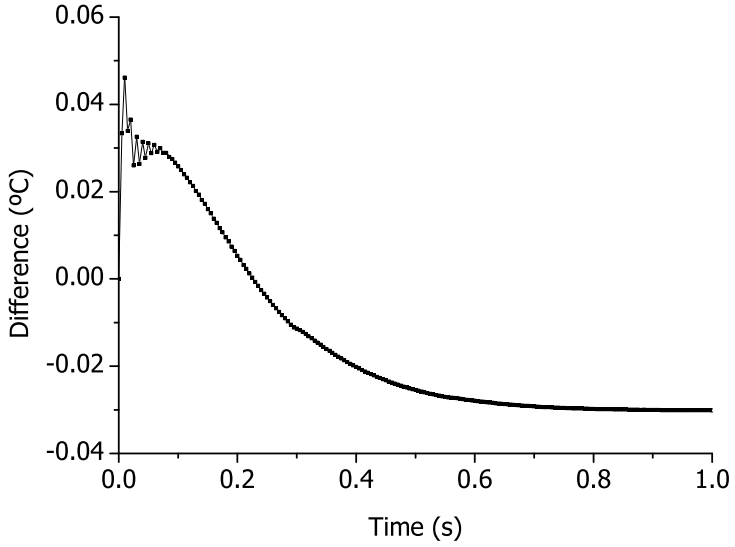


Figure 3.14: *Difference of simulated temperatures from FLOTHERM™ 4.2 and the simulation engine.*

than the maximum accuracy of the temperature probes. The quality of the thermal simulations of heat conduction obtained from the simulation engine are comparable with those from FLOTHERM™ 4.2. Moreover, it has been observed that the simulation time for the same thermal system and in the same computer is of the same order.

3.5.2 Auto-Consistence of the Parameter Identification Software

The identification algorithm, implemented into the identification program has been also validated. The objective is to test the capability of the application to derive thermal parameter values consistent with the simulation engine. The procedure is simple, a thermal simulation of a given thermal system is performed using the simulation program and the obtained heating curve is collected in a file, which is considered as experimental data. Then, introducing initial thermal parameter values into the identification program different than the original ones, it is expected that the nominal thermal properties will be restored. This procedure will check the validity of the identification procedure itself separating the computer-based analysis system *CescThParId 4.1* behavior from eventual disturbances linked to the experimental data extraction setup.

The thermal systems already used for the simulation engine validation of section 3.5.1 have been used this way. The objective is to evidence the capabilities of the algorithm as well as its limitations. Hence, selected successful identifications are exposed together with failed projects that will serve as examples for further discussion.

Single material: Aluminum bar example

The geometry of figure 3.9 with a given set of nominal thermal parameter values has been simulated and the resulting transient temperature curve from monitor point number one has been used as the experimental entry. The nominal thermal properties can be found on page 77. Thus, parameter identifications have been carried out giving preselected initial parameter values to the identification program. The same settings, with the exception of the material thermal properties, are used for the identification simulations.

The first example is an identification project with the parameters settled to values 50% below the nominal ones. The parameter evolution, transcribed here from the result file, is the following:

```
it.   k mat1       c mat1
0     100.500000 456.500000
1     151.205543 686.072988
2     189.003800 857.586715
3     200.386223 909.940463
4     201.003646 913.008008
5     200.999966 912.999972
6     201.000000 913.000000
7     201.000000 913.000000
```

First column '`it.`' is the iteration number, the iteration zero corresponds to the initial values. Next two columns labeled '`k mat1`' and '`c mat1`' are the values of the thermal conductivity k and the specific heat c of material number one. The identification has converged to the nominal parameter values after seven iterations.

The second example is very similar than the previous one, but with initial parameter values settled 100% above the nominal ones, converging to the exact values within ten iterations.

```
it.   k mat1       c mat1
0     402.000000   1826.000000
1     40.200000    187.924514
2     72.651548    338.039070
3     119.464266   552.163392
4     168.365525   771.665119
5     195.950291   891.812546
6     200.917971   912.625401
7     201.000710   913.002021
8     200.999993   912.999991
9     201.000000   913.000000
10    201.000000   913.000000
```

Multiple material stack: DCB Example

This multiple material example is much more illustrative of the *CescThParId 4.1* capabilities and drawbacks. The thermal system used here is the same DCB example of section 3.5.1 and the same simulation settings are used. Again, the system is simulated with the simulation program and the result is used as experimental data entry for the identification process. Thus, initial thermal parameter values are established expecting to obtain again the nominal values. The original nominal values are listed in table 3.2.

First, the thermal properties of the alumina ceramic (columns '`k mat4`' and '`c mat4`') have been identified obtaining the exact values. The parameters of the rest of materials are not included in the process but taken as

already known. The table 3.3 shows the transcription of the result file containing the evolution of the parameter values through consecutive iterations. The initial parameter values have been set to a value 10% below their nominal ones. After six iterations, the result with the exact original values has been reached. This shows how the unknown parameter values of a material can be extracted from a multiple material system heating curve.

Next, in order to check the limitations of the identification program, multiple parameter identification processes have been carried out. First, every parameter except those of the die attach ('`k mat2`' and '`c mat2`' in tables) have been identified setting the initial parameter values to values 10% below the nominal ones. From the result file, transcribed in the table 3.4, it is observed that after nine iterations the exact original values are reached for all the identified parameters. The advisability of the exclusion of the material two is discussed further on.

The same identification has been carried out with the initial parameter values settled 50% below the nominal ones. The table 3.5 contains the transcription of the parameter evolution through the consecutive iterations. After twelve iterations the process has reproduced the expected results.

When all parameters of the current geometry are simultaneously identified including those of the die-attach, a convergence of results is possible if the initial values are chosen to be 1% below the true ones. The process takes 14 iterations to get a definitive result, as shown in the table 3.6, containing the transcription of the result file. It can be observed that the original values of the experimental data are not reproduced. Instead, an alternative set of values is obtained with values very different to the original ones for the die attach. Other material thermal properties reach values acceptably close to the expected ones. Nevertheless, the resulting heating curve with the nominal values and that obtained from this identification result are identical. The curves coincide perfectly, having a maximum discrepancy of $-1.7 \times 10^{-4}\,^{\circ}\mathrm{C}$ at the time instant 0.04 s and an average discrepancy of $4.2 \times 10^{-7}\,^{\circ}\mathrm{C}$. Thus, no difference can be appreciated in a graphical representation of them. The identification program has found a combination of thermal parameter values that reproduce exactly the experimental curve; i.e., an alternative solution. Discussion about this is provided in section 3.6.

The last trial with the DCB geometry has consisted in identifying every parameter of the thermal system. This time, the initial parameter values are settled 10% below their ideal values. Wide fluctuations are observed in the table 3.7, showing the transcription of the result file, without any clear tendency during the initial iterations. The algorithm probably evolutes toward an alternative result, as in the previous example. If convergence was finally reached, the result would probably not coincide with the expected

values and, in any case, it would take an excessive number of iterations (unacceptable simulation time). After ten iterations, the process has been forced to terminate.

The identification tests exposed in this section demonstrate the capability of the identification algorithm to extract exact thermal parameter values from a simulated heating curve. Hence, *CescThParId 4.1* is self-consistent with its internal simulation engine, despite convergence problems and alternative results with no physical sense have been encountered. This is due to an excessive number of parameters to be identified simultaneously. A discussion on this problem and some advises to avoid it are given in section 3.6.

3.5.3 *CescThParId 4.1* vs FLOTHERM™ 4.2

Another virtual experimental validation of the computer-based part of the thermal parameter identification method is performed with the results of a thermal simulation carried out with FLOTHERM™ 4.2. These are used as the experimental data and the original parameter values are extracted with *CescThParId 4.1*. FLOTHERM™ 4.2 is considered here the best simulated approximation to real heat transfer phenomena. These tests will prove the capability of the proposed program to extract parameter values from a realistic heating curve before doing it on a real experiment with unknown materials.

Single material: Aluminum bar example

The same single material model described above is used here, this time the experimental data coming form a FLOTHERM™ 4.2 simulation. Here, the results setting the initial parameter values 50% below the nominal ones are presented:

it.	k mat1	c mat1
0	100.500000	456.500000
1	150.783262	687.750639
2	187.844777	862.202176
3	198.774822	916.379278
4	199.343469	919.649055
5	199.340042	919.640091
6	199.340075	919.640127
7	199.340074	919.640127
8	199.340074	919.640127

If the initial parameter values are set 100% above the nominal ones, the results converge to the same value as follows.

it.	k mat1	c mat1
0	402.000000	1826.000000
1	40.200000	238.310558
2	72.609314	416.600452
3	119.256427	647.435678
4	167.705425	842.386093
5	194.620606	914.544817
6	199.275961	919.666502
7	199.340692	919.639234
8	199.340068	919.640141
9	199.340074	919.640127
10	199.340074	919.640127

The obtained thermal conductivity value is 1.659926 W/ m K lower than the nominal value, or in relative terms, 0.826% below. Regarding the specific heat, it is obtained a value 6.640127 J/ kg K (or 0.727%) above the nominal parameter value. Thus, in practice it can be considered that the extracted parameter values fit the nominal ones with acceptable accuracy.

Multiple material stack: DCB Example

The DCB multiple material example has been also used with simulated results from FLOTHERM™ 4.2 as experimental data.

The thermal properties of the alumina material have been identified. Two identifications have been carried out. First with initial parameter values of the ceramic 50% below the nominal ones, second with the initial values 100% above. The result files are respectively transcribed in tables 3.8 and 3.9. The values in both identifications have converged to 15.927025 W/ m K for the thermal conductivity and 806.602081 J/ kg K for its specific heat. The obtained results coincide with the original values used for the FLOTHERM™ 4.2 simulation with small discrepancies. The obtained thermal conductivity is 0.072975 W/ m K below the original value, this is 0.458% lower. Regarding the specific heat, the obtained value is 41.602081 J/ kg K above the original parameter, this is 5.158% higher than the ideal value.

The differences observed between the identified parameter values and the nominal ones are due to the small discrepancies between the simulation engine and the FLOTHERM™ 4.2 results as discussed in section 3.5.1. The resulting value of specific heat is more sensitive to this than the values of the thermal conductivity, yielding a greater discrepancy. As observed in figure 3.4 the DT curves corresponding to the specific heat of materials reach absolute values lower than those corresponding to the thermal conductivity. Therefore, small heating curve discrepancies lead to greater specific heat value modifications.

As it was expected the obtained parameter values do not match exactly the nominal ones, as different applications have been used to generate the pseudo-experimental record and to identify them. The deduced values correspond, with acceptable differences, to the nominal ones. Therefore, it can be inferred that *CescThParId 4.1* identifies properly the parameter values from a heating curve coming from FLOTHERM™ 4.2.

it.	k mat1	c mat1	k mat2	c mat2	k mat3	c mat3	k mat4	c mat4
0	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	14.400000	688.500000
1	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	15.857411	772.282207
2	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	16.000182	765.050705
3	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	15.999998	765.000166
4	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	16.000000	764.999997
5	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	16.000000	765.000000
6	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	16.000000	765.000000

Table 3.3: *Self-consistence test with the DCB geometry. Initial parameter values 10% below the nominal ones, identifying the alumina ceramic parameters* `k mat4` *and* `c mat4`.

it.	k mat1	c mat1	k mat2	c mat2	k mat3	c mat3	k mat4	c mat4
0	105.750000	630.000000	50.900000	150.000000	346.500000	346.500000	14.400000	688.500000
1	117.963111	693.078366	50.900000	150.000000	302.400673	386.646093	15.942673	750.519678
2	117.710880	699.919161	50.900000	150.000000	364.127578	384.975923	16.009902	766.167436
3	117.492844	700.003623	50.900000	150.000000	384.430822	384.949394	15.999617	765.124943
4	117.500343	699.999858	50.900000	150.000000	384.989926	385.001408	16.000020	764.997449
5	117.499978	700.000008	50.900000	150.000000	385.001004	384.999932	15.999999	765.000092
6	117.500001	700.000000	50.900000	150.000000	384.999942	385.000004	16.000000	764.999996
7	117.500000	700.000000	50.900000	150.000000	385.000001	385.000000	16.000000	765.000000
8	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	16.000000	765.000000
9	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	16.000000	765.000000

Table 3.4: *Self-consistence test with the DCB geometry. Initial parameter values settled 10% below the nominal ones, identifying each parameter value except the die attach properties* `k mat2` *and* `c mat2`.

it.	k mat1	c mat1	k mat2	c mat2	k mat3	c mat3	k mat4	c mat4
0	58.750000	350.000000	50.900000	150.000000	192.500000	192.500000	8.000000	372.500000
1	93.261643	524.865684	50.900000	150.000000	59.761050	308.355778	12.292218	548.157281
2	118.095313	657.562637	50.900000	150.000000	108.782576	369.261458	15.247230	719.892285
3	120.778376	697.122311	50.900000	150.000000	193.759134	369.702364	15.963602	805.556274
4	118.067534	699.836522	50.900000	150.000000	297.217797	378.392525	16.002443	784.386123
5	117.651504	699.943026	50.900000	150.000000	365.259922	384.296238	16.004116	767.593426
6	117.499451	700.000843	50.900000	150.000000	384.299389	384.969151	15.999942	765.092830
7	117.499451	700.000843	50.900000	150.000000	384.299389	384.969151	15.999942	765.092830
8	117.499989	699.999997	50.900000	150.000000	385.005229	385.000305	16.000000	764.998967
9	117.500001	700.000000	50.900000	150.000000	384.999922	384.999999	16.000000	765.000007
10	117.500000	700.000000	50.900000	150.000000	385.000001	385.000000	16.000000	765.000000
11	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	16.000000	765.000000
12	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	16.000000	765.000000

Table 3.5: *Self-consistence test with the DCB geometry. Initial parameter values settled 50% below the nominal ones, identifying each parameter value except the die attach properties* `k mat2` *and* `c mat2`*.*

it.	k mat1	c mat1	k mat2	c mat2	k mat3	c mat3	k mat4	c mat4
0	116.325000	693.000000	50.391000	148.500000	381.150000	346.500000	15.840000	757.350000
1	123.183707	693.515613	46.416116	14.850000	368.238427	369.103953	15.833815	757.590145
2	124.180358	693.692358	45.973643	1.485000	365.877933	371.652677	15.835260	757.712701
3	124.685423	694.319702	45.654828	0.148500	369.192532	374.949622	15.845745	758.676083
4	129.780788	700.336994	40.819499	0.391710	433.663431	406.210246	15.919400	760.160790
5	127.508183	700.420515	44.658060	0.148017	365.111483	407.581267	15.978985	764.894090
6	127.822348	700.406270	44.453640	0.256123	384.658778	407.626757	15.967046	763.523560
7	127.398555	700.428202	45.082342	0.325851	375.988185	407.619714	15.976253	764.340381
8	127.404179	700.421810	45.112727	0.354407	375.290499	407.651591	15.977143	764.332057
9	127.408377	700.420284	45.113753	0.359451	375.125440	407.658684	15.977317	764.328375
10	127.409914	700.420032	45.112500	0.359855	375.120384	407.659911	15.977324	764.325796
11	127.410049	700.420013	45.112387	0.359875	375.120061	407.660013	15.977325	764.325574
12	127.410056	700.420012	45.112381	0.359876	375.120046	407.660018	15.977325	764.325563
13	127.410057	700.420012	45.112381	0.359876	375.120045	407.660018	15.977325	764.325563
14	127.410057	700.420012	45.112381	0.359876	375.120045	407.660018	15.977325	764.325563

Table 3.6: *Self-consistence test with the DCB geometry. Initial parameter values settled 1% below the nominal ones, identifying each parameter value.*

it.	k mat1	c mat1	k mat2	c mat2	k mat3	c mat3	k mat4	c mat4
0	105.750000	630.000000	45.810000	135.000000	346.500000	346.500000	14.400000	688.500000
1	44.166741	662.987449	89.467554	1485.000000	491.494930	160.162558	15.328419	728.486699
2	46.177180	664.514366	8.946755	1479.967394	527.813370	169.513080	15.454004	718.280805
3	91.744371	709.522624	8.748400	1219.128860	293.003534	228.469318	17.085484	836.656662
4	84.448650	703.475970	11.502000	2610.394264	364.456667	22.846932	17.853278	861.259199
5	94.067934	701.613973	21.153602	2305.270561	301.437604	24.655560	16.127087	840.847282
6	91.537454	702.651313	27.433665	2451.479267	390.514752	37.848960	16.594710	776.803411
7	92.242919	702.388332	27.589543	2404.013446	440.793302	56.845249	16.625812	747.363113
8	92.536204	702.284014	28.574882	2329.176741	459.498197	68.483711	16.574658	744.144422
9	92.699693	702.254047	29.569699	2272.608249	466.707863	76.606677	16.537387	743.210096
10	92.819993	702.231593	30.360890	2230.321438	470.577165	82.622082	16.511273	742.796985

*** Process killed ***

Table 3.7: *Self-consistence test with the DCB geometry. Initial parameter values settled 10% below the nominal ones, identifying each parameter value. The process is forced to terminate after iteration 10.*

it.	k mat1	c mat1	k mat2	c mat2	k mat3	c mat3	k mat4	c mat4
0	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	8.000000	382.500000
1	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	12.097857	1013.810907
2	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	15.102515	987.072903
3	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	15.902512	818.147101
4	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	15.927274	806.532668
5	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	15.927022	806.602714
6	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	15.927025	806.602074
7	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	15.927025	806.602081
8	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	15.927025	806.602081

Table 3.8: *Extraction of the Alumina thermal properties from a FLOTHERM™ 4.2 simulated heating curve. Initial parameter values of the ceramic are set 50% below the nominal values.*

it.	k mat1	c mat1	k mat2	c mat2	k mat3	c mat3	k mat4	c mat4
0	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	32.000000	1530.000000
1	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	3.200000	1663.738642
2	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	6.430986	3188.365083
3	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	11.870103	2561.149569
4	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	15.498180	1180.876581
5	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	15.940556	801.390489
6	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	15.926877	806.644710
7	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	15.927026	806.601702
8	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	15.927025	806.602084
9	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	15.927025	806.602081
10	117.500000	700.000000	50.900000	150.000000	385.000000	385.000000	15.927025	806.602081

Table 3.9: *Extraction the Alumina thermal properties from a FLOTHERM™ 4.2 simulated heating curve. Initial parameter values of the ceramic are set 100% above the nominal values.*

3.6 Identification Capabilities

Few illustrative examples have been described so far. Nevertheless, plenty of different models and identification settings have been carried out during the identification software development. With the technique of using simulation results from the simulation engine and from FLOTHERM™ 4.2 as experimental data entry, the *CescThParId 4.1* behavior has been tested at different situations giving a background knowledge before launching on true experimental data identification (chapter 4). Let us discuss the main issues:

Convergence of results

It has been shown that *CescThParId 4.1* works properly, being able to reproduce the original thermal property values of a given thermal simulation being self-consistent. The main drawback found in the identification process is that the numerical system can converge to an alternative set of resulting parameter values if its settings are not appropriate. The identification algorithm is a complex numerical system, consequently, it can present multiple solutions and even chaotic evolution. An in-depth study of the complex dynamics of this algorithm is out of the scope of this work. Nevertheless, a straightforward description of some hypothesis of the factors causing these disturbing effects, together with several advises to avoid them, are provided in this section.

The identification process evolutes properly when the parameter values do not experiment drastic variations of their values in an iteration and they show a tendency to approximate a final value. On the contrary, the numerical system might fall into the influence of an alternative numerical solution; i.e., another local error minimum. Examples of such effect are found in the identifications described for the self-consistence tests with the DCB geometry. Multiple parameters have been tried to identify, getting the following results:
The identification project of table 3.6 (page 91) identifies all the parameter values of the DCB geometry, the values of the found solution do not coincide with the nominal ones. The solution is mathematically valid since it reproduces the heating curve. However, it does make any physical sense in the current experimental context. The identification of table 3.7 (page 92) follows similar evolutions during its first ten iterations. The difference is that, expecting a prohibitive large number of iterations (and time) before convergence, the processes have been forced to terminate. A common hint when this happen is the fact that the numerical system becomes unstable and the parameter values experiment large fluctuations that can lead it to

another numerical solution.

Future efforts in the *CescThParId 4.1* improvement are channeled in part into the search of solutions to this problem, leading to a more stable and faster algorithm. However, the factors that work in favor of such runaway of values are identified so far and can be avoided. There is not a single factor that causes the system to fall into this dynamics by itself but a conjunction of conditions that increase the probability of the system to fall into the influence of an alternative local error minimum far from the expected solution.

Initial parameter values

It has been observed that initial parameter values in a multiple parameter identification closer to the final ones give rise, in general, to a faster convergence of results. If initial parameter values are too far from the nominal ones, the identification will take more time to converge as the values perform more fluctuations while approaching the final solution. Moreover, the probability of instability and run away of values will increase considerably. Even when only one of the parameter values is set too far in a multiple identification, the system may show this tendency.

Therefore, it is advisable to set the initial parameter values to an approximate known value, this would speed up the process and works in favor of its stability.

The identifications of tables 3.4 (page 89) and 3.5 (page 90) serve as example. They correspond to the same identification project with different initial values. When the initial parameter values are settled 50% below the final ones the solution converges within twelve iterations. Keeping the same setup but setting values only 10% below the solution, the identification of each parameter of the DCB geometry converges within nine iterations.

In conclusion, when identifying parameter values from an experimental record, the best approximation available for the initial values, such as typical values provided by the manufacturer or tables are recommended.

Number of Simultaneous Parameter Identification

An excessive number of thermal parameters to be identified facilitate the instability and gives rise to longer identifications (more iterations and more time each one). It has been observed that the numerical system specially tends to fall into the influence of an alternative local solution as larger number of parameters have to be identified. The increase of the dimensions of a dynamical system increases its complexity and facilitates complex dynamics.

The above described examples are useful again. The project of table 3.7 is attempting to identify simultaneously eight parameters but it has been terminated as no tendency to reach the right results has been detected. Simply excluding the parameters of the die attach material (the number two), the algorithm becomes stable reaching a solution as shown in the table 3.4. If only the two parameters of the alumina ceramic (the fourth material) are identified, the result converges quicker as shown in table 3.3. The same behavior is observed setting closer initial parameter values.

The advise is then to reduce the number of parameters to be identified. The complementary part of the identification software; i.e., the experimental setup which completes the proposed dynamical thermal parameter extraction system, must be designed having the fewest number of unknown materials. This will minimize the complexity of the resulting dynamical numerical system, facilitating a good parameter extraction. It has been also observed empirically that the algorithm is unconditionally stable leading to a unique solution if only one thermal conductivity and one specific heat are identified simultaneously. If possible, the best choice is a single material identification.

Thermal System Geometry and Material Properties

Another factor that may lead the process to the instability is the presence of materials with small influence on the transient heating record, such as thin layers with low specific heat and high thermal conductivity; e.g., the die attach and the copper layers of the DCB example system. The fact is that the variation curve $DT_{p,l}$ (defined in equation (3.1)) is very close to zero for such layers. Consequently, the component C_p (defined in equation (3.3)) corresponding to those parameters easily shoot up either to excessively positive or negative large values. The parameter values of such materials do not approach any fixed value while performing large fluctuations and they disturb other parameter evolution, inducing similar effects on the rest of parameter values.

In the identification examples this behavior is clearly observed, the presence of those parameters leads the identification algorithm to the instability. In table 3.6, the specific heat of the second material (the die attach) reduces drastically its value within the first three iterations and leads the system to an alternative solution. In table 3.7 the drastic decrease of the thermal conductivity and specific heat increase cause the run away of the parameter values. Similar fluctuations can be also observed for the copper (material three) specific heat.

Thus, it is derived that only materials having a significant contribution to the total thermal resistance or a significant contribution to the overall

heat capacity are recommendable for its parameter identification, otherwise, they may cause difficulties in the identification process. When the experimental setup involves materials with little relevance, it is worth to know their properties and do not include them in the identification.

Computation time

The implemented computer-based identification application can become easily extremely time-consuming. The process is iterative, taking a number of iterations n_{it} before reaching the definitive result. Each iteration of a process identifying n_{id} different parameter values performs $n_{id} + 1$ thermal simulations consuming t_{sim} each. Therefore, the whole process approximately takes t_{id}:

$$t_{id} = n_{it}(n_{id} + 1)t_{sim} \tag{3.47}$$

The rest of the identification process takes a negligible time in comparison with the thermal simulation time. Therefore, in order to reduce identifications time, actions must be addressed to the simulation time reduction. The thermal simulation time increases exponentially as narrower grid spacing and sampling period is settled, the user must choose this with caution when writing the geometry of the simulated model, performing test thermal simulations with the simulation program `simulation.exe` previous to the simulation launching permits the prediction of each iteration time.

As an example, the simulations carried out in the self-consistence tests of section 3.5.2 take about 27 min 35 s in a Pentium® 4 CPU 3 GHz, 1GB of RAM. Thus, for identifying two parameters, each iteration will take approximately 1 h 22 min 45 s. The simulations for the parameter identification from true experimental data of chapter 4 take about 9 h and 30 min. Thus, two parameter identification take 28 h 30 min 30 s for each iteration.

3.7 Conclusion

In this chapter, the computer-based parameter identification procedure is described. The algorithm has been implemented in MATLAB®, and its source code is transcribed in appendix A.

Firstly, the problem approach has been presented. Next, the identification algorithm has been described establishing the essential concepts and equations that are implemented in the software source code. The thermal simulation engine is also described. The software source code splitting into

several functions performing the different routines of the identification algorithm is presented in order to facilitate the interpretation of the transcribed code. Each compiled program operation is also described from the user point of view. Finally, several identification validation results are presented. The different aspects of the software are tested with three different experiences, the simulation capabilities of the simulation engine and the identification capabilities.

The correspondence of the thermal simulation engine with a reliable simulation application has been checked. It has been found that the simulation engine offers good simulation capabilities as the results agree with those obtained from FLOTHERM™ 4.2. The tests have been carried out on two geometry models, an aluminum bar and a DCB substrate with a dissipating semiconductor chip on it.

The same geometries have been analysed to check the self-consistence of the identification algorithm. The capability of the program to derive thermal parameter values consistent with its internal simulation engine have been proved. The algorithm can extract exact thermal parameter values from a simulated heating curve.

Finally, complementary tests have been performed extracting parameter values from a transient temperature curve simulated with FLOTHERM™ 4.2. The obtained results from *CescThParId 4.1* coincide within 5% with the nominal thermal parameter values.

The existence of multiple solutions of the complex numerical system have been illustrated in the described examples. This can lead the algorithm to alternative results with no physical sense but numerically correct. To avoid this, it is recommended to identify the thermal properties of materials with a significant contribution to the total thermal resistance and to the overall heat capacity. Materials with minor influence on the heating curve should be avoided. The best available approximation of the material parameter values should be set as initial values of an identification. This will speed up the convergence of results and reduce the probability of falling into a wrong local error minimum. It is worth to design experimental thermal systems with well known materials with the only exception of the material under test. It is also highly recommended to reduce the identification of parameters to a single specific heat and a single thermal conductivity; i.e., a single material. Identifications of more parameters may lead to instabilities.

Future efforts in the *CescThParId 4.1* improvement are channeled into the prevention of the described algorithm instabilities and to obtain fastest convergence of results both reducing the number of iterations and reducing each iteration time.

This may include to give the user the possibility of choosing the parameter

alteration amplitude, now fixed to be 1%, for the deduction of each parameter sensitivity within the identification algorithm. The algorithm could also detect the suitability of each parameter, setting automatically this variations, reducing them when the parameter values are close to the optimal solution. The algorithm could also detect excessive parameter value fluctuations as a criterion for these modifications. The convergence criterion could be also improved giving the user the possibility to choose the number of significant positions to be fixed before the algorithm considers that a solution is reached.

Another research line is the reduction of the time-consuming of the thermal simulations. In this sense, an adaptive time-step size in the thermal simulations could be implemented. The program could be also implemented in UNIX, since this operation system offers better calculation facilities than Windows. This could also allow performing the multiple thermal simulations with altered parameter values in different computers; i.e., several computers working in parallel performing a different thermal simulation each instead of a single computer.

Chapter 4

Thermal Conductivity and Specific Heat Measurements by Parameter Identification (Hardware Part)

Along the chapter 3, the measuring principle of the transient parameter identification method has been described. The computer-based analysis of experimental data for the identification of thermal parameters has been also presented. The current chapter explains how the heating curves are experimentally measured. The thermal and electrical setup, the power step-wise generation and the experimental procedure are described. Finally the parameter identification results for a representative power packaging substrate are also exposed.

4.1 Thermal Test Chip (TTC)

The described transient parameter identification method assumes that the temperature evolution is measured on the heating device in order to carry out the parameter identification process. Therefore, a thermal test chip which integrates heat power dissipation and temperature measurement on the same silicon die has been designed and implemented. The designed device can be soldered or attached to any substrate to be measured by identification. Its simple geometry facilitates its thermal simulation. The fact of having independent and well known dice instead of an already packaged commercial device permits to design a more detailed model for its thermal simulation. A low cost solution has been preferred to other more expensive possibilit-

ies implementing an active device. The sense resistor is made of platinum, which is not compatible with the standard DMOS technology available at *Centre Nacional de Microelectrònica - CSIC* facilities since it is highly contaminating. Manufacturing a MOS device together with a platinum resistor would involve the development of a new, expensive technology in external facilities. The device has been designed with the objective of making a versatile device, useful for other thermal experiences needing a heat source, a temperature sensor and a known geometry. For instance, the thermal test chip described here has been used for the IIR-LD (IR-laser deflection) technique validation [30, 31], that measures the internal temperature and the free-carrier concentration in power devices. All details concerning the design and implementation of the thermal test chip are described in this section.

4.1.1 TTC Design

The available technology (described further on, in section 4.1.2, page 106) in the *Centre Nacional de Microelectrònica - CSIC* allows to integrate polysilicon and platinum tracks on a silicon device. The chip dimensions are 6000µm×6000µm and the following design rules must be respected:

- The sheet resistance of polysilicon is $25\,\Omega/_{\square}$ and $2.5\,\Omega/_{\square}$ for the platinum.
- Minimum spacing between polysilicon tracks is 3µm, 17µm for the platinum (20µm optimum) and 20µm between polysilicon and platinum paths.
- The scribe line is 100µm and contact pads must be at least 120µm × 120µm.
- When evaluating the electrical resistance of a track, it has to be taken into account that a corner square is approximately equivalent to 0.6 squares:

 ≈ 0.6 squares

A platinum resistance temperature sensor is placed at the center of the chip, where higher temperatures are reached. This resistor is designed to be as tiny as possible in order to do not disturb the temperature distribution. Its resistance value is chosen to be around 1KΩ, instead of the 100Ω of most

of the commercial platinum thermal probes in order to maximize the sensitivity of temperature measurements. This resistor, referred as R_{sense}, sense resistor or temperature sensor from hereafter, has been designed as shown in figure 4.1. Four contact pads are used to perform the 4 wire resistance measurement. The total length of the structure is 362 squares between the sense pads. Hence the resistance value of the platinum resistor is approximately:

$$R_{sense} \simeq 905\Omega$$

A heating resistor is made up of several polysilicon tracks in parallel in order to spread the heat generation across the chip surface. From now on this heating polysilicon resistor is also referred as R_{heat} or heating resistor. Figure 4.2 outlines the design of this structure. Following the shown layout pattern, up to 130 paralleled polysilicon tracks go from one side to the other covering uniformly the whole chip area except that of the platinum area. Two contact pads of this resistor are placed on opposite sides of the chip. The R_{heat} value is chosen to be approximately 60Ω in order to maximize the power dissipation together with the available Keithley 2420 power supply which can stand 1A supplying 60V. The length of each track is, 6220µm and 20µm wide, hence having the same electrical resistance. The spacing between them is 17.2µm. As each line contains 310.2 squares and the resistor is made of 130 lines, the equivalent resistance is 2.38615 squares. Thus, the nominal heating resistance value is $R_{heat} \simeq 59.65\Omega$.

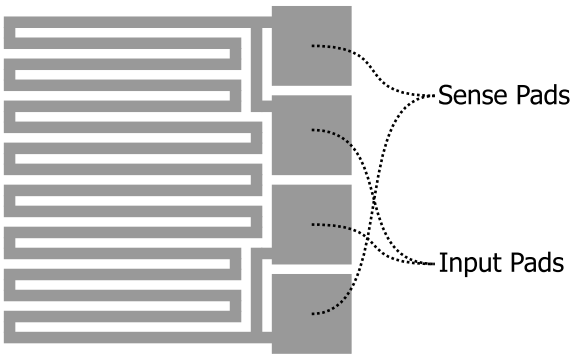


Figure 4.1: *Layout of the platinum resistor (R_{sense}).*

The thermal behavior of the designed thermal test chip has been analysed by thermal simulation with the FLOTHERM™ 4.2 application from Flomerics™ Limited. First, a simulation of the heat conduction from polysilicon tracks through the 525µm thick silicon bulk has been carried out. Figure 4.3 depicts the simulated temperature field of part of the silicon region below two

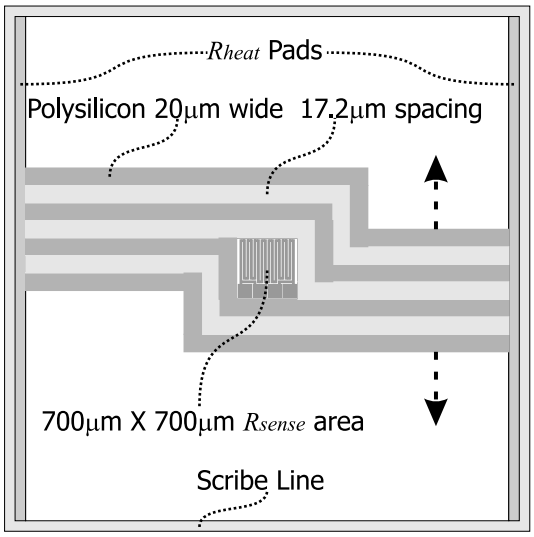


Figure 4.2: *Layout of the polysilicon resistor (R_{heat}) design (not to scale).*

polysilicon tracks. This estimation illustrates how the heat flux is uniformly spreaded few microns (less than 20μm) below the top surface, the isothermal surfaces being horizontal and parallel. Therefore, it can be considered as a good approximation for the identification simulations that heat flux is dissipated uniformly across the whole polysilicon heating resistor area. Secondly, a thermal simulation of the whole thermal test chip has been carried out dissipating 37W of total power at a room temperature of 20°C. Figure 4.4 shows a cross section zoom-in of the temperature field under the platinum resistor. There is no heat dissipation on the sense resistor area, therefore temperature at the center of the sensor is 3°C lower than the maximum temperature on the polysilicon area. The whole chip surface temperature is drawn in figure 4.5. Due to the fact that the presence of the platinum resistor area affects the temperature measurements, the heat dissipation can not longer be considered uniform across the chip surface. Thermal simulations of the thermal test chip in the parameter identification process will include the non-heating platinum resistor area in the model in order to provide results closer to the real experiment. Regarding this need, a multi-volume dissipation feature has been implemented in the thermal simulation engine.

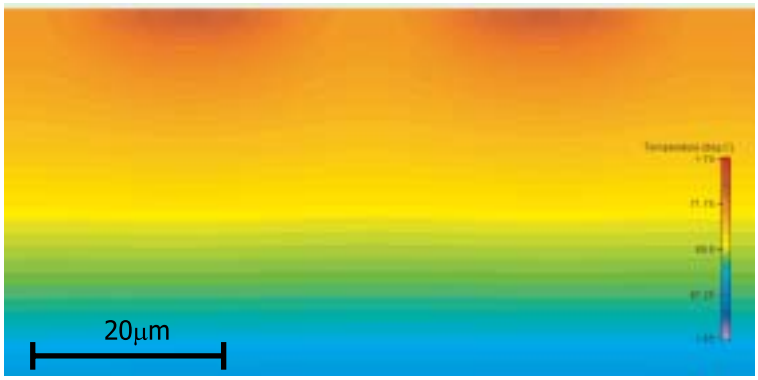


Figure 4.3: *Cross section of simulated temperature field close to two polysilicon tracks. Ambient temperature 20°C.*

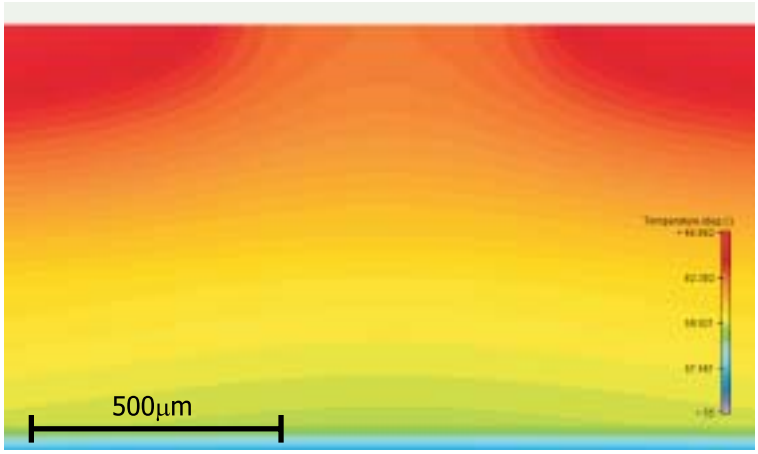


Figure 4.4: *Cross section of simulated temperature field under the platinum resistor area. Total heat power 37W. Ambient temperature 20°C.*

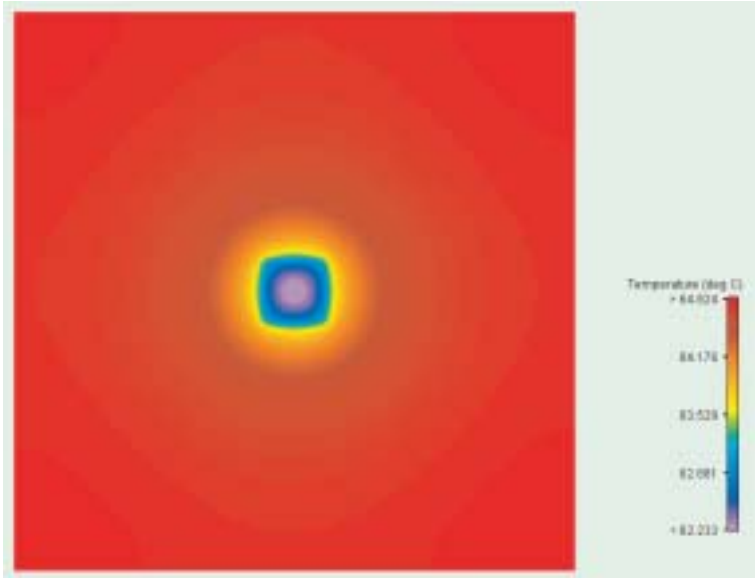


Figure 4.5: *Simulated surface temperature of the whole chip.*

4.1.2 TTC Implementation

Technology

Manufacturing of the thermal test chip has been carried out in the clean room facility of the *Centre Nacional de Microelectrònica - CSIC*. Five wafers with the designed device have been processed. Figure 4.6 shows two photographs under the microscope of the final result. The process architecture is drawn in figure 4.7. The process consists of 55 individual stages, including 7 lithography steps and can be summarized as follows:

300Å dry Oxidation

LOCOS oxidation: 1800Å nitride deposition, LOCOS mask lithography and 1800Å nitride dry etch off.

Polysilicon: 3500Å deposition of polysilicon and POCl3 doping, POLY mask lithography, 3500Å polysilicon dry etch off.

1.3μm of interlevel oxide deposition, BPSG.

Platinum region: CTPLT mask lithography, oxide dry etch off (1.3μm BPSG + 300Å thermal).

Al-polysilicon contact region: 300Å dry oxidation, CONT mask lithography, 1.3μm of BPSG on polysilicon dry etch off.

Metalization: 2μm Al/Cu deposition, METL mask lithography, 2μm Al/Si/Cu wet etched.

Platinum: LTOFF mask lithography, 200Å Ti deposition and 1500Å Pt deposition, photo resist lift-off.

Passivation: Oxide deposition (4000Å) + nitride (7000Å), P-ASS mask lithography.

Backline metalization: Ti/Ni/Au deposition.

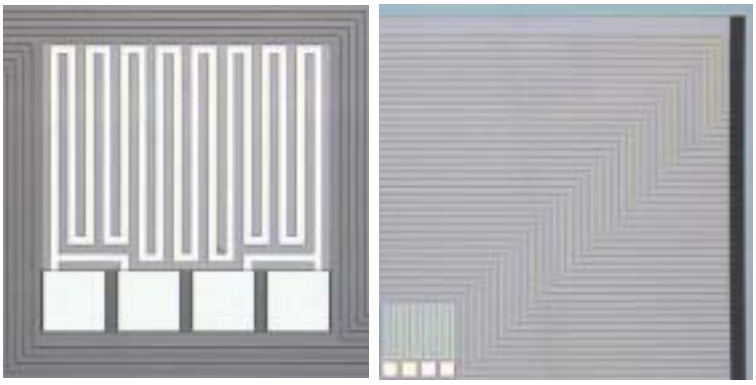


Figure 4.6: *R_{sense} and part of the R_{heat} structure arriving to the edges of the chip, under the microscope.*

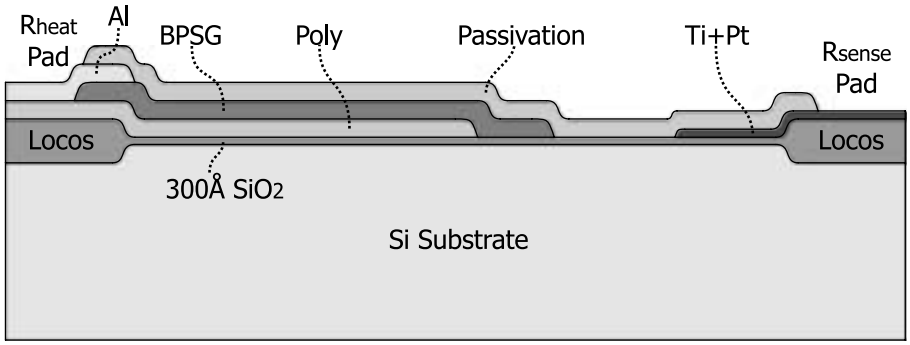


Figure 4.7: *Process architecture of the thermal test chip.*

All manufactured chips have been tested on wafer in order to know if they work properly and to monitor R_{heat} and R_{sense} values using an specific probe card. The probe card design is described on page 183 in the appendix C.

Wafer dicing, die attach to the substrates and wirebonding is carried out also at *Centre Nacional de Microelectrònica - CSIC*.

4.1.3 TTC Calibration

Calibration of the platinum resistance on the thermal test chip is necessary for each individual device. As it has been seen from the wafer's mapping carried out with the probe card, resistance values of each chip are a bit different to any other. Once each individual chip is attached to its substrate, the corresponding R_{sense} vs temperature relationship has to be found.

The experimental procedure is simple. The device is immersed into an environment in a steady and controlled temperature. Once the chip is in thermal equilibrium with this ambient, its R_{sense} is measured as shown in figure 4.8. All needed calibration data is obtained by repeating this acquisition at several temperatures within the desired temperature range of operation.

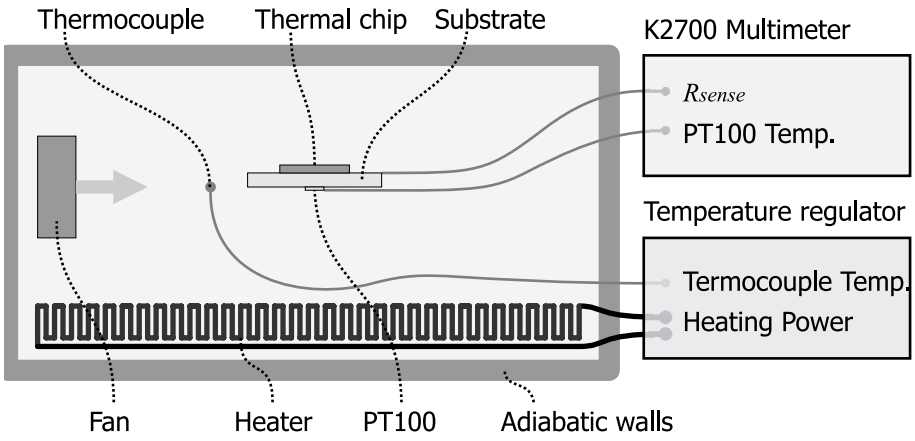


Figure 4.8: *The thermal test chip calibration setup.*

The cavity is a thermally isolated box with a mounted 250Ω heating resistor inside where 220V AC are applied. The power supply is switched by an Omron G3NA Solid-State Relay, controlled by an Omron E5CS-X Digital Temperature Controller. The temperature measuring for the control is performed by a thermocouple. The temperature is homogenized inside the box by a fan which continuously stirs the air. The substrate with its thermal test chip is mounted on a base that holds the DUT and a PT100 1/10DIN together and tied to the base of the box. This PT100 platinum resistance, together with the K2700 multimeter, provide high precision temperature measurement for the calibration. The K2700 multimeter also performs the accurate 4 probe measurement of the chip platinum resistance obtaining the wanted paired couple of data R_{sense} vs temperature. An overview of the calibration setup is shown on the photograph in figure 4.9.

As platinum resistance shows a linear dependence with the temperature, a linear fit of experimental data has been carried out. Once obtained series

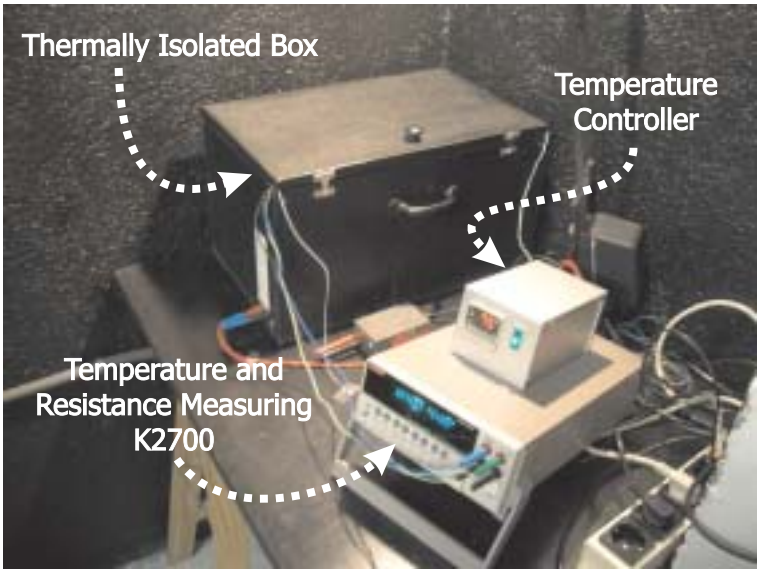


Figure 4.9: *Thermal test chip calibration setup.*

of pair data at different temperatures, a linear fit is done in order to obtain the parameters of the calibration formula (4.1). This is the expression used later to obtain the temperature values from R_{sense} measurements.

$$T = mR_{sense} + b \tag{4.1}$$

Let us take the calibration of the thermal test chip 4-30; i.e., the chip number 30 of the fourth wafer, mounted on a DCB substrate manufactured by IXYS, as an example. The calibration measurements are taken from room temperature up to nearly 85℃ at steps of 10℃. At each temperature point, fifty consecutive measurements have been taken. The regression carried out by Origin® 7.0 leads to the following values for the parameters of equation (4.1):

$$b = -622.85387\,℃ \quad \text{and} \quad m = 0.87855\,℃/\Omega$$

Data is represented on figure 4.10 together with the linear fit. Each temperature point contains fifty measurements, figure 4.11 represents a zoom-in in order to detail the measurements near a measurement temperature value.

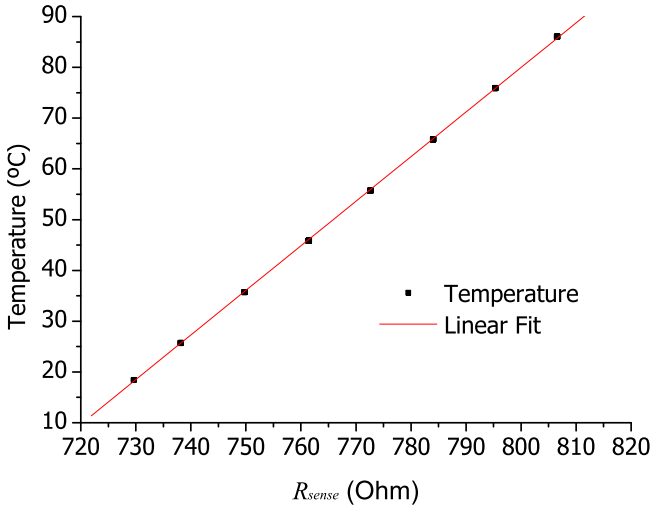


Figure 4.10: *Calibration measurements of the thermal test chip 4-30 and its linear fit.*

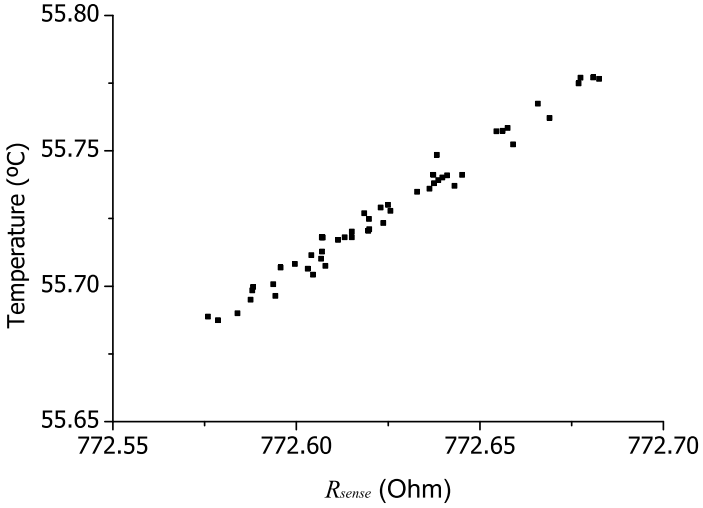


Figure 4.11: *Calibration measurements near 55°C.*

4.2 Transient Heating Curve Recording

Almost all parts of the experimental setup of the thermal conductivity measurement methods exposed on chapter 2 are used also for the transient temperature measurements. The innovation of this experimental setup lies on the use of the thermal test chip and the step-wise power generation method. Next, the procedure for the measurement of a transient temperature record is exposed.

4.2.1 Thermal Setup

A thermal test chip is attached to the power substrate under test, thus the material is ready to extract the temperature record for its parameter identification. Substrates used for measures in this work are mechanized into 15 mm × 20 mm pieces and a layout is etched on the top copper circuit layer in order to attach the thermal test chip. The etched layout must include six pads, two for wire bond the R_{heat} resistance and four for the four probe R_{sense} resistance measurement. Figure 4.12 shows the layout for the DCB substrates whose parameters have been identified in this work.

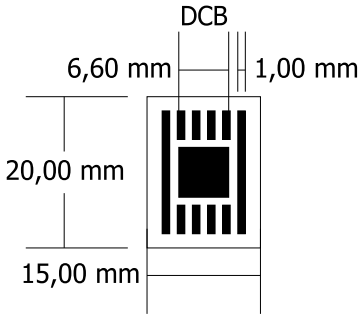


Figure 4.12: *Layout of the DCB substrate under test.*

Figure 4.13 shows a diagram of the thermal test chip on a substrate with all the wire bonds. The nominal current of each 25 µm diameter wire is 0.4 A. Eight wires are attached on both pads of the heating resistor, they are placed in parallel, its electrical resistance is estimated to be 0.033 Ω (considering 2 mm long each with an electrical resistance of 1.32 Ω/ cm). This will avoid significant heat loss in the wire bond in comparison with the dissipation in the heating resistor. Four single wires are used for the four sense resistance pads since only 1 mA will be needed for the temperature measurement.

The transient thermal parameter identification method uses basically the same thermal setup than the steady state method. It consists of an air cooled

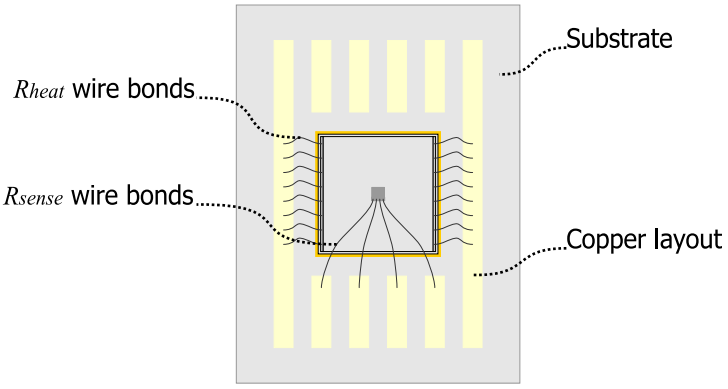


Figure 4.13: *Thermal test chip on a substrate and its wire-bond connections scheme.*

heat sink, a piece to press the sample and a vacuum cavity (see section 2.2). Figure 4.14 shows the transient method thermal setup, almost the same than that of figure 2.4 with some peculiarities. First, the power substrate with the attached thermal test chip is held tight by the pressure system adding a piece that applies the pressure in a way that the wire bonds do not become damaged. This piece is represented in figure 4.15 with its detailed geometry. It is made of wood due to its low thermal conductivity and its mechanical resistance in front of other plastic materials that would become deformed under the applied pressure endangering the thermal test chip. Second difference is the fact that no heat spreader is present since the current approach does not require to homogenize the heat flow.

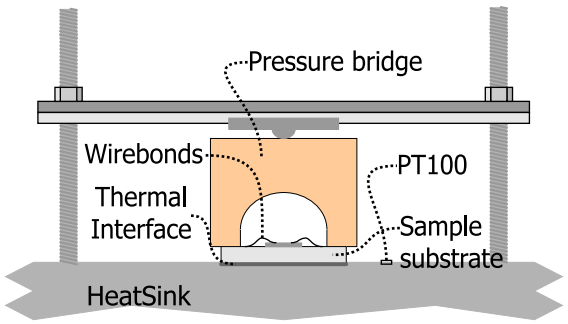


Figure 4.14: *Thermal setup for the transient measurement method.*

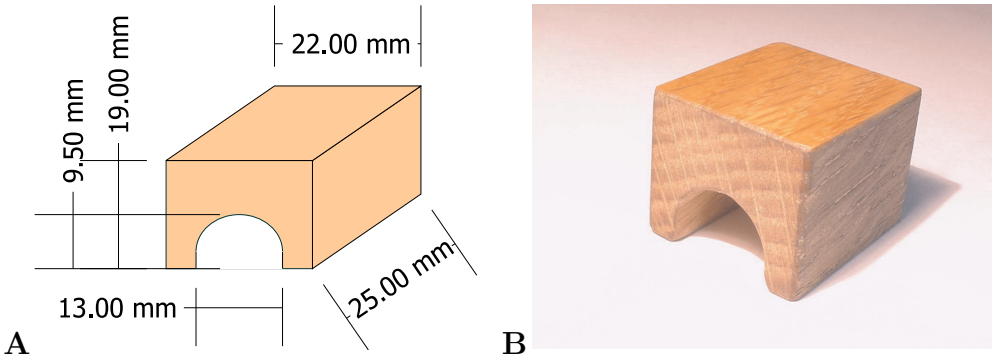


Figure 4.15: *A. Geometry of the bridge pressure piece. B. Photograph view.*

4.2.2 Step-Wise Power Control: The Electrical Setup

The heat dissipated by the R_{heat} resistor must be in the form of a step-wise function. The thermal test chip will increase its temperature during the heating operation and consequently, the heating resistance made of polysilicon will vary its electrical conductivity. Therefore, a feedback heat power control maintaining the heat power value with no fluctuations is needed. The controller described in section 2.3 will perform such control as it will be exposed next.

The electrical setup is described in figure 4.16, which represents the controller and the heat generation circuits interconnected. This figure stresses the particular wiring of the heating device and the controller in contrast with figure 2.16, which illustrates the steady state method of chapter 2. Although the heater is a simple electrical resistor, the control operation is basically the same. The MOSFET plays the role of a current source driven by the controller, which manages the current provided to the heating resistance, controlling it to meet the reference demand. The heating resistance voltage drop V_{heat} is measured in the controller. The shunt resistance signal, which measures the current I_{CC}, is connected to the V_{shunt} input. Thus, the controller circuit gets a reading of the heat generation on the R_{heat} resistance.

The V_{pwr} signal is the output of the multiplier part of the control circuit, whose inputs are the V_{heat} and V_{shunt} signals. As exposed before, the ground of the controller circuit is the S connection, this assures that the V_G signal that the circuit applies to the MOSFET gate is the V_{GS} parameter that controls the drain current of the device. This must be maintained to assure a correct device driving. To achieve this, the voltage measurement on the resistance is inverted as shown in figure 4.16; i.e., the multiplier input V_{heat} is a negative voltage. Consequently, the V_{shunt} measurement must be also

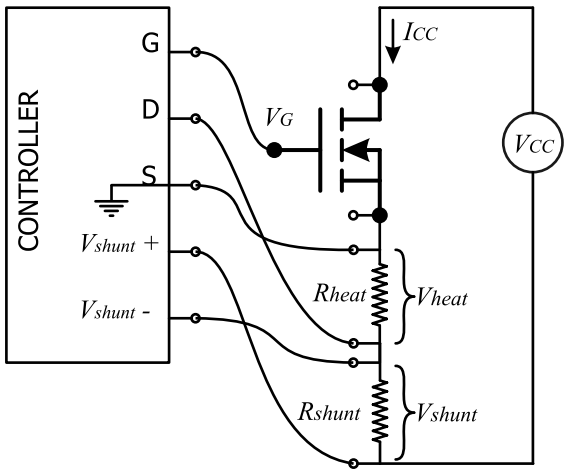


Figure 4.16: *Power control in the transient measurements.*

inverted in order to obtain a positive V_{pwr} signal from the multiplier block of the power controller. As it has been described in section 2.3.1, the V_{pwr} signal is proportional to the heat dissipated in the MOSFET when operating in steady state. Under the transient conditions it is proportional to the heat dissipated in the heating resistor R_{heat}. Assuming

$$V_{heat} = I_{CC} R_{heat} \quad \text{and} \quad V_{shunt} = I_{CC} R_{shunt}$$

it is deduced the equation (4.2) that is similar to equations (2.26) and (2.28), V_{heat} signal playing the role of V_{DS}.

$$V_{pwr} = K V_{Rheat} V_{shunt} = K R_{shunt} P \tag{4.2}$$

The V_{pwr} signal is proportional to the dissipated heat power P at the R_{heat} resistance with the same constant K as in equation (2.36). Given that, the feedback control operates as exposed in section 2.3.1 accomplishing the expression (2.31).

The electrical connections through the vacuum cavity lid and the connection box is represented in figure 4.17. In comparison with figure 2.17, only the wires and the MOSFET position have been changed. Nothing must be modified in the controlling circuit nor the connection box.

Given that, the MOSFET is in series with the heating resistance, each element dissipates the following power:

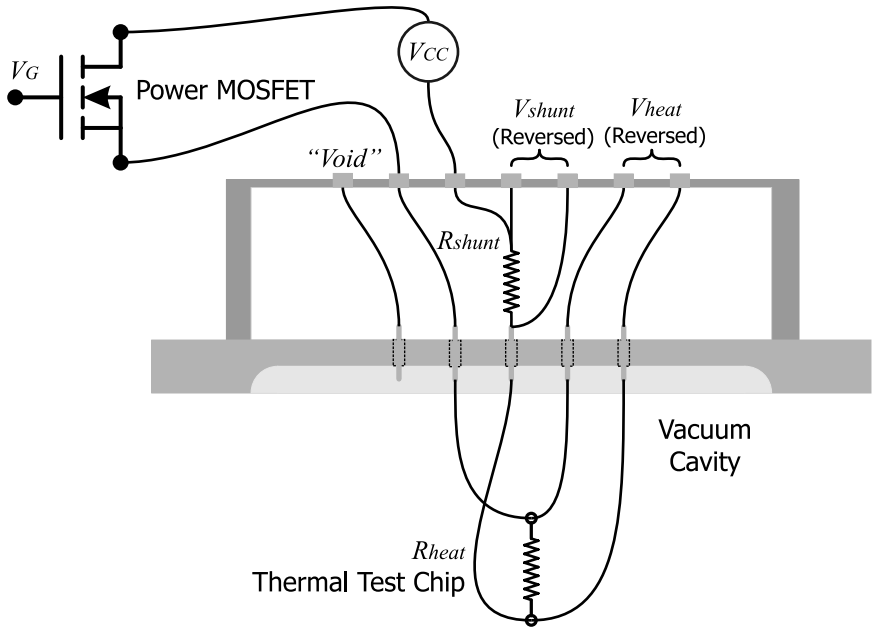


Figure 4.17: *Connections for the transient measurement.*

$$\begin{aligned} P_{shunt} &= R_{shunt} I_{CC}^2 \\ P &= R_{heat} I_{CC}^2 \\ P_{MOSFET} &= I_{CC}(v_{cc} - R_{shunt} I_{CC} - R_{Rheat} I_{CC}) \\ P_{total} &= I_{CC} V_{CC} \end{aligned} \tag{4.3}$$

P being the heat dissipated at the R_{heat} resistance, P_{shunt} that of R_{shunt}, P_{MOSFET} the heat dissipated at the transistor and P_{total} the total power supplied by the source. This is represented on a graph in figure 4.18 considering an I_{CC} current from 0 A to 1 A and $V_{CC} = 60$ V. The power MOSFET may dissipate then a considerable amount of heat, above 10 W, thus it must be attached to a heatsink in order to prevent an excessive self heating. The heat dissipated by the MOSFET does not affect the temperature inside the vacuum cavity since this device is placed outside.

In the framework of the described electrical setup, the heat power stepwise waveform is produced as follows. The origin of the step generation is the V_{ref} signal that performs a voltage step from the *enable/disable* switch. The signal is 0 V when the switch is in *disable* position and steps to a preset value when the switch turns to the *enable* position. Before the step, the fan and the vacuum are turned on, thus the heat level is adjusted previously in steady state as explained in section 2.3.2 (page 28) with the *Limiter*, *Range*

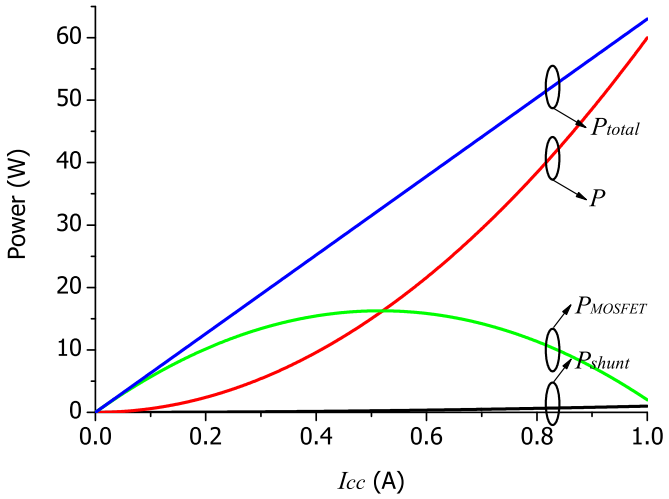


Figure 4.18: *Power dissipation of all components of the electrical setup of figure 4.16 versus the I_{CC} current.*

and *Control* potentiometers. This fixes the V_{ref} rise amplitude. At the instant when the *enable/disable* switch is set to *enable*, the controller circuit drives the I_{CC} current to generate a heat power step-wise at the heating resistance of the thermal test chip seeking the V_{ref} signal. Figure 4.19 shows the evolution of the signals V_{pwr}, V_G and V_{ref} during a power rise from 0 W to approximately 46 W. The V_{ref} signal rises in a small fraction of millisecond. Then the controller circuit immediately rises the V_G signal to its maximum; i.e., about the operational amplifier voltage supply, 15 V, in order to increase the I_{CC} current as fast as possible. The signal V_{pwr}, proportional to the heat power P, is also represented. Its rise time is about 1 ms with a 0.5 ms delay. Finally, the steady state value is settled in approximately 2 ms. This is negligible in comparison with the thermal time constants of the power packages that are higher than one second for small packages, being longer for large packages.

The I_{CC} compliance value can not be reached with the presented electrical settings. The fact that R_{heat} and R_{shunt} are connected in series with the power transistor causes a slope of $-(R_{heat}+R_{shunt})^{-1}$ in the working MOSFET I-V curve shown in figure 4.20. Therefore, the *limiter* potentiometer must be set to do not limit the V_G signal. This contributes to obtain a fast settlement time of the I_{CC} current (and the heat power).

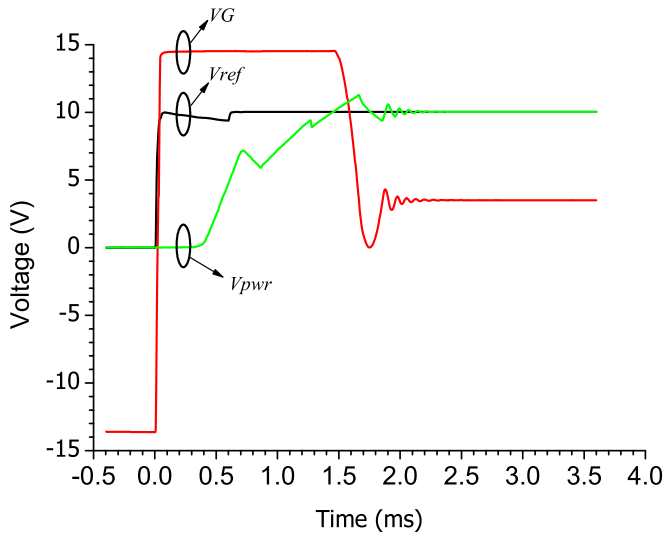


Figure 4.19: *Electrical signal performance of a* $46\,\mathrm{W}$ *heat power step.*

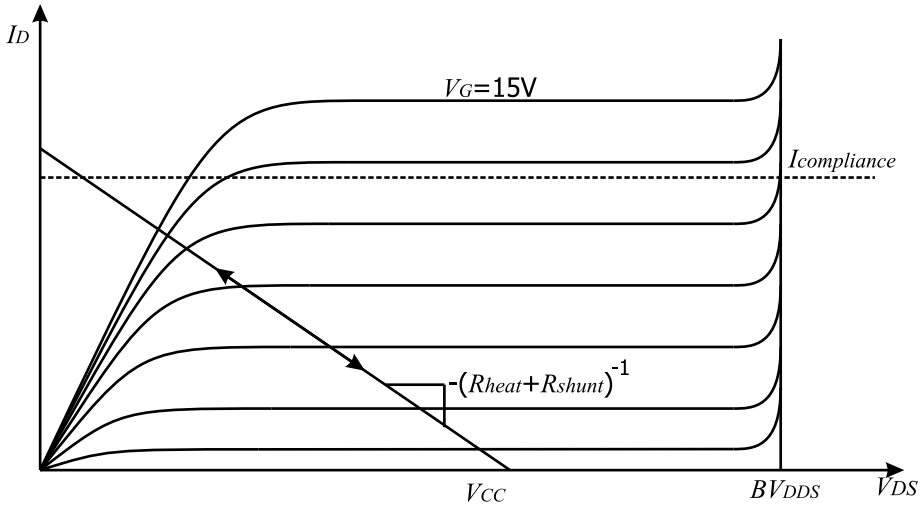


Figure 4.20: *The I-V characteristics of the MOSFET with the* $R_{shunt} + R_{heat}$ *load line.*

4.2.3 Transient Temperature Record Measurement

Temperature measurement is performed on the sense platinum resistor R_{sense} in the thermal test chip. It is assumed that a given calibrated chip is attached to a substrate, thus a transient temperature record is obtained as follows.

The K2700 multimeter, used in the thermal conductivity measurement, does not perform readings fast enough to have a significant number of temperature vs time readings. The minimum time step between temperature measurements that can be achieved with this multimeter is around 39 ms. The identification program manages with variable sampling period, the problem is that the time interval is too long and, consequently, fast temperature evolution details are not detected. As a consequence, the identification procedure may miss important information due to the poor sample rate of temperature records. It is used therefore an oscilloscope model TDS 744A from Tektronics to perform the R_{sense} transient measurement. The sampling period in the TDS 744A records is 4 ms, providing enough time resolution. In addition, the oscilloscope can get up to 4 channel measurements and has a trigger system that allows synchronizing the initial acquisition instant.

Although the readings are performed by an oscilloscope, the K2700 multimeter is still connected as shown in figure 4.21 acting as a current source. The four probe resistance measurement injects a small and very stable current I_{sense} through a pair of wires causing a voltage drop V_{sense}. The V_{sense} is measured by the oscilloscope, recording the signal in an adequate sample rate. The oscilloscope also measures the I_{CC} current through a passive isolated current probe (Tektronix P6021) in another channel. The sudden rise of the heating current causes the probe to introduce a voltage rising edge that triggers the oscilloscope data acquisition.

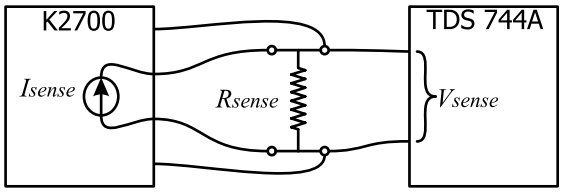


Figure 4.21: *R_{sense} measurement scheme with the K2700 acting as a current source and the TDS744A recording the V_{sense} evolution.*

The I_{sense} current is measured at each transient recording. When the power step-wise is triggered, the transient $V_{sense}(t_n)$ evolution is recorded in the oscilloscope. The thermal equilibrium state is achieved after few minutes, and V_{CC} and I_{CC} are noted down from the source SMU as well as V_{heat}. The

dissipated power is calculated accurately with the values of V_{heat} and I_{CC} as:

$$P = V_{heat} I_{CC} \tag{4.4}$$

Hence, a $V_{sense}(t_n)$ record, the power level of the step P and a steady sense current I_{sense} are available. The transient magnitudes are a finite collection of values at finite number of instants t_n. These data must be entered as experimental data to the *CescThParId 4.1*. Using Origin® 7.0, the transient sense resistance is deduced:

$$R_{sense}(t_n) = \frac{V_{sense}(t_n)}{I_{sense}} \tag{4.5}$$

Finally, with the calibration data of the thermal test chip, it is deduced the final temperature vs time heating curve $T(t_n)$ according to the expression:

$$T(t_n) = mR_{sense}(t_n) + b \tag{4.6}$$

This is in fact the calibration expression (4.1) of the thermal test chip. The parameters m and b are supposed to be obtained previously.

In order to reduce the noise of the V_{sense} signal, ten transient recordings are carried out. The final processed $V_{sense}(t_n)$ is a cleaner average curve of all the records. The ambient or initial temperature is deduced from the $T(t_n)$ readings prior the trigger ($time < 0\,\text{sec}$). An example of such $T(t_n)$ curve is shown in figure 4.22 in the next section.

4.3 Transient Identification on a Power DCB Substrate

One of the most used power substrates, a DCB (Direct Copper Bonding) from *IXYS Corporation*, has been chosen for the extraction of its thermal parameters. The substrate consists in two 0.3 mm thick copper layers directly bonded to a 0.65 mm thick alumina[1] slab. The thermal properties of the alumina ceramic are essential parameters for a good thermal management in the design of a package using such a DCB substrate. These are the material properties to be extracted from experimental records. The manufacturer provides a value for the alumina thermal conductivity of $[24 - 28]\,\text{W/m K}$ in its data sheet [32]. Other materials such as silicon, the soldering material and the copper are considered well known and their values do not need to be identified.

[1] aluminum oxide Al_2O_3

Transient Temperature Record

The identification of the DCB sample substrate thermal parameters has been carried out on an experimental transient temperature record obtained under the following conditions:
The sense current supplied by the K2700 is:

$$I_{sense} = 0,9797 \pm 0,0006\,\mathrm{mA}$$

The dissipated power when the power controller is turned on is:

$$P = 25,29 \pm 0,34\,\mathrm{W}$$

And the ambient temperature; i.e., the initial temperature of the thermal system at the beginning of the transient record is:

$$T_{initial} = 20.39\,^{\circ}\mathrm{C}$$

Hence, with the described procedure, the temperature curve represented in figure 4.22 has been obtained. These data are read by the identification program from the experimental data file, an ASCII file with a list of temperature vs time value pairs.

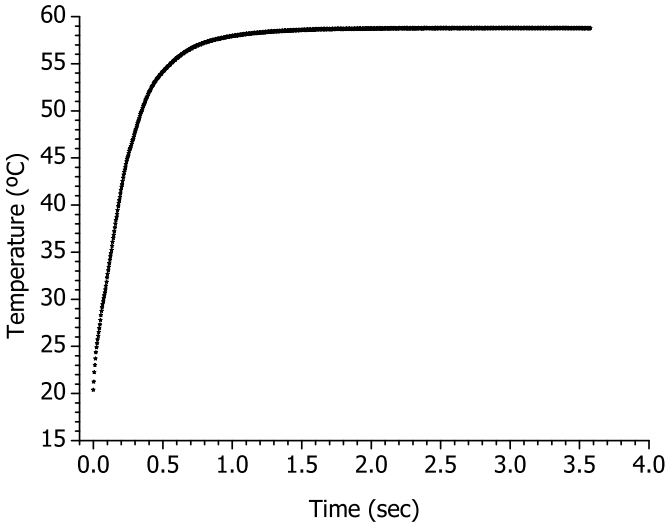


Figure 4.22: *Transient temperature record for the thermal test chip 4-30 on a DCB substrate.*

Thermal Simulation Geometry File

Once available the experimental data, the second input necessary for the thermal parameter identification is the 3D thermal simulation geometry file. The figure 4.23 shows the geometry to be simulated. The geometry file that inputs the simulation geometry and the initial parameters to the identification application *CescThParId 4.1*, are detailed in the appendix B. The material geometry is defined in the file (from line 14 to line 43). Taking advantage of the symmetry of the thermal system made up of the thermal test chip and the substrate, only a quarter has been simulated. This is a technique usually resorted to by users of thermal simulation CAD tools to reduce the computing time. As the thermal simulation engine implements adiabatic walls at the end of the simulation range, this strategy can be used. Thus, only the texturized area represented in figure 4.23, is computed.

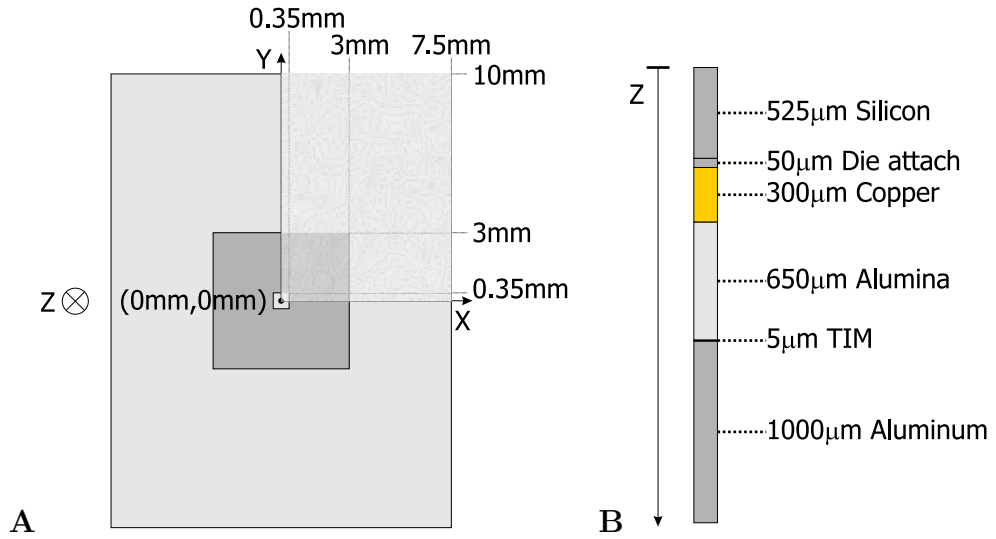


Figure 4.23: *The DCB substrate geometry: A.) an XY plane view, and B.) a Z view across the Z axis.*

The geometry for the simulation includes part of the aluminum heat sink as well as the thermal chip and the substrate. The simulation of the whole heat sink would involve a prohibitive large geometry leading to an unacceptable time-consuming simulation. However, there is a heat spreading effect near the contact with the substrate that must be taken into account. Nevertheless, in the heat sink bulk, the temperature becomes constant and equal to the ambient temperature. One millimeter thick aluminum layer has been considered as a first approach for reproducing this effect, leading to more detailed and realistic simulation of the heat conduction phenomenon.

Initial Parameters

The geometry file includes also the initial values for the physical properties of all materials involved in the simulation, from line 45 to line 67 in the example file transcription. The table 4.1 collects the initial parameter values of all materials used in the geometry. If no parameters can be found in any technical data publication or data-sheet, the typical values found in the FLOTHERM™ 4.2 material library of physical properties are considered appropriate for the identification. The data-sheet of the alumina DCB substrate sets its thermal conductivity within the range $[24 - 28]\,\mathrm{W/\,m\,K}$, thus a value of $26\,\mathrm{W/\,m\,K}$ has been used as initial value. No value is given for the copper parameter values in the substrate data-sheet, nevertheless the typical copper thermal properties are considered appropriate enough. Also typical physical properties for the die attach alloy (Sn63/Pb37) are used in the simulations. Regarding the silicon, its thermal conductivity depends on the working temperature. The simulation engine and the identification program does not permit this kind of parameter variation. Therefore, a compromise value has been settled taking into account that the initial die temperature is 20.39 °C and the maximum attained value is 58.75 °C; i.e., its thermal conductivity varies from $168\,\mathrm{W/\,m\,K}$ to $130\,\mathrm{W/\,m\,K}$ [33]. Therefore, an average thermal conductivity of $150\,\mathrm{W/\,m\,K}$ has been considered. The thermal interface material used is the RS HeatSink Compound, already used in the thermal conductivity steady state measurements. Its thermal conductivity has been found in its data-sheet, being $0.9\,\mathrm{W/\,m\,K}$. Nevertheless, its density and specific heat are not detailed. As it is an extremely thin layer in comparison with others, its thermal capacity is inappreciable. Consequently, the simulation value of its specific heat does not have any effect on the simulated heating curve shape. Thus, the same values of the alumina ceramic are used for the thermal simulation. The typical thermal properties of the aluminum, from the FLOTHERM™ 4.2 data base, have been considered.

Time Discretization

The total simulation time is programmed to be two seconds and it is divided into 200 time steps; i.e., a sampling period of 10 ms. This is seen from line 105 to line 108 in the DCB thermal simulation geometry file transcription.

Identification Configuration File

The third and last element that defines a given parameter identification is the identification configuration file. The current identification is transcribed also in appendix B. The file contains information about the names of the

	Thermal Conductivity (W/ m K)	Specific Heat (J/ kg K)	Density (kg/ m^3)
Silicon	150	700	2330
Die attach	50.9	150	8400
Copper	385	385	8930
Alumina	26	765	3970
RS TIM	0.9	765	3970
Aluminum	201	913	2710

Table 4.1: *Initial thermal properties of materials involved in the DCB identification.*

involved files, details of the experimental data file format and a maximum number of iterations if any definitive result is reached.

4.3.1 Identification of the Alumina Parameters

The identification of the thermal conductivity and specific heat of the alumina ceramic has been launched with the settings described above. In a first stage, only the alumina thermal properties are identified; i.e., one thermal conductivity and one specific heat. The following is a partial transcription of the result file:

```
it.  ...  k mat4    c mat4       ...
0    ...  26.000000 765.000000   ...
1    ...  2.600000  1477.416195  ...
2    ...  5.614808  1549.414975  ...
3    ...  9.709573  1484.720094  ...
4    ...  13.234074 1369.773181  ...
5    ...  13.658976 1353.631574  ...
6    ...  13.657700 1371.025236  ...
7    ...  13.658416 1369.464413  ...
8    ...  13.658338 1369.623868  ...
9    ...  13.658346 1369.607453  ...
10   ...  13.658345 1369.609142  ...
11   ...  13.658345 1369.608963  ...
12   ...  13.658345 1369.608979  ...
13   ...  13.658345 1369.608978  ...
14   ...  13.658345 1369.608978  ...
```

where the column "`it.`" is the iteration number, "`k mat4`" the thermal conductivity value of the alumina (the material number four in the geometry file) and "`c mat4`" its specific heat. Other parameters have been obviated as they are fixed to its initial value. The process has provides these results after fourteen iterations that have lasted 400 h and 8 min in a Pentium® 4 CPU

3 GHz, 1GB of RAM. The time consuming of every thermal simulation has been about 9 h and 30 min each.

The final parameter values; i.e., the thermal conductivity $k_{alumina}$ and the specific heat $c_{alumina}$ are:

$$k_{alumina} = 13.66\,\mathrm{W/m\,K}$$

$$c_{alumina} = 1369\,\mathrm{J/kg\,K}$$

A thermal simulation has been carried out with the initial parameters and another one using the resulting final collection of thermal parameters. They are represented in figure 4.24 together with the experimental temperature curve in order to provide a view of the evolution toward the curve coincidence of the process.

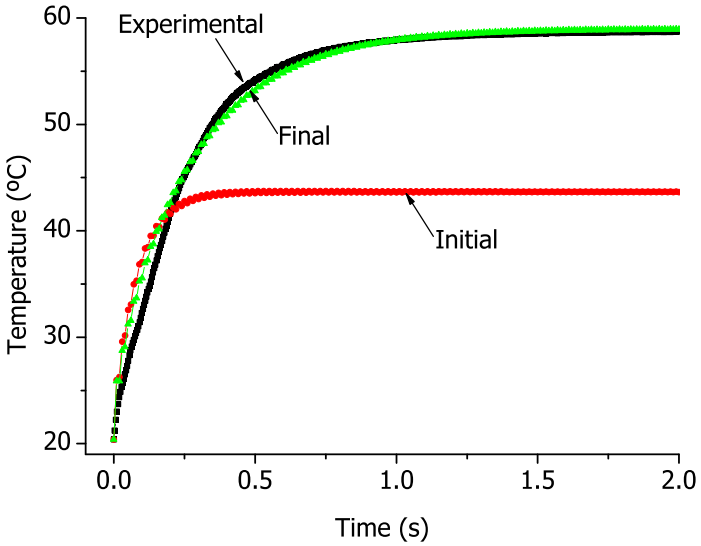


Figure 4.24: *Initial and final simulated transient temperature curves compared with the experimental one identifying the alumina thermal parameters.*

It can be observed that the specific heat obtained is much higher than the value reported in the literature. This is due to the fact that the simulated temperature increases faster than the experimental one. During the identification process, the algorithm tries to increase the specific heat of the alumina in order to correct this, but it is not possible modifying only the alumina thermal parameters. The identification finalizes when the best possible

curve adjustment is reached. This delay of the experimental temperature rise is due, for the most part, to the thermal capacity of the heat sink. The simulated portion of aluminum seems to be not enough for reproducing the thermal impedance of the heat spreader. But increasing its dimensions would easily lead to an excessive long identification process. However, another approach can be carried out identifying also the specific heat of the partially simulated aluminum heat sink. This approach will not lead to a realistic value of the aluminum specific heat but rather a correction of it, providing a suitable thermal capacity.

The same identification has been carried out, this time the specific heat of the aluminum is identified as well as the alumina parameters. The transcription of the result file of this identification process is the following:

```
it.  ... k mat4    c mat4       ... c mat6
0    ... 26.000000 765.000000   ... 913.000000
1    ... 25.279743 765.649714   ... 9103.894015
2    ... 13.277487 920.510468   ... 40473.012467
3    ... 14.821374 929.271477   ... 5047.397801
4    ... 14.825104 929.893268   ... 2776.265458
5    ... 14.826176 930.119795   ... 2981.980684
6    ... 14.840779 929.859733   ... 2904.215473
7    ... 14.843740 930.766484   ... 2892.311409
8    ... 14.842774 930.532668   ... 2889.753491
9    ... 14.842722 930.602714   ... 2888.491084
10   ... 14.842725 930.602074   ... 2887.031567
11   ... 14.842725 930.602081   ... 2887.669303
12   ... 14.842725 930.602081   ... 2887.635254
13   ... 14.842725 930.602081   ... 2887.642432
```

The specific heat values of the aluminum, the sixth material in the geometry file, are collected in the column "`c mat6`". The identification process has not arrived to definitive values. After 13 iterations it has been forced to terminate since the alumina parameters were already fixed and the first decimal position of the aluminum specific heat as well. The total 13 iteration process has taken about 496 h and 54 min; again, about 9 h and 30 min per simulation, in the same computer than the previous identification.

The final parameter values for the alumina thermal conductivity and specific heat have been the following:

$$k_{alumina} = 14.84\,\mathrm{W/\,m\,K}$$

$$c_{alumina} = 930.6\,\mathrm{J/\,kg\,K}$$

After the last iteration, a thermal simulation with the current thermal properties has been carried out. Again, for appreciation of the curve coincidence, the final simulated temperature curve and the experimental one are represented in figure 4.25.

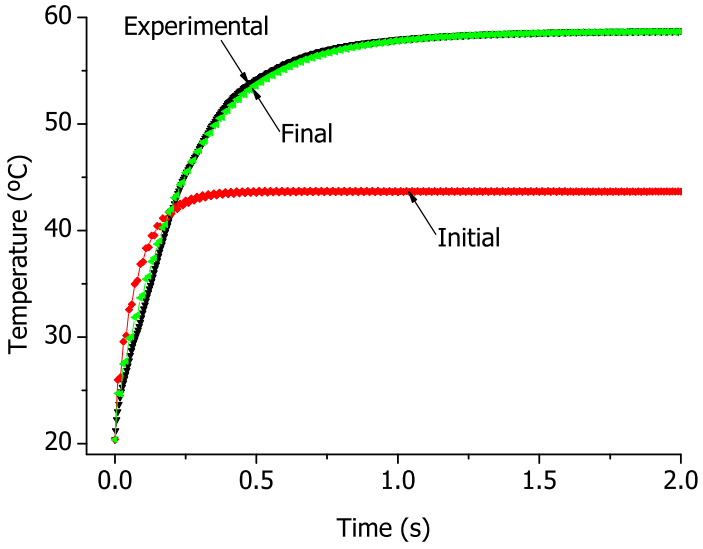


Figure 4.25: *Initial and final simulated transient temperature curves compared with the experimental one identifying the specific heat of the aluminum as well as the alumina thermal parameters.*

This time, the specific heat of the portion of aluminum in contact with the alumina substrate has been adapted by the identification algorithm obtaining a heating curve shape best fitted to the experimental one. Consequently, since the temperature rise delay due to the heat sink thermal capacity of the portion immediately in contact with the alumina has been reproduced, the obtained specific heat of the alumina can be considered more suitable than the presented above. Regarding the thermal conductivity, it is obtained a slightly higher value.

In conclusion, the obtained value for the thermal conductivity is lower than the typical one, and the specific heat is higher than expected. However, the obtained simulated curve with the identified parameters fits the experimentally measured curve better than the previous one. Moreover, the simulated curve using the identified thermal parameters fits the real experimental curve with an accuracy impossible to attain with typical values found in the literature. In this sense, the obtained thermal parameters are now optimal for the thermal simulation of such thermal system. This is the main objective of the developed thermal parameter measurement method: to find thermal parameters leading to accurate thermal simulation results.

However, there are still differences between the experimental and the simulated curve, probably due to the several hardware factors described next:

- The dependence of the silicon thermal conductivity on temperature has not been taken into account. Part of the discrepancy of the simulation during the first instants of heating may be due to this fact.

- The die attach material is well known but its structure may not be perfect. The presence of voids in this layer is frequent. Nevertheless, it is a very conductive and extremely thin layer with little influence on the heating.

- The substrate under test presents pads where relatively thick wires connect electrically the thermal test chip to the power source. There is a minor heat leak through this connection pads that has been neglected since these wires are not simulated. There is a small heat conduction through the wire bond that has not been taken into account as well. Also an insignificant heat could be conducted through the pressure piece.

- The heat sink has been partially simulated, it is considered that only a small part of the whole heat sink presents a representative temperature gradient that influence the temperature evolution.

- The thermal interface material used to thermally connect the substrate and the heat sink is approximatelly simulated. Although its bulk thermal conductivity is known, the contact thermal resistance [15, 22, 23] is not modelized and its thickness is only estimated while its contribution to the total thermal resistance is not negligible.

- The platinum sense resistance electrical response to a temperature variation is considered instantaneous. In addition and the sense resistance performs the temperature measurements across an extensive area while the simulations considers the temperature at one point.

It is concluded that further development of the experimental data acquisition setup is necessary. Firstly, better control of the experimental factors listed above is advisable. Secondly, with the present setup, the simulated fixed temperature boundary just under the alumina is not very accurate. Since the simulation engine implements a plane fixed temperature boundary, future improved experimental setup should be adapted to ensure that this approximation is adequate.

The analysis of the described experiment also suggests some improvements of the software part: Firstly, the simulation engine should be improved in order to manage with materials having a thermal conductivity depending on the temperature such as the silicon of the thermal test chip. It has been also derived that the simulation engine could include a detailed model of the platinum resistance temperature sense reproducing the experimental temperature readings across an extensive area. Also a more in deep study of the transient thermal behavior of the temperature sensor on the thermal test chip is advisable.

Another proposal in order to simplify the data acquisition is to perform a heat step-down; i.e., the system should be settled dissipating a fixed heat power and when the thermal equilibrium is reached, the power should be turned off at a given instant. This step involves a simpler experimental setup with no need of the power controller. However, the implementation of the software reproducing this by thermal simulation would become more complex. The steady state thermal equilibrium should be simulated prior every transient thermal simulation. This could become excessively time-consuming.

4.4 Conclusion

The experimental setup for transient heating curves acquisition has been described in this chapter. Firstly, as an essential part of the experimental setup, the thermal test chip that integrates the heat power dissipation and temperature measurement has been described. It has been conceived in the context of this work, thus its design and implementation have been described in detail. The design rules to implement the platinum temperature sensor and the heating polysilicon resistance of the device have been also described together with the technology process overview. As a part of the design, detailed thermal simulations of the device have been carried out in order to propose some tips for its thermal simulation by the *CescThParId 4.1*. It is concluded that it can be considered a uniformly distributed heat dissipation across the polysilicon area. However, the platinum resistance area geometry can not be simplified. Last, an experimental calibration procedure of the integrated platinum temperature sensor has been also proposed.

The complete thermal and electrical setup and procedure to obtain a temperature transient data record have been thus described. The thermal setup that is based on the one used in the thermal conductivity method, adding the thermal test chip and the substrate preparation, also addressed. The electrical setup that makes possible the step-wise power generation in the

thermal test chip is described in detail. It has been stated that the integral controller, which design is exposed within the thermal conductivity steady state measurement method description, is needed to perform the power step. It has been proved that, although the heater is an electrical resistance, the feedback control operation is essentially similar than for a power transistor. The complete procedure to obtain a transient temperature record together with the power generation level and initial temperature using the described equipment has been also described. The adaptation of the read data for its optimal use by the *CescThParId 4.1* software has been exposed.

A thermal parameter identification using the described thermal parameter identification program and the experimental setup has been carried out. The material under test has been the alumina ceramic (aluminum oxide) of a DCB substrate from *IXYS Corporation*. The details of the experimental transient temperature recording are exposed together with the detailed settings and options introduced to the identification program. This includes the geometry and the initial parameter values. The identification process results have been transcribed together with the interpretation of the obtained parameter values.

The objective of this identification method is to extract thermal parameter values that lead to thermal simulations fitting the real experimental behavior. The presented process has resulted in a set of thermal parameters that reproduce in detail the temperature experimental transient curve. This could not be achieved with the typical values from data sheets and literature. Thus, the described process has been concluded successfully. However, the experience acquired during the tests operation of the developed method give rise to other conclusions, which contemplate the improvement of both the software and the experimental setup of the system. Future work suggestions for the experimental setup and the simulation capabilities improvement have been also addressed.

Chapter 5

Conclusions

The work exposed in this document proposes two methods for the measurement of the thermal properties of materials, mainly for those devoted to the Power Electronics packaging. It has been developed at the *Centre Nacional de Microelectrònica - CSIC*.

In the first chapter, a brief introduction to the Thermal Management and its importance for the Power Electronics packaging has been presented together with a description of the motivations of this work. Thermal property values from literature may not fit the actual ones of a particular material used in a given design. Consequently, predictions from thermal simulation tools would not provide the necessary accuracy to achieve the design aims imposed by the increasing thermal performance requests. The fact that *in-situ* measurements are essential for obtaining accurate thermal parameter values for particular materials has been stated.

The first proposed method is an effective thermal conductivity extraction system which is described in chapter two. Its operation principle, design, implementation and experimental results are detailed. The effective thermal conductivity of the material is deduced from a thermal resistance differential measurement on two samples with different thicknesses. The setup is placed inside a vacuum cavity, using thermal interface materials with a pressure system to assure thermal contact between the heat source, the sample and the heat sink. Accurate temperature measurements are carried out with PT100 temperature probes. To assure an appropriate heat conduction through the sample, reducing heat flow leakage and performing accurate temperature readings have been important design issues that have lead to accurate measured values. Thus, the experimental error of the measurement has been evaluated and minimized. It has been found that the method can keep the relative experimental error below 10% if the measured virtual thermal resistance is higher than $0.04\,\mathrm{K/W}$. The method implementation includes an

innovative heat power generation controller, designed as part of the measurement system. This integral controller circuit, described in detail, is optimized to drive a power MOSFET in the context of this work. Nevertheless, since it is a versatile experimental tool, other voltage controlled or gated devices may be driven, being suitable for other experimental work needing a controlled heat power source. The implemented measurement method has been validated with thermal conductivity measurements on a BN anisotropic ceramic, an AlSiC composite and an AlN ceramic, evidencing its capability to obtain values for materials with thermal conductivities up to 170 W/ m K. Values with a good accuracy and very good repeatability, with variations within the experimental error, have been obtained. The method can extract thermal conductivity values at ambient temperature only. However, the main advantages of the proposed steady state method is its versatility and its low cost. The method permits having immediate characterization of materials during their development or their manufacturing process. The setup also gives the possibility to perform comparative material studies such as the one carried out over a set of eight different thermal interface materials included in this work.

Secondly, an innovative thermal parameter identification method, described along the third and fourth chapters has been developed. The method allows to determine both the specific heat and thermal conductivity of a sample material from an experimental transient temperature record. Chapter three describes in detail the principle of operation of the method, the identification algorithm and the implementation of the software with MATLAB®. An experimental transient temperature record is compared systematically with a 3D thermal simulation and the thermal parameter values for the simulation are identified in an iterative computer-based process. Other parameter identification methods compare experimental curves with RC circuits or simple models. However, the presented work uses the most realistic and complex model; the 3D thermal simulation. This involves the design of a thermal simulation engine as part of the developed thermal parameter identification software. The heat conduction equations are solved numerically using the Cranck-Nicholson scheme as it has been detailed in this work. The multiple parameter identification algorithm implementation has been also described in detail. Although this method is very time-consuming, the resulting parameter values permit reproducing experimental curves by thermal simulation with extreme accuracy. A validation of the process has been carried out, concluding that the 3D thermal simulation capabilities of the program offer fine results in accordance with a commercial program (FLOTHERM™ 4.2). The algorithm capability of extracting thermal parameter values consistent with its internal simulation engine and with FLOTHERM™ 4.2 has been also

checked succesfully. The existence of multiple results and complex dynamics have been encountered during the validation tests and some recommendations to avoid them have been derived. The identified material should hold a significant part of the overall thermal resistance and the total thermal capacity of the thermal system under test. The best available approximation of the thermal properties for the identified material must be used as initial values. Nevertheless, during the self-consistence tests initial values up to 100% above and 50% below the nominal ones have been introduced and correctly identified. The non identified thermal properties of the rest of materials of the thermal system have to be well known. It is also highly recommended to reduce the number of identified parameter values to a single thermal conductivity and a single specific heat value. Some future work suggestions are pointed out, mainly channeled into the result convergence improvement and the reduction of the time-consumption.

Chapter four describes the experimental setup for the acquisition of transient heating curves. A low cost thermal test chip, integrating heat power dissipation and temperature measurement in the same semiconductor device has been designed and manufactured in the *Centre Nacional de Microelectrònica - CSIC* facilities as an essential part of the experimental setup. The complete thermal and electrical setup with the procedure to obtain a transient record have been described. The power controller described in chapter two is used also to generate the step-wise heat power. A thermal parameter identification of the alumina ceramic of a DCB substrate has been carried out. All experimental and software settings are described in detail as well as the results of the identification. It has been found that the extracted thermal parameters for the alumina ceramic do not match exactly the expected ones (from the data-sheet). Nevertheless, they lead to thermal simulations that accurately reproduce the temperature experimental transient curve. Thus, the aim of the transient thermal parameter identification method has been achieved. Future work suggestions for the improvement of the method have been finally discussed. They are channeled into the improvement of the experimental setup, in order to facilitate its thermal simulation, and to the improvement of the simulation capabilities in reproducing the real experiment conditions.

The virtues of the proposed measurement methods are their low cost and the fact that most of their parts, designed in the context of this work, are versatile enough to give experimental solutions to other Thermal Management related works. In this sense, the power controller circuit and the thermal test chip are themselves two innovative experimental tools. The power controller circuit together with the thermal test chip could be used for the studies on the die attach quality. In a similar way, the thermal parameter identifica-

tion tool could be applied to other applications different from the exposed here. The parameters of virtually any thermal system that could be simulated by the simulation engine can be extracted. This opens a new possibility in modeling already working modules and packages under realistic working conditions, or even other thermal systems of interest. Modification of the original designs exposed here can greatly increase the number of alternative applications.

Appendix A

Source Code

A.1 File: identification.m

```
1  %Function IDENTIFICATION performs a parameter identification.

3  %Input variables:
4  %   nomfitxerconfigidentifica, is the name of the identification
       configuration file.
5  %   nomfitxerlog, is the name of the log file.

7  %Output variables:
8  %   tempstrigat, the time that the process has taken.
9  %   control, controls if an error is found.

11 function [ control, tempstrigat] = identification(
       nomfitxerconfigidentifica, nomfitxerlog)

13 %The current project is readed from the 'info' file.
14 fid=fopen('info', 'r');
15 if fid==-1
16     fprintf('\n***ERROR EN LA IDENTICICACIO***: No hi ha cap
           projecte definit.');
17     control=-1;
18     fclose('all');
19     return;
20 end
21 nomprojecte=fscanf(fid, '%s');
22 fclose(fid);

24 %Initiates a chronometer.
25 momentinicial=clock;

27 %The variables are initialized.
28 control=-1;
29 Tmesura=0;
30 Tvariada=0;
31 Tvariadaadap=0;
32 Tsimulada=0;
33 Tsimuladaadap=0;
34 tempstrigat=0;
35 DT=0;
36 T=0;

38 %Depending on the input parameters, the log file is opened or the
       log information is printed on the screen.
39 if nargin==0
40     control=-1;
41     fprintf('Has d''introduir el nom del fitxer de configuracio de
           la identificacio i opcionalment el nom d''un fitxer de log
           .');
```

```
42      fprintf('\nPer exemple: identificacio
           NomFitxerConfigIdentificacio.txt NomFitxerLog.txt\nLlegeix
           el fitxer d''instruccions adjunt.');
43      return
44  elseif nargin==1
45      fidlog=1;
46  elseif nargin==2
47      %Adaption of nomfitxerlog to the project path.
48      nomfitxerlog=[nomprojecte '/' nomfitxerlog];
49      fidlog=fopen( nomfitxerlog , 'w');
50  end

52  %The log is initiated.
53  fprintf(fidlog, '\n***INICI DE LA IDENTIFICACIO***\nProjecte:%s',
       nomprojecte);
54  fprintf(fidlog, '\n\nEs comproven els fitxers de configuracio i les
       dades de la mesura...');

56  %Adaption of nomfitxerconfigidentifica
57  nomfitxerconfigidentifica=[nomprojecte '/' nomfitxerconfigidentifica
       ];

59  %Reading of files to find eventual errors and extracting information
       .
60  %The identification configuration file is readed.
61  [ control, nomfitxerconfig, nomfitxermesura, liniesaignorar,
       caracterseparacio, nomfitxerresultat, nomfitxerconfigresultat,
       nomfitxerconfigvariacio, numiteracions ] = idfileread(
       nomfitxerconfigidentifica, fidlog );
62  if control==-1;
63      fprintf(fidlog, '\n***ERROR EN LA IDENTICICACIO***: Error al
           llegir la configuracio de la identificacio, no es pot
           prosseguir amb el proces.');
64      control=-1;
65      fclose('all');
66      return;
67  end

69  %All names of files are adapted to the project path.
70  nomfitxerconfig=[nomprojecte '/' nomfitxerconfig];
71  nomfitxermesura=[nomprojecte '/' nomfitxermesura];
72  nomfitxerresultat=[nomprojecte '/' nomfitxerresultat];
73  nomfitxerconfigresultat=[nomprojecte '/' nomfitxerconfigresultat];
74  nomfitxerconfigvariacio=[nomprojecte '/' nomfitxerconfigvariacio];

76  %The geomety file is readed.
77  [ control, nmaterials, dimmaterials, ndimensions, conductivitat,
       cespecifica, densitat, total, ntotal, npotencia, dompotencia,
       ndompotencia, potencia, ptemp, nptemp, tempinicial, temps,
```

```
    ntemps, factor, condaidentificar, cespaidentificar] =
    simfileread ( nomfitxerconfig, fidlog );
if control==-1;
    fprintf(fidlog, '\n***ERROR EN LA IDENTICICACIO***: Error al
        llegir dades del fitxer de configuracio del sistema, no es
        pot prosseguir amb el proces.');
    control=-1;
    fclose('all');
    return;
end

%The variable paramidentif is deduced (parameters to be identified).
paramidentif=[condaidentificar, cespaidentificar];
if paramidentif==0
    fprintf(fidlog, '\n***ERROR EN LA IDENTICICACIO***: No vols
        identificar cap parametre? El proces s''ha aturat.');
    control=-1;
    fclose('all');
    return;
end

%The experimental data file is readed.
[ control, Tmesura] = expfileread ( nomfitxermesura,
    caracterseparacio, liniesaignorar, fidlog);
if control==-1;
    fprintf(fidlog, '\n***ERROR EN LA IDENTICICACIO***: Error al
        llegir dades de la mesura, no es pot prosseguir amb el
        proces.');
    control=-1;
    fclose('all');
    return;
end
fprintf(fidlog, '\nOK');

%Checking that the measurement time is lower than the simulation
    time.
if max(Tmesura(:,1))>temps
     fprintf(fidlog, '\n***ERROR EN LA IDENTIFICACIO***: El temps de
         simulacio es menor que el temps de la mesura.');
     control=-1;
     return;
end

%End of the file checking.

%The result file is opened/created.
fidres=fopen(nomfitxerresultat, 'w');
fprintf(fidres, '***Fitxer de registre dels parametres resultats de
    la identificacio parametrica***');
```

```
116 fprintf(fidres, '\niteracio');
117 for n=1:nmaterials
118     fprintf(fidres, ['\tCond mat' int2str(n) '\tCalesp mat' int2str(
        n) ]);
119 end
120 fprintf(fidres, '\n0\t');
121 for n=1:nmaterials
122     fprintf(fidres, '\t%f\t%f', conductivitat(n), cespecifica(n));
123 end

125 %The variable config stores the file name of the current geometry
    file over parameter modifications.
126 config=nomfitxerconfig;

128 %BEGINING OF THE ITERATIVE PROCESS.
129 for iteracio=1:numiteracions
130     tempstrigat=etime(clock,momentinicial);
131     fprintf(fidlog, ['\n\nInici de l''iteracio num:' int2str(
        iteracio) ]);
132     fprintf(fidlog, '\nHan transcorregut %d hores %d minuts i %d
        segons des de l''inici de la identificacio.', floor(
        tempstrigat/3600), floor(mod(floor(tempstrigat),3600)/60),
        mod(floor(tempstrigat),60));

134     %The current geometry file is readed.
135     [control, nmaterials, dimmaterials, ndimensions, conductivitat,
        cespecifica, densitat, total, ntotal, npotencia, dompotencia
        , ndompotencia, potencia, ptemp, nptemp, tempinicial, temps,
         ntemps, factor, condaidentificar, cespaidentificar] =
        simfileread ( config, fidlog );
136     if control==-1;
137         fprintf(fidlog, '\n***ERROR EN LA IDENTICICACIO***: Error al
              llegir dades del fitxer de configuracio del sistema, no
              es pot prosseguir amb el proces.');
138         control=-1;
139         fclose('all');
140         return;
141     end

143     %Simulation of the current geometry with the current parameters.
144     fprintf(fidlog, '\n\nSimulacio del model...');
145     [control, tempssimulacio, Tsimulada] = simule(config,
        nomprojecte, fidlog);
146     if control==-1;
147         fprintf(fidlog, '\n***ERROR EN LA IDENTICICACIO***: Error al
              realitzar la simulacio, no es pot prosseguir amb el
             proces.');
148         control=-1;
149         fclose('all');
```

```
150         return;
151     end

153     %Simulations with variated parameters to obtain Tvariada
154     Tvariada=zeros(ntemps+1, nmaterials*2+1);
155     %1% is added to conductivities and specific heats.
156     condvariat=1.01*conductivitat;
157     cespvariat=1.01*cespecifica;
158     Tvariada(:,1)=Tsimulada(:,1);
159     for n=1:nmaterials
160         if condaidentificar(n)
161             condmodif=conductivitat;
162             condmodif(n)=condvariat(n);
163             varsimfilewrite(nomfitxerconfig, [
                    nomfitxerconfigvariacio 'Cond' int2str(n) '.txt'],
                    nmaterials, condmodif, cespecifica);

165             fprintf(fidlog, ['\n\nSimulacio de la variacio de la
                    Conductivitat del material ' int2str(n) '.']);
166             [ control, tempssimulacio, T] = simule( [
                    nomfitxerconfigvariacio 'Cond' int2str(n) '.txt'] ,
                    nomprojecte, fidlog);
167             Tvariada(:, n+1)=T(:,2);
168         end

170         if cespaidentificar(n)
171             cespmodif=cespecifica;
172             cespmodif(n)=cespvariat(n);
173             varsimfilewrite(nomfitxerconfig, [
                    nomfitxerconfigvariacio 'Cesp' int2str(n) '.txt'],
                    nmaterials, conductivitat, cespmodif);

175             fprintf(fidlog, ['\n\nSimulacio de la variacio de la
                    Calor especifica del material ' int2str(n) '.']);
176             [ control, tempssimulacio, T] = simule( [
                    nomfitxerconfigvariacio 'Cesp' int2str(n) '.txt'] ,
                    nomprojecte, fidlog);
177             Tvariada(:, n+nmaterials+1)=T(:,2);
178         end
179     end
180     %All results are stored.
181     fidres=fopen( [nomprojecte '/' int2str(iteracio) 'simul.txt'], '
            w');
182     fprintf(fidres, '***Resultat de la simulacio de la configuracio
            %s (iteracio %d)***', config, iteracio);
183     fprintf(fidres, '\n%s\t\t\t\t%s', 'Temps', 'Temperatura' );
184     for n=1:length(Tsimulada(:,1))
185         fprintf(fidres, '\n');
186         fprintf(fidres, '%.20f\t', Tsimulada(n,:));
```

```
187     end
188     fclose(fidres);
189     fidres=fopen( [nomprojecte '/' int2str(iteracio) 'simulvari.txt
            '], 'w');
190     fprintf(fidres, '***Resultat de la simulacio de la configuracio
           %s variada (iteracio %d)***', config, iteracio);
191     fprintf(fidres, '\n%s\t\t\t\t%s', 'Temps', 'Temperatures (
            conductivitats ; calors especifics)' );
192     for n=1:length(Tvariada(:,1))
193         fprintf(fidres, '\n');
194         fprintf(fidres, '%.20f\t', Tvariada(n,:));
195     end
196     fclose(fidres);


199     %The simulated temperatures are interpolated to the experimental
            time-steps.
200     Tsimuladaadap=Tmesura(:,1);
201     Tsimuladaadap(:,2)=interp1(Tsimulada(:,1), Tsimulada(:,2),
           Tmesura(:,1), 'spline');
202     Tvariadaadap=Tmesura(:,1);
203     for n=1:2*nmaterials
204         Tvariadaadap(:,n+1)=interp1(Tvariada(:,1), Tvariada(:,n+1),
               Tmesura(:,1), 'spline');
205     end

207     %Calculation of DT.
208     DT=Tsimuladaadap(:,1);
209     for n=find(paramidentif)+1
210         DT(:,n)=Tvariadaadap(:,n)-Tsimuladaadap(:,2);
211     end

213     %Direct solution of the equation system DT(:,2:2*nmaterials+1)*
           components = (Tmesura(:,2)-Tsimulada(:,2))
214     components=DT(:,find(paramidentif)+1)\(Tmesura(:,2)-
           Tsimuladaadap(:,2));

216     %Control over the variable components, the min and max is found.
217     [valormax,indexmax]=max(components);
218     if valormax>1000
219         components=components/abs(components(indexmax))*1000;
220         fprintf(fidlog, '\nUn o mes parametres proven d''incrmentar-
               se mes del 1000%, s''ha impedit');
221     end
222     [valormin,indexmin]=min(components);
223     if valormin<-90
224         components=components/abs(components(indexmin))*90;
225         fprintf(fidlog, '\nUn o mes parametres proven de disminuir a
                menys d''un 10%, s''ha impedit');
```

```
226     end

228     %The vector components is adapted for the parameter modification
            .
229     componentsmodif=zeros(2*nmaterials,1);
230     componentsmodif(find(paramidentif))=components;

232     %The paremeters are modified and are written with resultwrite,
            to the result file and resulting configuration file.
233     condmodif=(1+componentsmodif(1:nmaterials)/100)'.*conductivitat;
234     cespmodif=(1+componentsmodif((nmaterials+1):2*nmaterials)/100)
            '.*cespecifica;

236     resultwrite(iteracio, nomfitxerconfig, nomfitxerconfigresultat,
            nomfitxerresultat, nmaterials, condmodif, cespmodif);

238     config=nomfitxerconfigresultat;

240     %Control to exit the iterative process
241     if (abs(condmodif-conductivitat)<0.0000005 & abs(cespmodif-
            cespecifica)<0.0000005)
242         fprintf(fidlog, '\n***S''ha çforat la fi de les iteracions,
                tot ha convergit a la iteracio %d.***', iteracio);
243         break;
244      end

246 end %End of the iteration

248 %The temporal filea are deleted.
249 for n=find(condaidentificar)
250     delete( [nomfitxerconfigvariacio 'Cond' int2str(n) '.txt'] );
251 end
252 for n=find(cespaidentificar)
253     delete( [nomfitxerconfigvariacio 'Cesp' int2str(n) '.txt'] );
254 end

256 tempstrigat=etime(clock,momentinicial);
257 fprintf(fidlog, '\n\n***S''ha finalitzat amb exit l''identificacio
        parametrica en %d hores %d minuts i %d segons.***', floor(
        tempstrigat/3600), floor(mod(floor(tempstrigat),3600)/60), mod(
        floor(tempstrigat),60));

259 fclose('all');%Al files are closed.
260 control=0;%No errors found so far.
261 return %End of the function.
```

A.2 File: simulation.m

```
1  %Main file of the application SIMULATION that performs a 3D thermal
       simulation.

3  %Input variables are:
4  %   nomfitxersimulacio, the name of the simulation geometry file.
5  %   nomfitxerresultat, the name of the file where the result is
       written.
6  %   nomfitxerlog, the name of the log file.

8  %Output variable is:
9  %       control, controls if an error is found.

11 function [control] = simulation( nomfitxersimulacio,
       nomfitxerresultat, nomfitxerlog)

13 %Current project location is readed
14 fid=fopen('info', 'r');
15 if fid==-1
16     fprintf('\n***ERROR EN LA SIMULACIO***: No hi ha cap projecte
           definit.');
17     control=-1;
18     fclose('all');
19     return;
20 end
21 nomprojecte=fscanf(fid, '%s');
22 fclose(fid);

24 %Check the geometry and result files and opens the log file if a
       name is provided (otherwise the comments are printed on the
       screen).
25 if nargin==0|nargin==1
26     control=-1;
27     fprintf('Has d''introduir el nom del fitxer de configuracio de
           la simulacio, el d''on es registrara el resultat i
           opcionalment el nom d''un fitxer de log.');
28     fprintf('\nPer exemple: simulation NomFitxerConfigSimulacio.txt
           NomFitxerResultat.txt NomFitxerLog.txt\nLlegeix el fitxer d
           ''instruccions adjunt.');
29     return
30 elseif nargin==2
31     fidlog=1;
32 elseif nargin==3
33     nomfitxerlog=[nomprojecte '/' nomfitxerlog];
34     fidlog=fopen( nomfitxerlog , 'w'); %'w' vol dir que ignora el
           contingut anterior.
35 end
```

```
37 %Initiates the log file.
38 fprintf(fidlog, '\n***INICI DE LA SIMULACIO***\nProjecte:%s',
      nomprojecte);

40 %Inserts the location path of the project to the file name.
41 nomfitxersimulacio=[nomprojecte '/' nomfitxersimulacio];

43 %The function simule performs the 3d thermal simulation and returns
      the result. The success of the simulation is checked with the
      variable control.
44 [control, tempssimulacio, T] = simule( nomfitxersimulacio,
      nomprojecte, fidlog);
45 if control==-1
46     fclose('all');
47     return;
48 end

50 %The result is written to the result file.
51 nomfitxerresultat =[ nomprojecte '/' nomfitxerresultat];
52 fidres=fopen( nomfitxerresultat , 'w');
53 fprintf(fidres, '***Resultat de la simulacio de la configuracio %s
      ***', nomfitxersimulacio);
54 fprintf(fidres, '\n%s\t\t\t\t%s', 'Temps', 'Temperatura' );
55 for n=1:length(T(:,1))
56     fprintf(fidres, '\n');
57     fprintf(fidres, '%.20f\t', T(n,:));
58 end

60 fprintf(fidlog, '\n***S''ha finalitzat amb exit la simulacio***');

62 %All files are closed
63 fclose('all');

65 %So far no errors are found.
66 control=0;

68 return %End of the function simulation
```

A.3 File: project.m

```
%Function PROJECT that sets or reads the current project path in the 'info' file.

%Input variable is:
%       nomprojecte, the current project location path introduced by the user.

%Output variables are:
%       control, controls if an error is found.
%       nomprojecte, the current project location path given to the calling program.

function [ control, nomprojecte ] = project( nomprojecte )
if nargin==0
    fid=fopen('info', 'r');
    if fid==-1
        fprintf('\nNo hi ha cap projecte definit!\n\n');
        control=-1;
        fclose('all');
        return;
    end
    nomprojecte=fscanf(fid, '%s');
    fprintf(1, '\n\nEl projecte en curs es: %s\n\n', nomprojecte);
    fclose('all');
else
    nomprojecte=['../' nomprojecte];

    fid=fopen('info', 'w');
    fprintf(fid, nomprojecte);
    fprintf(1, '\n\nEl projecte en curs es: %s\n\n', nomprojecte);
    fclose('all');
end

control=0; %No errors found
return; %End of the function
```

A.4 File: simule.m

```
1  %Function SIMULE that performs a 3D thermal simulation. It is used
       by the functions simule and identification.

3  %Input variables are:
4  %       nomfitxer, the name of the simulation geometry file.
5  %       nomprojecte, the current project location path.
6  %       fidlog, the name of the log file.

8  %Output variables are:
9  %       control, controls if an error is found.
10 %       tempssimulacio, simulation time in seconds.
11 %       T, array with the temperature evolution at the measuring
       points.

13 function [ control, tempssimulacio, T] = simule( nomfitxer,
       nomprojecte, fidlog )
14 tic %Initiates a chronometer.

16 %The output variables are initialized in case of error.
17 tempssimulacio=toc;
18 T=0;

20 %Es llegeix la configuracio del sistema a simular amb la funcio
       simfileread.
21 [control, nmaterials, dimmaterials, ndimensions, conductivitat,
       cespecifica, densitat, total, ntotal, npotencia, dompotencia,
       ndompotencia, potencia, ptemp, nptemp, tempinicial, temps,
       ntemps, factor, condaidentificar, cespaidentificar ] =
       simfileread ( nomfitxer, fidlog );
22 if control==-1;
23     fprintf(fidlog, '\n***ERROR DURANT LA SIMULACIO TERMICA***: Hi
           ha algun error a l''arxiu de configuracio que impedeix
           realitzar la simulacio.');
24     control=-1;
25     return;
26 end

28 %Volume element dimensions.
29 dx=total(1)/ntotal(1);
30 dy=total(2)/ntotal(2);
31 dz=total(3)/ntotal(3);

33 %The T variable is initialized with its dimensions.
34 %       t=0 (n=1) up to t=temps (n=ntemps+1).
35 npt=size(nptemp,2);
36 T=zeros(ntemps+1,npt+1);
37 T(1,2:npt+1)=tempinicial;
```

```
39 %Power dissipation density.
40 volumpotencia=0;
41 for n=1:npotencia
42     volumpotencia=volumpotencia+((dompotencia(6,n)-dompotencia(5,n))
          *(dompotencia(4,n)-dompotencia(3,n))*(dompotencia(2,n)-
          dompotencia(1,n)));
43 end
44 dpot=potencia/volumpotencia;

46 %A z node is added to implement a Dirichlet boundary in z=Ztot.
47 ntotal(3)=ntotal(3)+1;

49 %File names where temporary data is stored.
50 nomtemp=[nomprojecte '/temp'];
51 nomcond=[nomprojecte '/cond'];
52 nomalfa1=[nomprojecte '/alfa1'];
53 nomalfa2=[nomprojecte '/alfa2'];

55 %Creating the matrices temperature (temp), thermal conductivity (
      cond), alfa1 and alfa2 in multiband files.
56 multibandwrite(tempinicial, nomtemp,  'bsq', [1 1 1], [ntotal(1)
      ntotal(2) ntotal(3)], 'precision', 'double', 'offset', 0, '
      fillvalue', tempinicial)
57 multibandwrite(0,           nomcond,  'bsq', [1 1 1], [ntotal(1)
      ntotal(2) ntotal(3)], 'precision', 'double', 'offset', 0, '
      fillvalue', 0)
58 multibandwrite(0,           nomalfa1, 'bsq', [1 1 1], [ntotal(1)
      ntotal(2) ntotal(3)], 'precision', 'double', 'offset', 0, '
      fillvalue', 0)
59 multibandwrite(0,           nomalfa2, 'bsq', [1 1 1], [ntotal(1)
      ntotal(2) ntotal(3)], 'precision', 'double', 'offset', 0, '
      fillvalue', 0)

61 %Giving values to all elements of matrices alfa1, alfa2 and cond.
62 for n=1:nmaterials %For every material
63     cond=conductivitat(n)*ones(ndimensions(2,n)-ndimensions(1,n)+1,
          ndimensions(4,n)-ndimensions(3,n)+1);
64     cesp=cespecifica(n)  *ones(ndimensions(2,n)-ndimensions(1,n)+1,
          ndimensions(4,n)-ndimensions(3,n)+1);
65     dens=densitat(n)     *ones(ndimensions(2,n)-ndimensions(1,n)+1,
          ndimensions(4,n)-ndimensions(3,n)+1);
66     alfa1=dens.*cesp*2;
67     for k=ndimensions(5,n):ndimensions(6,n)
68         multibandwrite(cond , nomcond    , 'bsq', [ndimensions(1,n)
              ndimensions(3,n) k], [ntotal(1) ntotal(2) ntotal(3)], '
              precision', 'double', 'offset', 0)
69         multibandwrite(alfa1, nomalfa1   , 'bsq', [ndimensions(1,n)
              ndimensions(3,n) k], [ntotal(1) ntotal(2) ntotal(3)], '
```

```
          precision', 'double', 'offset', 0)
70    end
71 end

73 for n=1:npotencia
74     for k=ndompotencia(5,n):ndompotencia(6,n) %Within the power
           dissipation volume
75         alfa2=2*dpot*ones(ndompotencia(2,n)-ndompotencia(1,n)+1,
               ndompotencia(4,n)-ndompotencia(3,n)+1);
76         multibandwrite(alfa2, nomalfa2, 'bsq', [ndompotencia(1,n)
               ndompotencia(3,n) k], [ntotal(1) ntotal(2) ntotal(3)], '
               precision', 'double', 'offset', 0 );
77     end
78 end

80 %End of the alfa1, alfa2 and temp multiband files initialization.

82 %Begin of the thermal simulation.
83 %Initialization of some variables to accelerate calculations in the
       'k slices' (Cranck Nicholson in X and Y directions).
84      alfa1xy=zeros(ntotal(1), ntotal(2));
85      condxy=alfa1xy;
86      tempxy=alfa1xy;
87      Ax=zeros(ntotal(1));
88      Bx=zeros(ntotal(1));
89      Bvectx=zeros(ntotal(1),1);
90      Ay=zeros(ntotal(2));
91      By=zeros(ntotal(2));
92      Bvecty=zeros(ntotal(2),1);
93 %Initialization of auxiliary variables (used when a void found)
94      alfa1auxx=0;
95      tempauxx=0;
96      condauxx=0;
97      rangauxx=0;
98      desdeauxx=0;
99      finsaauxx=0;
100     alfa1auxy=0;
101     tempauxy=0;
102     condauxy=0;
103     rangauxy=0;
104     desdeauxy=0;
105     finsaauxy=0;

107 %Initialization of some variables to accelerate calculations in the
       'i slices' (Cranck Nicholson in Z direction).
108     alfa1z=zeros(ntotal(3), ntotal(2));
109     alfa2z=alfa1z;
110     tempz=alfa1z;
111     condz=alfa1z;
```

```
112       Az=zeros(ntotal(3));
113       Bz=zeros(ntotal(3));
114       Bvectz=zeros(ntotal(3),1);
115 %Initialization of auxiliary variables (used when a void found)
116       alfa1auxz=0;
117       alfa2auxz=0;
118       tempauxz=0;
119       condauxz=0;
120       rangauxz=0;
121       desdeauxz=0;
122       finsaauxz=0;

124 %Initialization of variable tempptemp for the temperature '
      measurement' in the monitor point.
125 tempptemp=zeros(ntotal(1), ntotal(2), 2);

127 %Begin of the on screen progress counter.
128 fprintf(1, ['\n' nomfitxer ', complet:    ']);

130 %Begin of the simulation time discretization.
131 for n=1:ntemps %For every moment.

133 %Time step size.
134     dt=factor^(n-1)*temps*(factor-1)/(factor^ntemps-1);

136       %Cranck-Nicholson in Z direction.
137       for i=1:ntotal(1) %'Slices' in direction X.

139             tempz =multibandread(nomtemp,  [ntotal(3), ntotal(2),
                    ntotal(1)], 'double', 0, 'bil', 'native', {'Band', '
                    Direct', i});
140             alfa1z=multibandread(nomalfa1, [ntotal(3), ntotal(2),
                    ntotal(1)], 'double', 0, 'bil', 'native', {'Band', '
                    Direct', i});
141             alfa2z=multibandread(nomalfa2, [ntotal(3), ntotal(2),
                    ntotal(1)], 'double', 0, 'bil', 'native', {'Band', '
                    Direct', i});
142             condz =multibandread(nomcond,  [ntotal(3), ntotal(2),
                    ntotal(1)], 'double', 0, 'bil', 'native', {'Band', '
                    Direct', i});

144             for j=1:ntotal(2) %Covering every y in this 'slice'
145             %One dimension simulation through k values for fixed i
                    and j.
146                if size(nonzeros(alfa1z(:,j)),1)==(ntotal(3)-1)
147                %In case of no void.
148                   %Creation of matrices Az and Bz.
149                   for m=2:ntotal(3)-1
150                       Az(m,m-1)=-condz(m-1,j);
```

```
151                     Az(m,m)  =alfa1z(m,j)*dz^2/dt+condz(m,j)+condz(
                            m-1,j);
152                     Az(m,m+1)=-condz(m,j);
153                     Bz(m,m-1)=condz(m-1,j);
154                     Bz(m,m)  =alfa1z(m,j)*dz^2/dt-condz(m,j)-condz(
                            m-1,j);
155                     Bz(m,m+1)=condz(m,j);
156                 end
157                 %Boundaries in matrices Az and Bz
158                 %Adiabatic surface in Z=0
159                 Az(1,1)=alfa1z(1,j)*dz^2/dt+condz(1,j);
160                 Az(1,2)=-condz(1,j);
161                 Bz(1,1)=alfa1z(1,j)*dz^2/dt-condz(1,j);
162                 Bz(1,2)=condz(1,j);
163                 %Constant temperature in z=ZTotal.
164                 Az(ntotal(3), ntotal(3))=1;
165                 Bz(ntotal(3), ntotal(3))=1;
166                 %Vector Bvectz.
167                 Bvectz=Bz*tempz(:,j)+alfa2z(:,j)*dz^2;
168                 %Direct solution of the equation system AT=B => T=A
                        \B
169                 tempz(:,j)=Az\Bvectz;

171             elseif size(nonzeros(alfa1z(:,j)),1)~=0
172             %If void found (alfa1=0).
173                  %Auxiliary variables to create auxiliary matrices.
174                  alfa1auxz=nonzeros(alfa1z(:,j));
175                  rangauxz=size(alfa1auxz,1)+1;
176                  alfa1auxz(rangauxz)=0;
177                  desdeauxz=min(find(alfa1z(:,j)));
178                  finsaauxz=max(find(alfa1z(:,j)))+1;
179                  alfa2auxz=alfa2z(desdeauxz:finsaauxz, j);
180                  condauxz=condz(desdeauxz:finsaauxz, j);
181                  tempauxz=tempz(desdeauxz:finsaauxz, j);
182                  %Creation of auxiliary matrices Aauxz and Bauxz
183                  Aauxz=zeros(rangauxz);
184                  Bauxz=zeros(rangauxz);
185                  for m=2:rangauxz-1
186                     Aauxz(m,m-1)=-condauxz(m-1);
187                     Aauxz(m,m)  =alfa1auxz(m)*dz^2/dt+condauxz(m)+
                            condauxz(m-1);
188                     Aauxz(m,m+1)=-condauxz(m);
189                     Bauxz(m,m-1)=condauxz(m-1);
190                     Bauxz(m,m)  =alfa1auxz(m)*dz^2/dt-condauxz(m)-
                            condauxz(m-1);
191                     Bauxz(m,m+1)=condauxz(m);
192                 end
193                 %Boundaries for the matrices Aauxz i Bauxz
194                 %Adiabatic boundary in Z=begin of the material
```

```
                    media.
195                 Aauxz(1,1)=alfa1auxz(1)*dz^2/dt+condauxz(1);
196                 Aauxz(1,2)=-condauxz(1);
197                 Bauxz(1,1)=alfa1auxz(1)*dz^2/dt-condauxz(1);
198                 Bauxz(1,2)=condauxz(1);
199                 %Constant temperature in z=ZTotal.
200                 Aauxz(rangauxz, rangauxz)=1;
201                 Bauxz(rangauxz, rangauxz)=1;
202                 %Vector Bvectx.
203                 Bvectauxz=Bauxz*tempauxz+alfa2auxz*dz^2;
204                 %Direct solution of the equation system AT=B => T=A
                        \B
205                 tempz(desdeauxz:finsaauxz,j)=Aauxz\Bvectauxz;
206             else
207                 fprintf(fidlog, 'ADVERTENCIA DURANT LA SIMULACIO
                        TERMICA: Hi ha part de la seccio en Z sense
                        material\n');
208             end %End of the 'in case of void' condition
209         end %End of the Y covering.
210         %The temperature resulting of this slice is saved.
211         multibandwrite(tempz, nomtemp, 'bil', [1 1 i], [ntotal(3)
                ntotal(2) ntotal(1)], 'precision', 'double', 'offset
              ', 0);
212     end    %End of the 'slices' in X (i) and the Cranck-Nicholson
            in Z.

214     %Cranck-Nicholson in directions X and Y.
215     for k=1:ntotal(3)-1 %'slices' in direction Z
216         tempxy =multibandread(nomtemp , [ntotal(1), ntotal(2),
                ntotal(3)], 'double', 0, 'bsq', 'native', {'Band', '
                Direct', k});
217         alfa1xy=multibandread(nomalfa1, [ntotal(1), ntotal(2),
                ntotal(3)], 'double', 0, 'bsq', 'native', {'Band', '
                Direct', k});
218         condxy= multibandread(nomcond , [ntotal(1), ntotal(2),
                ntotal(3)], 'double', 0, 'bsq', 'native', {'Band', '
                Direct', k});

220         %Cranck-Nicholson in the X direction.
221         for j=1:ntotal(2) %Covering every j in this 'slice'.
222         %One dimension simulation through i values for fixed k and
                j.
223             if size(nonzeros(alfa1xy(:,j)),1)==ntotal(1)
224             %In case of no void.
225                 %Creation of matrices Ax and Bx.
226                 for m=2:ntotal(1)-1
227                     Ax(m,m-1)=-condxy(m-1,j);
228                     Ax(m,m)  =alfa1xy(m,j)*dx^2/dt+condxy(m,j)+
                            condxy(m-1,j);
```

```
229                     Ax(m,m+1)=-condxy(m,j);
230                     Bx(m,m-1)=condxy(m-1,j);
231                     Bx(m,m)  =alfa1xy(m,j)*dx^2/dt-condxy(m,j)-
                            condxy(m-1,j);
232                     Bx(m,m+1)=condxy(m,j);
233                 end
234                 %Boundaries in matrices Ax and Bx
235                 %Adiabatic boundary at X=0
236                 Ax(1,1)=alfa1xy(1,j)*dx^2/dt+condxy(1,j);
237                 Ax(1,2)=-condxy(1,j);
238                 Bx(1,1)=alfa1xy(1,j)*dx^2/dt-condxy(1,j);
239                 Bx(1,2)=condxy(1,j);
240                 %Adiabatic boundary at X=Xtot
241                 Ax(ntotal(1), ntotal(1)-1)=-condxy(ntotal(1)-1,j);
242                 Ax(ntotal(1), ntotal(1))=alfa1xy(ntotal(1),j)*dx^2/
                        dt+condxy(ntotal(1)-1,j);
243                 Bx(ntotal(1), ntotal(1)-1)=condxy(ntotal(1)-1,j);
244                 Bx(ntotal(1), ntotal(1))=alfa1xy(ntotal(1),j)*dx^2/
                        dt-condxy(ntotal(1)-1,j);
245                 %Vector Bvectx.
246                 Bvectx=Bx*tempxy(:,j);
247                 %Direct solution of the equation system AT=B => T=A
                        \B
248                 tempxy(:,j)=Ax\Bvectx;

250             elseif size(nonzeros(alfa1xy(:,j)),1)~=0
251             %If void found (alfa1=0).
252                       %Auxiliary variables to create auxiliary
                              matrices.
253                  alfa1auxx=nonzeros(alfa1xy(:,j));
254                  rangauxx=size(alfa1auxx,1);
255                  desdeauxx=min(find(alfa1xy(:,j)));
256                  finsaauxx=max(find(alfa1xy(:,j)));
257                  condauxx=condxy(desdeauxx:finsaauxx, j);
258                  tempauxx=tempxy(desdeauxx:finsaauxx, j);
259                  %Creation of auxiliary matrices Aauxx and Bauxx
260                  Aauxx=zeros(rangauxx);
261                  Bauxx=zeros(rangauxx);
262                  for m=2:rangauxx-1
263                     Aauxx(m,m-1)=-condauxx(m-1);
264                     Aauxx(m,m)  =alfa1auxx(m)*dx^2/dt+condauxx(m)+
                            condauxx(m-1);
265                     Aauxx(m,m+1)=-condauxx(m);
266                     Bauxx(m,m-1)=condauxx(m-1);
267                     Bauxx(m,m)  =alfa1auxx(m)*dx^2/dt-condauxx(m)-
                            condauxx(m-1);
268                     Bauxx(m,m+1)=condauxx(m);
269                 end
270                 %Boundaries for the matrices Aauxx and Bauxx
```

```
271                 %Adiabatic boundary at X=begin of the material
                        media.
272                 Aauxx(1,1)=alfa1auxx(1)*dx^2/dt+condauxx(1);
273                 Aauxx(1,2)=-condauxx(1);
274                 Bauxx(1,1)=alfa1auxx(1)*dx^2/dt-condauxx(1);
275                 Bauxx(1,2)=condauxx(1);
276                 %Adiabatic boundary at X=end of the material media.
277                 Aauxx(rangauxx, rangauxx-1)=-condauxx(rangauxx-1);
278                 Aauxx(rangauxx, rangauxx)=alfa1auxx(rangauxx)*dx^2/
                        dt+condauxx(rangauxx-1);
279                 Bauxx(rangauxx, rangauxx-1)=condauxx(rangauxx);
280                 Bauxx(rangauxx, rangauxx)=alfa1auxx(rangauxx)*dx^2/
                        dt-condauxx(rangauxx-1);
281                 %Vector Bvectx.
282                 Bvectauxx=Bauxx*tempauxx;
283                 %Direct solution of the equation system AT=B => T=A
                        \B
284                 tempxy(desdeauxx:finsaauxx,j)=Aauxx\Bvectauxx;
285             end %End of the 'in case of void' condition
286         end %End of the Cranck-Nicholson in X.

288         %Cranck-Nicholson in the Y direction.
289         for i=1:ntotal(1) %Covering every i in this 'slice'.
290         %One dimension simulation through j values for fixed k and i
                .
291             if size(nonzeros(alfa1xy(i,:)),1)==ntotal(2)
292             %In case of no void.
293                 %Creation of matrices Ay and By.
294                 for m=2:ntotal(2)-1
295                     Ay(m,m-1)=-condxy(i,m-1);
296                     Ay(m,m)  =alfa1xy(i,m)*dy^2/dt+condxy(i,m)+
                            condxy(i,m-1);
297                     Ay(m,m+1)=-condxy(i,m);
298                     By(m,m-1)=condxy(i,m-1);
299                     By(m,m)  =alfa1xy(i,m)*dy^2/dt-condxy(i,m)-
                            condxy(i,m-1);
300                     By(m,m+1)=condxy(i,m);
301                 end
302                 %Boundaries in matrices Ay and By
303                 %Adiabatic boundary at Y=0
304                 Ay(1,1)=alfa1xy(i,1)*dy^2/dt+condxy(i,1);
305                 Ay(1,2)=-condxy(i,1);
306                 By(1,1)=alfa1xy(i,1)*dy^2/dt-condxy(i,1);
307                 By(1,2)=condxy(i,1);
308                 %Adiabatic boundary at Y=Ytot
309                 Ay(ntotal(2), ntotal(2)-1)=-condxy(i,ntotal(2)-1);
310                 Ay(ntotal(2), ntotal(2))=alfa1xy(i,ntotal(2))*dy^2/
                        dt+condxy(i,ntotal(2)-1);
311                 By(ntotal(2), ntotal(2)-1)=condxy(i,ntotal(2)-1);
```

```
312                 By(ntotal(2), ntotal(2))=alfa1xy(i,ntotal(2))*dy^2/
                        dt-condxy(i,ntotal(2)-1);
313                 %Vector Bvecty.
314                 Bvecty=By*tempxy(i,:)';
315                 %Direct solution of the equation system AT=B => T=A
                        \B
316                 tempxy(i,:)=(Ay\Bvecty)';

318               elseif size(nonzeros(alfa1xy(i,:)),1)~=0
319               %If void found (alfa1=0).
320                       %Auxiliary variables to create auxiliary
                            matrices.
321                  alfa1auxy=nonzeros(alfa1xy(i,:));
322                  rangauxy=size(alfa1auxy,1);
323                  desdeauxy=min(find(alfa1xy(i,:)));
324                  finsaauxy=max(find(alfa1xy(i,:)));
325                  condauxy=condxy(i, desdeauxy:finsaauxy);
326                  tempauxy=tempxy(i, desdeauxy:finsaauxy);
327                  %Creation of auxiliary matrices Aauxy and Bauxy
328                  Aauxy=zeros(rangauxy);
329                  Bauxy=zeros(rangauxy);
330                  for m=2:rangauxy-1
331                     Aauxy(m,m-1)=-condauxy(m-1);
332                     Aauxy(m,m)  =alfa1auxy(m)*dy^2/dt+condauxy(m)+
                            condauxy(m-1);
333                     Aauxy(m,m+1)=-condauxy(m);
334                     Bauxy(m,m-1)=condauxy(m-1);
335                     Bauxy(m,m)  =alfa1auxy(m)*dy^2/dt-condauxy(m)-
                            condauxy(m-1);
336                     Bauxy(m,m+1)=condauxy(m);
337                 end
338                 %Boundaries for the matrices Aauxy and Bauxy
339                 %Adiabatic boundary at Y=begin of the material
                        media.
340                 Aauxy(1,1)=alfa1auxy(1)*dy^2/dt+condauxy(1);
341                 Aauxy(1,2)=-condauxy(1);
342                 Bauxy(1,1)=alfa1auxy(1)*dy^2/dt-condauxy(1);
343                 Bauxy(1,2)=condauxy(1);
344                 %Adiabatic boundary at Y=end of the material media.
345                 Aauxy(rangauxy, rangauxy-1)=-condauxy(rangauxy-1);
346                 Aauxy(rangauxy, rangauxy)=alfa1auxy(rangauxy)*dy^2/
                        dt+condauxy(rangauxy-1);
347                 Bauxy(rangauxy, rangauxy-1)=condauxy(rangauxy);
348                 Bauxy(rangauxy, rangauxy)=alfa1auxy(rangauxy)*dy^2/
                        dt-condauxy(rangauxy-1);
349                 %Vector Bvecty.
350                 Bvectauxy=Bauxy*tempauxy';
351                 %Direct solution of the equation system AT=B => T=A
                        \B
```

```
352                     tempxy(i,desdeauxy:finsaauxy)=(Aauxy\Bvectauxy)';
353                 end %End of the 'in case of void' condition
354             end %End of Cranck-Nicholson in Y.

356             %The temperature resulting of this slice is saved.
357             multibandwrite(tempxy, nomtemp, 'bsq', [1 1 k], [ntotal(1)
                    ntotal(2) ntotal(3)], 'precision', 'double', 'offset',
                    0)

359         end %End of the 'slices' in Z (k) and the Cranck-Nicholson in X
                and Y

361         %The evolution of the simulation is printed on the screen.
362         fprintf(1, '\b\b\b\b%-3i%%', floor(n/ntemps*100) )

364         %The temperature in the monitor points is recorded
365         T(n+1,1)=temps*(factor^n-1)/(factor^ntemps-1);
366         for i=1:npt
367             tempptemp=multibandread(nomtemp, [ntotal(1), ntotal(2),
                    ntotal(3)], 'double', 0, 'bsq', 'native', {'band', '
                    direct', nptemp(3,i)});
368             T(n+1,i+1)=tempptemp(nptemp(1,i),nptemp(2,i));
369         end

371 end  %End of the time step.

373 %The temporary files temp, alfa1 and alfa2 are deleted.
374 delete(nomtemp );
375 delete(nomalfa1);
376 delete(nomalfa2);
377 delete(nomcond);

379 tempssimulacio=toc; %The chronometer is stopped.

381 %The end of the simulation is anounced and the log file closed.
382 fprintf(fidlog, '\nCompletada la simulacio termica en %d hores %d
        minuts i %d segons.', floor(tempssimulacio/3600), floor(mod(
        floor(tempssimulacio),3600)/60), mod(floor(tempssimulacio),60));

384 control=0; %No errors encountered.
385 return %End of the function.
```

A.5 File: simfileread.m

```
1  %Function SIMFILEREAD reads the geometry file for the thermal
       simulations.

3  %Input variables:
4  %   nomfitxer, name of the geometry file
5  %   fidlog, name of the log file

7  %COMMENTS:
8  %   All longitudes are in meters.
9  %   The resolution of every longitude is 1um, fractions are rounded.

11 function [ control, nmaterials, dimensions, ndimensions,
       conductivitat, cespecifica, densitat, total, ntotal, npotencia,
       dompotencia, ndompotencia, potencia, ptemp, nptemp, tempinicial,
        temps, ntemps, factor, condaidentificar, cespaidentificar] =
       simfileread ( nomfitxer, fidlog )
12 fprintf(fidlog, ['\nLectura de la configuracio del model: '
       nomfitxer ]);

14 %Initialization of all variables.
15 control=-1;
16 nmaterials=0;
17 dimensions=0;
18 ndimensions=0;
19 conductivitat=0;
20 cespecifica=0;
21 densitat=0;
22 total=0;
23 ntotal=0;
24 dompotencia=0;
25 ndompotencia=0;
26 potencia=0;
27 ptemp=0;
28 nptemp=0;
29 tempinicial=0;
30 temps=0;
31 ntemps=0;
32 factor=0;
33 condaidentificar=0;
34 cespaidentificar=0;
35 npotencia=0;


38 %The geometry file is opened.
39 fid=fopen(nomfitxer, 'r');
40 if fid==-1;
41     fprintf(fidlog, '\n***ERROR AL FITXER DE CONFIGURACIO***: No es
```

```
        troba el fitxer de configuracio, possiblement no existeix.')
        ;
42      control=-1;
43      return;
44  end

46  %The number of materials nmaterials is readed. Only integers are
       readed.
47  nmaterials=0;
48  linia=seguentlinia(fid);
49  nmaterials=sscanf(linia, 'nmaterials=%d');
50  if strcmp(nmaterials,'')|nmaterials==0; %Error if nmaterials is zero
51      fprintf(fidlog, '\n***ERROR AL FITXER DE CONFIGURACIO***: No es
           pot llegir nmaterials.');
52      control=-1;
53      return;
54  end

56  %The dimensions of the materials are readed dimensions=(posicion,
       number of the material).
57  dimensions=zeros(6,nmaterials);
58  for n=1:nmaterials
59      linia=seguentlinia(fid);

61      lectura=sscanf(linia, ['XMaterial' int2str(n) '=(%f,%f)']);
62      if strcmp(lectura,'');
63          fprintf(fidlog, ['\n***ERROR AL FITXER DE CONFIGURACIO***:
               Error al llegir XMaterial' int2str(n) '.']);
64          control=-1;
65          return;
66      end
67      dimensions(1:2,n)=round(lectura*1000)/1000000;

69      linia=seguentlinia(fid);
70      lectura=sscanf(linia, ['YMaterial' int2str(n) '=(%f,%f)']);
71      if strcmp(lectura,'');
72          fprintf(fidlog, ['\n***ERROR AL FITXER DE CONFIGURACIO***:
               Error al llegir YMaterial' int2str(n) '.']);
73          control=-1;
74          return;
75      end
76      dimensions(3:4,n)=round(lectura*1000)/1000000;

78      linia=seguentlinia(fid);
79      lectura=sscanf(linia, ['ZMaterial' int2str(n) '=(%f,%f)']);
80      if strcmp(lectura,'');
81          fprintf(fidlog, ['\n***ERROR AL FITXER DE CONFIGURACIO***:
               Error al llegir ZMaterial' int2str(n) '.']);
82          control=-1;
```

```
83          return;
84      end
85      dimensions(5:6,n)=round(lectura*1000)/1000000;
86  end

88  %The properties of every material are readed, thermal conductivity,
        specific heat and density.
89  conductivitat=zeros(1,nmaterials);
90  cespecifica=zeros(1,nmaterials);
91  densitat=zeros(1,nmaterials);
92  for n=1:nmaterials
93      linia=seguentlinia(fid);
94      lectura=sscanf(linia, ['KMaterial' int2str(n) '=%f']);
95      if strcmp(lectura,'');
96          fprintf(fidlog, ['\n***ERROR AL FITXER DE CONFIGURACIO***:
                Error al llegir la conductivitat del material ' int2str(
                n) '.']);
97          control=-1;
98          return;
99      end
100     conductivitat(1,n)=lectura;
101 end
102 for n=1:nmaterials
103     linia=seguentlinia(fid);
104     lectura=sscanf(linia, ['CMaterial' int2str(n) '=%f']);
105     if strcmp(lectura,'');
106         fprintf(fidlog, ['\n***ERROR AL FITXER DE CONFIGURACIO***:
                Error al llegir la calor especifica del material '
                int2str(n) '.']);
107         control=-1;
108         return;
109     end
110     cespecifica(1,n)=lectura;
111 end
112 for n=1:nmaterials
113     linia=seguentlinia(fid);
114     lectura=sscanf(linia, ['DMaterial' int2str(n) '=%f']);
115     if strcmp(lectura,'');
116         fprintf(fidlog, ['\n***ERROR AL FITXER DE CONFIGURACIO***:
                Error al llegir la densitat del material ' int2str(n)
                '.']);
117         control=-1;
118         return;
119     end
120     densitat(1,n)=lectura;
121 end

123 %The dimensions of the total simulation domain are readed.
124 total=zeros(3,1);
```

```
125 linia=seguentlinia(fid);
126 lectura=sscanf(linia, 'Xtot=%f');
127 if strcmp(lectura,'');
128         fprintf(fidlog, '\n***ERROR AL FITXER DE CONFIGURACIO***:
               Error al llegir Xtot.');
129         control=-1;
130         return;
131 end
132 total(1)=round(lectura*1000)/1000000;
133 linia=seguentlinia(fid);
134 lectura=sscanf(linia, 'Ytot=%f');
135 if strcmp(lectura,'');
136         fprintf(fidlog, '\n***ERROR AL FITXER DE CONFIGURACIO***:
               Error al llegir Ytot.');
137         control=-1;
138         return;
139 end
140 total(2)=round(lectura*1000)/1000000;
141 linia=seguentlinia(fid);
142 lectura=sscanf(linia, 'Ztot=%f');
143 if strcmp(lectura,'');
144         fprintf(fidlog, '\n***ERROR AL FITXER DE CONFIGURACIO***:
               Error al llegir Ztot.');
145         control=-1;
146         return;
147 end
148 total(3)=round(lectura*1000)/1000000;

150 %The space discretization is readed. Only integers are readed.
151 ntotal=zeros(3,1);

153 linia=seguentlinia(fid);
154 lectura=sscanf(linia, 'nXtot=%d');
155 if strcmp(lectura,'');
156         fprintf(fidlog, '\n***ERROR AL FITXER DE CONFIGURACIO***:
               Error al llegir nXtot.');
157         control=-1;
158         return;
159 end
160 ntotal(1)=lectura;

162 linia=seguentlinia(fid);
163 lectura=sscanf(linia, 'nYtot=%d');
164 if strcmp(lectura,'');
165         fprintf(fidlog, '\n***ERROR AL FITXER DE CONFIGURACIO***:
               Error al llegir nYtot.');
166         control=-1;
167         return;
168 end
```

```
169 ntotal(2)=lectura;

171 linia=seguentlinia(fid);
172 lectura=sscanf(linia, 'nZtot=%d');
173 if strcmp(lectura,'');
174         fprintf(fidlog, '\n***ERROR AL FITXER DE CONFIGURACIO***:
               Error al llegir nZtot.');
175         control=-1;
176         return;
177 end
178 ntotal(3)=lectura;

180 %The number of dissipating heat volumes is readed. Only integers are
        readed.
181 npotencia=0;
182 linia=seguentlinia(fid);
183 npotencia=sscanf(linia, 'npotencia=%d');
184 if strcmp(npotencia,'')|npotencia==0; %Error if nothing readed or
       npotencia equal to zero.
185     fprintf(fidlog, '\n***ERROR AL FITXER DE CONFIGURACIO***: No es
           pot llegir npotencia.');
186     control=-1;
187     return;
188 end

190 %The dimensions of every material are readed .
191 dompotencia=zeros(6,npotencia);

193 for n=1:npotencia
194    linia=seguentlinia(fid);
195    lectura=sscanf(linia, ['XPotencia' int2str(n) '=(%f,%f)']);
196    if strcmp(lectura,'');
197         fprintf(fidlog, ['\n***ERROR AL FITXER DE CONFIGURACIO***:
               Error al llegir XPotencia.' int2str(n) '.']);
198         control=-1;
199         return;
200    end
201    dompotencia(1:2,n)=round(lectura*1000)/1000000;

203    linia=seguentlinia(fid);
204    lectura=sscanf(linia, ['YPotencia' int2str(n) '=(%f,%f)']);
205    if strcmp(lectura,'');
206         fprintf(fidlog, ['\n***ERROR AL FITXER DE CONFIGURACIO***:
               Error al llegir YPotencia.' int2str(n) '.']);
207         control=-1;
208         return;
209     end
210     dompotencia(3:4,n)=round(lectura*1000)/1000000;
```

```
212     linia=seguentlinia(fid);
213     lectura=sscanf(linia, ['ZPotencia' int2str(n) '=(%f,%f)']);
214     if strcmp(lectura,'');
215          fprintf(fidlog, ['\n***ERROR AL FITXER DE CONFIGURACIO***:
                Error al llegir ZPotencia.' int2str(n) '.']);
216          control=-1;
217          return;
218      end
219      dompotencia(5:6,n)=round(lectura*1000)/1000000;

221 end

223 linia=seguentlinia(fid);
224 potencia=sscanf(linia, 'PotenciaDissipada=%f');
225 if strcmp(potencia,'');
226          fprintf(fidlog, '\n***ERROR AL FITXER DE CONFIGURACIO***:
                Error al llegir PotenciaDissipada.');
227          control=-1;
228          return;
229 end

231 %The monitor points positions are readed.
232 ptemp=0;
233 n=1;
234 linia=seguentlinia(fid);
235 while strncmp(linia, 'PTemp', 5)
236     lectura=sscanf(linia, 'PTemp%*d=(%f,%f,%f)');
237     if strcmp(lectura,'');
238          fprintf(fidlog, '\n***ERROR AL FITXER DE CONFIGURACIO***:
                Error al llegir PTemp.');
239          control=-1;
240          return;
241     end
242     ptemp(1:3,n)=round(lectura*1000)/1000000;
243     n=n+1;
244     linia=seguentlinia(fid);
245 end

247 %The initial temperature is readed.
248 tempinicial=0;
249 tempinicial=sscanf(linia, 'TempInicial=%f');
250 if strcmp(tempinicial,'');
251          fprintf(fidlog, '\n***ERROR AL FITXER DE CONFIGURACIO***:
                Error al llegir TempInicial.');
252          control=-1;
253          return;
254 end

256 %The time of simulation and the time-step number is readed.
```

```
257 temps=0;
258 ntemps=0;
259 linia=seguentlinia(fid);
260 temps=sscanf(linia, 'Temps=%f');
261 if strcmp(temps,'');
262         fprintf(fidlog, '\n***ERROR AL FITXER DE CONFIGURACIO***:
               Error al llegir Temps.');
263         control=-1;
264         return;
265 end

267 linia=seguentlinia(fid);
268 ntemps=sscanf(linia, 'nTemps=%d');
269 if strcmp(ntemps,'');
270         fprintf(fidlog, '\n***ERROR AL FITXER DE CONFIGURACIO***:
               Error al llegir nTemps.');
271         control=-1;
272         return;
273 end

275 linia=seguentlinia(fid);
276 factor=sscanf(linia, 'Factor=%f');
277 if strcmp(factor,'');
278         fprintf(fidlog, '\n***ERROR AL FITXER DE CONFIGURACIO***:
               Error al llegir Factor.');
279         control=-1;
280         return;
281 end

283 %End of the file reading, beginning of checkings and data
       adaptations.

285 %ndimensions are the dimensions translated in number of space
       cuboids.
286 ndimensions=zeros(6,nmaterials);
287 for n=1:nmaterials
288     ndimensions(1,n)=dimensions(1,n)*ntotal(1)/total(1)+1;
289     ndimensions(2,n)=dimensions(2,n)*ntotal(1)/total(1);
290     ndimensions(3,n)=dimensions(3,n)*ntotal(2)/total(2)+1;
291     ndimensions(4,n)=dimensions(4,n)*ntotal(2)/total(2);
292     ndimensions(5,n)=dimensions(5,n)*ntotal(3)/total(3)+1;
293     ndimensions(6,n)=dimensions(6,n)*ntotal(3)/total(3);
294 end

296 %Checking that the material dimensions fit into the total spatial
       discretization.
297 %Checking that any material sticks out the total simulation domain.
298 for n=1:nmaterials
299     for i=1:6
```

```
300         if abs(round(ndimensions(i,n))-ndimensions(i,n))>=ntotal(
                ceil(i/2))/total(ceil(i/2))/2000000
301             fprintf(fidlog, ['\n***ERROR A LA CONFIGURACIO***: Les
                    dimensions del material ' int2str(n) ' no encaixen
                    amb la discretitzacio espaial.']);
302             control=-1;
303             return;
304         elseif dimensions(i,n)<0|dimensions(i,n)>total(ceil(i/2))
305             fprintf(fidlog, ['\n***ERROR A LA CONFIGURACIO***: El
                    material ' int2str(n) ' sobresurt del domini total
                    .']);
306             control=-1;
307             return;
308         end
309     end
310 end

312 %The ndimensions are rounded.
313 ndimensions=round(ndimensions);

315 %Checking that every heat dissipation volume fit into the total
        spatial discretization.
316 %Checking that any heat dissipation volume sticks out the total
        simulation domain.
317 ndompotencia=zeros(6,npotencia);
318 for n=1:npotencia
319     ndompotencia(1,n)=dompotencia(1,n)*ntotal(1)/total(1)+1;
320     ndompotencia(2,n)=dompotencia(2,n)*ntotal(1)/total(1);
321     ndompotencia(3,n)=dompotencia(3,n)*ntotal(2)/total(2)+1;
322     ndompotencia(4,n)=dompotencia(4,n)*ntotal(2)/total(2);
323     ndompotencia(5,n)=dompotencia(5,n)*ntotal(3)/total(3)+1;
324     ndompotencia(6,n)=dompotencia(6,n)*ntotal(3)/total(3);

326     for i=1:6
327         if abs(round(ndompotencia(i,n))-ndompotencia(i,n))>=ntotal(
                ceil(i/2))/total(ceil(i/2))/2000000 %Es comprova que
                encaixi (tenint en compte la resolucio i que la operacio
                 pot no ser exacta, 'sensibilitat'1/2um)
328             fprintf(fidlog, '\n***ERROR A LA CONFIGURACIO***: Les
                    dimensions del domini on es dissipa la potencia no
                    encaixen amb la discretitzacio espaial.');
329             control=-1;
330             return;
331         elseif dompotencia(i,n)<0|dompotencia(i,n)>total(ceil(i/2))
332             fprintf(fidlog, '\n***ERROR A LA CONFIGURACIO***: Les
                    dimensions del domini on es dissipa la potencia
                    sobresurten del domini total.');
333             control=-1;
334             return;
```

```
335         end
336     end
337 end
338 %ndompotencia is rounded.
339 ndompotencia=round(ndompotencia);

341 %Checkin that any monitor point is out of the simulation domain.
342 for n=1:size(ptemp, 2)
343     for i=1:3
344         if ptemp(i,n)<0|ptemp(i,n)>total(i)
345         fprintf(fidlog, ['\n***ERROR A LA CONFIGURACIO***: El punt
               de mesura de la temperatura ' int2str(n) ' cau fora del
               domini total.']);
346         control=-1;
347         return;
348         end
349     end
350 end

352 %The monitor point positions ptemp are translated into nptemp.
353 npt=size(ptemp,2);
354 nptemp=zeros(3,npt);
355 ptempcorregit=zeros(3,npt);
356 for n=1:npt
357 nptemp(:,n)=floor(ptemp(:,n).*ntotal(:)./total(:))+1;
358 ptempcorregit(:,n)=(nptemp(:,n)-0.5).*total(:)./ntotal(:);
359 end

361 %The parameters to be identified are stored into paidentificar.
362 condaidentificar=zeros(1,nmaterials);
363 cespaidentificar=zeros(1,nmaterials);
364 for n=1:nmaterials
365     linia=seguentlinia(fid);
366     if linia==-1
367         fprintf(fidlog, '\n***ERROR AL FITXER DE CONFIGURACIO***: Hi
              falten instruccions.');
368         control=-1;
369         return;
370     end
371     lectura=sscanf(linia, ['IdMaterial' int2str(n) ' %s']);
372     if strcmp(lectura, 'Cond')
373         condaidentificar(1,n)=1;
374     elseif strcmp(lectura, 'Cesp')
375         cespaidentificar(1,n)=1;
376     elseif strcmp(lectura, 'Cond&Cesp')
377         condaidentificar(1,n)=1;
378         cespaidentificar(1,n)=1;
379     elseif strcmp(lectura, 'nores')
380     else
```

```
381         fprintf(fidlog, ['\n***ERROR AL FITXER DE CONFIGURACIO***:
                Error al llegir els parametres a identificar del
                material ' int2str(n) '.']);
382         control=-1;
383         return;
384     end
385 end

387 %The file is closed
388 fclose(fid);
389 control=0;
390 return %End of the function

392 %Routine to obtain a new line of the read file.%
393 function [linia] = seguentlinia(fid)
394     linia=deblank(fgetl(fid));
395     while strcmp(linia,'') | strncmp(linia, '*', 1)
396         linia=deblank(fgetl(fid));
397     end
398 return
```

A.6 File: idfileread.m

```
1  %Function IDFILEREAD that reads the Identification configuration
       file and extracts all the information.

3  %Input variables are:
4  %       nomfitxer, the name of the Identification Configuration file
       .
5  %       fidlog, identifier of the log file.

7  %Output variables are:
8  %       control, indicates if an error is found.
9  %       nomfitxermodel, name of the Geometry Configuration file.
10 %       nomfitxermesura, name of the experimental data file.
11 %   liniesaignorar,
12 %   caracterseparacio,
13 %   nomfitxerresultat,
14 %   nomfitxerconfigresultat,
15 %   nomfitxerconfigvariacio,
16 %       numiteracions,

18 function [ control, nomfitxermodel, nomfitxermesura, liniesaignorar,
       caracterseparacio, nomfitxerresultat, nomfitxerconfigresultat,
      nomfitxerconfigvariacio, numiteracions ] = idfileread( nomfitxer
      , fidlog )

20 %The log file is opened
21 fprintf(fidlog, ['\nLectura del fitxer de configuracio de la
       identificacio: ' nomfitxer]);

23 %Initialization of all variables
24 control=-1;
25 nomfitxermodel=0;
26 nomfitxermesura=0;
27 liniesaignorar=0;
28 caracterseparacio=0;
29 nomfitxerresultat=0;
30 nomfitxerconfigresultat=0;
31 nomfitxerconfigvariacio=0;
32 numiteracions=0;

34 %The Identification Configuration file is opened.
35 fid=fopen(nomfitxer, 'r');
36 if fid==-1;
37     fprintf(fidlog, '\n***ERROR AL FITXER DE CONFIGURACIO
          IDENTIFICACIO***: No es troba el fitxer de configuracio,
          possiblement no existeix.');
38     control=-1;
39     return;
```

```
40 end

42 %The name of the Geometry file is read
43 linia=seguentlinia(fid);
44 nomfitxermodel=sscanf(linia, 'NomFitxerModel=%s');
45 if strcmp(nomfitxermodel,''); %Error if nothing is read
46     fprintf(fidlog, '\n***ERROR AL FITXER DE CONFIGURACIO
          IDENTIFICACIO***: No es pot llegir NomFitxerModel');
47     control=-1;
48     return;
49 end

51 %The name of the experimental data file is read
52 linia=seguentlinia(fid);
53 nomfitxermesura=sscanf(linia, 'NomFitxerMesura=%s');
54 if strcmp(nomfitxermesura,''); %Error if nothing is read
55     fprintf(fidlog, '\n***ERROR AL FITXER DE CONFIGURACIO
          IDENTIFICACIO***: No es pot llegir NomFitxerMesura');
56     control=-1;
57     return;
58 end

60 %The number of lines to be ignored in the experimental data file is
      read
61 linia=seguentlinia(fid);
62 liniesaignorar=sscanf(linia, 'LiniesAIgnorar=%d');
63 if strcmp(liniesaignorar,''); %Error if nothing is read
64     fprintf(fidlog, '\n***ERROR AL FITXER DE CONFIGURACIO
          IDENTIFICACIO***: No es pot llegir LiniesAIgnorar');
65     control=-1;
66     return;
67 end

69 %The char between columns in the experimental data file is read.
70 linia=seguentlinia(fid);
71 caracterseparacio=sscanf(linia, 'CaracterDeSeparacio=%s');
72 if strcmp(caracterseparacio,''); %Error if nothing is read
73     fprintf(fidlog, '\n***ERROR AL FITXER DE CONFIGURACIO
          IDENTIFICACIO***: No es pot llegir CaracterDeSeparacio');
74     control=-1;
75     return;
76 elseif strcmp(caracterseparacio,'tabulat')
77     caracterseparacio='\t';
78 elseif strcmp(caracterseparacio,'espai')
79     caracterseparacio=' ';
80 end

82 %The name of the file where the results are going to be written is
      read.
```

```
83 linia=seguentlinia(fid);
84 nomfitxerresultat=sscanf(linia, 'NomFitxerResultat=%s');
85 if strcmp(nomfitxerresultat,''); %Error if nothing is read
86     fprintf(fidlog, '\n***ERROR AL FITXER DE CONFIGURACIO
          IDENTIFICACIO***: No es pot llegir NomFitxerResultat');
87     control=-1;
88     return;
89 end

91 %The name of the resulting geometry file after parameter
      modification is read
92 linia=seguentlinia(fid);
93 nomfitxerconfigresultat=sscanf(linia, 'NomFitxerConfigResultat=%s');
94 if strcmp(nomfitxerconfigresultat,''); %Error if nothing is read
95     fprintf(fidlog, '\n***ERROR AL FITXER DE CONFIGURACIO
          IDENTIFICACIO***: No es pot llegir NomFitxerConfigResultat')
          ;
96     control=-1;
97     return;
98 end

100 %The prefix of the different geometry files with variated parameters
       is read
101 linia=seguentlinia(fid);
102 nomfitxerconfigvariacio=sscanf(linia, 'NomFitxerConfigVariacio=%s');
103 if strcmp(nomfitxerconfigvariacio,''); %Error if nothing is read
104     fprintf(fidlog, '\n***ERROR AL FITXER DE CONFIGURACIO
          IDENTIFICACIO***: No es pot llegir NomFitxerConfigVariacio')
          ;
105     control=-1;
106     return;
107 end

109 %The maximum number of permitted iterations is read
110 linia=seguentlinia(fid);
111 if linia==-1
112     fprintf(fidlog, '\n***ERROR AL FITXER DE CONFIGURACIO***: Hi
          falten instruccions.');
113     control=-1;
114     return;
115 end
116 numiteracions=sscanf(linia, 'NumIteracions=%d');
117 if strcmp(numiteracions,''); %Error if nothing is read
118     fprintf(fidlog, '\n***ERROR AL FITXER DE CONFIGURACIO
          IDENTIFICACIO***: No es pot llegir NumIteracions');
119     control=-1;
120     return;
121 end
```

```
123 control=0; %No errors found
124 fclose(fid); %The file is closen
125 return %End of the function

127 %Routine to obtain a new line of the read file.%
128 function [linia] = seguentlinia(fid)
129     linia=deblank(fgetl(fid));
130     while strcmp(linia,'') | strncmp(linia, '*', 1)
131         linia=deblank(fgetl(fid));
132     end
133 return
```

A.7 File: expfileread.m

```
%Function EXPFILEREAD reads the experimental data file.

%Input variables are:
%   nomfitxer, the name of the experimental data file.
%   caracterseparacio, the char separating data columns.
%   liniesaignorar, number of lines to be skipped before data.
%   fidlog, name of the log file.

%Output variables are:
%       control, controls if an error is found.
%       Tmesura, array with the readed temperature evolution.

function [ control, Tmesura ] = expfileread ( nomfitxer,
   caracterseparacio, liniesaignorar, fidlog )

fprintf(fidlog, ['\nLectura del fitxer de mesura: ' nomfitxer ]);

%The data file is opened
fid=fopen(nomfitxer, 'r');
if fid==-1;
    fprintf(fidlog, '\n***ERROR EM LA LECTURA DEL FITXER DE MESURA
       ***: No es pot llegir el fitxer de mesura, possiblement no
       existeix.');
    control=-1;
    Tmesura=0;
    return;
end

%The array T(time, temperature) is readed
[T(:,1),T(:,2)]=textread(nomfitxer, ['%f' caracterseparacio '%f'], '
   headerlines', liniesaignorar);

%The file is closed
fclose(fid);

%The array T is sorted in chronological order.
Tmesura=sortrows(T, 1);

%No errors found.
control=0;

return %end
```

A.8 File: varsimfilewrite.m

```
%Function VARSIMFILEWRITE writes a copy of the geometry file of the current identification iteration with an altered thermal property of a given material.

%Input variables are:
%       nomfitxerconfig, the name of the current simulation geometry file.
%       nomfitxerconfigvariacio, name of the altered parameter geometry file.
%       nmaterials, number of different materials.
%       condmodif, thermal conductivity of materials to be written (may be altered).
%       cespmodif, specific heat of materials to be written (may be altered).

function [control] = varsimfilewrite( nomfitxerconfig, nomfitxerconfigvariacio, nmaterials, condmodif, cespmodif)

control=-1;

%New and original geometry files are opened.
fid=fopen(nomfitxerconfig);
fidres=fopen( nomfitxerconfigvariacio, 'w');

fprintf( fidres, '*CONFIGURACIO PER LA VARIACIO\n');

%Substitution of the thermal conductivity.
for n=1:nmaterials
    linia=fgetl(fid);
    while ~strncmp(deblank(linia), 'KMaterial', 9)
        fprintf(fidres, [linia '\n']);
        linia=fgetl(fid);
    end
    fprintf(fidres, 'KMaterial%d=%f\n', n , condmodif(n));
end

%Substitution of the specific heat.
for n=1:nmaterials
    linia=fgetl(fid);
    while ~strncmp(linia, 'CMaterial', 9)
        fprintf(fidres, [linia '\n']);
        linia=fgetl(fid);
    end
    fprintf(fidres, 'CMaterial%d=%f\n', n , cespmodif(n));
end

%Transcribing the rest of the file.
```

```
41 linia=fgetl(fid);
42 while ischar(linia)
43     fprintf(fidres, [linia '\n']);
44     linia=fgetl(fid);
45 end

47 %Closing the files.
48 fclose(fid);
49 fclose(fidres);

51 control=0; %No errors found.

53 return %End of function.
```

A.9 File: resultwrite.m

```
1  %Function RESULTWRITE that writes the result of current iteration in
       the identification process in the results file. Also writes the
       resulting geometry file transcribing the current geometry file
      with new parameters for next iteration.

3  %Input variables are:
4  %       iteracio, the current iteration number.
5  %       nomfitxerconfig, name of the current iteration geometry file
       .
6  %       nomfitxerconfigresultat, name of the resulting geometry file
       .
7  %       nomfitxerresultat, name of the results file.
8  %       nmaterials, number of materials.
9  %       condmodif, modified thermal conductivity.
10 %       cespmodif, modified specific heat.

12 function [  ] = escriuresultat(iteracio, nomfitxerconfig,
      nomfitxerconfigresultat, nomfitxerresultat, nmaterials,
      condmodif, cespmodif)

14 %Resulting parameters are written in the results file.
15 fidres=fopen(nomfitxerresultat, 'a');
16 fprintf(fidres, '\n%d\t', iteracio);
17 for n=1:nmaterials
18 fprintf(fidres, '\t%f\t%f', condmodif(n), cespmodif(n));
19 end
20 fclose(fidres);

22 %The resulting geometry file with new parameters for the next
      iteration is written.
23 fid=fopen(nomfitxerconfig);
24 fidres=fopen(nomfitxerconfigresultat, 'w');

26 fprintf(fidres, ['*CONFIGURACIO PER LA ITERACIO ºN' int2str(iteracio
      +1) ' DE LA IDENTIFICACIO.\n']);

28 %Substitution of the thermal conductivity.
29 for n=1:nmaterials
30     linia=fgetl(fid);
31     while ~strncmp(deblank(linia), 'KMaterial', 9)
32         fprintf(fidres, [linia '\n']);
33         linia=fgetl(fid);
34     end
35     fprintf(fidres, 'KMaterial%d=%f\n', n , condmodif(n));
36 end

38 %Substitution of the specific heat.
```

```
39 for n=1:nmaterials
40     linia=fgetl(fid);
41     while ~strncmp(linia, 'CMaterial', 9)
42         fprintf(fidres, [linia '\n']);
43         linia=fgetl(fid);
44     end
45     fprintf(fidres, 'CMaterial%d=%f\n', n , cespmodif(n));
46 end

48 %Transcribing the rest of the file.
49 linia=fgetl(fid);
50 while ischar(linia)
51     fprintf(fidres, [linia '\n']);
52     linia=fgetl(fid);
53 end

55 %Closing the file.
56 fclose(fid);
57 fclose(fidres);
58 return
```

Appendix B

Geometry and Identification Configuration Files

B.1 DCB Geometry Configuration file

```
1  *This is the geometry file for DCB identifications.
2  *Comment lines are introduced by *.

4  *COMMENTS:
5  *       All dimensions are introduced in millimeters.
6  *       Be sure materials are in contact.
7  *       Materials must be cuboids.
8  *       Material dimensions must be multiple of the space grid
       dimensions.
9  *       Spatial resolution is 1um (0.001mm), a fraction of
       micrometer will be rounded.

11 *Number of materials (only a integer is read).
12 nmaterials=6

14 *Material dimensions (only 1/4 part is implemented).
15 *Material 1 is silicon (0.525mm thick).
16 XMaterial1=(0,3)
17 YMaterial1=(0,3)
18 ZMaterial1=(0,0.525)

20 *Material 2 is the die attach (Sn62Pb36Ag2, 0.05mm thick).
21 XMaterial2=(0,3)
22 YMaterial2=(0,3)
23 ZMaterial2=(0.525,0.575)

25 *Material 3 is copper (0.300mm thick).
26 XMaterial3=(0,3)
27 YMaterial3=(0,3)
28 ZMaterial3=(0.575,0.875)

30 *Material 4 is alumina (Al2O3, 0.64mm thick).
31 XMaterial4=(0,7.5)
32 YMaterial4=(0,10)
33 ZMaterial4=(0.875,1.515)

35 *Material 5 is the TIM (RS HeatSink Compound, 0.005mm thick).
36 XMaterial5=(0,7.5)
37 YMaterial5=(0,10)
38 ZMaterial5=(1.515,1.520)

40 *Material 6 is Aluminum (1mm thick).
41 XMaterial6=(0,7.5)
42 YMaterial6=(0,10)
43 ZMaterial6=(1.520,2.520)

45 *Thermal conductivity of every material in W/mK.
```

```
KMaterial1=150
KMaterial2=50.9
KMaterial3=385
KMaterial4=26
KMaterial5=0.9
KMaterial6=201

*Specific heat of every material in J/KgK.
CMaterial1=700
CMaterial2=150
CMaterial3=385
CMaterial4=765
CMaterial5=765
CMaterial6=913

*Density of every material in Kg/m^3.
DMaterial1=2330
DMaterial2=8400
DMaterial3=8930
DMaterial4=3970
DMaterial5=3970
DMaterial6=2710

*Dimensions of the total simulated domain.
Xtot=7.5
Ytot=10
Ztot=2.520

*Spatial discretization of the domain. Only integers are valid.
nXtot=75
nYtot=100
nZtot=504

*Total number of power dissipation volumes. Integers only.
npotencia=3

*Power dissipation volume dimensions, DO NOT overlap.
XPotencia1=(0,0.3)
YPotencia1=(0.3,3)
ZPotencia1=(0,0.05)

XPotencia2=(0.3,3)
YPotencia2=(0,0.3)
ZPotencia2=(0,0.05)

XPotencia3=(0.3,3)
YPotencia3=(0.3,3)
ZPotencia3=(0,0.05)
```

```
95  *Total generated power in W.
96  PotenciaDissipada=6.3232003382079

98  *Points where the temperature is evaluated and written on the output
        'Temp vs Time' result.
99  *Several points can be set but only one is used for the parameter
       identification.
100 PTemp1=(0,0,0)

102 *Initial temperature in Celsius degrees. Also the bottom constant
       temperature boundary.
103 TempInicial=20.393526020813100

105 *Simulated time in seconds, time step number and factor for
       increasing logarithmic time steps.
106 Temps=2
107 nTemps=200
108 Factor=1.00000000001

110 *Parameters to be identified. For every material:
111 *  'nores' no parameter is tried to identify.
112 *  'Cond' thermal conductivity of the material.
113 *  'Cesp' specific heat of the material.
114 *  'Cond&Cesp' thermal conductivity and specific heat.
115 IdMaterial1 nores
116 IdMaterial2 nores
117 IdMaterial3 nores
118 IdMaterial4 Cond&Cesp
119 IdMaterial5 nores
120 IdMaterial6 nores

122 *FILE END.
```

B.2 DCB Identification Configuration file

```
1  *This file contains the DCB parameter identification options.
2  *Comment lines are introduced by *

4  *Geometry file name.
5  NomFitxerModel=GeometryDCB430.txt

7  *Experimental data file name.
8  NomFitxerMesura=Experiment2sDCB430.txt

10 *Number of lines to skip before any data reading in the experimental
       data file.
11 LiniesAIgnorar=2

13 *Separation character between the 'time' and 'temperature' columns.
14 *For tabulator enter 'tabulat' and 'espai' for the space.
15 CaracterDeSeparacio=tabulat

17 *Name of the file where the resuling parameters after every
      iteration are stored.
18 NomFitxerResultat=ResultFile.txt

20 *Name of the resulting geometry file.
21 NomFitxerConfigResultat=ConfigResultat.txt

23 *Prefix of temporary geometry files with altered parameters.
24 NomFitxerConfigVariacio=ConfigVar

26 *Maximum number of iterations if no definitive result is reached.
27 NumIteracions=20

29 *FILE END.
```

Appendix C

Supplementary Material

Excel spreadsheet

TIM: RS Thermal Grease
Torque: 1Nm
Material: AlN07 INASMET

k= 91,73079 ± 5,2916703 W/mK

L virtual	0,00237	±	0,00002	m
Area	0,0003	±	0,00000175	m2
Rshunt	1,05	±	0,001	Ohm
R2	10,21	±	0,0001	KOhm

Thick Sample

Vcc	63	±	0,0126	V
Icc	983,7	±	0,344295	mA
Power	60,5809567	±	0,03418114	W

T spreader				T heatsink				Rth			
54,3126793	±	0,05715634	℃	24,9766979	±	0,042488349	℃	0,48 4244274	±	0,001918	℃/W
54,3898468	±	0,05719492	℃	25,0274448	±	0,042513722	℃	0,48 4680394	±	0,001919	℃/W
54,4325104	±	0,05721626	℃	25,0617676	±	0,042530884	℃	0,48 4818075	±	0,00192	℃/W
54,4605865	±	0,05723029	℃	25,0983734	±	0,042549187	℃	0,48 4677277	±	0,001921	℃/W
54,5010681	±	0,05725053	℃	25,1217403	±	0,04256087	℃	0,484 959786	±	0,001921	℃/W
54,5237465	±	0,05726187	℃	25,1409378	±	0,042570469	℃	0,48 5017245	±	0,001922	℃/W
54,5535927	±	0,0572768	℃	25,1814003	±	0,0425907	℃	0,48484 2003	±	0,001922	℃/W
54,5579987	±	0,057279	℃	25,1760292	±	0,042588015	℃	0,4850 03392	±	0,001922	℃/W
54,5814323	±	0,05729072	℃	25,2062817	±	0,042603141	℃	0,48 4890835	±	0,001923	℃/W
54,6119156	±	0,05730596	℃	25,2330341	±	0,042616517	℃	0,48 4952419	±	0,001923	℃/W
54,6050301	±	0,05730252	℃	25,2182617	±	0,042609131	℃	0,48 5082606	±	0,001923	℃/W
54,6262817	±	0,05731314	℃	25,2392693	±	0,042619635	℃	0,48 5086636	±	0,001923	℃/W
54,6112404	±	0,05730562	℃	25,2385216	±	0,042619261	℃	0,48 4850693	±	0,001923	℃/W
54,6352692	±	0,05731763	℃	25,2501087	±	0,042625054	℃	0,48 5056065	±	0,001923	℃/W
54,6441612	±	0,05732208	℃	25,2640934	±	0,042632047	℃	0,48 4972002	±	0,001924	℃/W

Rth average	0,48487558	±	0,00192178	℃/W

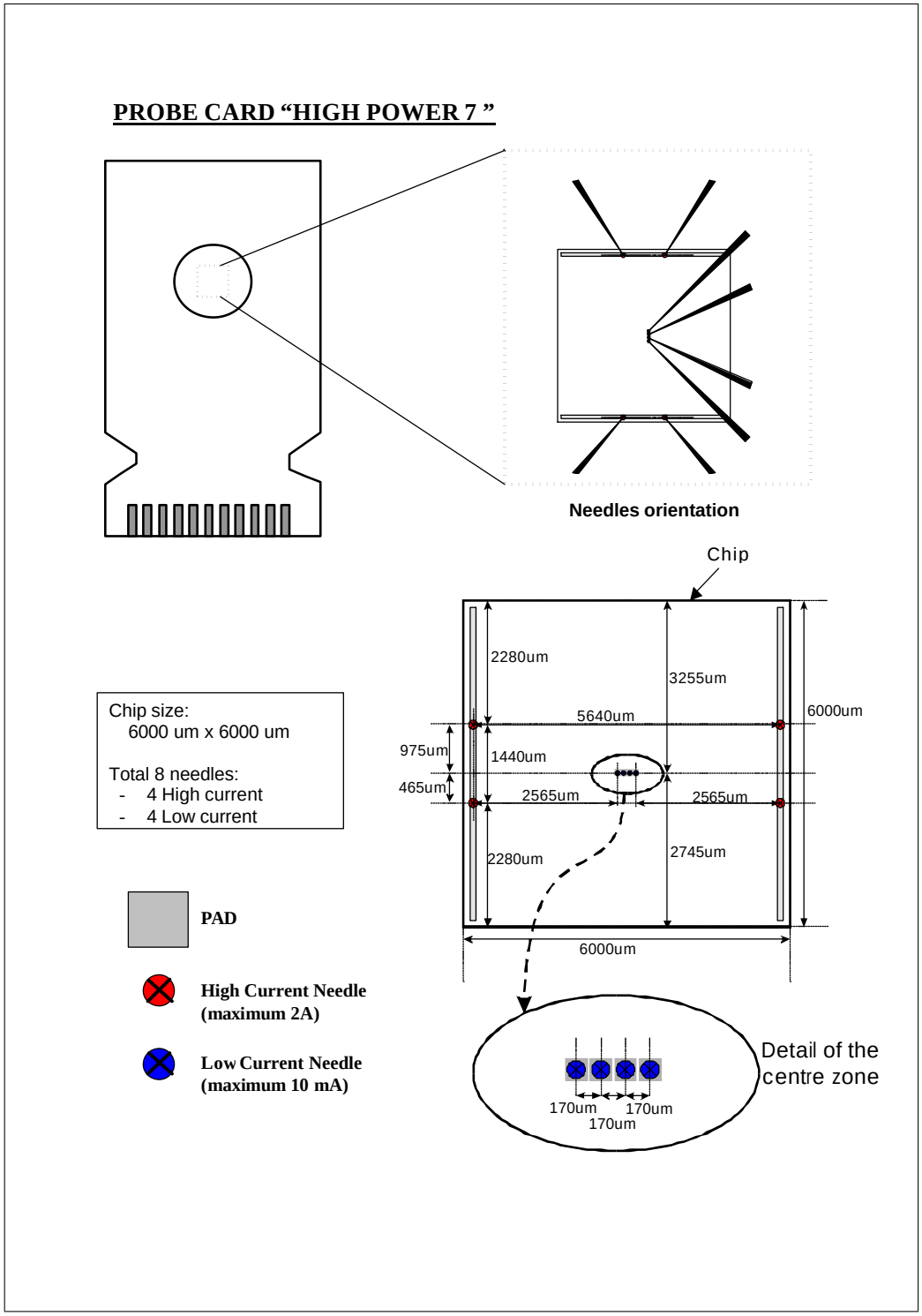
Thin Sample

Vcc	63	±	0,0126	V
Icc	983,7	±	0,344295	mA
Power	60,5809567	±	0,03418114	W

T spreader				T heatsink				Rth			
49,0798187	±	0,05453991	℃	24,9499683	±	0,042474984	℃	0,39 8307516	±	0,001826	℃/W
49,1243134	±	0,05456216	℃	24,9937115	±	0,042496856	℃	0,39 8319921	±	0,001827	℃/W
49,1922684	±	0,05459613	℃	25,0474682	±	0,042523734	℃	0,39 855429	±	0,001828	℃/W
49,230545	±	0,05461527	℃	25,1029358	±	0,042551468	℃	0,398 270522	±	0,001829	℃/W
49,2742691	±	0,05463713	℃	25,1318111	±	0,042565906	℃	0,39 8515627	±	0,001829	℃/W
49,284832	±	0,05464242	℃	25,1511669	±	0,042575583	℃	0,398 370485	±	0,00183	℃/W
49,3149834	±	0,05465749	℃	25,1753216	±	0,042587661	℃	0,39 8469472	±	0,00183	℃/W
49,3386421	±	0,05466932	℃	25,1965065	±	0,042598253	℃	0,39 8510307	±	0,00183	℃/W
49,341156	±	0,05467058	℃	25,1968803	±	0,04259844	℃	0,3985 45632	±	0,00183	℃/W
49,3762474	±	0,05468812	℃	25,227684	±	0,042613842	℃	0,398 616409	±	0,001831	℃/W
49,396595	±	0,0546983	℃	25,2465878	±	0,042623294	℃	0,3986 40242	±	0,001831	℃/W
49,4212036	±	0,0547106	℃	25,2694435	±	0,042634722	℃	0,398 669176	±	0,001832	℃/W
49,4331894	±	0,05471659	℃	25,2767811	±	0,042638391	℃	0,39 8745904	±	0,001832	℃/W
49,458847	±	0,05472942	℃	25,2971401	±	0,04264857	℃	0,3988 33367	±	0,001832	℃/W
49,4527245	±	0,05472636	℃	25,305933	±	0,042652966	℃	0,398 58716	±	0,001832	℃/W

Rth màx	0,40036044

Probe Card Design Sheet



Thermal Interface Materials Data-Sheets

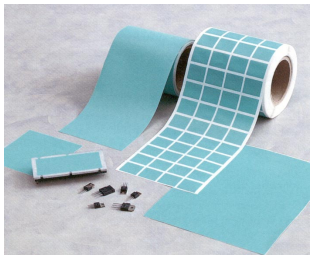
THE BERGQUIST COMPANY

Hi-Flow®625

Electrically Insulating, Thermally Conductive Phase Change Material

Features and Benefits

- Thermal impedance 0.71°C-in²/W (@25 psi)
- Electrically isolating
- 65°C phase change compound coated on Bergquist film
- Tack free and scratch resistant



Hi-Flow 625 is a film reinforced phase change material. The product consists of a thermally conductive 65 °C phase change compound coated on Bergquist film. Hi-Flow 625 is designed to be used as a thermal interface material between electronic power devices that require electrical isolation and a heat sink. The reinforcement makes Hi-Flow 625 easy to handle, and the 65 °C phase change temperature of the coating material eliminates shipping and handling problems. The Bergquist film has a continuous use temperature of 150 °C.

Hi-Flow 625 is Tack Free and Scratch Resistant at production temperature and does not require a protective liner in most shipping situations.

Hi-Flow 625 has the thermal performance of 2-3 mil mica and grease assemblies.

Typical Properties of Hi-Flow 625

Property	Imperial Value	Metric Value	Test Method
Color	Green	Green	Visual
Reinforcement Carrier	Film	Film	***
Thickness, (inch) / (mm)	0.005	0.127	ASTM D374
Elongation, (%45° to Warp & Fill)	60	60	ASTM D882A
Tensile Strength, (psi) / (Mpa)	30000	206	ASTM D882A
Continuous Use Temp., (°F) / (°C)	302	150	***
Phase Change Temp., (°F) / (°C)	149	65	DSC
Electrical	**Imperial Value**	**Metric Value**	**Test Method**
Dielectric Breakdown Voltage, (VAC)	4000	4000	ASTM D149
Dielectric Constant, (100 Hz)	3.5	3.5	ASTM D150
Volume Resistivity, (Ohm-meter)	$>10^{10}$	$>10^{10}$	ASTM D257
Flame Rating	94 V-O	94 V-O	U.L.

Thermal Impedance vs. Pressure

Pressure (psi)	10	25	50	100	200
TO-220 Thermal Performance, (°C/W)	2.26	2.10	2.00	1.93	1.87
Thermal Impedance, (°C-in²/W) (1)	0.79	0.71	0.70	0.67	0.61

1). The ASTM D5470 (Bergquist Modified) test fixture was used and the test sample was conditioned at 70°C prior to test. The recorded value includes interfacial thermal resistance. These values are given to the customer for reference only. Actual application performance is directly related to the surface roughness, flatness and pressure applied.

2). This is the measured thermal conductivity of the Hi-Flow coating only (per Bergquist Modified ASTM-D5470). This compound is equally coated to both surfaces of Bergquist's T-600 film carrier. This lamination typically includes two layers (one to each side) of 2-mil Hi-Flow compound coated to 1.0-mil T-600 film. The Hi-Flow coatings are phase-change, thixotropic compounds and thus respond via compressive flow to heat and pressure induced stress. Knowing the average final thickness of the interface, the overall apparent thermal conductivity of the laminate can be estimated via back-calculation (ref: L = K •) of the Bergquist Modified ASTM-D5470 test results stated. This statement assumes negligible interfacial thermal resistance. Please contact your Bergquist Sales Representative or Bergquist Inside Sales if additional specifications are required.

Typical Applications Include

- Clip mounted
- Spring mounted
- Power semiconductors
- Power modules

Configurations

Available:
- Sheet form
- Die-Cut parts
- Roll form
- With or without pressure sensitive adhesive

We produce thousands of specials. Tooling charges vary depending on tolerances and complexity of the part.

Hi- Flow®: U.S. Patent 4,950,066 and others.

www.bergquistcompany.com

Product Data Sheet / PDS-0602-001-01; Rev 01

The Bergquist Company
North American Headquarters
18930 West 78th Street
Chanhassen, MN 55317
Phone: 800-347-4572
Fax: 952-835-4156

The Bergquist Company-Europe
Bramenberg 9a
3755 BT Eemnes
Netherlands
Phone: 31-35-5380684
Fax: 31-35-5380295

The Bergquist Company-Asia
9F-1, No. 314, Section 4
Ren Ai Road
Taipei, Taiwan, ROC 106
Phone: 886-2-2700-7796
Fax: 886-2-2700-7795

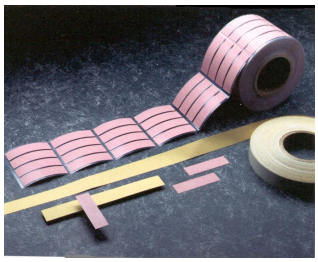
THE BERGQUIST COMPANY

Sil-Pad®800

High Performance Insulator for Low Pressure Applications

Features and Benefits

- Thermal impedance 0.45°C-in²/W (@50 psi)
- Low mounting pressures
- Smooth and highly compliant surface
- Electrically isolating



The Sil-Pad 800 family of thermally conductive insulation materials is designed for applications requiring high thermal performance and electrical isolation. These applications also typically have low mounting pressures for component clamping.

Sil-Pad 800 material combines a smooth and highly compliant surface characteristic with high thermal conductivity. These features optimize the thermal resistance properties at low pressure.

Applications requiring low component clamping forces include discrete semiconductors (TO-220, TO-247 and TO-218) mounted with spring clips. Spring clips assist with quick assembly but apply a limited amount of force to the semiconductor. The smooth surface texture of Sil-Pad 800 minimizes interfacial thermal resistance and maximizes thermal performance.

Typical Properties of Sil-Pad 800

Property	Imperial Value	Metric Value	Test Method
Color	Gold	Gold	Visual
Reinforcement Carrier	Fiberglass	Fiberglass	***
Thickness, (inch) / (mm)	0.005	0.127	ASTM D374
Hardness, (Shore A)	91	91	ASTM D2240
Elongation, (%45° to Warp & Fill)	20	20	ASTM D412
Tensile Strength, (psi) / (Mpa)	1700	12	ASTM D412
Continuous Use Temp., (°F) / (°C)	-76 to 356	-60 to 180	***
Electrical	**Imperial Value**	**Metric Value**	**Test Method**
Dielectric Breakdown Voltage, (VAC)	1700	1700	ASTM D149
Type 3 Electrodes	3000	3000	ASTM D149
Dielectric Constant, (1000 Hz)	6.0	6.0	ASTM D150
Volume Resistivity, (Ohm-meter)	10^{10}	10^{10}	ASTM D257
Flame Rating	94 V-O	94 V-O	U.L.
Thermal	**Imperial Value**	**Metric Value**	**Test Method**
Thermal Conductivity, (W/m-K)	1.6	1.6	ASTM D5470
Thermal Impedance vs. Pressure			
Pressure (psi)	10	25	50	100	200
TO-220 Thermal Performance, (°C/W)	3.56	3.01	2.45	2.05	1.74
Thermal Impedance, (°C-in²/W) (1)	0.92	0.60	0.45	0.36	0.29

1). The ASTM D5470 (Bergquist Modified) test fixture was used. The recorded value includes interfacial thermal resistance. These values are given to the customer for reference only. Actual application performance is directly related to the surface roughness, flatness and pressure applied.

Typical Applications Include

- Power supplies
- Automotive electronics
- Motor controls
- Power semiconductors

Configurations

Available:
- Sheet form
- Die-Cut parts
- Roll form
- With or without pressure sensitive adhesive

We produce thousands of specials. Tooling charges vary depending on tolerances and complexity of the part.

Sil-Pad®: U.S. Patents 4,574,879; 4,602,125; 4,602,678; 4,685,987; 4,842,911 and others

The Bergquist Company
North American Headquarters
18930 West 78th Street
Chanhassen, MN 55317
Phone: 800-347-4572
Fax: 952-835-4156

The Bergquist Company-Europe
Bramenberg 9a
3755 BT Eemnes
Netherlands
Phone: 31-35-5380684
Fax: 31-35-5380295

The Bergquist Company-Asia
9F-1, No. 314, Section 4
Ren Ai Road
Taipei, Taiwan, ROC 106
Phone: 886-2-2700-7796
Fax: 886-2-2700-7795

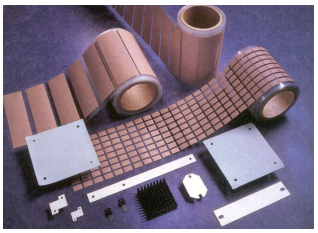
THE BERGQUIST COMPANY

Sil-Pad® A2000

High Performance, High Reliability Insulator

Features and Benefits

- Thermal impedance
  0.32°C-in²/W (@50 psi)
- Optimal heat transfer
- High thermal conductivity
  3.5 W/m-K



Sil-Pad A2000 is a conformable elastomer with very high thermal conductivity that acts as a thermal interface between electrical components and heat sinks. Sil-Pad A2000 is for applications where optimal heat transfer is a requirement.

This thermally conductive silicone elastomer is formulated to maximize the thermal and dielectric performance of the filler/binder matrix. The result is a "grease-free", conformable material capable of meeting or exceeding the thermal and electrical requirements of high reliability electronic packaging applications.

Typical Applications Include

- Motor Drive Controls
- Avionics
- High Voltage Power Supplies
- Power Transistor / Heat Sink Interface

Typical Properties of Sil-Pad A2000

Property	Imperial Value	Metric Value	Test Method
Color	White	White	Visual
Reinforcement Carrier	Fiberglass	Fiberglass	***
Thickness, (inch) / (mm)	0.011 to 0.020	0.279 to 0.508	ASTM D374
Hardness, (Shore A)	90	90	ASTM D2240
Heat Capacity, (J/g-K)	1.0	1.0	ASTM C351
Continuous Use Temp., (°F) / (°C)	-76 to 392	-60 to 200	***
Electrical	**Imperial Value**	**Metric Value**	**Test Method**
Dielectric Breakdown Voltage, (VAC)	4000	4000	ASTM D149
Dielectric Constant, (1000 Hz)	7.0	7.0	ASTM D150
Volume Resistivity, (Ohm-meter)	10^{11}	10^{11}	ASTM D257
Flame Rating	94 V-O	94 V-O	U.L.
Thermal	**Imperial Value**	**Metric Value**	**Test Method**
Thermal Conductivity, (W/m-K)	3.0	3.0	ASTM D5470

Thermal Impedance vs. Pressure

	Pressure (psi)	10	25	50	100	200
TO-220 Thermal Performance, (°C/W)	0.011"	2.06	1.93	1.82	1.77	1.71
TO-220 Thermal Performance, (°C/W)	0.015"	2.05	1.94	1.86	1.79	1.72
TO-220 Thermal Performance, (°C/W)	0.020"	2.51	2.46	2.41	2.34	2.24
Thermal Impedance, (°C-in²/W) (1)	0.011"	0.46	0.36	0.32	0.28	0.25
Thermal Impedance, (°C-in²/W) (1)	0.015"	0.53	0.40	0.32	0.28	0.26
Thermal Impedance, (°C-in²/W) (1)	0.020"	0.62	0.52	0.51	0.44	0.41

1). The ASTM D5470 (Bergquist Modified) test fixture was used. The recorded value includes interfacial thermal resistance. These values are given to the customer for reference only. Actual application performance is directly related to the surface roughness, flatness and pressure applied.

Configurations

Available:
- Sheet form
- Die-Cut parts
- With or without pressure sensitive adhesive
- Variety of thickness gages to meet customer requirements
  Preferred thickness includes 0.011", 0.015", and 0.020"

We produce thousands of specials. Tooling charges vary depending on tolerances and complexity of the part.

Sil-Pad®: U.S. Patents 4,574,879; 4,602,125; 4,602,678; 4,685,987; 4,842,911 and others

www.bergquistcompany.com

Product Data Sheet / PDS-0602-001-01; Rev 01

The Bergquist Company
North American Headquarters
18930 West 78th Street
Chanhassen, MN 55317
Phone: 800-347-4572
Fax: 952-835-4156

The Bergquist Company-Europe
Bramenberg 9a
3755 BT Eemnes
Netherlands
Phone: 31-35-5380684
Fax: 31-35-5380295

The Bergquist Company-Asia
9F-1, No. 314, Section 4
Ren Ai Road
Taipei, Taiwan, ROC 106
Phone: 886-2-2700-7796
Fax: 886-2-2700-7795

INSULATORS

CHOMERICS

Parker Seals

- *Phase-change thermal interface materials*
- *Thermally conductive adhesive tapes*
- *Thermally conductive insulator pads*
- *Thermally conductive gap fillers*
- *Thermally conductive silicone compounds*
- *Flexible heat spreaders • Thermal management for BGAs*

LEADER IN THERMAL MANAGEMENT: DESIGN, INNOVATION AND MATERIALS

TECHNICAL BULLETIN 68

CHO-THERM® 1680 Thermally Conductive Insulators For Surface Mount Applications

MATERIAL DESCRIPTION & PRODUCT FORM

CHO-THERM 1680 thermal interface insulators efficiently transfer heat away from mounted components without the need for clamping force. These pads have a high tack, pressure-sensitive adhesive on one or both sides. This configuration is optimized for mounting multichip modules and other surface mount components to PCBs. CHO-THERM 1680 thermal pads are composed of a polyimide film beneath a layer of boron nitride-filled silicone, with adhesive on one or both sides.

The 0.001 in. thick polyimide film offers excellent resistance to cut-through. The film is coated on one side with a resilient boron nitride-filled silicone, providing an effective thermal path. The adhesive bonds components to provide good heat transfer under low mounting pressures. This is especially useful in applications such as hybrid, ceramic, and flat packages.

CRITICAL PERFORMANCE CHARACTERISTICS

There are generally two objectives that must be satisfied in the interface between power semiconductor devices and their heat sinks:

1. To enhance the flow of heat from the device to the metal heat sink.
2. To electrically isolate the device from the metal heat sink.

CHO-THERM 1680 material performs both functions simultaneously.

	TYPICAL PROPERTIES	1680	TEST METHOD
CONSTRUCTION	Binder	Silicone	—
	Filler	Boron Nitride	—
	Carrier	Polyimide	—
	Color	White/Gold	Visual
	Thickness, inch (mm)	0.007 (0.18)	ASTM D374
THERMAL	Thermal Impedance °C-in²/W(°C-cm²/W)	0.40 (2.6)	ASTM D5470
	Thermal Conductivity, W/m-K	0.65	ASTM D5470
	Operating Temperature Range, °C	–60 to +200	—
ELEC.	Voltage Breakdown, Vac	6000	ASTM D149
	Volume Resistivity, ohm-cm	1×10^{14}	ASTM D257
MECHANICAL	Tensile Strength psi, (MPa)	4000 (27.6)	ASTM D412
	Tear Strength, lb/in (kN/m)	200 (35.02)	ASTM D624
	Elongation, %	25	ASTM D412
	Hardness (Shore A)	10	ASTM D2240
	Specific Gravity	1.45	ASTM D792

THERMAL INTERFACE IMPEDANCE

Thermal impedance is the measure of a material's ability to conduct heat. In addition, thermal impedance describes a material's ability to conform to irregular surfaces and minimize contact resistance.

VOLTAGE BREAKDOWN CHARACTERISTICS

When using thermal interface pads to electrically isolate a component from a metal heat sink or chassis, the critical material property for the pad is its dielectric strength. Dielectric strength is a measure of how well a material can prevent the voltage on the component case from arcing through the material and allowing an electrical short circuit between the component and the metal mounting surface. This property is commonly presented as the voltage breakdown shown in the Typical Properties Table and is determined by electrical testing of multiple flat sheet samples in accordance with the test procedures detailed in ASTM D149. The higher the value of voltage breakdown, the better the material is at withstanding applied voltages.

The dielectric strength of a material can also be affected by many external factors including: insulator thickness, area of the contact surfaces, temperature, humidity, mechanical stress applied to the insulator, the presence of partial discharge, etc. Contact Chomerics Applications Engineering for details of test methods and assistance with the electrical requirements of your specific application.

continued

INSULATORS

CHO-THERM® 1680 Thermally Conductive Insulators *continued*

CLEANING & WAVE SOLDERING COMPATIBILITY

CHO-THERM 1680 material is compatible with standard PCB wave soldering processes. Because of the relatively short exposure times, CHO-THERM 1680 material can be safely used with fluorocarbon or water based cleaning solutions. Cleaning steps should be performed after soldering.

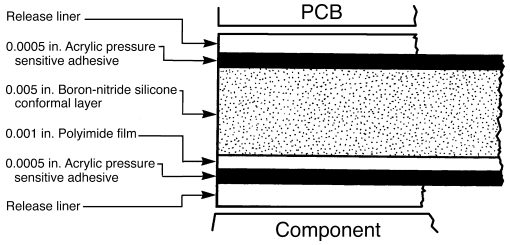
ORDERING INFORMATION

CHO-THERM 1680 thermally conductive mounting pad material is available in standard roll sizes of 12 in. (30.5 cm) width x 100 ft. (30.5 m) length. This material is also available in kiss-cut rectangular shapes. Contact Chomerics Sales Department for details on custom sizes or die-cut parts. Part numbers for standard rolls are:

66-10-1200-1680 Adhesive One Side

67-10-1200-1680 Adhesive Two Sides

FIG. 1 Cross Section of CHO-THERM 1680 Insulator



CHOMERICS *US Headquarters* **TEL:** 781-935-4850 **FAX:** 781-933-4318 • *www.chomerics.com*
Europe **TEL:** +(44) 1628 404000 **FAX:** +(44) 1628 404090
Asia Pacific **TEL:** +(852) 2 423 8008 **FAX:** +(852) 2 423 8253
Parker Seals

INSULATORS

CHOMERICS

Parker Seals

- *Phase-change thermal interface materials*
- *Thermally conductive adhesive tapes*
- *Thermally conductive insulator pads*
- *Thermally conductive gap fillers*
- *Thermally conductive silicone compounds*
- *Flexible heat spreaders • Thermal management for BGAs*

LEADER IN THERMAL MANAGEMENT: DESIGN, INNOVATION AND MATERIALS

TECHNICAL BULLETIN 69

CHO-THERM® T500
Highly Thermally Conductive Elastomer Insulators

MATERIAL DESCRIPTION & PRODUCT FORM OPTIONS

CHO-THERM T500 material is a highly thermally conductive interface insulator designed for use where the lowest possible thermal impedance is required.

One-component CHO-THERM T500 insulators combine a silicone binder with a boron nitride filler to deliver thermal impedance of just 0.19°C-in²/watt.

This elastomer is designed for use at temperatures ranging from -60°C to 200°C.

Fiberglass cloth reinforcement strengthens CHO-THERM T500 pads against tear, cut-through and punctures. The material is available in sheet form and die-cut configurations. An optional pressure-sensitive adhesive is available on one side.

In contrast to conventional insulators with thermal grease, CHO-THERM T500 material does not crack, tear or otherwise fail when torqued between mating metal surfaces. CHO-THERM T500 material eliminates the problems associated with silicone migration, contamination or drying out.

CRITICAL PERFORMANCE CHARACTERISTICS

There are generally two objectives that must be satisfied in the interface:

1. To enhance the flow of heat from the device to the heat sink.
2. To provide electrical isolation.

CHO-THERM T500 material performs both functions simultaneously, effectively replacing combinations of beryllium oxide or mica wafers and silicone grease.

	TYPICAL PROPERTIES	T500	TEST METHOD
CONSTRUCTION	Binder	Silicone	—
	Filler	Boron Nitride	—
	Carrier	Fiberglass	—
	Color	Green	Visual
	Thickness, inch (mm)	0.010 (0.25)	ASTM D374
THERMAL	Thermal Impedance, °C-in²/W (°C-cm²/W)	0.19 (1.23)	ASTM D5470
	Thermal Conductivity, W/m-K	2.07	ASTM D5470
	Operating Temperature Range, °C	–60 to +200	—
ELEC.	Voltage Breakdown, Vac	5000	ASTM D149
	Volume Resistivity, ohm-cm	1×10^{14}	ASTM D257
MECHANICAL	Tensile Strength, psi (MPa)	1000 (6.89)	ASTM D412
	Tear Strength, lb/in (kN/m)	100 (17.5)	ASTM D624
	Elongation, %	10	ASTM D412
	Hardness (Shore A)	82	ASTM D2240
	Specific Gravity	1.55	ASTM D792
	UL Recognized	File No. 57104	QMFZ2
	Outgassing: % TML % CVCM	0.40 0.10	ASTM E595

TML= Total Mass Loss
CVCM= Collected Volatile Condensible Materials
Note: Pressure-sensitive adhesive may increase thermal impedance by as much as 0.05°C-in²/W (0.32°C-cm²/W). Contact Chomerics for further information.

CONTACT PRESSURE & MOUNTING TORQUE

The optimum contact pressure range for CHO-THERM materials is 300-500 psi ($2.07 \times 10^6 - 3.45 \times 10^6$ N/m²). Beyond this range, thermal performance gains are negligible.

To convert mounting torque into contact pressure, use the following equation:

$$P = \frac{(T)(N)}{(0.2)(D)(A)}$$

P = Contact Pressure (psi or N/m²)
T = Torque (in-lbs or N-m)
N = Number of Fasteners
(0.2) = Average Friction Factor
D = Diameter (in. or m)
A = Contact Area (in² or m²)

IMPROVEMENT IN THERMAL IMPEDANCE WITH TIME

The thermal impedance characteristics of CHO-THERM materials can be expected to improve during use due to stress relaxation of the elastomer and consequent additional filling of the microscopic voids in the interface. Improvement can be as much as 10-15% after the first few weeks of use.

continued

CHOMERICS
US Headquarters **TEL:** 781-935-4850 **FAX:** 781-933-4318 • *www.chomerics.com*
Europe **TEL:** +(44) 1628 404000 **FAX:** +(44) 1628 404090
Asia Pacific **TEL:** +(852) 2 423 8008 **FAX:** +(852) 2 423 8253
Parker Seals

CHO-THERM® T500 Thermally Conductive Insulators *continued*

THERMAL INTERFACE IMPEDANCE

The thermal performance of interface materials is generally characterized by the thermal impedance across the interface in °C-in²/watt. The lower the value of thermal interface impedance, the better the thermal performance.

The thermal impedance of an interface depends greatly on a number of different parameters, including the flatness and smoothness of the mating surfaces forming the interface and the contact pressure between them, as well as the thickness of the interface material, its thermal conductivity and conformability.

VOLTAGE BREAKDOWN CHARACTERISTICS

When using thermal interface pads to electrically isolate a component from a metal heat sink or chassis, the critical material property for the pad is its dielectric strength. Dielectric strength is a measure of how well a material can prevent the voltage on the component case from arcing through the material and allowing an electrical short circuit between the component and the metal mounting surface. This property is commonly presented as the voltage breakdown shown in the Typical Properties Table, and is determined by electrical testing of multiple flat sheet samples in accordance with the test procedures detailed in ASTM D149. The higher the value of voltage breakdown, the better the material is at withstanding applied voltages.

The dielectric strength of a material can also be affected by many external factors including: insulator thickness, area of the contact surfaces, temperature, humidity, mechanical stress applied to the insulator, the presence of partial discharge, etc. Contact Chomerics Applications Engineering for details of test methods and assistance with the electrical requirements of your specific application.

CHEMICAL & SOLVENT RESISTANCE

Exposure to petrochemicals or chlorinated solvents, such as trichlorethylene, freon, toluene, trichlorethane and other cleaning agents, chemicals and solvents used in vapor degreasing, defluxing and cleaning operations is not harmful to CHO-THERM T500 materials, although exposed edges do tend to swell. The amount of swelling is a function of exposure time and type of solvent. After drying out, the exposed edges will return to their former size and conditions with no effect on thermal or electrical properties.

CHO-THERM T500 material is recognized under the Components Program of Underwriters Laboratories, Inc.

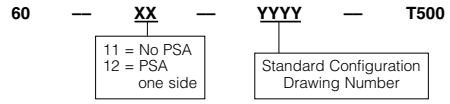
ORDERING INFORMATION

Using the diagram below, construct the appropriate part number. Part numbers for non-standard configurations will be assigned by Chomerics

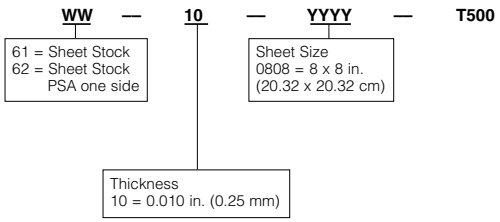
For customized die-cut parts, submit a detailed drawing of the desired part, including all dimensions, tolerances, hole locations and profile. When specifying pressure-sensitive adhesive (PSA) on non-symmetrical die-cut parts, indicate to which side PSA is to be applied.

All CHO-THERM products are available through local Chomerics distributors. Contact Chomerics for the distributor in your area.

Standard Die-Cut Parts



Standard Sheet Stock



Silicone Heat Transfer Compound Plus

TECHNICAL DATA SHEET

Product Code: HTSP

PRODUCT DESCRIPTION

HTSP provides the ultimate in thermal conductivity together with the very wide temperature range obtained by using silicone base oils. The exceptional properties obtained from HTSP are due to the novel use of various metal oxide (ceramic) powders. These materials are electrically insulative to ensure that leakage currents can not be formed if the paste should come into contact with other parts of the assembly.

HTSP should be used where a large amount of heat needs to be dissipated quickly and effectively. The heat dissipation from the source (e.g. semiconductor barrier layer) is achieved through many layers of different material before the heat is dissipated through free or forced convection. It should be noted that the use of a thermally conductive paste will only aid the dissipation of heat if the interface where it is used has the lowest thermal conductivity within the system, i.e. is the rate determining step. This is usually the case.

The rate at which heat flows is dependant on the temperature differential, the thickness of the layer, and the thermal conductivity of the material.

A full range of heat transfer products are available from Electrolube. This range includes silicone and non-silicone based pastes (HTS & HTC), an RTV rubber (TCR), an adhesive epoxy (TBS) and an epoxy based potting resin (ER2074).

A non-silicone version of this material is also available, order code HTCP.

FEATURES

* Superior thermal conductivity even at high temperatures.
* Excellent non-creep characteristics.
* Wide operating temperature range with low evaporation weight loss.
* Easy to handle, economic in use and low in toxicity.

ELECTROLUBE, A division of H K Wentworth, Kingsbury Park, Midland Road, Swadlincote, Derbyshire, DE11 0AN
Tel: +44(0) 1283-222111 Fax: +44(0) 1283-550177 www.electrolube.com
ISO 9002 Registered Firm. Certificate No. FM 32082

Silicone Heat Transfer Compound Plus – Page 2

TECHNICAL DATA SHEET

TYPICAL PROPERTIES

Colour:	White
Base:	Silicone Oil
Thermoconductive Components:	Powdered Metal Oxides
Density @ 20°C:	3 g/cm^3
Temperature Range:	-50°C to +200°C
Thermal Conductivity:	3.0 W/m.K
Weight Loss after 96 hours @ 100°C:	= 0.80%
Permitivity @ 10^6 Hz:	4.9
Specific Resistance:	1 x 10^{15} Ohms/cm
Dielectric Strength:	18 kV/mm
Viscosity:	Paste

DIRECTIONS FOR USE

Apply in a thin film, to the base and mounting studs of diodes, transistors, thyristors, heat sinks, silicone rectifiers and semi-conductors, thermostats, power resistors and radiators.

PACKAGING	ORDER CODE
50 ml Tube (150g)	HTSP50T
1 Kg Bulk	HTSP01K

ADDITIONAL INFORMATION

Some useful conversion factors are as follows:

1 cal	=	0.003968 BTU (British Thermal Unit)
1 cal/cm x sec x K	=	0.04964 BTU/in x h x ?F
	=	416.8 W/m x K
1 BTU/h x ft x ?F	=	12 BTU x in/h x sq ft x ?F
	=	0.04134 cal/sec x cm x K
1 BTU x in/h x sq ft x F?	=	0.0003445 cal/sec x cm x K
	=	0.1437 W/m x K
1 BTU/h x ft x ?F	=	1.724 W/m x K
1 W/in x K	=	22.75 BTU/h x ft x ?F
1 cal/sec x cm	=	10.6 W/in x K

ELECTROLUBE, A division of H K Wentworth, Kingsbury Park, Midland Road, Swadlincote, Derbyshire, DE11 0AN
Tel: +44(0) 1283-222111 Fax: +44(0) 1283-550177 www.electrolube.com
ISO 9002 Registered Firm. Certificate No. FM 32082

TITAN

RUNNING FASTER IS THE SYSTEM'S JOB
KEEPING THEM COOLER IS OUR BUSINESS

Thermal Grease

TTG-S104

Specification :

Length : 130 mm

Thermal Conductivity : > 7.5 Watt/m-K

Thermal Impedance : > 0.06 °C-in²/Watt

Specific Gravity : 2.5 g

Dielectric Constant A : 5.1 ohm-cm

Dissipation Factor A : 0.005

Dielectric Constant B : 5.0 ohm-cm

Dissipation Factor B : 0.001

Bleed : 0 %

Evaporation : 0.001 %

Operation Temperature : -50 °C ~ +240 °C

Ingredients :

Silicone Compounds : 50 %

Carbon Compounds : 20 %

Metal Oxide Compounds : 20 %

Silver Oxide Compounds : 10 %

Manufacturer & Exporter
TITAN Computer CO., LTD.
(Taiwan Headquarters)
25F, No. 29, Sec. 2, Jungjang E. Rd., Danshuei Jen, Taipei,Taiwan, R.O.C.
Tel: 886-2-28081189 Fax: 886-2-28084567
Email: titan@seed.net.tw

TITAN Computer GmbH
(Germany Branch office)
An der Feuerwache 9,41836 Hückelhoven, Germany
Tel:+49-2433-940160 Fax:+49-2433-9401899
Email: titan@titan-computer.de

	Unidades por Lote = 1					
	tipo	**código RS**	**precio unitario**			
			1-9	*10-35*	*36+*	
Pedido	tubo de 20ml	554-311	4,17 €	3,58 €	3,25 €	Info técnica
			1-9	*10-49*	*50+*	
Pedido	jeringa de 35ml	503-357	6,59 €	5,65 €	5,12 €	Info técnica

Dos tipos de pasta rellenas de óxido metálico, en las que se combinan las propiedades de gran conductividad térmica y alto grado de aislamiento, lo cual asegura una excelente transmisión de calor entre los dispositivos semiconductores y los disipadores de calor. Ambas mejoran el aislamiento eléctrico cuando se emplean de forma normal con las arandelas aislantes y reducen los intervalos de tiempo en los termostatos. Son químicamente inertes, amortiguan los golpes, repelen la humedad y son estables a largo plazo. Un compuesto contiene una base de silicona mientras que el otro se basa en fluidos sintéticos, que reducen cualquier riesgo de contaminación.

Características técnicas

Propiedades normales	Sin silicona (jeringa de 35ml)	Silicona (tubo de 20ml)	Unidades
SG	2,04	2,10	
Resistencia dieléctrica	42	18	kV/mm
Resistividad volumétrica	10^{14}	10^{15}	O cm
Conductividad térmica	0,9	0,9	W/m K
Temp. de funcionamiento	-100/+130	-100/+200	°C

Bibliography

[1] Bimal K. Bose. Energy, environment, and advances in power electronics. *IEEE Transactions on Power Electronics*, Vol.15(No.4):688–701, July 2000.

[2] Ned Mohan, Tore M. Undeland, and William P. Robbins. *Power Electronics*, chapter 1 and 29. John Wiley & Sons, second edition, 1995.

[3] Jerry Sergent and Al Krum. *Thermal Management Handbook*, chapter 1, 5 and 7. McGraw-Hill, 1998. ISBN 0-07-026699-9.

[4] Clemens J.M.Lasance. The thermal conductivity of ceramics. *Electronics Cooling*, Vol.5(No.3):12, September 1999.

[5] Seibersdorf research GmbH. W. Hohenauer. Austrian Research Centers. Thermophysics. [online] http://www.arcs.ac.at/AMTT/facilities/-thermophysics.html [cited 2004-July-6].

[6] M.O'Flaherty, C.Cahill, K.Rodgers, and O.Slattery. Thermal resistance measurement protocols. *Microelectronics Journal*, 29(4-5):199–208, April-May 1998.

[7] L.Kubičár and V.Boháč. Review of several dynamic methods of measuring thermophysical parameters. In *Proc. Of the 24th Int. Thermal Conductivity Conference / 12th Int. Thermal Expansion Symposium*, pages 135–149, 1999.

[8] The UK's National Measurement Laboratory. Standards for contact transient measurements of thermal properties. [online] http://-www.npl.co.uk/thermal/ctm/ [cited 2004-July-6].

[9] Silas E.Gustafsson. Transient plane source techniques for thermal conductivity and thermal diffusivity measurements of solid materials. *Review of Scientific Instruments*, Vol.62(No.3):797–804, March 1991.

[10] L.Kubičár and V.Boháč. A step-wise method for measuring thermophysical parameters of materials. *Meas. Sci. Tech.*, 11:252–258, 2000.

[11] Max Jakob. *Heat Transfer*, volume 1. John Wiley & Sons, 10 edition, 1967.

[12] M.Necati Özişik. *Heat Conduction*, chapter 1. John Wiley & Sons, 2 edition, 1996. ISBN 0-471-53256-8.

[13] Paul A. Tipler. *Mecánica. Oscilaciones y ondas. Termodinámica.*, volume 1, chapter 21. Reverté, 4 edition, 1999. ISBN 84-291-4381-5.

[14] John W. Sofia. Fundamentals of thermal resistance measurement. Technical report, Analysis Tech, 1995.

[15] Ravi Mahajan, Chia-Pin Chiu, and Ravi Prasher. Thermal interface materials. *Electronics Cooling*, Vol.10(No.1):10–18, February 2004.

[16] Flomerics Limited. *Flotherm, Introduction to Version 3.2, Tutorials*, flotherm/lc/09/01 issue 1.0 edition.

[17] Flomerics Limited. *FlothermTM On-line Help*, issue 1.0 edition, May 2003.

[18] International Electrotechnical Commission. *IEC60751: Industrial platinum resistance thermometer sensors*, 1.0 edition, 1983. Website: http://www.iec.ch.

[19] Honeywell. Temperature sensors: Platinum rtd resistance vs. temperature function. [online] http://content.honeywell.com/sensing/prodinfo/temperature/technical/c15_136.pdf [cited 2004-7-14].

[20] Duncan A. Grant and John Gowar. *Power MOSFETs, Theory and Applications*. John Wiley & Sons, 1989. ISBN 0-471-82867-X.

[21] Katsuhiko Ogata. *Modern control engineering*, chapter 10. Prentice Hall, 2002. ISBN 0-13-060907-2.

[22] Daniel Blazej. Thermal interface materials. *Electronics Cooling*, Vol.9(No.4), November 2003.

[23] Carol A. Latham. Thermal resistance of interface materials as a function of pressure. *Electronics Cooling*, Vol.2(No.3), September 1996.

[24] F.Madrid, X.Jordà, P.Godignon, M.Vellvehí, and J.Rebollo. A simple thermal conductivity measurement system. In *10th International Power Electronics and Motion Control Conference (EPE-PEMC 2002)*, page 202, 9-11 September 2002.

[25] F.Madrid, X.Jordà, P.Godignon, M.Vellvehí, and J.Rebollo. Sistema de extracción de conductividad térmica para simulación de módulos de potencia. In *Seminario Anual de Automática, Electrónica Industrial e Instrumentación (SAAEI '02)*, page 56, 9-11 September 2002.

[26] F.Madrid, X.Jordà, M.Vellvehi, C.Guraya, and J.Rebollo. An effective thermal conductivity measurement system. *Review of Scientific Instruments*, 75(11):4505–4510, November 2004.

[27] F.Madrid, X.Jordà, M.Vellvehi, X.Perpiñà, and P.Godignon. Heat power source controller circuit. *Review of Scientific Instruments*, 75(12):5123–5125, December 2004.

[28] The MathWorks, Inc. *Matlab Help.* Release 13.

[29] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C, The Art of Scientific Computing.* Cambridge University Press, 2nd edition, 1997.

[30] X. Perpiñà, X. Jordà, F. Madrid, D. Flores, S. Hidalgo, and N. Mestres. Measuring device internal temperature by means of iir-ld technique. In *24th Electronique de Puissance du futur (EPF04)*, September 2004.

[31] X.Perpiñà, X.Jordà, F.Madrid, S.Hidalgo, D.Flores, M.Vellvehi, and N.Mestres. A calibration procedure for an iir-ld equipment using a thermal test chip and an analitical model. In *10th International Workshop on Thermal Investigations of ICs and Systems (THERMINIC04)*, pages 251–255, September 2004. ISBN: 2-84813-036-9.

[32] IXYS Corporation. Direct copper bonded (dcb) ceramic substrates. [online] http://www.ixys.com/9d003.pdf [cited 2004-September-21].

[33] C. Y. Ho, R. W. Powell, and P. E. Liley. Thermal conductivity of the elements. *J.Phys.Chem.Ref.Data*, Vol.1(No.2):pp. 279–422, 1972.