ON RADIATION-BASED THERMAL SERVOING: PERCEPTION, MODELLING, CONTROL, AND EXPERIMENTS

HU LUYIN

MPhil

The Hong Kong Polytechnic University

2022

The Hong Kong Polytechnic University

Department of Mechanical Engineering

On Radiation-Based Thermal Servoing: Perception, Modelling, Control, and Experiments

HU Luyin

A thesis submitted in partial fulfilment
of the requirements for the degree of

MASTER OF PHILOSOPHY

January 2022

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

Signature:

Name: Hu Luyin

Abstract

Robotic thermal servoing (TS) is a new sensor-based temperature control method that regulates heat energy transfer processes by actively changing the robot configuration. This control method is essential for creating machines with thermo-motor intelligence for industrial, surgical, exploration, and rescue applications. Despite its practical benefits, state-of-the-art methods typically address this problem in an openloop fashion (i.e. with no thermal feedback) and with static source-surface configurations (i.e. no robot controls). The main challenge is to derive a geometrical-thermal-motor model that describes the relation between the active robot configuration and the produced dynamic thermal response. The general objective of our research is to implement different types of TS techniques and explore their practical applications. Specifically, this thesis focuses on the robotic TS scenarios where the heat radiation is dominant.

We started by investigating devices that enable the robot to perceive temperature of its surrounding environment. To this end, an RGB-depth-thermal camera system that registers an object's temperature profile to its geometrical information was developed to obtain abundant multimodal feedback in real-time.

Then, we elucidates the formulation of a "fire-to-hand" robotic TS problem, where multiple objects (with unknown thermophysical properties) attached to the same robot end-effector are controlled to move around a radiative heat source. This experimental setup is a generalization of many practical industrial applications. To effectively and simultaneously regulate the temperature values the objects, two asymptotically stable controllers, one model-based and one adaptive, were designed. The experimental results validate the feasibility of our proposed method, and the unfeasible temperature target problem is analysed in depth.

Next, we explored a "fire-in-hand" robotic TS problem, where a robot system that autonomously tracks the sun and concentrates the solar power through a Fresnel lens was developed. Relying on the optical simulations and the heat pyrolysis model, we could actively regulate the temperature of the target surface and the induced alteration of the surface property by robot motion. This functional robotic platform is a prototype of a new type of light-weight and energy-efficient field robot that effectively utilizes solar power for electricity generation and high temperature operations.

Acknowledgement

I wish to express my sincere appreciation to my supervisor, Dr David Navarro-Alarcon, whose insight, knowledge, and encouragement steered me through this research. Without his persistent help, the current result of this project would not have been achieved. It is a great honor to study and work under his guidance.

Many thanks also to Dr Andrea Cherubini with LIRMM, University of Montpellier CNRS, who provided constructive suggestions and thorough comments for improving the research paper .

The physical and technical contribution of The Hong Kong Polytechnic University, the Research Grants Council of Hong Kong, and the Jiangsu Industrial Technology Research Institute is truly appreciated. Without their support and funding, this research could not have achieved the current result.

My thanks also go out to the accompany, innovative ideas, practical suggestions, and encouragements from all members of ROMI Lab.

Last and but not the least, I would like to thank my family, my parents Hu Zhong and Lu Honglian, for supporting me spiritually throughout writing this report and my life in general.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Temperature is a ubiquitous and crucial factor in scientific research fields. Biologists find animals actively regulate their body temperatures by behavioural and physiological responses under varying environmental conditions [1]. For humans, we not only intuitively ad-

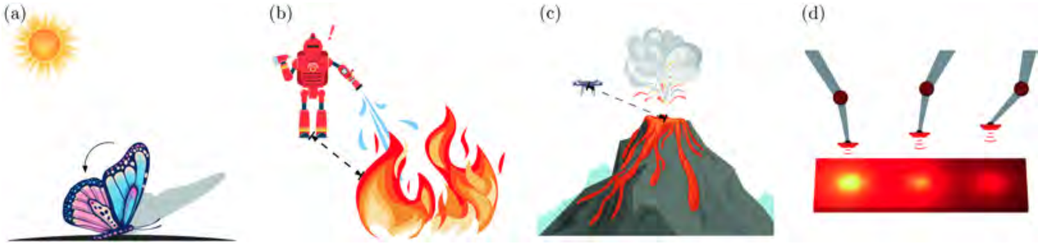


Figure 1.1: Creatures and robots with “thermo-motor intelligence”. Butterflies intuitively adjusting the angles between their wings and sunshine is depicted in (a). Future robotic systems integrated with thermal servoing algorithms for volcano exploration, firefighting and industrial applications are depicted in (b), (c), (d), respectively. Images are designed by the authors.

just the heat exchange between ourselves and environments, but also invent delicate tools and algorithms for controlling the temperatures of other physical systems (e.g., industrial temperature control systems [2]–[4]). Recently, a number of studies have also shown that the temperature, both of the robot itself and of the object being manipulated, is an important factor that affects and constrains the performance of robots for industrial [5], rescue [6], surgical [7] and exploration purposes [8].

In previous studies, temperature control was dominantly achieved by altering power supply to robotic systems. However, there has been little discussion on a robotic temperature control system that concerns or exploits heat energy transferred from the environment (like what animals usually do). This capability is important for robots that work under high temperature environments [6], contain temperature-sensitive parts[7], or equipped with heat energy harvesting devices [9]. In addition, in some industrial applications where output power of heaters is not controllable or reaches its maximum (when heating large-scale industrial products [3]), a different temperature control algorithm is needed.

1.2 Robot Thermal Servoing

To this end, we introduce robotic thermal servoing, a feedback control problem that deals with the regulation of an object's temperature by means of motor actions of a rigid robot, which can either manipulate the object or the heat source. It is a frontier problem that has numerous important applications (e.g. in industrial process control, cosmetic dermatology, fire-fighting missions, etc.) where temperature needs to be dynamically controlled and the environment is uncertain. The quality, performance and safety of these (otherwise open-loop) applications can be improved by incorporating thermal sensorimotor capabilities.

From a control systems perspective, the automation of this type of temperature-critical tasks requires: (a) the computation of a geometric-thermal-motor (GTM) model[1] describing the relation between the robot's motion and the consequent thermal response, and (b) the development of a sensor-based strategy (that relies on the thermal interaction matrix)[2] to autonomously impose a desired heat profile

---

[1] The GTM model is analogous to the geometric-image-motor model used in visual servoing to control the robot's motion [10]. Its derivation relies on thermophysical principles (to be introduced in Sec. 4.2), which correspond to the role of a camera model in the visual servoing formulation.

[2] Similar to the interaction (Jacobian) matrix of servoing problems, a thermal interaction matrix relates the heat energy inflow/outflow towards/from the object of interest (that causes a temperature change) with the robot motion.

onto the surface of interest. Note that unlike other perception modalities for robot control (e.g. vision [11], proximity [12], touch[13], audition [14] and even smell [15]), thermoception has not been fully formalized in the literature as a bona fide feedback signal for motion control. In the robotics community, we still lack the framework to fully exploit it. Up to now, the overwhelming use of thermoception in robotics has been to monitor processes (e.g., image-based visual servoing with thermal cameras [16]), but not to establish *explicit thermal servo-loops* [17], which are needed to accurately control temperature. Our aim in this paper is to develop the necessary framework that enables the design of thermal servoing controls with radiative heat sources. In general, there are two types of robotic TS techniques: When the heat source is independent of the robot (wildfire, volcano, sunshine, etc.) and temperature of the robot itself or objects attached to the robot is to be controlled, we may classify it as “fire-to-hand” type; when the heat source is attached to the robot (a common set up of power based robotic temperature control systems), we may classify it as “fire-in-hand” type (in analogy to the well-known configuration in the classical visual servoing [18]).

1.2.1 Fire-to-hand

Design of “fire-to-hand” type of algorithms can be inspired by animals (especially ectotherms) who have learned various thermoregulation behavioural responses through evolution. To name a few, marine animals that habit around hydrothermal vents actively regulate their proximity to vents according to their energetic requirements [19]; butterflies increase their body temperature by basking dorsally or laterally in the sun, or to avoid overheating by turning the thin edge of wings towards the sun [20], see Fig. 1.1 (a); ground-dwelling insects vary distances between their bodies and ground surfaces to regulate their body temperatures by heat conduction and convection [1].

Inspired by these biological examples, we may design TS algorithms for existing robotic systems. In [6], firefighting robots equipped with thermal cameras automatically locate and classify fires in burning structures; by integrating the proposed method, the firefighting robots can instruct themselves and human firefighters to work at an optical safe distance depending on everchanging fireground conditions, see Fig. 1.1 (b). Aerial volcano exploration robots are equipped with thermal cameras for data collection purposes [8]; integrating the proposed method to these drones can make them automatically adjust their configurations according to the eruption stage of the volcano, see Fig. 1.1 (c). In [21], the functional behavior of a flapping-wing robot's hinge (made of temperature-sensitive material) was controlled by varying its working temperature. In future applications, robots with temperature-sensitive parts may integrate the proposed method to exploit the heat energy transferred from environment to save energy and to improve robustness.

1.2.2 Fire-in-hand

Development of "fire-in-hand" type of algorithm is more for industrial applications. For example, in automobile industry, uniform heating of an aluminium mould during surface processing is necessary to ensure its uniform structure and coloring; in [3], the optimal configurations of multiple heaters were calculated under a simulation environment to complete the task. However, if the heater configurations are predetermined and fixed, uniform heating may not be ensured as the real heat transfer processes are likely to be different from simulations. Reformulating it as a "fire-in-hand" type of problem (where the heaters are installed on movable platforms), an adaptive algorithm that controls heater configurations according to the feedback temperature profile of the mould can be designed, which also works for moulds of different shapes. We believe "fire-in-hand" temperature control algorithm can be widely applied in similar industrial applications where the temperatures of complex-shaped or large-scale objects are regulated by multiple heaters cooperatively. Note that although robotic TS controls object temperature only by motion, if other temperature control techniques (power based, heating duration based, etc.) are available, they can also be integrated to improve efficiency.

In its most fundamental form, robotic TS involves: (a) the computation of a geometric-thermal-motor (GTM) model that describes the relation between the active robot configuration and the produced dynamic thermal response, and (b) the development of a sensor-guided motion strategy to autonomously impose a desired target temperature to the body or the surface of interest.

1.3 Related Work

Although thermal sensing is a mature technology and has a rich history in the automation of many tasks (see e.g. [22]–[25]), its use as a feedback signal for robot control has not been sufficiently studied in the literature [26], where only a few works have addressed this challenging servo-control problem. Some representative works that deal with explicit thermal control include: [27], where a fuzzy controller is developed to regulate the temperature of a fuel cell actuator; [7], where the influence of temperature in the deformation behavior of a surgical robot is investigated, and an explicit thermal regulator is designed; [5], where a control method is designed to maintain a constant tool temperature by adjusting the spindle speed in a stir friction welding robot. However, in these types of methods, temperature control is achieved by directly modulating the power of the heat-generating components. This approach is not suitable when considering external heat sources, e.g. wildfires [6] and sunlight [9], or when the source's power should not be varied, e.g. in cosmetic procedures [28].

A different strategy is to use sensor-based control, i.e. to dynamically change the source-object geometric configuration to achieve a desired thermal response (similar to what many organisms do [29]). This can be easily done by rigid robots, since their basic function is motion control. Such approach demands the development of appropriate models that can effectively capture the system's GTM relations. This idea has been partially demonstrated in [3], where the optimal fixed location of multiple radiating heaters in a process is automatically calculated to evenly imprint a desired thermal profile onto a surface. Yet, the heater is static and the method requires exact knowledge of all thermodynamic parameters (which are generally unknown). The proposed approach has also the potential to be used e.g. in fire-fighting [30] or volcano exploration robots [8] to calculate optimal trajectories that avoid overheating or damaging the robot's components.

The dynamic coupling between temperature and motion may seem unintuitive for humans [31]. Yet, many organisms extensively exploit these relations. For instance, marine animals that inhabit hydrothermal vents manage their energetic demands by controlling their proximity to the source [32]. Butterflies increase their body temperature by basking dorsally/laterally to maximize sun expose [20]. Ground-dwelling insects adaptively change their body configuration with respect to the sun-heated ground to regulate their temperature [1]. Such advanced thermoception-based behaviors can be used to solve many real-world problems (see Fig. 1.1). However, these capabilities have not yet been fully incorporated in robot control, a discipline with good track record of borrowing inspiration from nature [33]–[35], but which seems to be lagging in this direction.

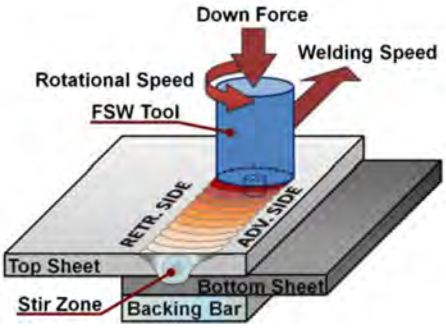


Figure 1.2: Temperature control of robotic friction stir welding [5]

1.3.1 Industrial Temperature Control System

Temperature control system for industrial applications has been well-studied due to its economical importance. Temperature control systems for heat treatment processes was designed due to its significant impact on the mechanical and structural properties of the end products [36]. Precise regulation of the heating and cooling temperatures during different stages is essential to ensure a good product quality. A number of studies implemented numerical simulation based method to analyze the heat transfer process, which serves as a general guidance for designing temperature control algorithms. In [37], researchers adopted the broadly used FVM (Finite volume method) method to predict the radiative heat transfer for a transient slab heating analysis. In [38], A new reheating furnace heat transfer model that divides the heating volume into small regions depending on the thermophysical conditions is proposed and verified by comparing to the benchmark numerical simulation. The advantage of implementing such numerical simulations of heat transfer processes is its high fidelity. In industrial environments, the conditions of the heat source, environments can be strictly controlled; thus, results of numerical simulations can usually reflect the true heat transfer process. However, from the robotics research point of view, the uncertainty of the natural environment is very common in outdoor applications; in addition, the dynamic movement of the robot during operation will change the static heat transfer model. To cope with problem, a real-time control method based on simplified and robust heat transfer model needs to be designed.

1.3.2 Potential Robotic Thermal Seroving Systems

Researchers from robotics fields also found that temperature is a import factor that constrains and affect the performance of the robotic

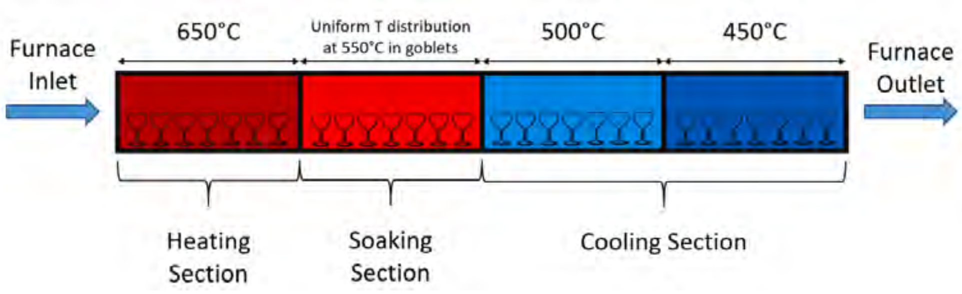


Figure 1.3: Schematic of the considered furnace model including heating, soaking and cooling sections [36]

system. In [39], a novel robotic system for skin photo-rejuvenation is developed [39] (See Fig.1.4).

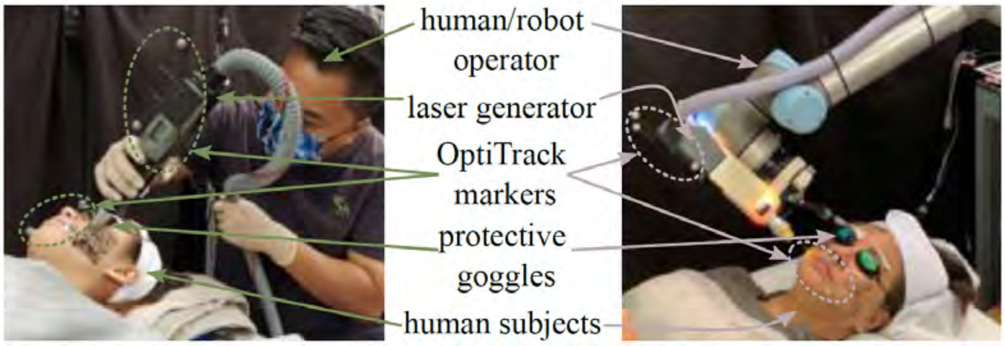


Figure 1.4: Skin photo-rejuvenation treatment is conducting by human and robot operator [39].

The working principle of the skin treatment is to stimulate the skin tissue by a laser shot with exact amount of energy. If the energy level is too low, the treatment will not have a significant effect; If the energy level is too large, it may cause damage to the skin. Due to the various skin conditions of human, it is hard to design a static algorithm that works perfectly for every circumstance. However, if

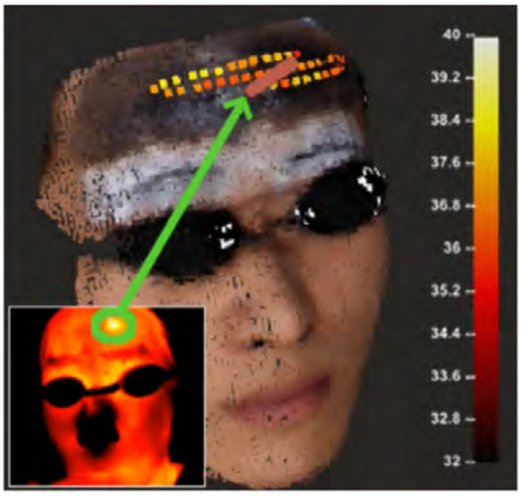


Figure 1.5: Thermal registration and feedback on control interface [39]

the feedback temperature profile of human face from the thermal camera is available (See Fig. 1.5), a dynamic calibration algorithm based on a preliminary test conducted before the treatment and a reliable skin tissue heat transfer model can be designed, which ensure the treatment result for different types of skins.

For other types of robotic systems that may benefit from the integration of the robotic thermal servoing algorithm, we have demonstrated a few of examples in Chapter 1. The potential applications include but not limited to robots equipped with thermoelectric components[40], robots for rescue mission in a fire-ground[41], robots for high temperature environment exploration[42], and robots for food preparation tasks[43].

1.3.3 RGB-D-T Multimodal Camera System

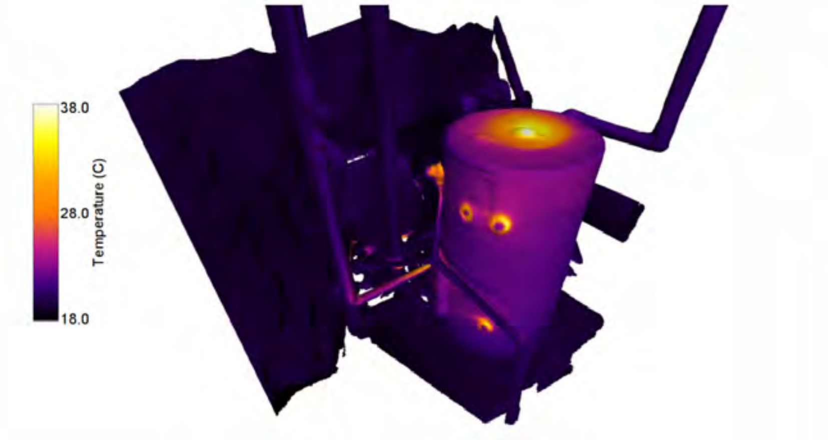


Figure 1.6: Result of a 3D thermal mapping of an underground hot water system. [44]

There are various types of temperature measurement devices that can be used for collecting the feedback temperature information. A thermisotor, which measures the temperature of the point it contacts, is widely used in industrial applications and research. However, if a temperature profile of an object surface is to be controlled, instead of using multiple thermistors to the surface, using a thermal camera to observe the target surface is a better choice.

In robotics research filed, depth camera has been integrated with robotic systems for collecting visual and geometrical information about its environment. In recent years, there has been an increasing interest in integrating a thermal camera with a depth camera to develop a multimodal perceptual system that exploits the hidden information of the object temperature distribution. The core technique required for building such a camera system is to design an accurate and effective calibration algorithm for the two independent cameras. A common method for geometric camera calibration uses a checkerboard to estimate the intrinsic parameters of the camera [45]. However, checkerboard patterns are not visible in the thermal images (there is no temperature difference between the white and black blocks of the checkerboards, thus, no distinguish characteristics). Modified approach needs to be developed to calibrate a thermal camera. In [44], 3D thermal mappings of building interiors are generated using a calibrated camera system for energy efficiency improvement. In [46], researchers suggest that the calibrated camera system can be used for power station maintenance. In [47], a mobile robot equipped with the camera system automatically surveillance the temperature control and electricity supply system in a building. In [48], the camera system is used for augmented reality design.

1.4 Objectives

This research aims to develop the first thorough formulation of the robotic thermal servoing problem. The main objectives include:

- To develop a multimodal perception system that is composed of a depth camera and a thermal camera, which enable the robot to obtain geometric and temperature information of the environment in real-time.

- To formulate a radiation-based “fire-to-hand” robotic TS method that regulates the temperature of multiple objects simultaneously. The aspects of the formulation include the heat transfer modeling, non-linear control design, experimental validation, and target feasibility analysis.

- To develop a robotic system that automatically concentrates solar energy to a target point, which is guided by the feedback information from a thermal camera and an RGB camera.

1.5 Research Outline

The remaining part of the thesis is organized as follows. In Chapter 2, we demonstrate the proposed multi-camera calibration workflow. The principles of multiview geometry and camera calibration is introduced, the design of the customised calibration tool is explained, the image processing method for feature extraction is introduced, and the visualization of the produced real-time thermal point cloud is illustrated. In Chapter 3, we formulate the “fire-in-hand” radiation-based robot TS method. We introduce the knowledge of heat transfer modeling with an emphasis on its connection between the robot servoing formulation. Multiple scenarios of the object configuration are discussed, and a novel real-time numerical method for computing the view factor for irregularly shaped objects at arbitrary configuration is introduced. In Chapter 4, the formulation and stability analysis of a model-based controller and an adaptive controller is demonstrated. Detailed experiments are reported and analysed to validate the performance of the proposed controllers. Subsequently, the target feasibility problem is addressed from optimization, analytical and experimental perspectives. In Chapter 5, the development of a robot system that automatically concentrates solar energy to a target point is reported. The geometric optics of the utilized Fresnel lens is demonstrated, as it is the foundation of the algorithm designs of the system. The forward and inverse kinematics that guides the robot motion is analysed, and the results of creating desired patterns on the wood plane using concentrated solar energy are reported. Finally, the conclusion and future works are discussed in Chapter 6.

Chapter 2

Multimodal Camera System

2.1 Background

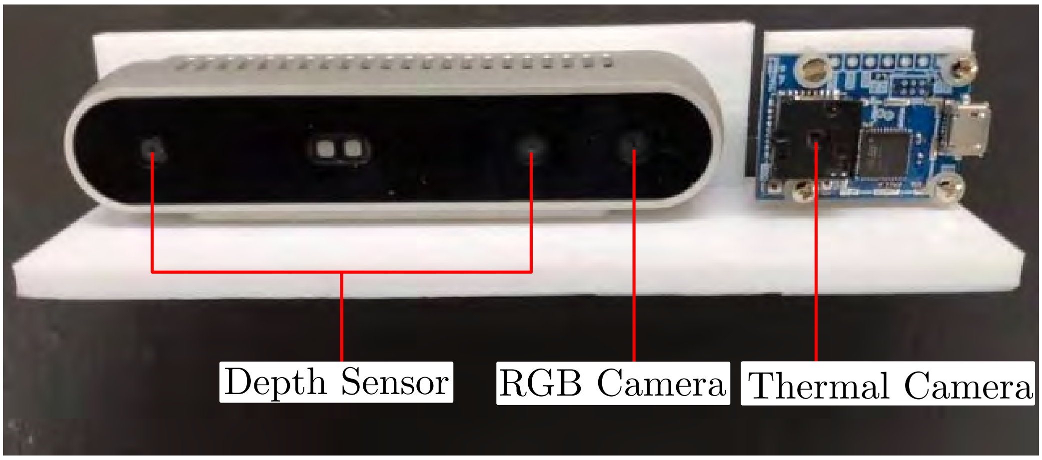


Figure 2.1: The developed multimodal camera system.

Depth cameras have been widely used in various robotic applications such as object manipulation, navigation, and human-robot cooperation. The depth cameras output RGB images and depth images of the same size simultaneously. The two images are calibrated such that the pixel values in the depth image represent the depth values of the pixels in the RGB image. Relying on the standard camera model, camera parameters, and fixed extrinsic transformations, an RGB image and a depth image are fused to output a RGB-D point cloud, which is a valuable representation of the environment: The robot could utilize the color information for feature extraction and object recognition, and the geometrical information for object manipulation and path planning.

For a robot system designed for thermal servoing, except for the color and geometric information obtained by the depth camera, it is necessary to equip it with devices that can perceive the temperature information of its surroundings. Depending on the application scenario, we have used two types of temperature sensors in this thesis: (a) Thermistor, a traditional contact temperature measurement unit that relies on the change of material's resistance to indicate the temperature change at its contact point. (b) Thermal camera, a non-contact temperature measurement device that receives the long wave infrared radiation emitted by objects, and formulates a thermal image that represents the temperature profile of the environment.

Comparing with the thermistor, the thermal camera outputs more abundant surface temperature distribution information within a shorter response time; however, its performance is limited by the possible occlusion between objects, the smaller range of working temperature, the limited field of view, and the different object surface emittances in the scene. These two types of devices can be used independently or cooperatively in thermal servoing applications depending on the specific task requirements. In this section, we present the integration of a consumer-grade (160 × 120 pixels resolution) thermal camera FLIR Letpon 3.5 with a common depth camera RealSense D415 for robot thermal servoing applications. The developed system is depicted in Figure 2.1. In Chapter 4 and 5, we use the measurements from thermistors instead as the feedback information to the robot thermal servoing system to avoid the aforementioned limitations of the thermal camera.

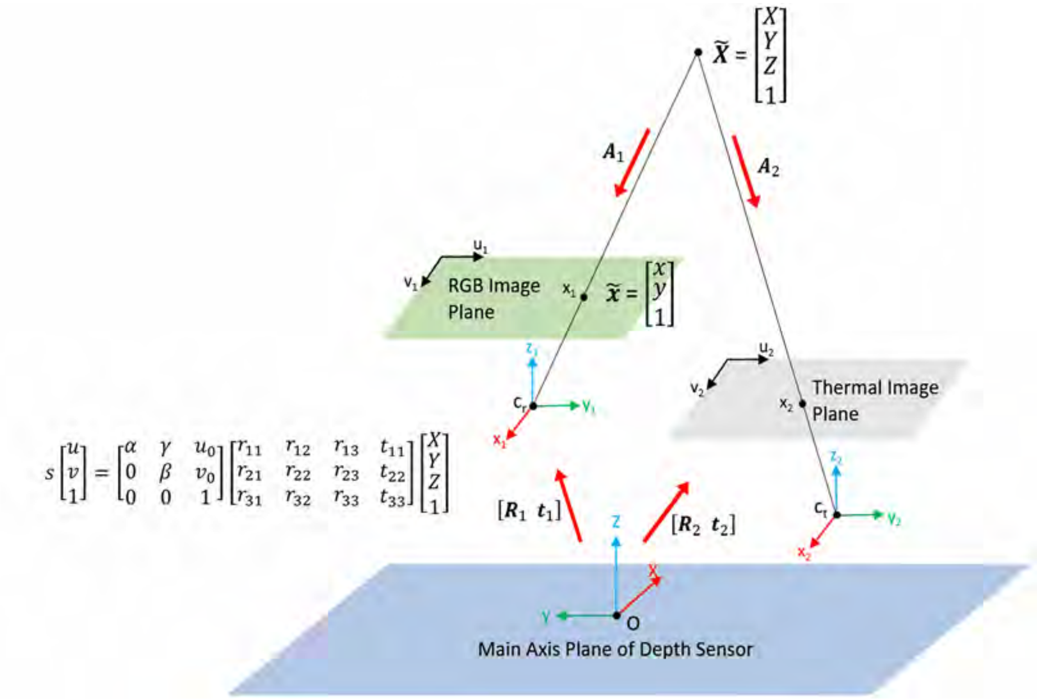


Figure 2.2: Mathematical notations of the transformations and coordinate systems in the camera system.

2.2 Methodology

2.2.1 Camera System Model

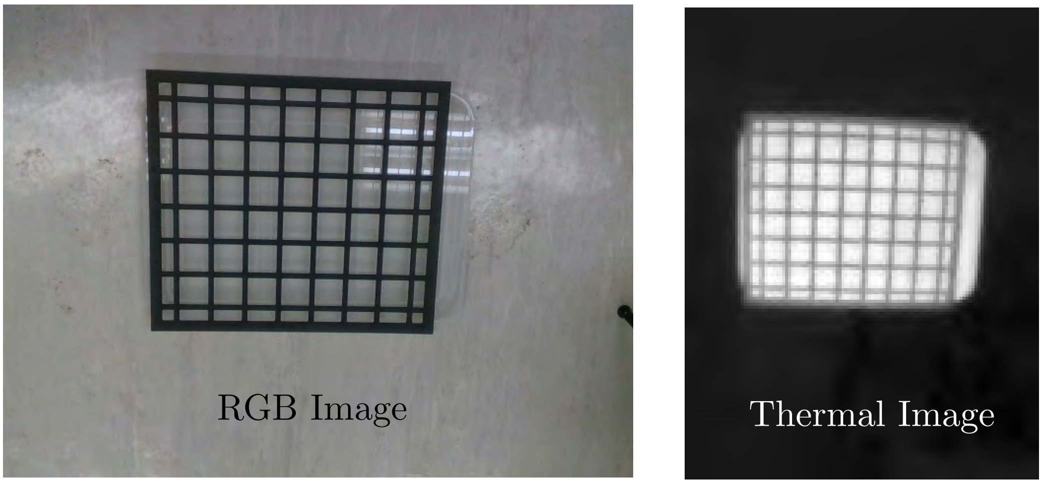
To integrate the feedback information from the two independent cameras, we developed a system calibration algorithm that determines the color and temperature information of a 3D point in real-time. The essential information required for the designed algorithm includes two sets of intrinsic parameters of the cameras and the transformations between coordinate systems. As depicted in Figure 2.2, the three major coordinate systems of the system are the RGB camera frame with the origin $\mathbf{C}_r$, the thermal camera frame with the origin $\mathbf{C}_t$, and the depth sensor frame with the origin $\mathbf{O}$. We denote the image coordinate of a pixel in the RGB image by $\mathbf{x}_r = \begin{bmatrix} u_r & v_r & 1 \end{bmatrix}^\mathsf{T}$, the image coordinate of a pixel in the thermal image by $\mathbf{x}_r = \begin{bmatrix} u_t & v_t & 1 \end{bmatrix}^\mathsf{T}$, and the corresponding 3D point of the two pixels by $\mathbf{X} = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^\mathsf{T}$. According to the standard camera projection model [49], the 3D world points and the 2D image points satisfy the following equations:

$$s_r \mathbf{x}_r = \mathbf{A}_r \mathbf{E}_r \mathbf{X}, \tag{2.1}$$

$$s_t \mathbf{x}_t = \mathbf{A}_t \mathbf{E}_t \mathbf{X} \tag{2.2}$$

where $\mathbf{A}_r, \mathbf{A}_t$ and $\mathbf{E}_r, \mathbf{E}_t$ are the intrinsic and extrinsic matrices of the two cameras, and s_r, s_t are two arbitrary scaling factors. It is critical to notice that in our proposed method, the RGB camera frame (which moves along with the motion of the camera system platform) is selected as the world coordinate system. Since the two cameras are rigidly attached to the platform, $\mathbf{A}_r, \mathbf{A}_t$ and $\mathbf{E}_r, \mathbf{E}_t$ are constant matrices.

2.2.2 Customzied Calibration Board



(a) (b)

Figure 2.3: The RGB image and the thermal image of the designed multimodal camera system calibration tool.

The standard calibration algorithm for obtaining the intrinsic and extrinsic parameters of a regular RGB camera has been implemented in various programming languages.It requires a specific calibration tool which contains distinct visual features that can be detected in the visual spectrum. Common calibration patterns like chessboard, circles, and ChArUco pattern utilize black and white color to formulate the required patterns and features. However, these calibration tools are not applicable for thermal camera calibration since a thermal camera is not capable of differentiating colors, but is designed to detect the temperature information. To this end, for our multimodal camera system, a calibration board that presents recognizable features in both RGB images (color difference) and thermal images (temperature difference) needs to be manufactured.

There are a few thermal camera calibration tools that have been designed in previous studies. In [50], a traditional chessboard was heated by a flood lamp and the temperature difference appeared since the black block area absorbs more radiation energy than the white block area. We tried to use this type of calibration tool for our camera system calibration, but the obtained thermal images are not desirable. Our thermal camera has a much lower resolution than the one used in [50], which makes the thermal image blurry and hard to detect the corner features in the chessboard patterns. Due to the resolution limitation, the line type calibration tool proposd in [51] is also not appropriate. In [52], a board with multiple circular holes was heated such that the centers of the circles are selected as features in a thermal image. It inspired the design of the calibration tool used in our study as shown in. The difference is that instead of using the centers of circles as the feature, we use the centers of crossings as the target feature. The advantage of this selection is that the depth information of the selected feature is available (for center of circular wholes, depth is not measurable), so we can directly obtain the 3D coordinates of the features to obtain the static extrinsic transformation. In addition, to uniformly heating up the board, we place the calibration board above a flat box that contains hot water, which easily creates a steady and uniform temperature difference between the calibration board and the background. The RGB image and the thermal image of our calibration tool are shown in Figure 2.3.

2.2.3 Calibration and RGB-D-T Point Cloud

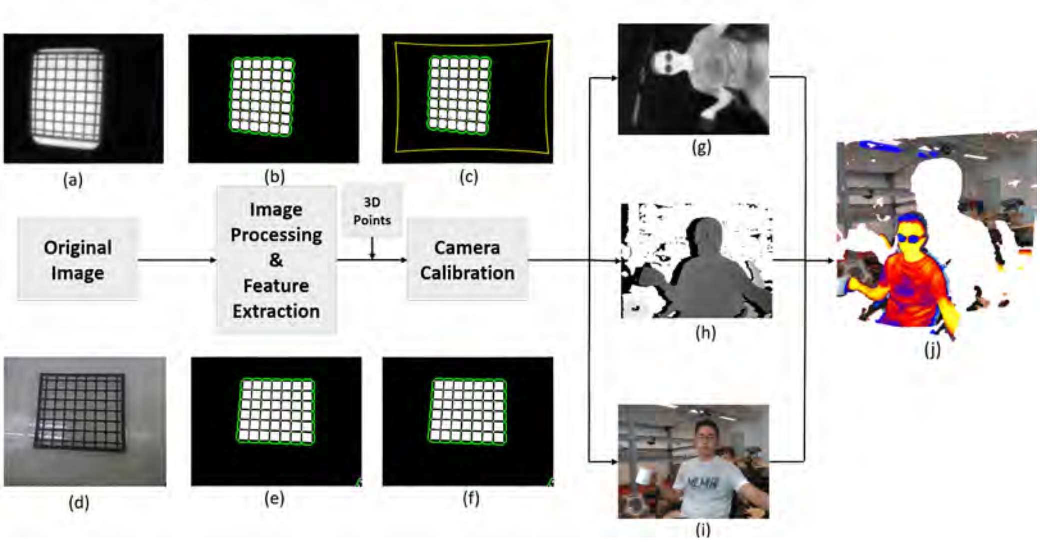


Figure 2.4: The workflow of the implemented algorithm.

In this section, we introduce the implementation of our calibration algorithm and output of the camera system, RGB-depth-thermal point cloud. The overall workflow of the method is depicted in Figure 2.4. We first collect 30 pairs of RGB and thermal images of the calibration board at different orientations and distances. Then, we temporally select the calibration board plane as the world plane to implement the feature extraction algorithm and camera calibration algorithm to obtain the intrinsic parameters of the RGB and thermal camera. For the feature extraction, we transform the collected images in gray color scale and apply the Gaussian filter to remove noise, then select an appropriate threshold to extract the whole pattern of the calibration board. Next, we extract the structuring element, i.e. the crossings in the image by using the following kernel (the actual size is 13 by 13):

$$\begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ 1 & \cdots & 1 & 1 & 1 & \cdots & 1 \\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix} \tag{2.3}$$

We then find the centers of the contours of the remaining features and record their image coordinates in a pre-defined order for the camera calibration. An example of the feature extraction process is demonstrated in Figure 2.5.

Once the intrinsic parameters of the RGB camera is calculated,

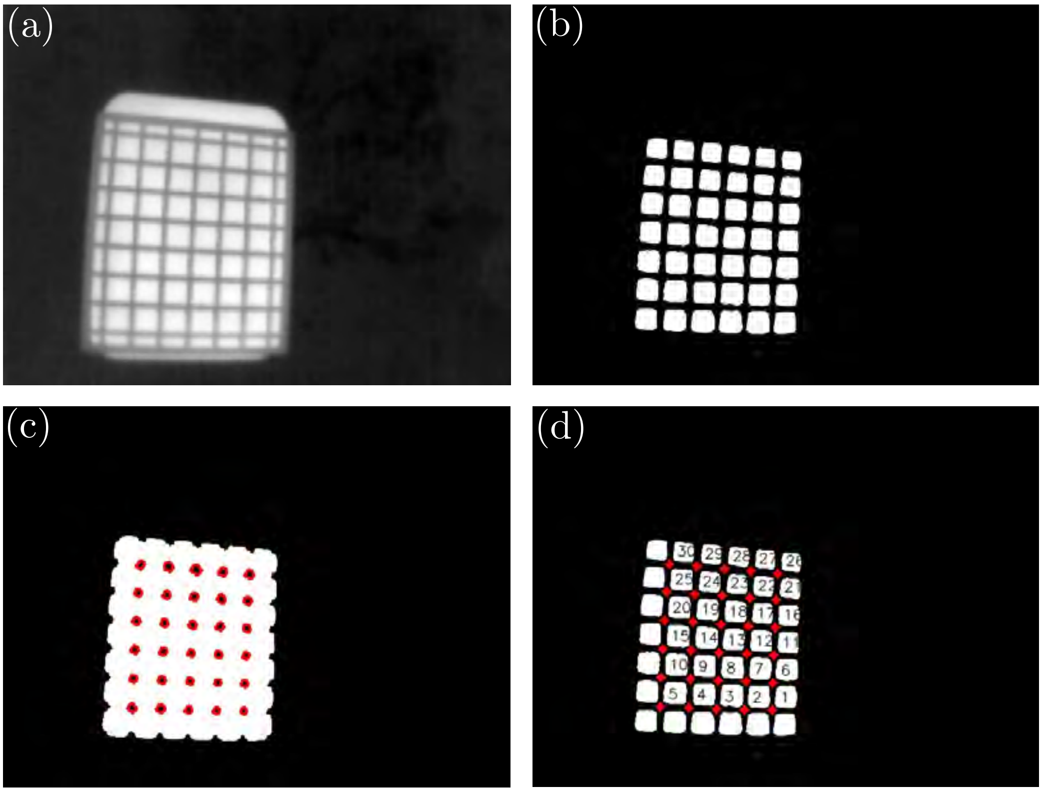


Figure 2.5: Feature extraction from a thermal image.

we can use the aligned depth and RGB image to generate the 3D coordinates of the features in the world (RGB camera) coordinate. Note that although the depth information is obtained by the depth sensors, the align function provided by the SDK of the depth camera automatically completed the transformation from the depth sensor frame to the RGB camera frame. We then implement the calibration method using the collected thermal images with the real world coordinates to obtain the static extrinsic transformation from the RGB camera frame to the thermal camera frame. Thereby, we can project each 3D point (in the RGB camera frame) to the RGB image and to the thermal image to find its corresponding color and temperature information. For any pair of collected RGB, depth, and thermal image, the information could be integrated within 0.14 second and visualized using the Ploty library in the form of RGB-D-T point cloud as shown in Figure 2.4.

Chapter 3

“Fire-to-Hand” Thermal Servoing: Mathematical Modelling

3.1 Notation

Throughout this report, we use the following mathematical notation. We denote column vectors and matrices by small bold and capital bold letters, eg. $\mathbf{v} \in \mathbb{R}^n$ and $\mathbf{M} \in \mathbb{R}^{m \times n}$. We denote scalar and vector errors by $\Delta T = T - T^* \in \mathbb{R}, \Delta \mathbf{T} = \mathbf{T} - \mathbf{T}^* \in \mathbb{R}^n$ where $T^*, \mathbf{T}^*$ are scalar and vector of constant target temperatures.

3.2 Heat Transfer Model

In following subsections, we introduce the basic thermodynamic knowledge for developing a heat transfer model referring to [53] and [54], but in a different way such that the effect of motion on varying the radiative heat transfer is emphasized. Consider a robot manipulator with end-effector configuration denoted by the vector $\mathbf{x} \in \mathbb{R}^n$. The robot rigidly grasps a planar object through an adiabatic layer. The object surface temperature should be controlled by changing its geometric configuration relative to a planar heat source. To this end, we first derive the relevant heat transfer model that is composed of the following three parts: (i) heat source, (ii) heat collector (i.e., the object), and (iii) surrounding environment, see Fig. 3.1. Thermophysical parameters of different parts are denoted by the same symbol with different subscripts. We denote the heat source temperature by T_1. Temperature of the object is denoted by T_2 and assumed to be spatially uniform at any instant during the heat transfer process (a common practice of using the *lumped capacitance method* for solids experiencing sudden thermal changes[53]). The environment temperature is denoted by T_3 and assumed to be constant.

Heat energy transfer occurs amongst the three parts whenever T_1, T_2, T_3 have different values. The direction of heat transfer is always from a high temperature part to a low temperature part. We denote the net energy transferred to the object by Q_2, where a positive value indicates the energy inflow. We introduce $q_2 = \mathrm{d}Q_2/\,\mathrm{d}t$ to represent the surface's net heat transfer rate and $v = \mathrm{d}T_2/\,\mathrm{d}t$ to describe the temporal change of measurement T_2. According to the energy conservation laws, these quantities satisfy the relation

$$v = \frac{1}{m_2 c_2} q_2 \tag{3.1}$$

where m_2 denotes the object's mass and c_2 denotes the material's specific heat. To synthesise a thermal servoing controller, it is useful to find an expression of the following form:

$$v = f(\mathbf{x}, T_2) : \mathbb{R}^6 \times \mathbb{R} \mapsto \mathbb{R} \tag{3.2}$$

which provides the thermal-geometric relation between the robot's configuration and the temperature changes. In the following subsections, we derive such a model by using the laws of thermodynamics.

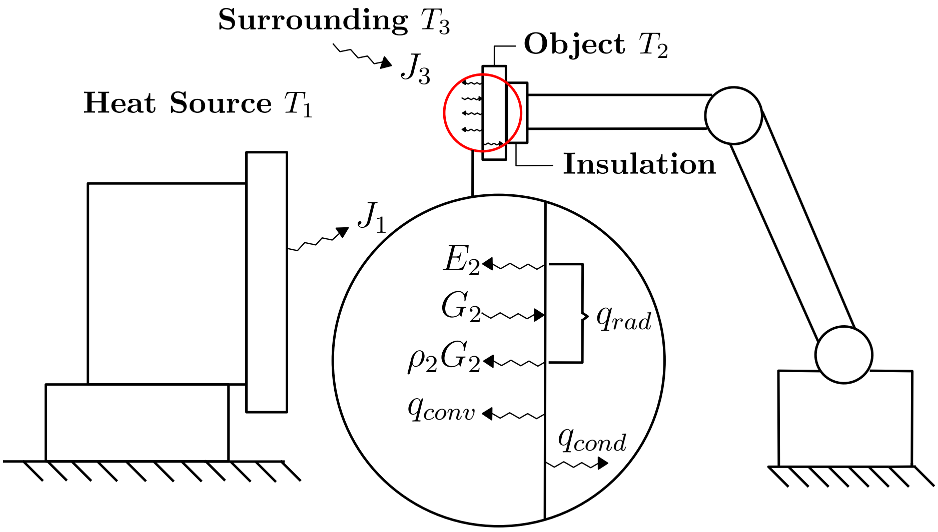


Figure 3.1: Conceptual representation of the heat transfer model. A part of the object surface is magnified to show different heat transfer processes.

3.3 Radiation Exchange Between Planar Surfaces

In this subsection, we demonstrate how to calculate q_2 between planar surfaces when the thermophysical properties (temperature, mass, specific heat, emissivity, absobpivity, etc.) of the parts are known. According to the different mechanisms (random microscopic motion and forced macroscopic motion of particles [55]) involved in heat transfer processes, the heat transfer rate q is the sum of the radiative heat transfer rate q_{rad}, the convective heat transfer rate q_{conv}, and the conductive heat transfer rate q_{cond}. In our case, we assume q_{cond} is negligible since the object is attached to the end-effector through an adiabatic layer. Since the constant heat source is kept at a much higher temperature (relative to the other parts), q_{rad} is the dominant part of the heat transfer process. Therefore, we also neglect the effect of q_{conv} in computing q_2.

Heat radiation is the emission of energy in the form of electromagnetic waves from all matter that has a temperature greater than absolute zero. Considering heat radiation exchange between the object and the other parts, we derive q_2 as:

$$q_2 = q_{rad} = A_2(\alpha_2 G_2 - E_2) \tag{3.3}$$

where A_2 denotes the object's surface area, G_2 denotes the rate of the radiation incident to the surface per unit area (irradiation, $\mathrm{W/m^2}$), $\alpha_2 \in [0,1]$ denotes the object's absorptivity (a unitless quantity describing what portion of irradiation is aborbed by the object surface), E_2 denotes the rate of radiation emitted by a surface per unit area (emissive power). The emissive power of an object is approximated using Stefan-Boltsman law as:

$$E_i = \varepsilon_i \sigma T_i^4 \tag{3.4}$$

where $\varepsilon_i \in [0,1]$ denotes the material's emissivity, T_i (measured in

K) denotes the temperature of the object ($\sigma = 5.67 \times 10^{-8}$ is the Stefan-Boltzmann constant). For an opaque surface, a portion α_i of the irradiation is absorbed by the surface while the other portion ρ_i is reflected to other parts; these two quantities must satisfy $\rho_i + \alpha_i = 1$. We introduce the radiosity, $J_i = E_i + \rho_i G_i$, which models the rate of radiation that leaves surface i. To use related thermodynamics principles to build a model for control purposes, we first make the following assumptions:

1. The surface of the heat source and object are diffuse emitters and receivers, and have uniform radiosity.
2. The environment (i.e., the room) is large enough so that it can be modelled as a blackbody ($\varepsilon_3 = \alpha_3 = 1$).
3. For the heat source, $E_1 \gg \rho_1 G_1$, therefore, we fairly approximate its radiosity as $J_1 \approx E_1$.

Remark 1. Here, we introduce the view factor F_{ij}, which is defined as the fraction of J_i that is incident to the j surface. Calculation of view factor is essential for developing radiation-based heat transfer models. The detailed derivation of F_{ij} for different cases are demonstrated in later subsections. Here, we assume F_{ij} are known scalars and focus on the derivation of q_2.

The radiation incident to a surface is the summation of the corresponding portion of radiation from other surfaces. Thus, G_i can be evaluated from the following expression:

$$A_2G_2 = \sum_{j=1}^{3} F_{j2}A_jJ_j = F_{12}A_1J_1 + F_{22}A_2J_2 + F_{32}A_3J_3 \tag{3.5}$$

Note that for planar surfaces, $F_{ii} = 0$. In our case, $F_{11} = F_{22} = 0$. To further simplify (3.5), we introduce the reciprocity relation $A_iF_{ij} = A_jF_{ji}$ and the summation rule $\sum_{j=1}^{N} F_{ij} = 1$ (Interested readers may refer to the book [53] for detailed explanations of these two principles). According to reciprocity relation, summation rule, and the assumptions we made, (3.5) is transformed to:

$$\begin{aligned} A_2G_2 &= F_{21}A_2J_1 + F_{23}A_2J_3 \\ &= F_{21}A_2E_1 + (1 - F_{21})A_2E_3 \end{aligned} \tag{3.6}$$

By injecting (3.6) in (3.3), we have

$$q_2 = A_2\alpha_2(E_1 - E_3)F_{21} + A_2\alpha_2E_3 - A_2E_2 \tag{3.7}$$

By injecting (3.7) and (3.4) in (3.1), we have

$$v = \lambda_1 F_{21} - \lambda_2 T_2^4 + \lambda_3 \tag{3.8}$$

Where the scalars λ_1, λ_2, and λ_3 satisfy:

$$\lambda_1 = \frac{A_2 \alpha_2 \sigma (\varepsilon_1 T_1^4 - T_3^4)}{m_2 c_2},\ \lambda_2 = \frac{A_2 \varepsilon_2 \sigma}{m_2 c_2},\ \lambda_3 = \frac{A_2 \alpha_2 \sigma T_3^4}{m_2 c_2} \tag{3.9}$$

Notice in , the only variable is F_{21}, which depends on the robot configuration $\mathbf{x}$

Remark 2. Intuitively, F_{12} is the parameter which directly relates to q_2, as it is defined as the portion of the radiation from the heat source that is incident to the object. However, recall the reciprocity relation, $A_1 F_{12} = A_2 F_{21}$, which shows F_{21} is nothing but F_{12} after scaling. In our case, A_1 is much larger than A_2, and makes F_{21} a better choice for further analyses and visualization purposes.

3.4 Uniform Temperature Assumption

In the formulation, we need to clarify that we only consider "small enough" objects, such that their temperature can be fairly approximated by a single sensing point. The "uniform temperature distribution” assumption is a prior condition of the “Lumped Capacitance Method”, see [56, Chapter 5], which is commonly used to simplify **transient** heat conduction problems. We would like to clarify that the essence of the assumption is to neglect the **internal** heat conduction effect, such that the heat transfer process can be concisely modeled by equation (1).

Necessity: Without this assumption, information about object’s surface temperature profile and the internal temperature distribution of the object is required to model the heat transfer process. The surface temperature profile could be obtained by a thermal camera; However, it is inapplicable in this thesis since the objects and the heat source sometimes get very close and the object surface is occluded. For the internal temperature distribution, it is usually estimated using finite-element methods since it is intricate to be measured directly. Yet, that might significantly increase the computation time for each step, and require an accurate estimation of the object’s thermophysical properties.

Consequence: A direct consequence of applying the uniform temperature assumption is that it differs from the real heat transfer process. Nevertheless, according to the **Biot number criterion**, the assumption reasonably approximates the real situation if the Biot number $Bi < 0.1$. In this thesis, the **largest** Biot number is $Bi_{rad} = 0.058 < 0.1$, which is calculated according to the following formula:

$$Bi_{rad} = \frac{R_{cond}}{R_{rad}} = \frac{L\varepsilon\sigma F_{12}(T_1^2 + T_2^2)(T_1 + T_2)}{k} \tag{3.10}$$

Where $L = 0.05\,\mathrm{m}$ is the object characteristic length, $\varepsilon = 0.25$ is the emittance of the heat source, σ is the Stefan-Boltzmann constant, $F_{12} \in [0, 0.041]$ is the view factor (heat source to object), $T_1 = 473.15\,\mathrm{K}$ is the heat source temperature, $T_2 = 333.15\,\mathrm{K}$ is the object temperature, and $k = 0.13\,\mathrm{W}/(\mathrm{m}\cdot\mathrm{K})$ is the heat conductivity of the PLA material. An intuitive explanation of the validity of the assumption and the Biot number is: When the internal heat conduction transfers **fast** enough in the solid body with respect to the external heat flow, it is fair to approximate the temperature throughout the object changes simultaneously (and uniformly).

Another consequence of the assumption is that instead of controlling the entire surface temperature profile, we only control the temperature of the surrounding area where the thermistor is attached (within the range where the Biot number criterion is satisfied). If the whole surface temperature profile is to be controlled, multiple sensors attached to the same object can be used as feedback. The formulation should be similar to Section 3.8. However, we would like to point out that if a **uniform** temperature profile needs to be precisely controlled, it is more appropriate to use other heat sources (e.g., a hair dryer) that generates relative uniform heat flow, since the spatial heat intensity of radiative heat source is essentially non-uniform.

3.5 View Factor

We introduce the mathematical definition of the view factor F_{ij} in this subsection. As shown in Fig. 3.2, consider an elementary area $\mathrm{d}A_1$ on the heat source, and an elementary area $\mathrm{d}A_2$ on the object; these two elementary areas are separated by a length r which forms the polar angles θ_1 and θ_2. The view factor is defined as follows:

$$F_{21} = \frac{1}{A_2} \int_{A_2} \int_{A_1} \frac{\cos\theta_2 \cos\theta_1}{\pi r^2} \mathrm{d}A_2 \, \mathrm{d}A_1 \tag{3.11}$$

The analytical solution of the above double surface integral is usually complicated to derive. A variety of analytical and numerical solutions are proposed to calculate the view factor [57]–[59]. Here, we implement an analytical method introduced in [60], which implements Stokes' theorem to convert the double surface integrals to double contour integrals as follows:

$$F_{21} = \frac{1}{2\pi A_2} \oint_{\Gamma_1} \oint_{\Gamma_2} \ln s \, \mathrm{d}\mathbf{s}_2 \, \mathrm{d}\mathbf{s}_1 \tag{3.12}$$

where Γ_i denotes the contour of the ith surface, $\mathbf{s}_i$ is the position vector of an arbitrary point on boundary Γ_i, and $s = \|\mathbf{s}_2 - \mathbf{s}_1\|_2$ is the distance between two points on contours. The advantage of using this approach is its further application in the calculation of view factors between complex geometries in arbitrary configurations.

3.6 Parallel Circular Surfaces

We denote the circular surface centre and radius as $\mathbf{c}_i, r_i$, respectively (see Fig. 3.2). We set the origin of the coordinate system $\vec{\mathbf{i}}_1\vec{\mathbf{j}}_1\vec{\mathbf{k}}_1$ at $\mathbf{c}_1$, with the unit base vector $\vec{\mathbf{k}}_1$ along the surface normal $\vec{\mathbf{n}}_1$, and the unit base vector $\vec{\mathbf{i}}_1$ perpendicular to the ground. We set $\vec{\mathbf{i}}_2\vec{\mathbf{j}}_2\vec{\mathbf{k}}_2$ as the translation of $\vec{\mathbf{i}}_1\vec{\mathbf{j}}_1\vec{\mathbf{k}}_1$, with the origin located at $\mathbf{c}_2$. We set ω_1 as the angle between $\tilde{\mathbf{i}}_1$ and $\mathbf{s}_1$, and ω_2 as the angle between $\vec{\mathbf{i}}_2$ and $\mathbf{s}_2$. We denote $\mathbf{c}_1 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\mathsf{T}$ and $\mathbf{c}_2 = \begin{bmatrix} p_1 & p_2 & p_3 \end{bmatrix}^\mathsf{T}$ under $\vec{\mathbf{i}}_1\vec{\mathbf{j}}_1\vec{\mathbf{k}}_1$

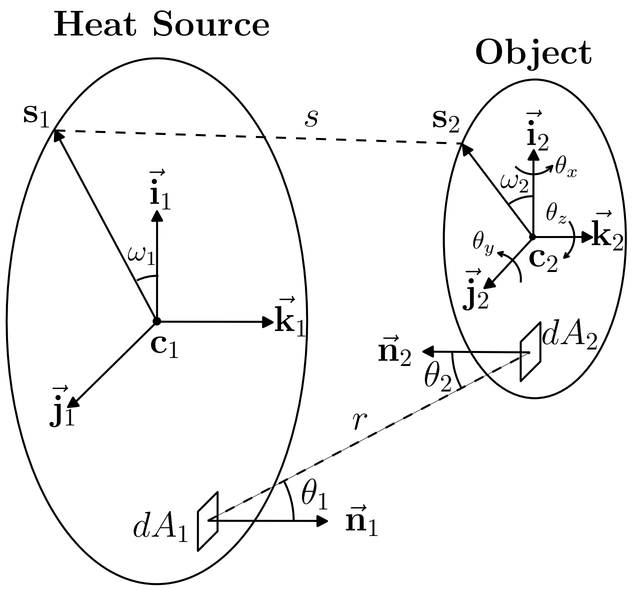


Figure 3.2: Notations for calculating the view factor between two elementary surfaces.

coordinate system. Parametric position vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ are:

$$\mathbf{s}_1 = \begin{bmatrix} r_1 \cos\omega_1 & r_1 \sin\omega_1 & 0 \end{bmatrix}^T \tag{3.13}$$

$$\mathbf{s}_2 = \begin{bmatrix} r_2 \cos\omega_2 + p_1 & r_2 \sin\omega_2 + p_2 & p_3 \end{bmatrix}^T \tag{3.14}$$

for $0 \leq \omega_1 \leq 2\pi$ and $0 \leq \omega_2 \leq 2\pi$. According to (3.13), (3.14), we replace $\mathrm{d}\mathbf{s}_1\,\mathrm{d}\mathbf{s}_2$ in (3.12) by $\mathrm{d}\omega_1\,\mathrm{d}\omega_2$:

$$\begin{aligned}
\mathrm{d}\mathbf{s}_1 &= \begin{bmatrix} -r_1 \sin\omega_1\,\mathrm{d}\omega_1 & r_1 \cos\omega_1\,\mathrm{d}\omega_1 & 0 \end{bmatrix}^T \\
\mathrm{d}\mathbf{s}_2 &= \begin{bmatrix} -r_2 \sin\omega_2\,\mathrm{d}\omega_2 & r_2 \cos\omega_2\,\mathrm{d}\omega_2 & 0 \end{bmatrix}^T \\
\mathrm{d}\mathbf{s}_1\,\mathrm{d}\mathbf{s}_2 &= r_1 r_2 \cos(\omega_1 - \omega_2)\,\mathrm{d}\omega_1\,\mathrm{d}\omega_2
\end{aligned} \tag{3.15}$$

Recall $s = \|\mathbf{s}_2 - \mathbf{s}_1\|_2$, we derive the expression of s as:

$$s^2 = (r_2 \cos\omega_2 + p_1 - r_1 \cos\omega_1)^2 + \\ (r_2 \sin\omega_2 + p_2 - r_1 \sin\omega_1)^2 + {p_3}^2 \quad (3.16)$$

Then, we rewrite (3.16) to derive a function of s as follows:

$$s = s(p_1, p_2, p_3, \omega_1, \omega_2) = ({p_1}^2 + {p_2}^2 + {p_3}^2 + \\ 2p_1(r_2 \cos\omega_2 - r_1 \cos\omega_1) + 2p_2(r_2 \sin\omega_2 - r_1 \sin\omega_1) \\ + {r_1}^2 + {r_2}^2 - 2r_1 r_2 \cos(\omega_2 - \omega_1))^{\frac{1}{2}} \quad (3.17)$$

By substituting (3.15) and (3.17) into (3.12), the view factor F_{21} can be calculated with the expression:

$$F_{21} = \frac{1}{2\pi \mathrm{A}_2} \int_0^{2\pi} \int_{2\pi}^{0} r_1 r_2 \cos(\omega_1 - \omega_2) \\ \ln s(p_1, p_2, p_3, \omega_1, \omega_2)\, \mathrm{d}\omega_2\, \mathrm{d}\omega_1 \quad (3.18)$$

Notice in (3.18), for a time instant, we assume the instantaneous end-effector position $\mathbf{x} = \begin{bmatrix} p_1 & p_2 & p_3 \end{bmatrix}^T$ is known exactly and A_2, r_1, r_2 are known constants.

As the object is rigidly grasped by the robot, we consider the instantaneous end-effector position to be located at the center of the object as $\mathbf{x} \equiv \mathbf{c}_2 = \begin{bmatrix} p_1 & p_2 & p_{3,} \end{bmatrix}^\mathsf{T}$. By substituting (3.18) to (1), we can finally determine the system's thermal-geometric relation:

$$v = f(\mathbf{x}, T_2) = \lambda_1 F_{21} - \lambda_2 {T_2}^4 + \lambda_3 \tag{3.19}$$

where the function $f : \mathbb{R}^3 \times \mathbb{R} \mapsto \mathbb{R}$ is the same as mentioned in (3.2). Take the time derivative of (3.19), we obtain the following dynamic system:

$$\dot{v} = \mathbf{l}\mathbf{u} - 4\lambda_2 {T_2}^3 v \tag{3.20}$$

where $\mathbf{l} = \lambda_1 \begin{bmatrix} \frac{\partial F_{21}}{\partial p_1} & \frac{\partial F_{21}}{\partial p_2} & \frac{\partial F_{21}}{\partial p_3} \end{bmatrix}$ is the system's interaction (Jacobian) vector, and $\mathbf{u} = \dot{\mathbf{x}}$ denotes the robot's motion input. This equation is used for designing control laws by taking the psudo-inverse of $\mathbf{l}$, and for compensating drift factors. By using the Leibniz integral rule [61], the interaction matrix can be derived as:

$$\mathbf{l}^\mathsf{T} = \begin{bmatrix} \displaystyle\int_0^{2\pi} \int_{2\pi}^{0} h(p_1 + r_2 \cos \omega_2 - r_1 \cos \omega_1)\, \mathrm{d}\omega_2\, \mathrm{d}\omega_1 \\ \displaystyle\int_0^{2\pi} \int_{2\pi}^{0} h(p_2 + r_2 \sin \omega_2 - r_1 \sin \omega_1)\, \mathrm{d}\omega_2\, \mathrm{d}\omega_1 \\ \displaystyle\int_0^{2\pi} \int_{2\pi}^{0} h p_3\, \mathrm{d}\omega_2\, \mathrm{d}\omega_1 \end{bmatrix} \tag{3.21}$$

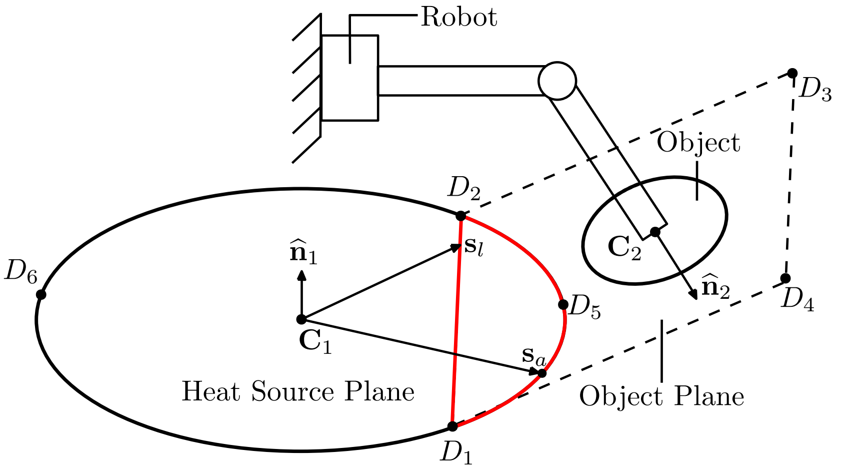


Figure 3.3: Conceptual representation of a self-obstruction case.

with h defined as:

$$h = \lambda_1 \frac{r_1 r_2 \cos(\omega_1 - \omega_2)}{2\pi A_2 s^2} \tag{3.22}$$

Since it is hard to derive the analytical solution for the above double integral, we use the numerical integral tool provided by *SciPy* [62] to online approximate **l**. See Appendix 3.35 for the the mathematical explanation.

3.7 Circular Surfaces in Arbitrary Configurations

In this subsection, we extend the previous parallel surface condition to a 6 DOF scenario. We denote the rotation about $\vec{\mathbf{i}}_2, \vec{\mathbf{j}}_2, \vec{\mathbf{k}}_2$ by $\theta_x, \theta_y, \theta_z$, and the rotation and translation matrices as $\mathrm{Rot}(\vec{\mathbf{i}}_2, \theta_x)$,

---

$\mathrm{Rot}(\vec{\mathbf{j}}_2, \theta_y), \mathrm{Rot}(\vec{\mathbf{k}}_2, \theta_z), \mathrm{Trans}(p_1, p_2, p_3) \in \mathbb{R}^{3\times 3}$. For simplicity, we use the abbreviation of cosine and sine as $\sin\omega = S_\omega, \cos\omega = C_\omega$. Most of the formulation is similar to the previous subsections, except for the parameterization of the heat source and object contours that are more complicated.

We first discuss the parameterization of the heat source. When the heat source plane and the object plane are not parallel, self-obstruction may occur and change the region of interest of the heat source (see Fig. 3.3). Concretely, in certain configurations, radiation from a part of the heat source can only incident to the backside of the object and will not contribute to the heat inflow to the object. We denote the object plane as $D_1D_2D_3D_4$, where D_1, D_2 are the intersubsections between the heat source plane and the object plane. The heat source is divided into two parts, the red part $\overparen{D_1D_5D_2}D_2D_1$ and the black part $\overparen{D_2D_6D_1}D_1D_2$. The black part of the heat source only "sees" the backside of the object and should be omitted from the calculation of F_{21}. To this end, we only consider the effect of the red part and rewrite the view factor formula (3.12) as:

$$F_{21} = \frac{1}{2\pi A_2}\left(\oint_{\Gamma_a}\oint_{\Gamma_2} \ln s_a \, d\mathbf{s}_2 \, d\mathbf{s}_a + \oint_{\Gamma_l}\oint_{\Gamma_2} \ln s_l \, d\mathbf{s}_2 \, d\mathbf{s}_l\right) \qquad (3.23)$$

where Γ_a, Γ_l denote the arc $\widehat{D_1D_5D_2}$, line D_1D_2, and other variables carries the same meaning as defined in subsection 3.5. Now we discuss how to derive $\mathbf{s}_a$, $\mathbf{s}_l$. The normal vector of the object plane $\mathbf{n}_2$ is calculated as:

$$\begin{aligned} \mathbf{n}_2 = & \operatorname{Rot}(z, \theta_z) \cdot \operatorname{Rot}(y, \theta_y) \cdot \operatorname{Rot}(x, \theta_x) \\ & \cdot \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} = \begin{bmatrix} -S_{\theta_x}S_{\theta_z} - C_{\theta_x}C_{\theta_z}S_{\theta_y} \\ C_{\theta_z}S_{\theta_x} - C_{\theta_x}S_{\theta_y}S_{\theta_z} \\ -C_{\theta_x}S_{\theta_y} \end{bmatrix} \end{aligned} \tag{3.24}$$

For simplicity, we denote $\mathbf{n}_2 = \begin{bmatrix} n_2^1 & n_2^2 & n_2^3 \end{bmatrix}^T$. The equation of the plane $D_1D_2D_3D_4$ is:

$$n_1^1 x + n_1^2 y + n_1^3 z - n_1^1 p_1 - n_1^2 p_2 - n_1^3 p_3 = 0 \tag{3.25}$$

To find the intersubsection points, we substitute $x = r_1 C_\varphi$, $y = r_1 S_\varphi$, $z = 0$ where φ is the unknown variable in (3.25). Self-obstruction only occurs when there exist two solutions φ_1, φ_2. Then, the parametric vectors can be easily derived as:

$$\mathbf{s}_a = \begin{bmatrix} r_1 C_\varphi \\ r_1 S_\varphi \\ 0 \end{bmatrix}, \mathbf{s}_l = \begin{bmatrix} x_l \\ k_l(x_l - r_1 C_{\varphi_2}) + r_1 S_{\varphi_2} \\ 0 \end{bmatrix} \tag{3.26}$$

for $\varphi \in [\varphi_1, \varphi_2]$, $x_l \in [r_1 C_{\varphi_2}, r_1 C_{\varphi_1}]$, and k_l is the slope of the line $D_1 D_2$:

$$k_l = \frac{r_1 S_{\varphi_2} - r_1 S_{\varphi_1}}{r_1 C_{\varphi_2} - r_1 C_{\varphi_1}} \tag{3.27}$$

To derive F_{21}, we first obtain $\mathrm{d}\mathbf{s}_a, \mathrm{d}\mathbf{s}_a$ as:

$$\mathrm{d}\mathbf{s}_a = \begin{bmatrix} -r_1 S_\varphi & r_1 C_\varphi & 0 \end{bmatrix}^T \mathrm{d}\varphi, \mathrm{d}\mathbf{s}_l = \begin{bmatrix} 1 & k_l & 0 \end{bmatrix}^T \mathrm{d}\varphi \tag{3.28}$$

Now we derive the parametric vector on the object contour after translation end rotation as:

$$\begin{aligned} \mathbf{s}_2 &= \mathrm{Rot}(z, \theta_z) \cdot \mathrm{Rot}(y, \theta_y) \cdot \mathrm{Rot}(x, \theta_x) \cdot \mathrm{Trans}(p_1, p_2, p_3) \\ &\cdot \begin{bmatrix} r_2 \cos \omega_2 \\ r_2 \sin \omega_2 \\ 0 \end{bmatrix} = \begin{bmatrix} a_1 r_2 C_{\omega_2} + a_2 r_2 S_{\omega_2} + p_1 \\ a_3 r_2 C_{\omega_2} + a_4 r_2 S_{\omega_2} + p_2 \\ a_5 r_2 C_{\omega_2} + a_6 r_2 S_{\omega_2} + p_3 \end{bmatrix} \end{aligned} \tag{3.29}$$

where the parameters determined by rotation angles are defined as follows:

$$\begin{aligned} a_1 = C_{\theta_y} C_{\theta_z}, a_2 = C_{\theta_z} S_{\theta_x} S_{\theta_y} - C_{\theta_x} S_{\theta_z}, a_3 = C_{\theta_y} S_{\theta_z}, \\ a_4 = C_{\theta_x} C_{\theta_z} + S_{\theta_x} S_{\theta_y} S_{\theta_z}, a_5 = -S_{\theta_y}, a_6 = C_{\theta_y} S_{\theta_x} \end{aligned} \tag{3.30}$$

Similarly, $\mathrm{d}\mathbf{s}_2$ is:

$$\mathrm{d}\mathbf{s}_2 = \begin{bmatrix} a_2 r_2 C_{\omega_2} - a_1 r_2 S_{\omega_2} \\ a_4 r_2 C_{\omega_2} - a_3 r_2 S_{\omega_2} \\ a_6 r_2 C_{\omega_2} - a_5 r_2 S_{\omega_2} \end{bmatrix} \mathrm{d}\omega_2 \tag{3.31}$$

Finally, distance between two points on $\mathbf{s}_2, \mathbf{s}_a$ can be derived as $s_a = \|\mathbf{s}_2 - \mathbf{s}_a\|_2$, $s_l = \|\mathbf{s}_2 - \mathbf{s}_l\|_2$. We substitute (3.26), (3.28), (3.29), and (3.31) to (3.23) and obtain the view factor formula for self-obstruction case as:

$$\begin{aligned} F_{21} = \frac{1}{2\pi \mathrm{A}_2} \int_{\varphi_1}^{\varphi_2} \int_{2\pi}^{0} \ln s_a \, \mathrm{d}\omega_1 \, \mathrm{d}\varphi \\ + \frac{1}{2\pi \mathrm{A}_2} \int_{r_1 C_{\varphi_2}}^{r_1 C_{\varphi_1}} \int_{2\pi}^{0} \ln s_l \, \mathrm{d}\omega_1 \, \mathrm{d}x_l \end{aligned} \tag{3.32}$$

Similar to the previous subsection, we obtain the following dynamic system:

$$\dot{v} = \mathbf{lu} - 4\lambda_2 {T_2}^3 v \tag{3.33}$$

Where the interaction vector $\mathbf{l}$ is defined as

$$\mathbf{l} = \lambda_1 \begin{bmatrix} \frac{\partial F_{21}}{\partial p_1} & \frac{\partial F_{21}}{\partial p_2} & \frac{\partial F_{21}}{\partial p_3} & \frac{\partial F_{21}}{\partial \theta_x} & \frac{\partial F_{21}}{\partial \theta_y} & \frac{\partial F_{21}}{\partial \theta_z} \end{bmatrix} \tag{3.34}$$

Except for the numerical integration approximation technique discussed in Appendix section, we also apply the numerical partial differentiation technique to approximate $\mathbf{l}$ in real time as:

$$\mathbf{l}^{\mathsf{T}} = \lambda_1 \begin{bmatrix} \dfrac{F_{21}\left(p_1 + \mathrm{d}p_1, p_2, \ldots, \omega_2\right) - F_{21}}{\mathrm{d}p_1} \\ \vdots \\ \dfrac{F_{21}\left(\ldots, \theta_z + \mathrm{d}\theta_z, \omega_1, \omega_2\right) - F_{21}}{\mathrm{d}\theta_z} \end{bmatrix} \tag{3.35}$$

3.8 Multiple Objects

In the previous subsection, we presented the general approach to derive $\mathbf{l} \in \mathbb{R}^6$ between one object and the heat source. Following the same approach, the interaction matrix $\mathbf{L} \in \mathbb{R}^{N \times 6}$ for N objects can be formulated easily. Note that we only consider the case where the heat radiation between N objects can be ignored; otherwise, a different form of thermal-geometrical relation (3.19) needs to be derived. We denote the interaction vector for each object by $\mathbf{l}_n$ for

$n = 1, 2, ..., N$, and the overall interaction matrix can be written as:

$$\mathbf{L} = \begin{bmatrix} \mathbf{l}_1^\intercal & \mathbf{l}_2^\intercal & \cdots & \mathbf{l}_N^\intercal \end{bmatrix}^\intercal \in \mathbb{R}^{N\times 6} \tag{3.36}$$

We constrain the number of objects to be less than the DOF number of the robot motion input. We denote the temperature, the temperature rate and the target temperature of each object by T_2^n, v^n, T^{*n} for $n = 1, 2, ..., N$. We then construct the following vector structures:

$$\tau = \begin{bmatrix} T_2^1 & T_2^2 & \cdots & T_2^N \end{bmatrix}^\intercal, \tag{3.37}$$

$$\mathbf{v} = \begin{bmatrix} v^1 & v^2 & \cdots & v^N \end{bmatrix}^\intercal, \tag{3.38}$$

$$\tau^* = \begin{bmatrix} T^{*1} & T^{*2} & \cdots & T^{*N} \end{bmatrix}^\intercal \in \mathbb{R}^N \tag{3.39}$$

We denote the thermophysical properties constant λ_2 defined in (3.9) of each object by λ_2^n, for $n = 1, 2, ..., N$. We construct the following structures:

$$\lambda = \operatorname{diag}(\lambda_2^1, \lambda_2^1, \cdots, \lambda_2^N) \in \mathbb{R}^{N\times N} \tag{3.40}$$

$$\tau' = \begin{bmatrix} (T_2^1)^3 & (T_2^1)^3 & \cdots & (T_2^N)^3 \end{bmatrix}^\intercal \in \mathbb{R}^N \tag{3.41}$$

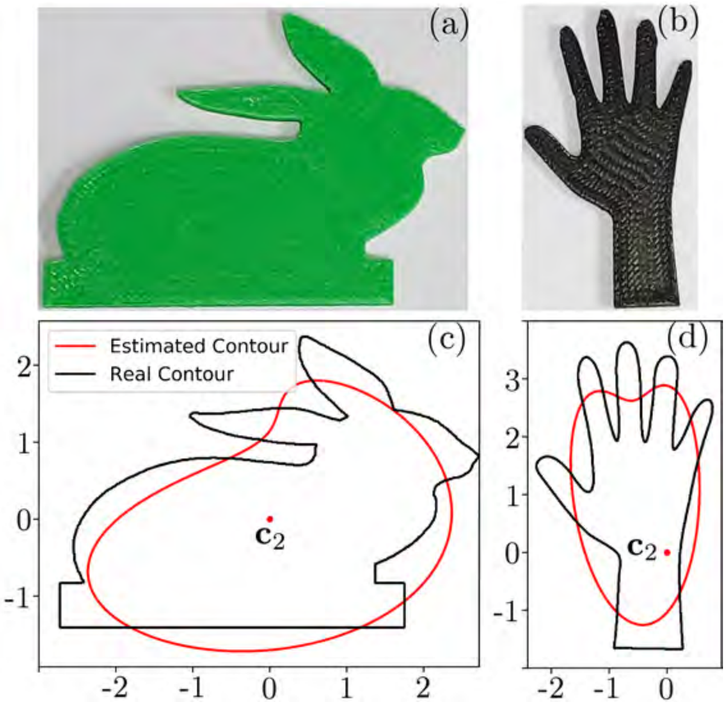


Figure 3.4: Contours of the two object are approximated by the truncated Fourier series with 5 harmonics.

Accordingly, we can extend (3.33) to multiple objects case:

$$\dot{\mathbf{v}} = \mathbf{L}\mathbf{u} - 4\tau'^{\mathsf{T}}\lambda\mathbf{v} \tag{3.42}$$

3.9 Arbitrary Surfaces at Arbitrary Configurations

In this subsection, we demonstrate how to develop radiation-based GTM model for arbitrary surfaces in arbitrary configurations. The core concept is to use the truncated Fourier series [63] for approximating the parametric position vectors $\mathbf{s}_1, \mathbf{s}_2$. After obtaining the approximation of $\mathbf{s}_1, \mathbf{s}_2$, the remaining derivations of F_{21} are very similar to the previous subsections. Thus, we only demonstrate how to derive the parametric equation of an irregular-shaped contour.

We take the case where the objects are a bunny-shaped and a hand-shaped artifact as an example, see Fig. 3.4 (a), (b). The cases where both the object and the heat source surfaces are irregular can be easily extended. The artifacts are 3D printed according to *Solidworks* models which are generated referring to the 2D contours shown in Fig. 3.4 (c), (d). The parametric equations of irregular-shaped contours are first derived in the complex number form, then transferred back to the real number form. We select m points on the contour in a counterclockwise order as $c_k = u_k + iv_k$ for $k = 0, 1, 2, \ldots, m-1$. Assume there is a function with parameter $x \in [0, 1)$, when $x = k/m$ for $k = 0, 1, 2, \ldots, m-1$, corresponding function values are $c(x) = c_{m \cdot x}$. We can then approximate $c(x)$ with the following truncated Fourier series:

$$c(x) = \sum_{n=-N}^{N} d_n e^{-2\pi inx} \tag{3.43}$$

where d_n are approximated Fourier coefficients calculated following the standard methods in [64], and $2N+1$ harmonics are used to approximate the contour (We set N=2 in this study). We denote $a_n = \mathrm{Re}(d_n)$, $b_n = \mathrm{Im}(d_n)$, and rewrite (3.43) as:

$$\begin{aligned} c(x) = \sum_{n=-N}^{N} a_n \cos 2\pi nx - b_n \sin 2\pi nx \\ + i \sum_{n=-N}^{N} a_n \sin 2\pi nx + b_n \cos 2\pi nx \end{aligned} \tag{3.44}$$

According to the above equation, we transform it back to real number form and derive the parametric position vector of $\mathbf{s}_2$ as

$$\mathbf{s}_2 = \begin{bmatrix} p_1 + \sum_{n=-N}^{N} a_n \sin 2\pi nx + b_n \cos 2\pi nx \\ p_2 + \sum_{n=-N}^{N} a_n \cos 2\pi nx - b_n \sin 2\pi nx \\ p_3 \end{bmatrix} \tag{3.45}$$

for $x \in [0, 1)$. Similarly, we can derive $\mathrm{d}\mathbf{s}_2$ as:

$$\mathrm{d}\mathbf{s}_2 = \begin{bmatrix} \sum_{n=-N}^{N} 2\pi n(a_n \cos 2\pi nx - b_n \sin 2\pi nx) \\ \sum_{n=-N}^{N} -2\pi n(a_n \sin 2\pi nx + b_n \cos 2\pi nx) \\ 0 \end{bmatrix} \mathrm{d}x \tag{3.46}$$

Following the similar procedures presented in the previous subsections, the heat transfer model and the corresponding Jacobian matrix can be derived easily.

Chapter 4

"Fire-to-Hand" Thermal Servoing: Controls and Experiments

Problem statement. Given a desired constant object target temperature vector $\tau^* \in \mathbb{R}^n$, design a velocity controller input $\mathbf{u}$ which asymptotically minimizes the temperature error $\Delta\tau = \tau - \tau^*$.

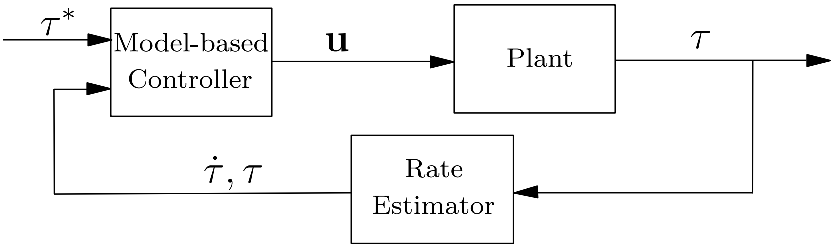


Figure 4.1: Schematic representation of the model-based controller.

4.1 Model-Based Controller

In this section, we discuss the model-based controller for controlling N objects to achieve the target temperatures. We design the following feedback velocity input:

$$\mathbf{u} = \mathbf{L}^{+}(-D\mathbf{v} - K\Delta\tau + 4{\tau_2'}^{\intercal}\lambda_2\mathbf{v}) \tag{4.1}$$

where $\mathbf{L}^{+} = \mathbf{L}^{\intercal}(\mathbf{L}\mathbf{L}^{\intercal})^{-1}$ represents the Moore-Penrose pseudoinverse. The state variables $\tau_2, \mathbf{v}$ are obtained from real-time sensor feedback. By applying the above input to the nonlinear dynamic system (3.20), it transforms to classical mass-spring-damper systems as:

$$\dot{\mathbf{v}} = -D\mathbf{v} - K\Delta\tau \tag{4.2}$$

Where D resembles the damping coefficient and K resembles the spring stiffness.

Consider at time instant t, assume the discrete integral of $\mathbf{L}$ (see Appendix for more details), and the prior values of combined thermophysical parameters well approximate their true values. The control input (4.1) enforces an energetically passive closed-loop system which asymptotically minimizes $\Delta\tau$.

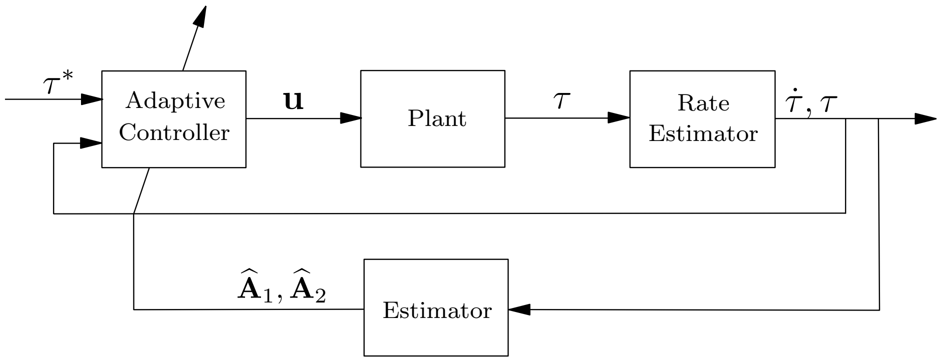


Figure 4.2: Schematic representation of the adaptive controller.

Proof. Consider the quadratic Lyapunov function

$$H = \frac{1}{2}\mathbf{v}^T\mathbf{v} + \frac{1}{2}K\Delta\tau^T\Delta\tau \tag{4.3}$$

whose time derivative is

$$\dot{H} = -D\mathbf{v}^T\mathbf{v} \leq 0 \tag{4.4}$$

By applying Krasovskii-LaSalle principle [65], the convergence can be proved.

4.2 Adaptive Controller

In the model-based controller design, we assume the thermophysical properties are known exactly. However, due to the differences in material and surface conditions, the true values of those properties are hard to be determined. We introduce two unknown parameters $a_1 = 1/\lambda_1$, $a_2 = \lambda_2/\lambda_1$ whose estimated values are $\widehat{a}_1, \widehat{a}_2$. To this end, we denote the difference between the estimated values and true values by $\widetilde{a}_1 = \widehat{a}_1 - a_1, \widetilde{a}_2 = \widehat{a}_2 - a_2$. For multiple objects, we use superscripts to distinguish between different objects such as $a_1^n, a_2^n, \widehat{a}_1^n, \widehat{a}_2^n, \widetilde{a}_1^n, \widetilde{a}_2^n$ for $n = 1, 2, ..., N$. We then construct the following matrix structures:

$$\begin{aligned}
\mathbf{A}_1 &= \mathrm{diag}(a_1^1, \cdots, a_1^N), \mathbf{A}_2 = \mathrm{diag}(a_2^1, \cdots, a_2^N), \\
\widehat{\mathbf{A}}_1 &= \mathrm{diag}(\widehat{a}_1^1, \cdots, \widehat{a}_1^N), \widehat{\mathbf{A}}_2 = \mathrm{diag}(\widehat{a}_2^2, \cdots, \widehat{a}_2^N), \\
\widetilde{\mathbf{A}}_1 &= \mathrm{diag}(\widetilde{a}_1^1, \cdots, \widetilde{a}_1^N), \widetilde{\mathbf{A}}_2 = \mathrm{diag}(\widetilde{a}_2^2, \cdots, \widetilde{a}_2^N)
\end{aligned}$$

Divide the dynamic equation (3.20) by λ_1 and combine it to vector form as:

$$\mathbf{A}_1 \dot{\mathbf{v}} + 4\tau'^{\mathsf{T}} \mathbf{A}_2 \mathbf{v} = \mathbf{J}\mathbf{u} \tag{4.5}$$

Where $\mathbf{J}$ is defined as:

$$\mathbf{J} = \begin{bmatrix} \mathbf{l}_1/\lambda_1^1 & \mathbf{l}_2/\lambda_1^2 & \cdots & \mathbf{l}_N/\lambda_1^N \end{bmatrix} \in \mathbb{R}^{N \times 6} \tag{4.6}$$

Where a general definition of interaction vectors $\mathbf{l}_n$ for $n = 1, 2, ..., N$ is provided in (3.34). Note that the unknown combined thermophysical properties λ_1^n are divided from $\mathbf{l}_n$, $\mathbf{J}$ is purely dependent on the values of view factors F_{21} of objects. We then introduce the combined error vector $\zeta = \begin{bmatrix} \zeta^1 & \dots & \zeta^N \end{bmatrix}^\mathsf{T}$ as

$$\zeta = \Delta\dot{\tau} + \mu_0 \Delta\tau = \mathbf{v} + \mu_0 \Delta\tau \tag{4.7}$$

Where μ_0 is a positive real number. Consider the feedback velocity input

$$\mathbf{u} = \mathbf{J}^+(-\mu_0 \widehat{\mathbf{A}}_1 \mathbf{v} - K\zeta + 4\tau'^\mathsf{T} \widehat{\mathbf{A}}_2 \mathbf{v}) \tag{4.8}$$

where K is a positive real number. Substitute (4.7) to (4.5) and add $\mu_0 \mathbf{A}_1 \mathbf{v}$ to both side, we obtain

$$\mathbf{A}_1 \dot{\zeta} + K\zeta = -\mu_0 \widetilde{\mathbf{A}}_1 \mathbf{v} + 4\tau'^\mathsf{T} \widetilde{\mathbf{A}}_2 \mathbf{v} \tag{4.9}$$

For slow and smooth motion of the manipulator, the parameter update rule

$$\dot{\hat{\mathbf{a}}}_1 = \gamma_1 \mu_0 \begin{bmatrix} (v^1 \zeta^1 & \dots & v^N \zeta^N) \end{bmatrix}^\mathsf{T} \tag{4.10}$$

$$\dot{\hat{\mathbf{a}}}_2 = -4\gamma_2 \begin{bmatrix} v^1 \zeta^1 (T_2^1)^3 & \dots & v^N \zeta^N (T_2^N)^3 \end{bmatrix}^\mathsf{T} \tag{4.11}$$

where γ_1, γ_2 are positive real numbers representing the learning rates, which ensures the following conditions:

1. The asymptotic minimization of the error $\Delta\mathbf{T}$.

Proof. We construct following vectors:

$$\widehat{\mathbf{a}}_1 = \begin{bmatrix} a_1^1 & a_1^2 & \cdots & a_1^N \end{bmatrix}^\mathsf{T}, \widehat{\mathbf{a}}_2 = \begin{bmatrix} a_2^1 & a_2^2 & \cdots & a_2^N \end{bmatrix}^\mathsf{T}, \tag{4.12}$$

Notice that $\dot{\widetilde{\mathbf{a}}}_1 = \dot{\widehat{\mathbf{a}}}_1, \dot{\widetilde{\mathbf{a}}}_2 = \dot{\widehat{\mathbf{a}}}_2$. Recall the definition of λ_1 in (3.9), we know every element in $\mathbf{a}_1$ is nonnegative. Consider following quadratic Lyapunov function:

$$H = \frac{1}{2}\zeta^\mathsf{T}\mathbf{A}_1\zeta + \frac{1}{2\gamma_1}\widetilde{\mathbf{a}}_1^\mathsf{T}\widetilde{\mathbf{a}}_1 + \frac{1}{2\gamma_2}\widetilde{\mathbf{a}}_2^\mathsf{T}\widetilde{\mathbf{a}}_2 \tag{4.13}$$

We take the time derivative of (4.13) and obtain

$$\dot{H} = \zeta^\mathsf{T}\mathbf{A}_1\dot{\zeta} + \frac{1}{\gamma_1}\dot{\widehat{\mathbf{a}}}_1^\mathsf{T}\widetilde{\mathbf{a}}_1 + \frac{1}{\gamma_2}\dot{\widehat{\mathbf{a}}}_2^\mathsf{T}\widetilde{\mathbf{a}}_2 \tag{4.14}$$

Substituting (4.9) into (4.14) and rearrange the equation, we have

$$\dot{H} = -K\zeta^\mathsf{T}\zeta - \mu_0\zeta^\mathsf{T}\widetilde{\mathbf{A}}_1\mathbf{v} + \frac{1}{\gamma_1}\dot{\widehat{\mathbf{a}}}_1\widetilde{\mathbf{a}}_1 + 4\zeta^\mathsf{T}\tau'\widetilde{\mathbf{A}}_2\mathbf{v} + \frac{1}{\gamma_2}\dot{\widehat{\mathbf{a}}}_2\widetilde{\mathbf{a}}_2 \tag{4.15}$$

Substitue update rules (4.10) into (4.15), we have

$$\dot{H} = -K\zeta^\mathsf{T}\zeta \tag{4.16}$$

By applying Krasovskii-LaSalle principle [65], convergence can be proved.

4.3 Target Feasibility

In the previous sections, we proved $|\Delta\tau|$ can be asymptotically minimized by the two controllers. However, it is not guaranteed that we can always enforce $|\Delta\tau|$ to decrease to zero. In some cases, $|\Delta\tau|$ will only asymptotically approach a large local minimum. In other words, the object temperature can not be controlled to reach the target temperature sometimes. Those failure cases are caused by the choice of unfeasible target temperatures: Intuitively, if the target temperatures are set to be too high or too low, it will be physically unachievable at first; in addition, for objects fixed to the same end-effector, their target temperatures should not be too different from each other (e.g. one is set to be 30 °C while the other is set to be 70 °C is unfeasible). In this section, we analytically discuss two necessary but insufficient conditions which ensure the feasibility of the target temperature.

We consider the case where two objects, object 1 and object 2 are fixed to the end effector (n object cases can be easily extended). We first derive boundary values of the target temperatures. Recall the thermal-geometrical equation (3.19) and rewrite it as follows:

$$v = -\lambda_2 {T_2}^4 + \lambda_1 F_{21}(\mathbf{x}_o) + \lambda_3 \tag{4.17}$$

Where $\mathbf{x}_o$ denotes the geometrical configuration of the object. We assume there exists $T_2 = T_{v0}$ that makes the temperature rate v equals to zero. Note that the temperature value is always nonnegative, T_{v0} can be solved as:

$$T_{v0}(F_{21}) = ((\lambda_1 F_{21} + \lambda_3)/\lambda_2)^{\frac{1}{4}} \tag{4.18}$$

As $\lambda_1, \lambda_2, \lambda_3$ are all positive and $F_{21} \in [0,1]$, T_{v0} always exists. Note that for $T_2 > T_{v0}$, $v < 0$; for $T_2 < T_{v0}$, $v > 0$. Thus, the temperature of an object at a specific configuration will always asymptotically approach T_{v0}. In other words, T_{v0} is the steady state temperature of the object at a specific configuration $\mathbf{x}_o$. Note that T_{v0} is a function of F_{21} with range $F_{21} \in [0,1]$, and it can be easily shown that $\mathrm{d}T_{v0}/\,\mathrm{d}F_{21}$ is always positive. Thus, when $F_{21} = 0$, we can calculate the minimum value of T_{v0}:

$$\min(T_{v0}) = (\lambda_3/\lambda_2)^{1/4} = (\alpha_2 T_3^4/\varepsilon_2)^{\frac{1}{4}} \tag{4.19}$$

According to Kirchhoff's law of thermal radiation[53], at thermodynamic equilibrium, $\alpha_2 = \varepsilon_2$. Thus, the minimum of T_{v0} is simply:

$$\min(T_{v0}) = T_3 \tag{4.20}$$

When $F_{21}(\mathbf{x}_o) \to 1$, the maximum value of T_{v0} approaches:

$$\max(T_{v0}) \to (\lambda_1 + \lambda_3)/\lambda_2 = (\alpha_2 \varepsilon_1 T_1^4 / \varepsilon_2)^{\frac{1}{4}} = \varepsilon_1^{\frac{1}{4}} T_1 \tag{4.21}$$

According to (4.20), (4.21), we derive the first condition as:

$$T^{1*}, T^{2*} \in [T_3, \varepsilon_1^{\frac{1}{4}} T_1) \tag{4.22}$$

Now we discuss the limitation of the difference between target temperatures $|\Delta T^*| = |T^{1*} - T^{2*}|$. We denote the geometrical configuration of object 1, object 2 as $\mathbf{x}_{o1} = \mathbf{x} + \Delta\mathbf{x}_1, \mathbf{x}_{o2} = \mathbf{x} + \Delta\mathbf{x}_2$ where $\mathbf{x}$ is the end-effector configuration, and $\Delta\mathbf{x}_1, \Delta\mathbf{x}_2$ are constant displacement vectors determined by the geometrical design of the end-effector. The corresponding view factor values of objects 1, 2 are denoted by $F_{21}(\mathbf{x}_{o1}), F_{21}(\mathbf{x}_{o2})$. We denote the robot workspace as $\mathbf{W}$. We denote the steady-state temperatures of objects 1, 2 by T_{v0}^1, T_{v0}^2. According to (4.18), $|\Delta T_{v0}| = |T_{v0}^1 - T_{v0}^2|$ can be expressed as a function:

$$|\Delta T_{v0}(\mathbf{x}_{o1},\mathbf{x}_{o2})| = |n_1(F_{21}(\mathbf{x}_{o1})) - n_2(F_{21}(\mathbf{x}_{o2}))| \tag{4.23}$$

with function $n_1(F_{21}), n_2(F_{21})$ defined as:

$$n_1(F_{21}) = (\frac{\lambda_1^1 F_{21} + \lambda_3^1}{\lambda_2^1})^{\frac{1}{4}}, n_2(F_{21}) = (\frac{\lambda_1^2 F_{21} + \lambda_3^2}{\lambda_2^2})^{\frac{1}{4}} \tag{4.24}$$

where $\lambda_1^1, \lambda_2^1, \lambda_3^1$ are thermophysical parameters of object 1, $\lambda_1^2, \lambda_2^2, \lambda_3^2$ are thermophysical parameters of object 2 (general definition in (3.9)). For the function $\Delta T_{v0}(\mathbf{x}+\Delta\mathbf{x}_1, \mathbf{x}+\Delta\mathbf{x}_2)$ where $\mathbf{x} \in \mathbf{W}$ and $\Delta\mathbf{x}_1, \mathbf{x}_2$ are constant vectors, as the function ΔT_{v0} is continuous, there must exist a minimum value $\min(\Delta T_{v0}) = \Delta T_{v0}(\mathbf{x}^{min})$ and a maximum value $\max(\Delta T_{v0}) = \Delta T_{v0}(\mathbf{x}^{max})$ that encompass all possible values of ΔT_{v0}, where $\mathbf{x}^{min}$ and $\mathbf{x}^{max}$ are the end-effector positions corresponding to the two extreme conditions. We can then derive the second condition for feasible target temperatures as:

$$\Delta T^* \in [\min(\Delta T_{v0}), \max(\Delta T_{v0})] \tag{4.25}$$

A numerical (geometrical) interpretation of $\mathbf{x}^{min}$ and $\mathbf{x}^{max}$ will be discussed in Section 4.11.

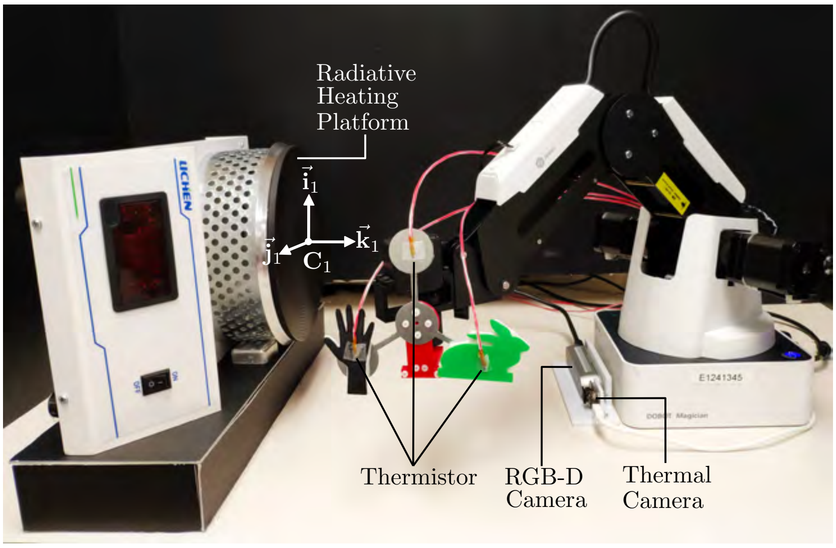


Figure 4.3: Experimental setup.

4.4 Experimental Setup

We conduct a series of experiments on a 4-DOF robot (3 translations and 1 rotation) to evaluate the proposed method. Fig. 4.3 shows the experimental robot, *Dobot Magician*, whose end-effector is replaced

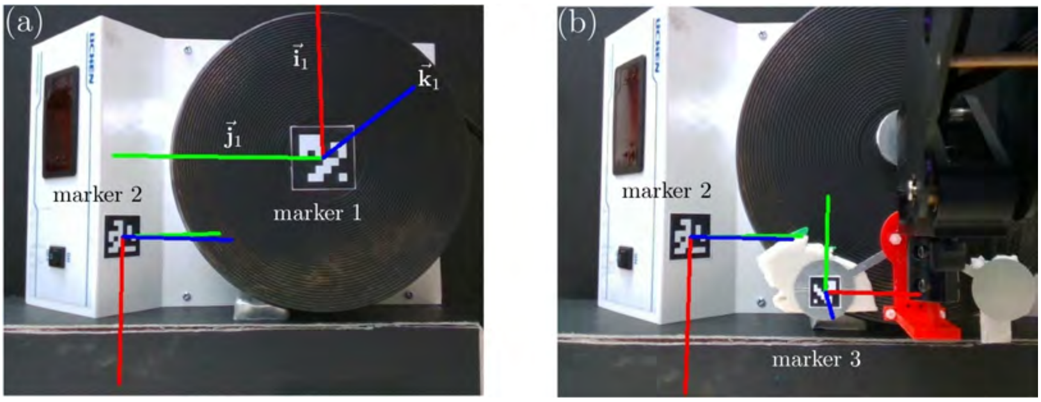


Figure 4.4: Auto-calibration using ArUco markers before experiments.

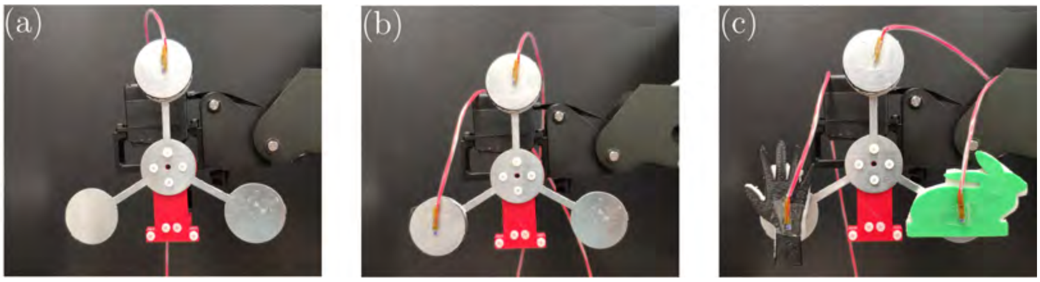


Figure 4.5: Different object configurations for the experiments.

by a red 3D printed connector connected to an aluminium holder for fixing objects. The triangular holder with a side length of 10 cm can hold up to 3 objects. We prepared three different kinds of objects for temperature control experiments: An aluminium circular sheet with 1.5 cm radius and 3 mm thickness; a green bunny-shaped sheet with 1 mm thickness which is 3D printed using the polylactic acid (PLA) material and with a 30% infill density; a black hand-shaped sheet with 1 mm thickness which is 3D printed using PLA material and with a 50% infill density. The dimensions of the two 3D printed objects are shown in Fig. 3.4 in centimeters. The three objects are attached to the holder through a layer of insulation foam to minimize the effect of conductive heat transfer. For the aluminium sheet, the values of its thermophysical properties can be approximated from standard tables[53]. The emissivity, absorptivity, specific heat, and density of aluminium objects are estimated as 0.04, 0.04, $903\,\mathrm{J}\cdot\mathrm{K}^{-1}\cdot$

kg^{-1}, and $2702\,\mathrm{kg/m^3}$.

For the two 3D printed objects, different infill densities, colors, and uncertain surface conditions make their thermophysical properties hard to be estimated. Thus, in model-based controller experiments, we only control the temperature of the aluminium sheet. While in adaptive controller experiments, we control the temperatures of three different objects. We use a heating platform with adjustable temperature output as the heat source. The heating surface is the black disk with a radius of 9.25 cm. The environment temperature is assumed to be constant as 23 °C.

At the beginning of the experiments, we use the RGB camera module of the RealSense 415 depth camera and ArUco markers to auto-calibrate the geometrical configuration of the heat source and the end-effector (see Fig. 4.4). Intrinsic parameters of the RGB camera are obtained by using Opencv [66] camera calibration functions. Marker 1 is attached to the heating plate at a specific location and orientation such that its center coincides with the $\mathbf{C}_1$ (the center of the heating plate), and its x, y, z axes (represented by red, blue, and green) align with $\vec{\mathbf{i}}_1, \vec{\mathbf{j}}_1, \vec{\mathbf{k}}_1$ axes depicted in Fig. 3.2. In other words, we make the marker 1 coordinate system the same as the heat source coordinate system which is used for calculating the view factor. Since marker 1 needs to be removed during the heating process, marker 2 is attached to the metal shell of the heating source. We denote the homogeneous coordinates of points under marker 1, marker 2 coordinate system by $\mathbf{m}_1, \mathbf{m}_2 \in \mathbb{R}^4$. As shown in Fig. 4.4 (a), a fixed the transformation between $\mathbf{m}_1, \mathbf{m}_2$ can be obtained as $\mathbf{m}_1 = (\mathbf{T}_1^c)^{-1}\mathbf{T}_2^c\mathbf{m}_2$ Where $\mathbf{T}_1^c, \mathbf{T}_2^c \in \mathbb{R}^{4\times 4}$ are transformation matrices between the marker 1, 2, and the camera coordinate systems. After this fixed transformation is determined, we detach marker 1 to prevent it from burning during the heating process. Marker 3 is attached to the back of the aluminum holder for determining the geometrical configurations of objects. We control the robot end-effector to move into the field of view of the RGB camera, see Fig. 4.4 (b). We denote the homogeneous coordinates of points under the new heat source, marker 2, and marker 3 coordinate systems by $\mathbf{m}_1^{'}, \mathbf{m}_2^{'}, \mathbf{m}_3 \in \mathbb{R}^4$. Transformation between $\mathbf{m}_2^{'}, \mathbf{m}_3$ is $\mathbf{m}_2^{'} = (\mathbf{T}_2^{c'})^{-1}\mathbf{T}_3^c\mathbf{m}_3$ Where $\mathbf{T}_2^{c'}, \mathbf{T}_3^c \in \mathbb{R}^{4\times 4}$ are transformation matrices between marker 2, 3, and camera coordinate systems. Finally, we can derive the transformation between $\mathbf{m}_1^{'}$ and $\mathbf{m}_3$ is :

$$\mathbf{m}_1^{'} = (\mathbf{T}_1^c)^{-1}\mathbf{T}_2^c(\mathbf{T}_2^{c'})^{-1}\mathbf{T}_3^c\mathbf{m}_3 \tag{4.26}$$

Similarly, the rotation angle between the object plane and the heat source plate plane can be calculated. Except for auto-calibration before the experiments, marker 2 is also used for online tracking of a moving heat source during thermal servoing processes. We report the performance of the integrated system in Section 4.9.

To obtain feedback temperatures from objects, we attach a PT100 platinum thermistor with 0.3 °C accuracy and 0.1 °C precision to each object. The raw data obtained by thermistors is processed by a current-temperature transformation module and sent to a Linux computer as the feedback data. The control output is calculated by the computer and conducted by the robot using the position-stepping mode.

4.5 Online Estimation of $\mathbf{T}_2$ and $\mathbf{v}$

Recall (4.1), (4.8), temperature rate $\mathbf{v}$ is also required for calculating appropriate control inputs. Due to the limitation of the sensor ac-

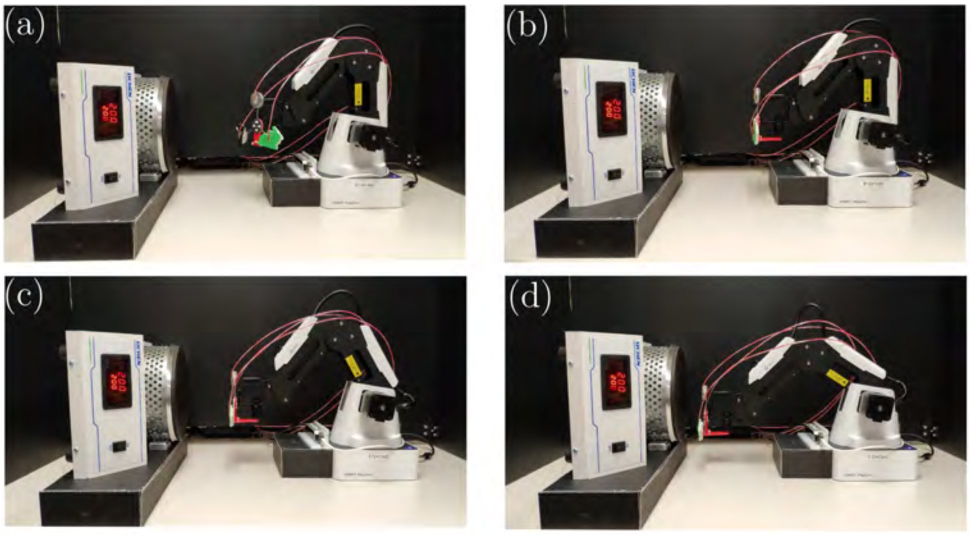


Figure 4.6: Snapshots of the experiment with the adaptive controller where the target temperatures is set to $\tau = \begin{bmatrix}40 & 45 & 50\end{bmatrix}^\mathsf{T}$ (measured in °C).

curacy, precision, and sampling rate, the raw feedback temperatures are discrete and relative noisy, which is undesirable for estimating the true values of $\mathbf{v}$ for calculating desirable control inputs. To this end, we implement polynomial fitting with sliding window method to estimate the values of τ and $\mathbf{v}$ from the raw feedback data. See Appendix section 7.2 for more details.

4.6 Experiments with the Model-Based Controller

We designed a series of experiments to evaluate the performance of the model-based controller. We first conduct experiments to control the temperature of one object. The aluminium circular sheet is attached to the end-effector through an adiabatic layer as shown in Fig. 4.5 (a).

In theory, we enforce the heat transfer system to resemble a mass-spring-damper system by applying the controlled motion input $\mathbf{u}$. Accordingly, the choices of stiffness and damping coefficients K, D will affect the performance of the system. Fig. 4.7 (a) demonstrates the effect of the stiffness-like coefficient K. To follow the standard form of error variation figures in visual servoing papers, we depict the change of $-\Delta T$ in the figures in this study. We set $D = 0.2$ and $T^* = 50\,°\mathrm{C}$ as constants and conduct three experiments with different K values. It shows that when $K = 0.005$ (red curve), $-\Delta T$ asymptotically decreases to zero with a relative slow speed; when K increases to 0.05 (blue curve), $-\Delta T$ decreases faster and a small overshoot occurs; when K increases to 0.5 (black curve), $-\Delta T$ oscillates near zero with an approximate $3\,°\mathrm{C}$ amplitude. This verifies that the designed system resembles the mass-spring-damper system. When the value of K increases, the system varies from over-damped to under-damped. Thus, the coefficients should be thoughtfully designed for different heat transfer systems according to the system performance requirements in real applications. We also conducted experiments with the same controller coefficients ($K = 0.05$, $D = 0.2$) but different target temperatures T^* which is depicted in Fig. 4.7 (a). It shows $-\Delta T$ is asymptotically minimized to zero for different target temperatures

For the multiple objects scenario, we conduct experiments to control the temperatures of two aluminium circular sheet as shown in Fig. 4.5 (b). We denote object temperature vector by $\tau = \begin{bmatrix} T_2^1 & T_2^2 \end{bmatrix}^\mathsf{T}$, the target temperature vector by $\tau = \begin{bmatrix} T^{*1} & T^{*2} \end{bmatrix}^\mathsf{T}$, and individual errors by $\Delta T^1 = T_2^1 - T^{*1}, \Delta T^2 = T_2^2 - T^{*2}$. We designed 4 experiments where the patterns of τ^* (measured in °C) are different. In Fig. 4.8, we depict $-\Delta T$ in the 4 experiments where target temperatures are set to $\tau^* = \begin{bmatrix} 50 & 40 \end{bmatrix}^\mathsf{T}$, $\tau^* = \begin{bmatrix} 60 & 40 \end{bmatrix}^\mathsf{T}$, $\tau^* = \begin{bmatrix} 50 & 50 \end{bmatrix}^\mathsf{T}$, and $\tau^* = \begin{bmatrix} 80 & 40 \end{bmatrix}^\mathsf{T}$ in (a), (b), (c), and (d). For the first three experiments where the differences between two target temperatures $|T^{*1} - T^{*2}|$ are small (or equals to zero), $||\Delta\tau||$ asymptotically decreases to zero. However, when $|T^{*1} - T^{*2}|$ is large, which is shown in Fig. 4.8 (d), two objects can not be controlled to reach the target temperatures, and $||\Delta\tau||$ can only decrease to a large local minimum. This failure can be explained by the second condition for feasible target temperatures discussed in Section 4.3.

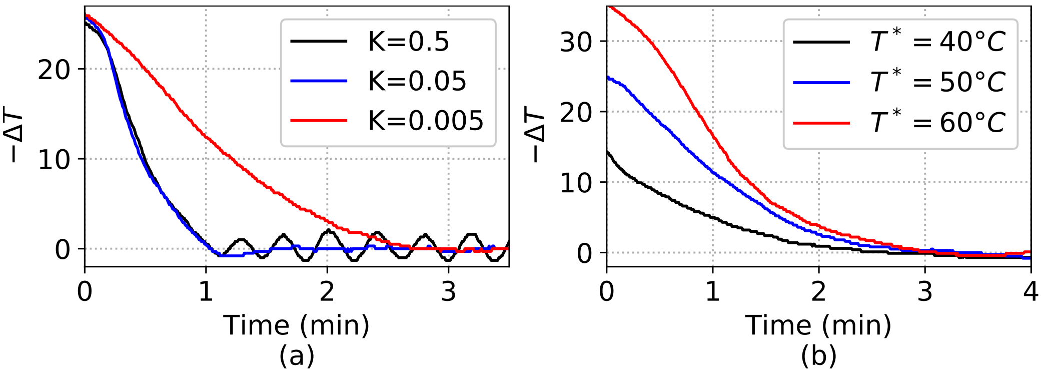


Figure 4.7: Evolution of the temperature error of the aluminium sheet ΔT in the experiments with the model-based controller.

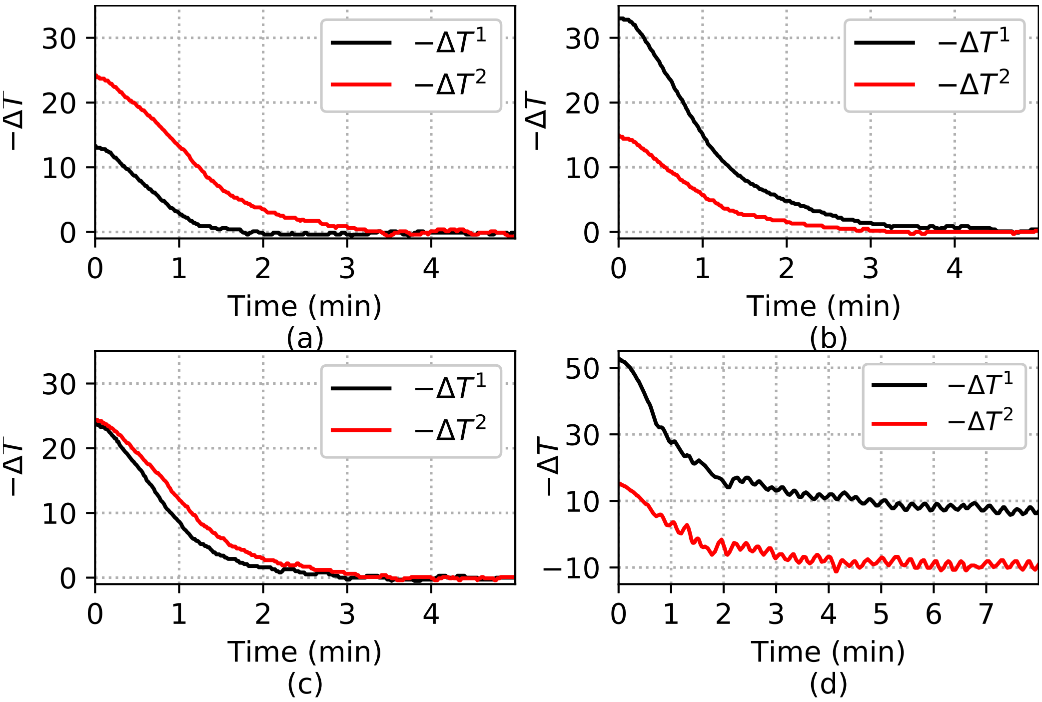


Figure 4.8: Evolution of the temperature errors of the two aluminium sheet $\Delta T^1, \Delta T^2$ in the experiments with the model-based controller.

4.7 Experiments with the Adaptive Controller

We designed a series of experiments to evaluate the performance of the adaptive controller. For the truncated Fourier series approximation, the more harmonics we use, the better the approximated contour will align with the original contour. However, for real time applications, we can use a small number of harmonics to achieve a relatively good approximation result (5 in this study). In these experiments with the adaptive controller, the temperatures of the aluminium circular sheet, the bunny-shaped sheet, and the hand-shaped sheet are to be controlled. We denote the vector of the target temperatures of the circular, bunny, and hand sheet by $\tau^* = \begin{bmatrix} T^{*1} & T^{*2} & T^{*3} \end{bmatrix}^\intercal$, and the corresponding vector of object temperatures by $\tau = \begin{bmatrix} T_2^1 & T_2^2 & T_2^3 \end{bmatrix}^\intercal$. The error vector is denoted by $\Delta\tau = \tau - \tau^* = \begin{bmatrix} \Delta T^1 & \Delta T^2 & \Delta T^3 \end{bmatrix}^\intercal$. The controller coefficients are set as $\mu = 0.05$, $K = 0.15$. We use learning gains $\gamma_1 = 0.005$, $\gamma_2 = 10^{-22}$ for the update rules (4.10),(4.11) to constrain $\dot{\hat{\mathbf{a}}}_1$, $\dot{\hat{\mathbf{a}}}_2$ to the same order of magnitude in each iteration. We set the initial values of the combined parameters as $a_1^1 = a_1^2 = a_1^3 = 3.4$, $a_2^1 = a_2^2 = a_2^3 = 4.2 \times 10^{-11}$, which are estimated according to the thermophysical parameters of aluminium, the heat source temperature $T_1 = 200\,^{\circ}\mathrm{C}$, and the environment temperature $T_3 = 25\,^{\circ}\mathrm{C}$.

In this study, we report eight temperature control experiments with different target temperatures and heat source conditions. Fig. 4.9 shows the variation of the negative individual temperature errors $-\Delta T^1$, -ΔT^2, and $-\Delta T^3$ in black, red, and blue colors. For ease of expression, we name these 8 experiments as *exp 1, exp 2, . . . , exp 8*, and denote the corresponding target temperatures of each experiment by $\tau^{*1}, \tau^{*2}, \ldots, \tau^{*8}$ (Values are listed in the caption part of Fig. 4.9). We set different values and patterns of object target temperatures in conducted experiments. In *exp 1*, we set the three target temperatures to the same value. In *exp 2, exp 3, exp 4*, we set two target temperatures to the same value, and the other to a lower or a higher value. In *exp 5, exp 6*, we set the target temperatures of the three objects to different values, with a non-uniform gap in *exp 5* and a uniform gap in *exp 6*. In *exp 7, exp 8*, the three target temperatures are set to the same value, while the heat source condition is different from the previous experiments. The heat source temperature T_1 is set to $200\,^{\circ}\mathrm{C}$ in *exp 1* $\sim$ *exp 6*, to $300\,^{\circ}\mathrm{C}$ in *exp 7*, and to vary from $200\,^{\circ}\mathrm{C}$ to $300\,^{\circ}\mathrm{C}$ in *exp 8*. In all of the 8 experiments, the temperature error magnitude $|\tau|$ asymptotically decreases to zero. In general, we validate that the adaptive controller is able to control the temperatures of three objects with different shape, material, color and surface condition without exact knowledge of the thermophysical properties of the objects, the heat source, and the environment.

In Fig. 4.10, we depict the 3D trajectories of the three objects in *exp 1, exp 2, ..., exp 8*. The boundary of the circular heat source is depicted as a black circle (and ellipse). The color of a trajectory point represents the temperature of the object at that position. A variation of color from blue to red corresponds to a change of temperature from low to high. To show the 3D trajectories clearly, we depict two sets of trajectory visualizations from different viewing angles: For Fig. 4.10 $(a_1),(b_1),\ldots,(h_1)$, the trajectories are viewed in $-\vec{\mathbf{k}}_1$ direction; for Fig. 4.10 $(a_2),(b_2),\ldots,(h_2)$, the trajectories are viewed in $\vec{\mathbf{i}}_1$ direction.

From the visualization of the trajectories, we find when the target temperatures are set to different values, the final position of the objects with higher target temperatures are closer to the center of the heat source, which can be explained either intuitively (the object closer to the center of the heat source receives more heat radiation

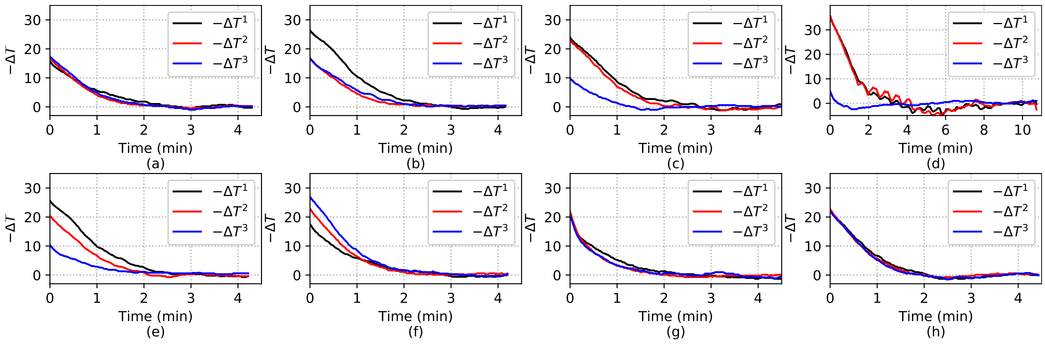


Figure 4.9: Evolution of the temperature errors of the three objects in the 8 experiments with the adaptive controller, ΔT^1, ΔT^2, and ΔT^3 (measured in °C). The target temperatures are set as: $\tau^{*1} = \begin{bmatrix}40 & 40 & 40\end{bmatrix}^\intercal$, $\tau^{*2} = \begin{bmatrix}50 & 40 & 40\end{bmatrix}^\intercal$, $\tau^{*3} = \begin{bmatrix}50 & 50 & 35\end{bmatrix}^\intercal$, $\tau^{*4} = \begin{bmatrix}60 & 60 & 30\end{bmatrix}^\intercal$, $\tau^{*5} = \begin{bmatrix}50 & 45 & 35\end{bmatrix}^\intercal$, $\tau^{*6} = \begin{bmatrix}40 & 45 & 50\end{bmatrix}^\intercal$, $\tau^{*7} = \begin{bmatrix}45 & 45 & 45\end{bmatrix}^\intercal$, $\tau^{*8} = \begin{bmatrix}45 & 45 & 45\end{bmatrix}^\intercal$.

energy), or by the spatial distribution of the view factor that we will discuss in the following Section 4.10; when the target temperatures are set to be the same value, the final position of the circular aluminium sheet is always closer to the center of the heat source, which is caused by the difference between the thermophysical properties of the objects. From this phenomenon, we obtain the intuition that the aluminium circular sheet is harder to be heated comparing the other two objects. This is caused by the absorptivity of the metal is usually much smaller than the absorptivity of non-metal. In fact, when we observing the aluminium sheet using a thermal camera, it looks like a mirror, which indicates the aluminium surface reflects most of the incident heat radiations.

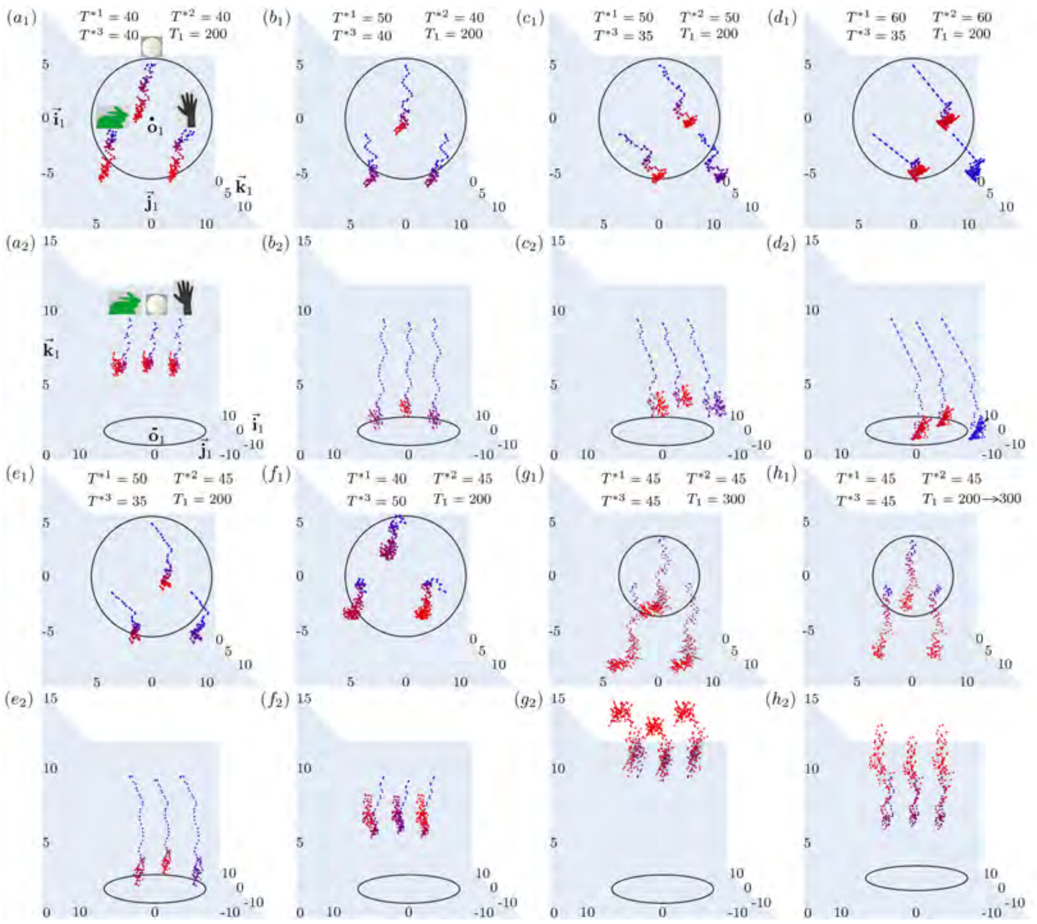
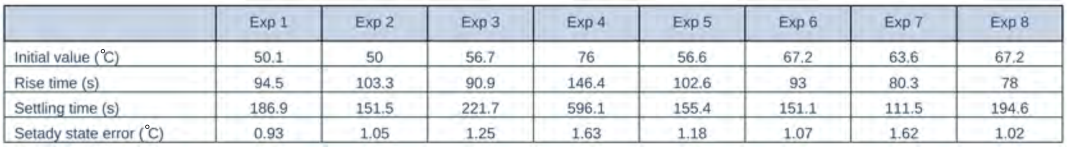


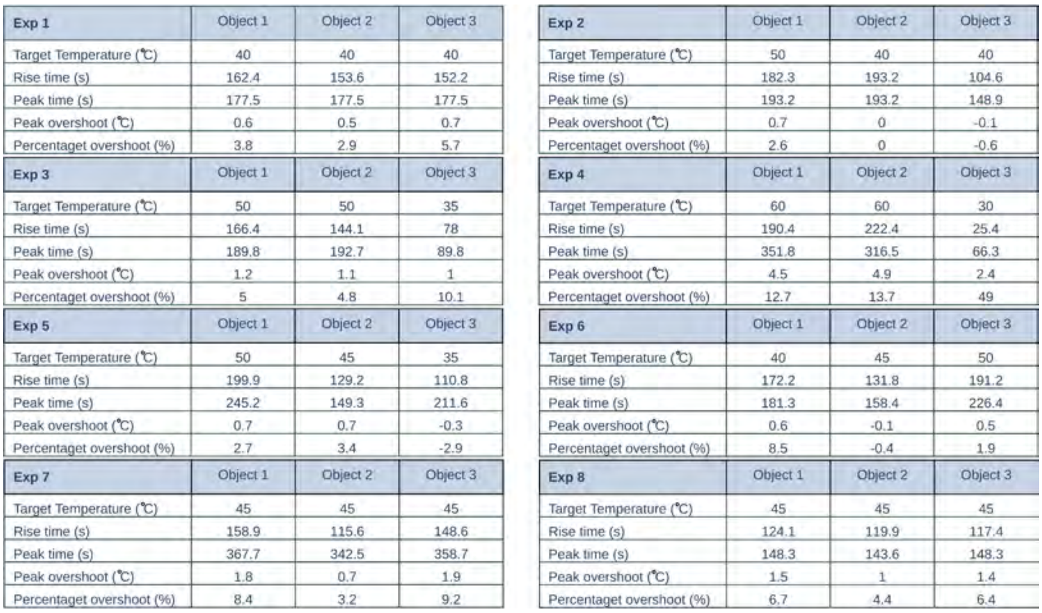
Figure 4.10: Spatial displacements of the three objects in the 8 experiments with the adaptive controller visualized from two viewing angles. We use change of colors to visualize the change of temperatures during the experiments.

4.8 Evaluation of the Controller Performance



	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6	Exp 7	Exp 8
Initial value (℃)	50.1	50	56.7	76	56.6	67.2	63.6	67.2
Rise time (s)	94.5	103.3	90.9	146.4	102.6	93	80.3	78
Settling time (s)	186.9	151.5	221.7	596.1	155.4	151.1	111.5	194.6
Setady state error (℃)	0.93	1.05	1.25	1.63	1.18	1.07	1.62	1.02

Figure 4.11: Evaluation of the adaptive controller's performance on the magnitude of temperature error vector $|\Delta\tau|$.



Exp 1	Object 1	Object 2	Object 3
Target Temperature (℃)	40	40	40
Rise time (s)	162.4	153.6	152.2
Peak time (s)	177.5	177.5	177.5
Peak overshoot (℃)	0.6	0.5	0.7
Percentaget overshoot (%)	3.8	2.9	5.7

Exp 2	Object 1	Object 2	Object 3
Target Temperature (℃)	50	40	40
Rise time (s)	182.3	193.2	104.6
Peak time (s)	193.2	193.2	148.9
Peak overshoot (℃)	0.7	0	-0.1
Percentaget overshoot (%)	2.6	0	-0.6

Exp 3	Object 1	Object 2	Object 3
Target Temperature (℃)	50	50	35
Rise time (s)	166.4	144.1	78
Peak time (s)	189.8	192.7	89.8
Peak overshoot (℃)	1.2	1.1	1
Percentaget overshoot (%)	5	4.8	10.1

Exp 4	Object 1	Object 2	Object 3
Target Temperature (℃)	60	60	30
Rise time (s)	190.4	222.4	25.4
Peak time (s)	351.8	316.5	66.3
Peak overshoot (℃)	4.5	4.9	2.4
Percentaget overshoot (%)	12.7	13.7	49

Exp 5	Object 1	Object 2	Object 3
Target Temperature (℃)	50	45	35
Rise time (s)	199.9	129.2	110.8
Peak time (s)	245.2	149.3	211.6
Peak overshoot (℃)	0.7	0.7	-0.3
Percentaget overshoot (%)	2.7	3.4	-2.9

Exp 6	Object 1	Object 2	Object 3
Target Temperature (℃)	40	45	50
Rise time (s)	172.2	131.8	191.2
Peak time (s)	181.3	158.4	226.4
Peak overshoot (℃)	0.6	-0.1	0.5
Percentaget overshoot (%)	8.5	-0.4	1.9

Exp 7	Object 1	Object 2	Object 3
Target Temperature (℃)	45	45	45
Rise time (s)	158.9	115.6	148.6
Peak time (s)	367.7	342.5	358.7
Peak overshoot (℃)	1.8	0.7	1.9
Percentaget overshoot (%)	8.4	3.2	9.2

Exp 8	Object 1	Object 2	Object 3
Target Temperature (℃)	45	45	45
Rise time (s)	124.1	119.9	117.4
Peak time (s)	148.3	143.6	148.3
Peak overshoot (℃)	1.5	1	1.4
Percentaget overshoot (%)	6.7	4.4	6.4

Figure 4.12: Evaluation of the adaptive controller's performance on the magnitude of temperature error vectors $|\Delta tau_1|, |\Delta tau_2|, |\Delta tau_3|$.

We have evaluated the performance of the adaptive controller by analyzing the collected data from the eight experiments and summarized the results in a table form (See Fig. 4.11, Fig. 4.12). The purpose of the developed controller is to asymptotically reduce $\Delta\tau$, which is a vector that is composed of the temperature errors of the

3 objects. To evaluate the controller performance, we determine the rise time, settling time, and the steady-state error of $|\Delta\tau|$. In this study, the rise time is defined as the time to decrease $|\Delta\tau|$ to 10% of its initial value, the settling time is defined as the time it takes to constrain $|\Delta\tau|$ within the 3% error band of its initial value, and the steady state error is calculated as an average of $|\Delta\tau|$ at each time step within the 3% error band. We find that the developed controller effectively drives the temperature error near the zero point in a short time. Furthermore, we include several detailed tables of controller performance on the individual temperature errors in the updated video, please check for more details.

4.9 Experiments with Interference

In this section, we report an integrated experiment where the adaptive temperature controller is combined with an online ArUco tracking algorithm to achieve temperature control tasks while the heat source is moving. In the Section 4.4, the ArUco markers attached to the heat source are used to obtain a fixed geometrical relationship between the robot and the heat source. This geometrical relationship is important for calculating a correct value of view factor, which is

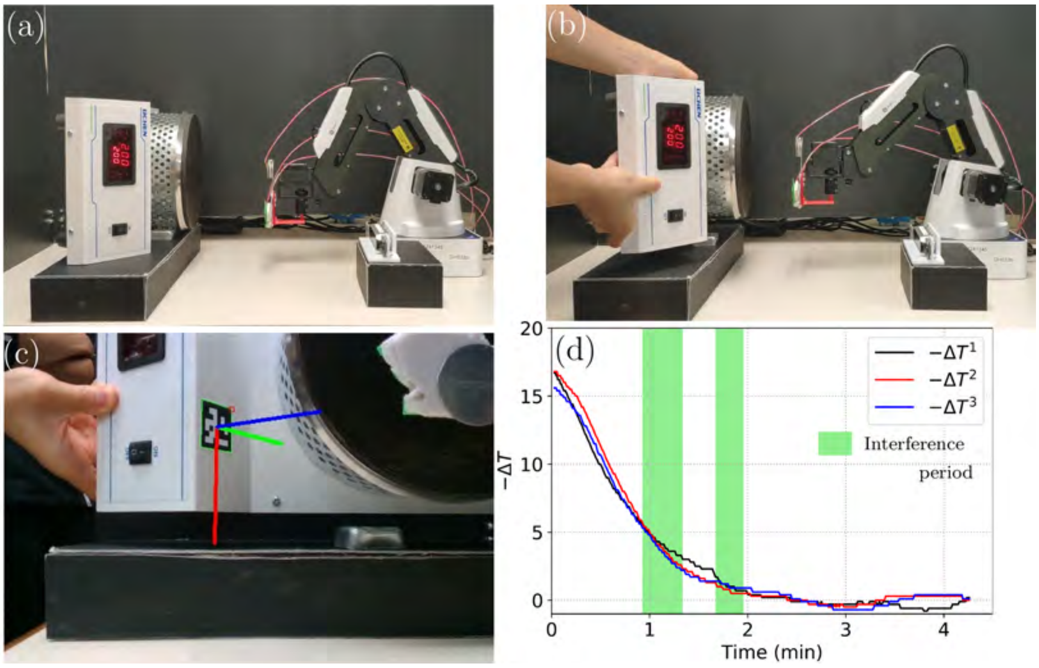


Figure 4.13: Experiments with Interference.

then used to online estimate the Jacobian matrix $\mathbf{J}$. However, when the heat source or the robot is moving, the new configuration of the parts has to be updated to the adaptive temperature controller. For simplicity, we study the case where robot is fixed and the heat source is moved manually (Since we only use one depth camera which is observing the heat source). We track the marker 2 attached to the heat source to online update the heat source configuration, and set the target temperature vector as $\tau = \begin{bmatrix} 40 & 40 & 40 \end{bmatrix}^\mathsf{T}$ (measured in °C). Figure 4.13 (a) shows the initial calibrated experiment set up, (b) shows the manual interference applied to the heat source during

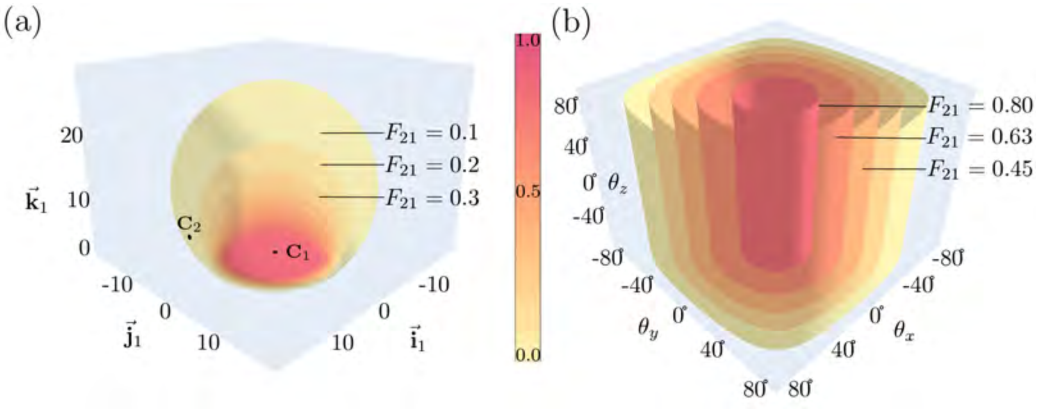


Figure 4.14: Isosurfaces visualization of the two view factor subsets. The translation subset is depicted in (a), and the rotation subset is depicted in (b).

the experiment, (c) shows the detected marker 2 during the interference period, and (d) shows the change of individual temperature errors (Notations are the same as the experiments with the adaptive controller). We find that during the manual interference period, decrease trends of magnitudes of the individual temperature errors are not affected, and $|\Delta\tau|$ asymptotically decreases to zero at the end of the experiment. This integrated experiment is to show a simple example of how to integrate the newly proposed method with the traditional robotic algorithms (like visual servoing).

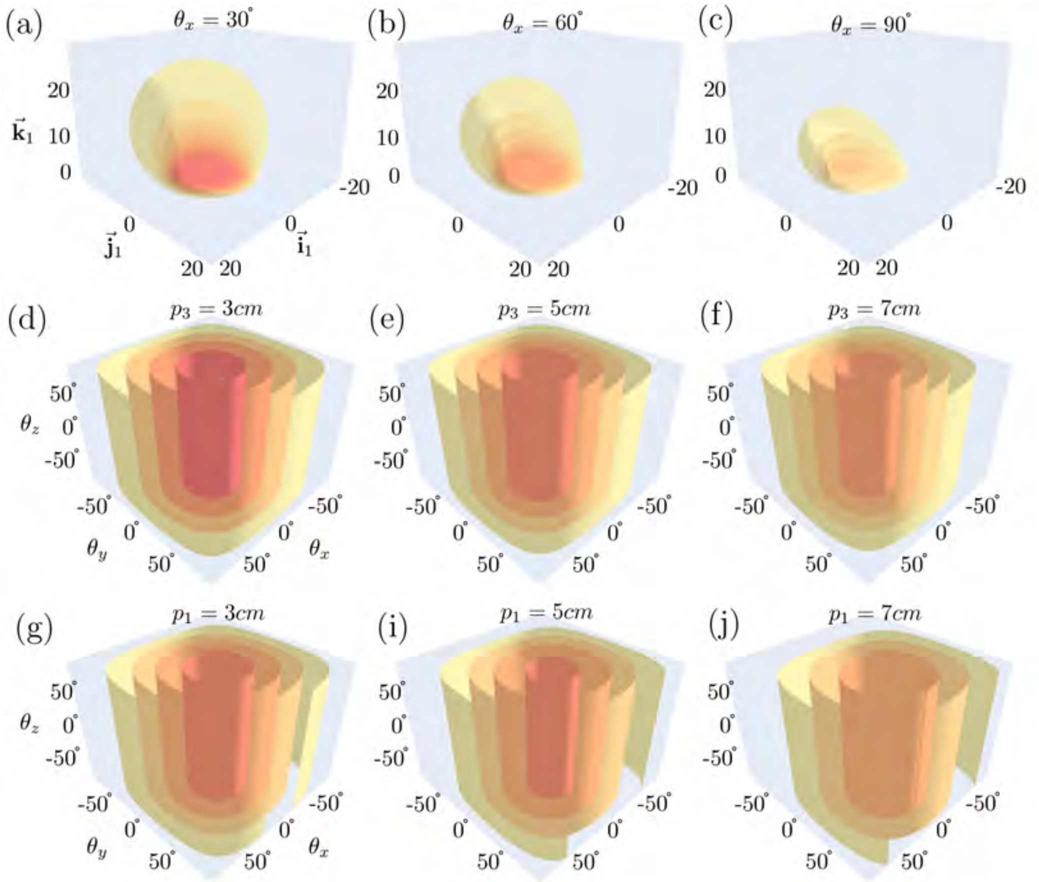


Figure 4.15: Controlled variable visualization of the view factor isosurfaces.

4.10 View Factor Visualization

In this section, we discuss the visualization of the view factor F_{21} with respect to the end-effector configuration $\mathbf{x}$, which could be a useful tool for analyzing and understanding radiation-based robotic TS problems. In previous sections, we design asymptotic stable temperature controllers based on derived heat transfer models. However, the equations relating F_{21} and $\mathbf{x}$ in these models are complex, which makes a part of the controlled system still remain like a "black box" to us. Thus, we introduce the visualization of F_{21} with respect to $\mathbf{x}$, and discuss the meaning and potential applications of it.

We take the view factor visualization of the scenario "circular surfaces in arbitrary configurations" discussed in previous sections as an example. We implement the controlled variable method to split the 6-DOF variable $\mathbf{x}$ into two 3-DOF subset: the translation subset where p_1, p_2, and p_3 (measured in centimeters) are controlled variables, and the rotation subset where θ_x, θ_y, θ_z (measured in degrees) are controlled variables. In the translation subset, the rotations are set to constants as $\theta_x = 0$, $\theta_y = 0$, $\theta_z = 0$. We then compute F_{21} for points in a selected working range of controlled variables

$p_1, p_2 \in [-20, 20]$, $p_3 \in [0, 30]$ with a step of 1 ($48,000$ points in total). In the rotation subset, the translations are set to fixed constants as $p_1 = 0, p_2 = 0, p_3 = 5$. We then compute points in range $\theta_x, \theta_y, \theta_z \in [-90, 90]$ with an incremental step of 2 ($729,000$ points in total).

We then use isosurface visualization tool provided by *Plotly* library [67] to visualize the calculated data. The translation subset is visualized in Fig. 4.14 (a), and the rotation subset is visualized in Figure 4.14 (b). Concretely, 3-DOF end-effector configurations are represented by points in 3D space, and 1-DOF view factor values are represented by isosurfaces with different colors. Isosurfaces are formed by points which have the same (or close) value of F_{21}. The isosurface visualization method is inspired by (3.34) which reveals that the Jacobian matrix is positive proportional to the directional derivative of $F_{21}(\mathbf{x})$ along $\mathbf{x}$ as $\mathbf{L} = \lambda_1 \nabla_{\mathbf{x}} F_{21}(\mathbf{x})$. According to the definition of isosurface, the surface normal of every point on the surface also points in $\nabla_{\mathbf{x}} F_{21}(\mathbf{x})$ direction. In addition, the distances between isosurfaces with a equal value difference show the magnitudes of the elements of $\nabla_{\mathbf{x}} F_{21}(\mathbf{x})$. A larger distance represents a smaller magnitude.

To show the usefulness of the view factor visualization, we take the analyze of the translation subset shown in Fig. 4.14 (a) as an example. For the single object scenario, according to the property of vector pseudo-inverse and designed control inputs (4.1), (4.8), the surface normal at a point on the isosurfaces also indicates the direction of the end-effector movement at that point. Before visualization, there are some obvious characteristics of isosurfaces that we can predict, including the symmetric spatial distribution of F_{21} (due to the circular shape of the heat source), and F_{21} decreases when the object moves further from the heat source surface. However, we discover some useful hidden information after visualization: The centres of the isosurface spheres are shifting upwards when F_{21} decreases. This means at some points, the movement in $\vec{\mathbf{k}}_1$ direction will cause a decrease of F_{21}, which is counter-intuitive (people may think moving the object towards the heat source will always increase the heating speed). See $\mathbf{C}_2$ on the $F_{21} = 0.1$ isosurface in Fig. 4.14 (a). When the end-effector is at $\mathbf{C}_2$, it needs to move backwards in $\widehat{\mathbf{k}}_1$ direction to heat up faster; in the regions where the end-effector is far from the heat source, the distance between isosurfaces with the same view factor value difference are comparatively larger. Thus, when the end-efector is far from the heat source, the control inputs will have a larger magnitude and the end-effector moves faster.

Previously, 3 DOF of $\mathbf{x}$ are set to be controlled variables in the two subsets for visualization. For further understanding the effect of motion on heat transfer process, we present a visualization of the view factor with respect to 4 DOF $\mathbf{x}$ in Fig. 4.15. For the previous 3 DOF translation subset, we add one more controlled variable θ_x to extend it to 4 DOF. Concretely, we vary the rotation θ_x (which is set to constant in 3 DOF visualization) from 0 to 90 and depict the change of isosurfaces in Fig. 4.15 (a), (b), (c). We find when the geometrical relationship between the object surface and the heat source surface changes from parallel to perpendicular, there is a decrease trend of view factor values and a shape change of isosurfaces. For the previous 3 DOF rotation subset, we vary the translation p_3 from 1 to 10, and depict the change of isosurfaces in Fig. 4.15 (d), (e), (f). We find a decrease trend of view factors when the end-effector moves in $\vec{\mathbf{k}}_1$ direction, while the shape of isosurfaces maintain the same. Similarly, we vary the translation p_1 from 0 to 10 and depict the change of isosurfaces in Fig. 4.15 (g), (h), (i). We also find the values of the isosurfaces decrease when the end-effector moves in $\vec{\mathbf{i}}_1$

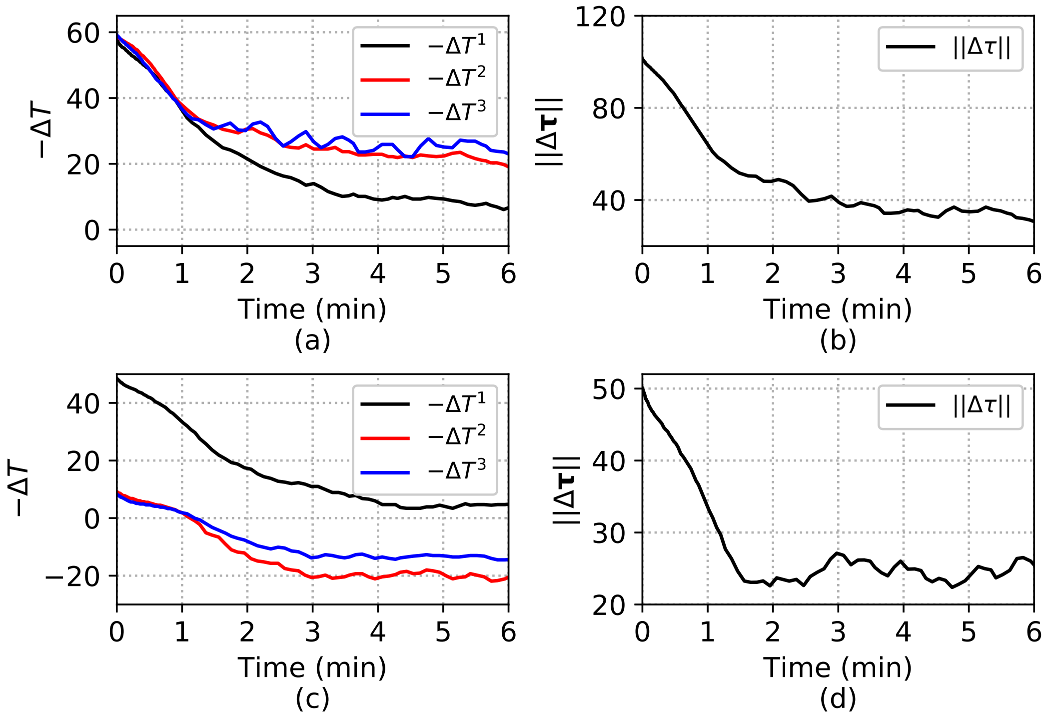


Figure 4.16: Evolution of the temperature errors of the three objects when unfeasible target temperatures are set.

direction.

4.11 Unfeasible Thermal Targets

In Section 4.3, we discussed two necessary but insufficient conditions for feasible target temperatures. When one of the two conditions is not fulfilled, the magnitude of the temperature error can not be minimized to zero, and we classify such cases as failed experiments caused by the unfeasible target temperatures. We report two failed experiments with adaptive controller where the target temper-

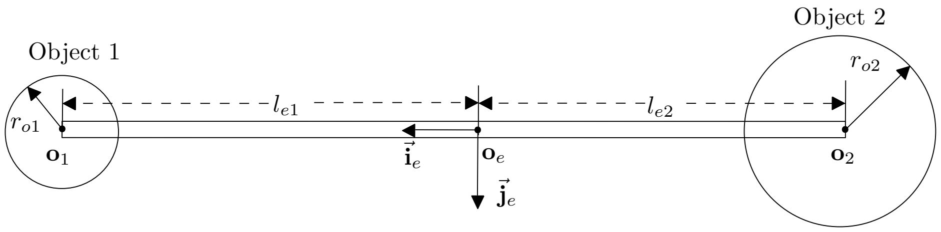


Figure 4.17: Conceptual illustration of two objects fixed to an end-effector for analyzing unfeasible target temperatures.

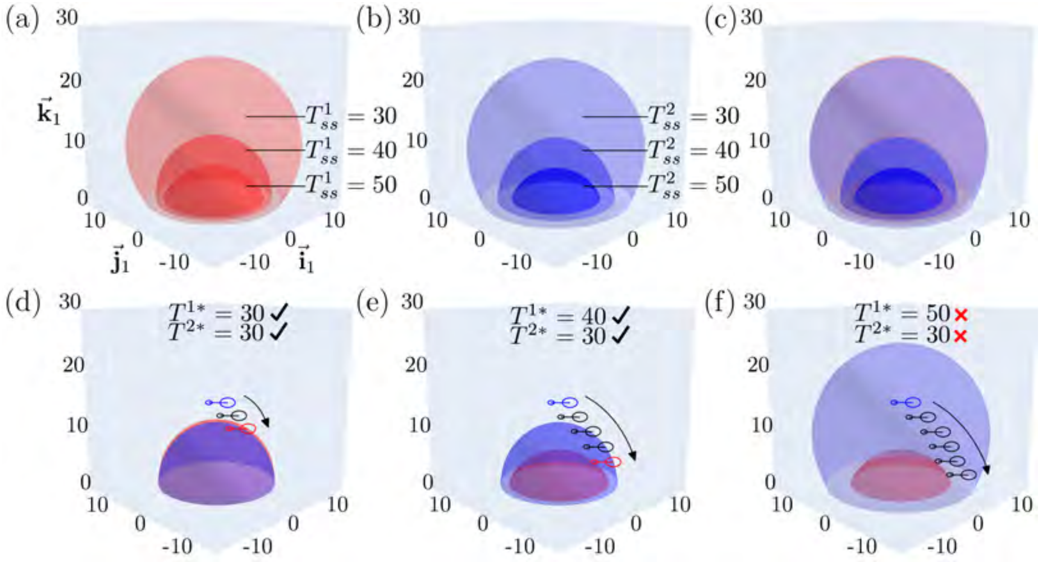


Figure 4.18: Geometrical explanation of the target temperature feasibility using the steady-state temperature isosurfaces.

atures are set as $\tau^* = \begin{bmatrix} 80 & 80 & 80 \end{bmatrix}$, $\tau^* = \begin{bmatrix} 70 & 35 & 35 \end{bmatrix}$, and depict the individual temperature errors ΔT^1, ΔT^2, and ΔT^3 in Fig. 4.16 (a), (c). We also depict the magnitude of the aggregate temperature error $\|\Delta\tau\|$ of the two experiments in Fig. 4.16 (b), (d). We find that when the target temperatures are set too high, all temperature errors for independent objects will not decrease to zero, and $\|\Delta\tau\|$ will asymptotically decrease to a large local minimum; when the difference between individual target temperatures is too large, the object with the highest initial individual temperature error will finally reach its target temperature, while the other individual temperature errors and overall temperature error will asymptotically decrease to a large local minimum.

In Section 4.3, we proof that the theoretical steady state temperature of an object heated by a heat source is positively proportional to the view factor F_{21}. Thus, the visualization of view factor isosurfaces is also useful for understanding the two conditions of feasible target temperatures. We discuss a simple but representative case where two aluminium circular sheet with radius $r_{o1} = 1.5\,\text{cm}, r_{o2} = 4.5\,\text{cm}$ are attached to the end-effector at $\mathbf{o}_1, \mathbf{o}_2$ (see Fig. 4.17) and heated by the same circular heat source used in the experiment which is maintained at 200 °C. The center of the end-effector is at $\mathbf{o}_e$, and $l_{e1}, l_{e2} = 2\,\mathrm{cm}$ are the distances between $\mathbf{o}_1, \mathbf{o}_2$ and $\mathbf{o}_e$. The basis vectors of end-effector coordinate system $\vec{\mathbf{i}}_e, \vec{\mathbf{j}}_e$ are depicted in the figure and $\vec{\mathbf{k}}_e$ can be determined accordingly. The origin of the coordinate system used for calculating the view factors is set at the center of the heat source and the basic vectors $\vec{\mathbf{i}}_1, \vec{\mathbf{j}}_1, \vec{\mathbf{k}}_1$ are set to be parallel to $\vec{\mathbf{i}}_e, \vec{\mathbf{j}}_e, \vec{\mathbf{k}}_e$. To explore the unfeasible temperature cases caused by the fixed geometrical configuration of multiple objects on the same end-effector, we select the visualization method where three translations are controlled variables. Thus, we restrict the robot can only conduct 3 DOF translations to ensure the object surface is always parallel to the heat source surface (When the objects are not parallel to the heat source surface, the shape of the isosurfaces will change as shown in Fig. 4.15).

View factors of objects 1, 2 with respect to the heat source coordinates $\vec{\mathbf{i}}_1, \vec{\mathbf{j}}_1, \vec{\mathbf{k}}_1$ were calculated within the range of $p_1, p_2 \in [-15, 15]$, $p_3 \in [0, 30]$ with a step of 1 (measured in centimeters). According to (4.18) and assuming the thermophysical properties of the objects, the heat source and the environment are the same as mentioned in Section 4.4, view factor values corresponding to $30\,°\mathrm{C}, 40\,°\mathrm{C}, 50\,°\mathrm{C}$

steady state temperatures are calculated as 0.17, 0.43, and 0.61. Since there exists a one-to-one correspondence between the view factor and and the steady state temperature, controlling an object to reach the target temperature T^* can be geometrically interpreted as controlling the object center to approach the view factor isosurface that corresponds to T^*. Accordingly, determining the feasibility of target temperatures T^{*1}, T^{*2} of two objects attached to the same end-effector is identical to finding whether there exists an end-effector configuration that makes objects 1, 2 land on the isosurfaces corresponding to target temperatures simultaneously.

An example of how the target temperature feasibility can be visualized and interpreted using isosurfaces is shown in Fig. 4.18. We denote the steady state temperatures of objects 1, 2 as T_{ss}^1, T_{ss}^2. Fig. 4.18 (a), (b) show steady state temperature isosurfaces of object 1, 2, where $T_{ss}^1, T_{ss}^{12} = 30\,°\text{C}, 40\,°\text{C}, 50\,°\text{C}$. We denote the three isosurfaces of object 1 as set 1, and the three isosurfaces of object 2 as set 2. Since the two objects are circular disks with different radius, the shapes of sets 1, 2 are similar but different (see the bottom part of the two sets). To show the differences between the two sets, set 1, 2 are colored in red, blue and depicted in the same coordinate system in Fig. 4.18 (c).

In Fig. 4.18 (d), (e), (f), different combinations of target tmeperatures T^{1*}, T^{2*} and their corresponding isosurfaces are depicted. We find that for the cases where $T^{*1} = T^{*2} = 30\,^{\circ}\mathrm{C}$ (depicted in Fig. 4.18 (d)) and $T^{*1} = 30\,^{\circ}\mathrm{C}, T^{*2} = 40\,^{\circ}\mathrm{C}$ (depicted in Fig. 4.18 (e)), the two target isosurfaces are relatively close to each other; the end-effector starting from its initial configuration (colored in blue) can find an appropriate configuration (colored in red) where the two objects land on the target isosurfaces simultaneously, which proves that these two combinations of T^{*1}, T^{*2} are feasible. While for the case where $T^{*1} = 50\,^{\circ}\mathrm{C}, T^{*2} = 30\,^{\circ}\mathrm{C}$ (depicted in Fig. 4.18 (f)), we find the minimum distance between two isosurfaces is larger than $l_{e1} + l_{e2}$. Thus, this combination of T^{*1}, T^{*2} is not feasible. Similarly, if $l_{e1} + l_{e2}$ is larger than the maximum distance between two target isosurfaces, that combination of T^{*1}, T^{*2} is unfeasible. In general, thermophysical properties, view factors, and the fixed distances between objects are the three main factors that affect the feasibility of target temperature. Considering the feasibility problem as an identical geometrical problem could help human to understand the unseen heat transfer processes, and might be useful for

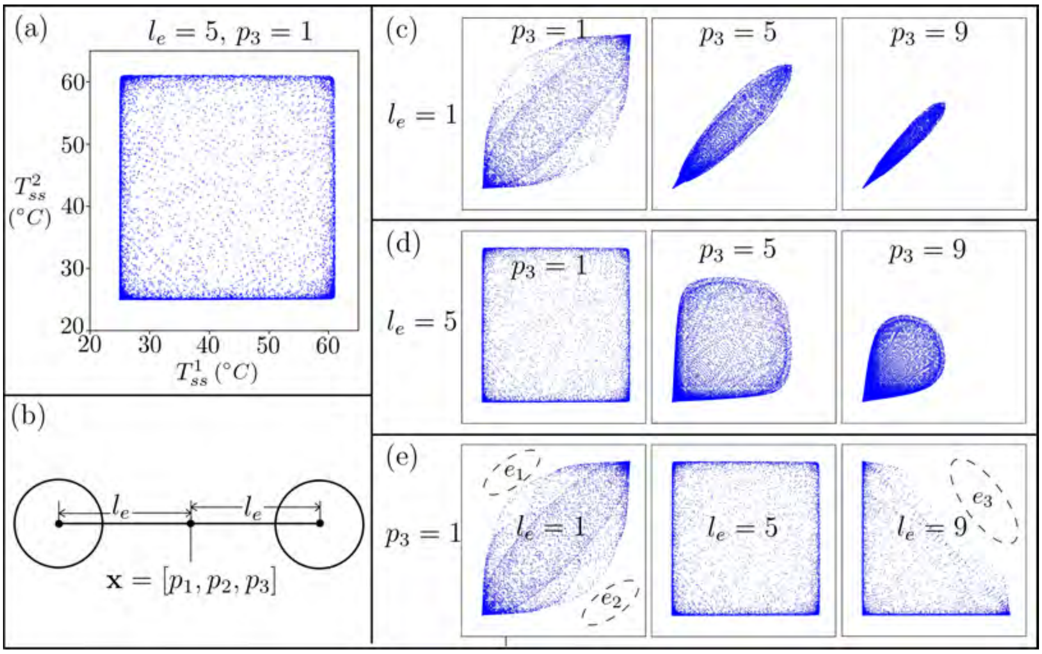


Figure 4.19: Visualization of the target temperature space

path-planning related thermal servoing problems.

In addition to the aforementioned geometric explanation, we conduct an analysis of the characteristics of the entire **feasible temperature space** Θ_{temp}, which is defined as the collection of all sets of steady-state temperature

$\mathbf{T}_{ss}(\mathbf{x}) = [T_{ss}^1(\mathbf{x}_{obj}^1), \ldots, T_{ss}^n(\mathbf{x}_{obj}^n)], \mathbf{T}_{ss}(\mathbf{x}) \in \Theta_{temp}$ of N objects attached to the same end effector, where $\mathbf{x}$ is the end-effector configuration, $\mathbf{x}_{obj}^n, n = 1, \ldots, N$ is the object configuration, and $T_{ss}^n(\mathbf{x}_{obj}^n)$ is the steady-state temperature of an object when its center is at $\mathbf{x}_{obj}^n$. Since there is a fixed geometric relationship between $\mathbf{x}$ and $\mathbf{x}_{obj}^n$, values of $T_{ss}^n(\mathbf{x}_{obj}^n)$ can be calculated.

Here, we take the set-up in Fig. 4.19 (b) where 2 identical circular aluminium objects are attached to a 3-DOF end-effector with an identical distance l_e as a case of study. We uniformly sample a discrete end-effector configuration space where p_1, $p_2 \in [-30\,\text{cm}, 30\,\text{cm}]$, $p_3 = 1\,\text{cm}$ with a step of $0.2\,\text{cm}$ (90,000 points) and calculate the view factor values. Accordingly, $\mathbf{T}_{ss}(\mathbf{x}) \in \mathbb{R}^2$, and $\Theta_{temp} \in \mathbb{R}^{90000\times 2}$ can be obtained. For each $\mathbf{T}_{ss}(\mathbf{x})$ in a feasible temperature space Θ_{temp}, we depict it as a 2D point such as in Fig. 4.19 (a). Although $\mathbf{x}$ is uniformly sampled, the distribution of $\mathbf{T}_{ss}(\mathbf{x})$ is not uniform, which is caused by the non-linear thermal-geometric coupling. By changing the values of l_e, p_3, several representative combinations are also depicted in Fig. 4.19 (c), (d), (e). We analyze these figures and discover the following thermal-geometrical coupling characteristics:

Consider Figure 4.19 (c), (d), if the end-effector moves away from the heat source (p_3 increases), the area of feasible temperature space Θ_{temp} shrinks, which means the range of feasible temperature is smaller. However, the density of points increases; It indicates the robot motion induces a smaller change on heat transfer to the object, which to some extent increases the accuracy of temperature control when p_3 increases. Secondly, refer to Fig. 4.19 (e), we discover the geometric coupling between objects (l_e in this case) also affects Θ_{temp}. If objects are close (e.g. $l_e = 1$), and the difference between the target temperature of the objects is large, these combinations of target temperature are not feasible (see the blank area e_1, e_2); If objects are far away (e.g. $l_e = 9$), controlling two objects to reach high temperature simultaneously is not feasible (see the blank area e_3).

For cases where $N > 3$, visualization is not practical, and some advanced data analysis is required (which is beyond the scope of this thesis). The analysis of the feasible temperature space reveals the physical essence of the thermal-geometric coupling, and could shed some light on practical thermal servoing system design. Nevertheless, it takes hours of computation even with parallel-computing to obtain the required data. To quickly verify the feasibility of a specific set of target temperatures, we find reformulating the problem from an optimization perspective is more effective.

Consider N objects attached to the same end-effector, the steady-state temperature of each object is denoted by $T_{ss}^{n}(\mathbf{x}_{obj}^{n})$, $n \in \{1, 2, \ldots, N\}$. We denote the target temperature of each object by T^{n*}, then the target feasibility problem can be solved by solving the following optimization problem:

$$\min_{\mathbf{x}} c(\mathbf{x}) = \sum_{n=1}^{N} |T_{ss}^{n}(\mathbf{x}) - T^{n*}| \quad \text{s.t. } \mathbf{x} \in \mathscr{W} \tag{4.27}$$

where $\mathscr{W}$ is the robot working space. If the global minimum of $c(\mathbf{x})$ equals to 0, the set of target temperature is feasible. We use a simple homology global optimization [68] algorithm, which is implemented in *SciPy* library, to conduct a verification of the method's feasibility. It turns out that the global minimum of $c(\mathbf{x})$ can be found effectively (in less than 1 minute) for randomly selected target temperatures and geometric relationships between objects.

4.12 Independent Control of each Feedback Temperature

When the objects are attached to the same end-effector, the motion of these objects is coupled and therefore, the evolution of their temperature is also coupled. Nevertheless, according to our analysis on the spatial distribution of the view factor, independent control of different feedback temperatures is **possible**, yet, this condition is local, subject to constraints, and limited to three sensing points (i.e.

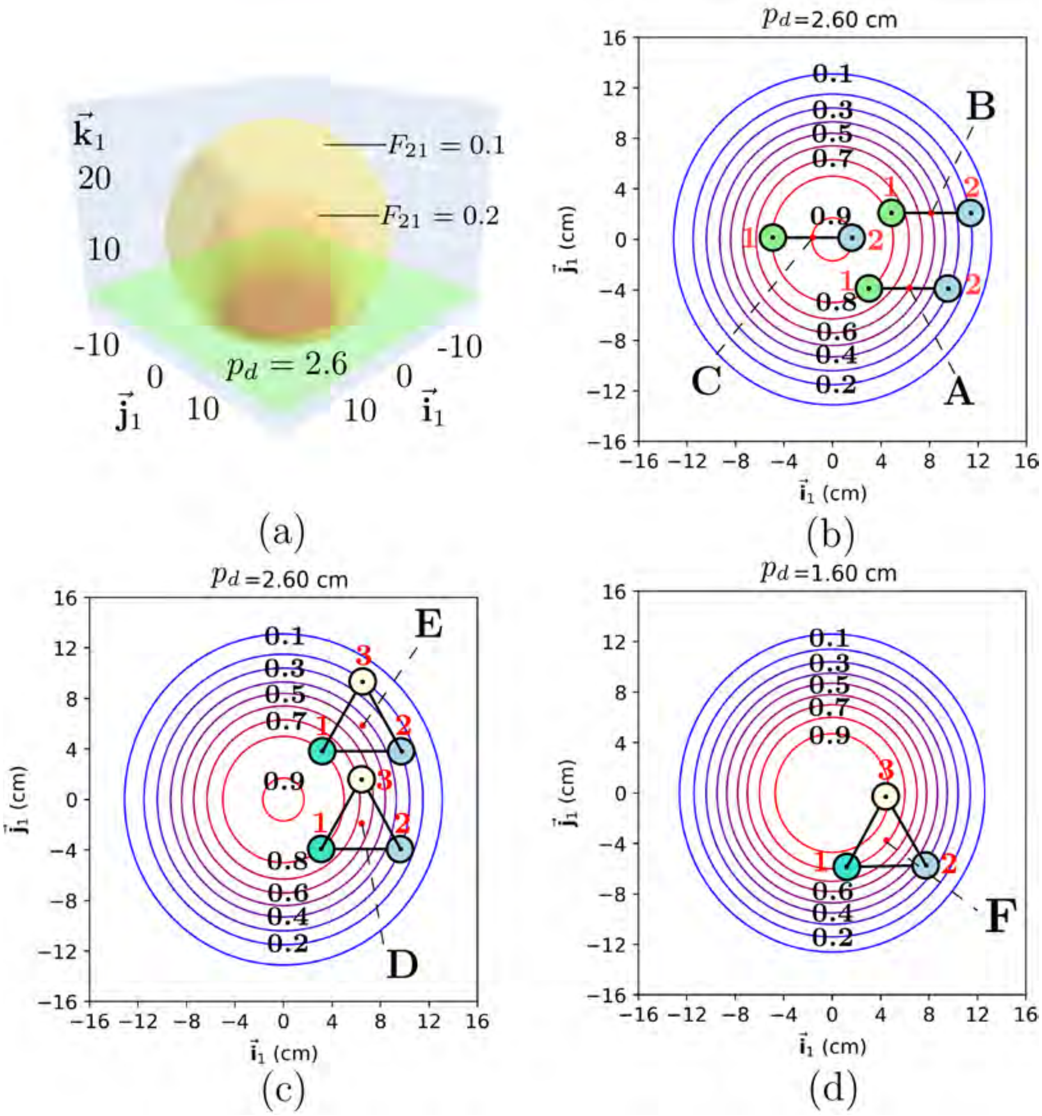
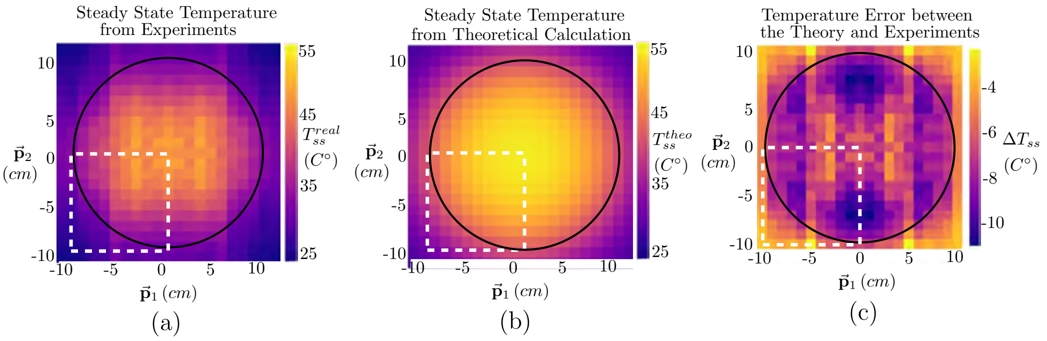


Figure 4.20: Representative examples of the independent control of the temperature of the objects. The spatial positioning of the rigidly grasped objects on to "desired" isosurfaces limits the number of independent thermal sensing points to 3 at most.

small objects) at most. The feasibility for independent control is determined by the geometry of the view factor isosurfaces and the rigid configuration amongst the manipulated objects. According to Section 4.3, if the view factors can be (locally) manipulated independently by the robot motion, we can fairly state that the objects' temperature can also be independently controlled (as the view factor is responsible for determining the heat transfer rate at a specific point). In the following, we use a simple 2D scenario to demonstrate various representative examples of this situation.

Referring to Figure 4.20 (a) below, we take a horizontal 2D slice of the 3D "view factor isosurfaces" (introduced in Section IV E and F) with height $p_d = 2.6\,\text{cm}$ and depict it in Figure 4.20 (b). The view factor contours with values ranging from 0.1 to 0.9 are depicted and labeled. Consider the original configuration of the end-effector holding two objects is at position **A**, and the view factor values of object 1, 2 are $F_{21}^1 = 0.8, F_{21}^2 = 0.3$. By moving object 1 along the contour where $F_{21}^1 = 0.8$, the heat transfer rate to object 1 is held constant, and we could **independently** control the heat transfer rate of object 2. For example, if the end-effector moves from **A** to **B**, F_{21}^2 decreases from 0.3 to 0.2 while F_{21}^1 is held constant; if the end-effector moves from **A** to **C**, F_{21}^2 increases from 0.3 to 0.9 while F_{21}^1 is held constant.

We can also extend this idea to a 3-object scenario (although, the problem further complicates). As depicted in Figure 4.20 (c), the end-effector holding object 1, 2, 3 is at the original configuration **D**, and the view factors are $F_{21}^1 = 0.8, F_{21}^2 = 0.3, F_{21}^3 = 0.65$. By holding F_{21}^1, F_{21}^2 constant, we independently control the view factor of object 3 to decrease from 0.65 to 0.2 by "sliding" the end-effector from **D** to **E**, along the specified contours. As shown in Figure 4.20 (d), by moving the robot in the off-plane $\vec{\mathbf{k}}_1$ direction, the shapes of view factor contours vary, and F_{21}^3 is independently controlled to increase from 0.65 to 0.9. Note that these simple examples only use translations. By incorporating controllable orientations, other isosurfaces can be reached. The *spatial* positioning of the rigidily grasped objects onto "desired" isosurfaces clearly limits the number of independent thermal sensing points to 3 at most.



(a) (b) (c)

Figure 4.21: Comparison between the steady state temperature collected from experiments and from theoretical calculation.

4.13 Comparison between Optimization Control and Robot Thermal Servoing

As mentioned in Section 4.3, the steady state temperature of the object at a specific location can be calculated. It seems intuitive to define the constraints of the problem, solve the optimization problem to find the adequate location of the object with respect to the heater to get the desired steady temperature and just control the robot to the equilibrium pose.

Temperature control achieved by position optimization (open-loop control) is feasible for industrial thermal engineering applications. Consider for instance [69], where uniform heating was achieved by finding the optimal heater configurations, and [70] where the optimization principles were discussed for energy-saving thermal system design. To implement these classical *simulation-based* optimization methods, we require accurate knowledge of the thermophysical properties of the object, the heat source, and the environment. If any of these parameters is uncertain or changes during the process, a position optimization method (without online adaptation) will perform poorly.

We conducted a simple test to verify this statement, where we uniformly sampled a discrete end-effector configuration space where $p_1, p_2 \in [-10\,\mathrm{cm}, 0\,\mathrm{cm}]$, $p_3 = 5\,\mathrm{cm}$ with a step of $1\,\mathrm{cm}$ (121 points in total) and collected the steady state temperature of the aluminium plate used in the thesis from experiments and denote it by T_{ss}^{real}. We then calculated the corresponding theoretical steady state temperature according to equation (49) and denote it by T_{ss}^{theo}. The error between the theory and the experiment is denoted by $\Delta T_{ss} = T_{ss}^{real} - T_{ss}^{theo}$. We visualized $T_{ss}^{real}, T_{ss}^{theo}, \Delta T_{ss}$ in the form of heat map in Figure 4.21. Note that the actual configuration space is enclosed by the white dashed box, which is only $1/4$ of the plane. For a better visualization, we completed the other regions symmetrically. The contour of the heat source is depicted by the black circle.

We find a considerably large error between the theoretical calculation and the experiments. The simplified heat transfer model is not the only reason for the error; in practical situations, the emittance of the heat source and the absorptance of the object are also hard to be accurately estimated in advance. To directly control the robot to the equilibrium pose, online estimation of the thermophysical properties of the objects, and an effective algorithm to recalculate the whole steady state temperature space are required. We think it is an alternative temperature control strategy, which has a comparable complexity to the proposed thermal servoing approach, however, it is not robust to uncertainties.

Chapter 5

“Fire-in-Hand” Thermal Servoing: Feedback Solar Concentration

5.1 Introduction

In the past few decades, exploitation of solar energy has been attracting people’s attention since it is a sustainable, powerful and ubiquitous energy source that is a promising alternative to fossil fuels. Large scale solar power plants were developed in rural areas like deserts and flatland; In urban areas, solar power collection devices have been deployed in infrastructures for electricity generation or redirecting natural sunlight inside buildings; Research on solar powered automobiles has also been reaching monuments one after another .

In robotics community, there are a large amount of work that utilizes solar power as electricity supply to the robotic systems. Such as in [71], a pan-tilt platform that carries a solar panel automatically followed the orientation of the sun according to the feedback signal from the calibrated light sensor series was designed. In [72], a locomotion robotic platform equipped with a thermal camera, a laser scanning device, and a solar panel navigated itself within a certain territory to generate a solar energy distribution map, which was then used to optimize its performance on path planning with respect to the expected energy consumption and collection trade-off.

Except for the generation of electricity that relies on the photoelectric effect, the potential of heat generation from the solar power did not raise enough attention among the robotics community. The sun emits heat radiation to the earth, of which the intensity is approximately 1000 watt per square meters. Inspired by the design of solar thermal power plants, which coordinates thousands of mirrors to focus and heat up a traditional steam electricity generation system, we propose to develop a new type of field robot that effectively utilize the thermal effect of solar energy. There are at least 3 essential functions that needs to be carefully designed: (a) The robot should be capable of tracking the orientation of the sun in real-time to maximize the net amount of the collected solar energy; (b) The robot should be capable of controlling the direction of solar rays such that the point of interest could be accurately heated; (c) The robot should be able to manipulate the heat power intensity transferred to the target point, such that a desired steady state temperature or an appropriate heating process could be achieved.

5.2 Methodology

5.2.1 Notation

Throughout this manuscript, we denote all *column* vectors by small bold letters, e.g. $\mathbf{r} \in \mathbb{R}^{n\times 1}$, and matrices by capital bold letters, e.g. $\mathbf{T} \in \mathbb{R}^{m\times n}$. We use a pre-superscript to indicate the coordinate system of a positional vector, e.g. ${}^{a}\mathbf{r} \in \mathbb{R}^{4\times 1}$ is a homogeneous position vector in the frame $(xyz)_a$. A homogeneous transformation matrix $\mathbf{T}_b^a \in \mathbb{R}^{4\times 4}$ describes the transformation from frame $(xyz)_a$ to frame $(xyz)_b$.

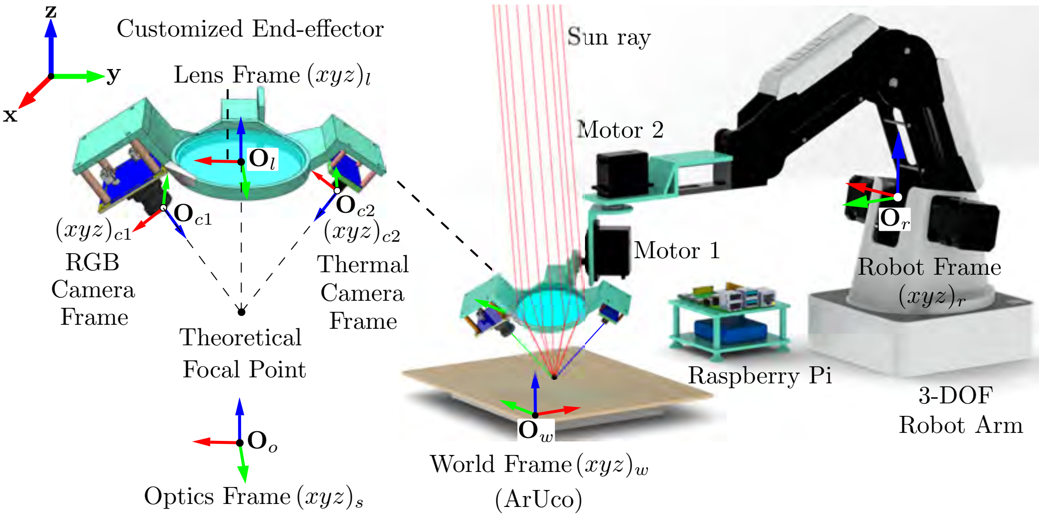


Figure 5.1: A schematic of the developed system.

5.2.2 System Design

The main components of the developed system are depicted in Figure 5.1. As will be discussed in Section 5.2.3, the robotic system needs at least 5-DOF to achieve the aforementioned objectives. To this end, we integrate a 3-DOF robotic arm *Dobot Magician* with two servo motors (controlled by a *Raspberry Pi*) to cooperatively manipulate a customized end-effector, which carries a spherical Fresnel lens, an RGB camera, and a thermal camera *FLIR Lepton 3.5*. We specifically design the structure of the end-effector so that the optical axis of the Fresnel lens and the optical axes of the two cameras intersect at the theoretical lens focal point, which creates a large overlapping of the FOVs of the two cameras. The pitch angle and the yaw angle of the lens are independently controlled by motor 1 and 2, while the 3-DOF translation of the lens is controlled by the robotic arm. Consequently, the configuration of the end-effector is denoted by $\mathbf{x} \in \mathbb{R}^5$. Relying on the system kinematics, the distance between the lens center and an arbitrary target point can be set while maintaining the desirable pitch and yaw angles of the lens.

With the aid of Figure 5.1, we clarify several frequently used notations. We denote the static robot base frame by $(xyz)_r$. The world frame, which is determined by the ArUco marker attached to the wood plank, is denoted by $(xyz)_w$. We denote the origin of the lens frame $(xyz)_l$ by $\mathbf{O}_l$ and set it at the center of the lens upper surface. We align the initial axes of $(xyz)_l$ with the axes of $(xyz)_r$ so that the initial lens yaw and pitch angles are set to zero. The RGB camera frame $(xyz)_{c_1}$ and the thermal camera frame $(xyz)_{c_2}$ are set according to the conventions [73]. Relying on the system kinematics, the robot-lens transformation matrix $\mathbf{T}_l^r$ can be derived. Since the cameras and the Fresnel lens are rigidly attached to the customized end-effector, $\mathbf{T}_{c_1}^l$, $\mathbf{T}_{c_2}^l$, $\mathbf{T}_{c2}^{c1}$ are known constant matrices.

At the beginning of an task, the world-robot transformation $\mathbf{T}_r^w$ is obtained by detecting the ArUco marker attached to the wood plane. The robot then moves to a predefined configuration to capture an RGB image of the light spot, which is streamed to our proposed algorithm to determine the sun's azimuth angle and altitude angle. With the estimated solar angles, an evaluation experiment is conducted to obtain the optimal focal length of the lens based on the variation of the light spot area in the RGB images. Subsequently, we estimate the unknown thermophysical parameters of the system by fitting the theoretical heat transfer model to temperature data recorded by the thermal camera. A model-based controller is then proposed to regulate the temperature variation of the target point. To demonstrate the potential applications of the developed system, we design an experimental study in which the robot creates a desired pattern on a wood plane by feedback-controlled solar energy concentration.

5.2.3 Geometric Optics

In this section, we demonstrate the modeling and analysis of the developed ray tracing simulation, which guides the design of the solar angle estimation and the robot path generation algorithms. The Fresnel lens adopted in this paper relies on the same working prin-

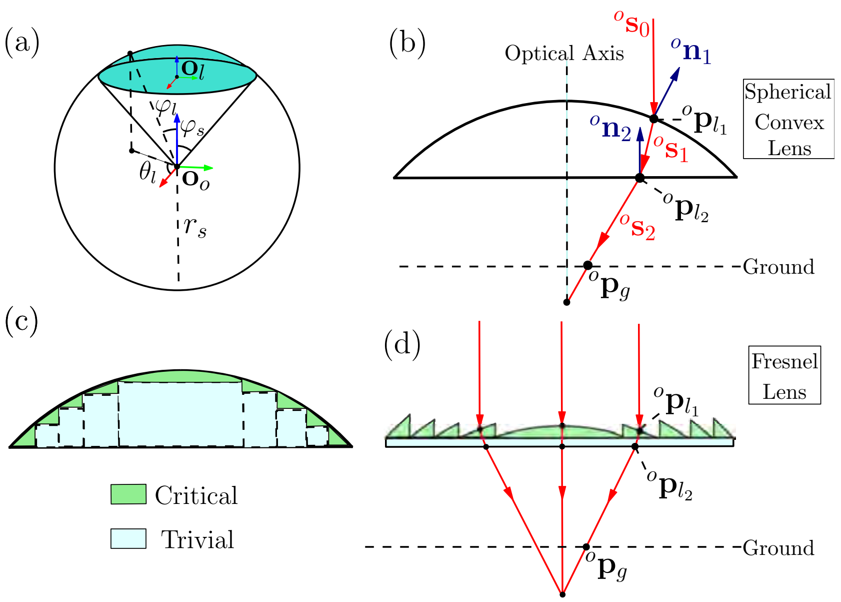


Figure 5.2: Ray tracing diagrams of the plano-convex lens and the Fresnel lens.

ciples as the traditional spherical convex lens; Yet, it is more cost- and energy-efficient ascribed to its delicate design [74]. As depicted in Figure 5.2 (a), the curvature design of the convex lens takes a notional sphere with center $\mathbf{O}_o$ and radius r_s as reference. The interested region (colored in turquoise) is bounded by a polar angle $\varphi_l \in [0, \varphi_{lb}]$ and an azimuthal angle $\theta_l \in [0, 2\pi]$. As shown in Figure 5.2 (b), the incident and refracted rays are denoted by ${}^o\mathbf{s}_0$, ${}^o\mathbf{s}_1$, ${}^o\mathbf{s}_2$. Note the ray only refracts at the boundary surfaces, the inner part of the lens (colored in light blue in Figure 5.2 c) merely contributes to concentration, thus can be removed to alleviate energy loss. The Fresnel lens exploits this feature as shown in Figure 5.2 (d), where the critical curved surface (colored in green) is divided into pieces while maintaining the curvature and moved towards the flat boundary surface.

Following the conventions in [75], the origin of the optics frame $(xyz)_o$ is set at the center of the notional sphere. We set the axes of $(xyz)_o$ to be parallel to the axes of $(xyz)_l$. Note that the thickness of the Fresnel lens is small (2 mm in this study), we fairly assume the sun ray enters and leaves the lens at the same point (See Figure 5.2 d) to simplify the derivation. Consequently, the positional vector of an incident point ${}^o\mathbf{p}_{l_1}$ and its corresponding leaving point ${}^o\mathbf{p}_{l_2}$ are denoted by:

$$\begin{aligned} {}^o\mathbf{p}_{l_1} = {}^o\mathbf{p}_{l_2} &= \begin{bmatrix} p_{l_{1x}} & p_{l_{1y}} & p_{l_{1z}} & 1 \end{bmatrix}^\mathsf{T} \\ &= \begin{bmatrix} r_s\mathrm{S}\varphi_l\mathrm{C}\theta_l & r_s\mathrm{S}\varphi_l\mathrm{S}\theta_l & r_s\mathrm{C}\varphi_{lb} & 1 \end{bmatrix}^\mathsf{T} \end{aligned} \tag{5.1}$$

According to the nomenclature of the spherical coordinate, the unit directional vector of the incident ray is denoted by:

$${}^o\mathbf{s}_0 = \begin{bmatrix} \mathrm{S}\varphi_0\mathrm{C}\theta_0 & \mathrm{S}\varphi_0\mathrm{S}\theta_0 & \mathrm{C}\varphi_0 & 0 \end{bmatrix}^\mathsf{T} \tag{5.2}$$

where $\varphi_0 \in [\frac{\pi}{2}, \pi], \theta_0 \in [0, 2\pi]$ are the incident polar angle and the azimuthal angle of the ray. For an incident point ${}^{o}\mathbf{p}_{l_1}$ on the convex boundary, and an incident point ${}^{o}\mathbf{p}_{l_2}$ on the flat boundary, the corresponding active surface normals are:

$$\begin{aligned} {}^{o}\mathbf{n}_{l_1} &= \begin{bmatrix} \mathrm{S}\varphi_l \mathrm{C}\theta_l & \mathrm{S}\varphi_l \mathrm{S}\theta_l & \mathrm{C}\varphi_l & 0 \end{bmatrix}^{\mathsf{T}} \\ {}^{o}\mathbf{n}_{l_2} &= \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^{\mathsf{T}} \end{aligned} \tag{5.3}$$

The incident angle between ${}^{o}\mathbf{s}_0$ and the convex boundary normal ${}^{o}\mathbf{n}_1$ is calculated by $\gamma_1 = \cos^{-1}(-{}^{o}\mathbf{s}_0 \cdot {}^{o}\mathbf{n}_{l_1})$. To make a compact derivation, we define the following variable:

$$\omega_1 = N_1 \mathrm{C}\gamma_1 - \sqrt{1 - N_1^2 + (N_1 \mathrm{C}\gamma_1)^2} \tag{5.4}$$

where $N_1 = \xi_0/\xi_1$, and ξ_0, ξ_1 are the refractive index (with respect to a specific wavelength of the light) of the air and the material of the lens, respectively. According to [75], the refracted ray ${}^{o}\mathbf{s}_1$ is derived as:

$${}^{o}\mathbf{s}_1 = \begin{bmatrix} s_{1_x} & s_{1_y} & s_{1_z} & 0 \end{bmatrix}^{\mathsf{T}} = \omega_1 {}^{o}\mathbf{n}_{l_1} + N_1 {}^{o}\mathbf{s}_0 \tag{5.5}$$

Similarly, the refracted ray ${}^{o}\mathbf{s}_2$ can be calculated. After the ray exits the flat boundary, it travels rectilinearly until it hits the target plane with the surface normal ${}^{o}\mathbf{n}_g$. We denote the intersection point

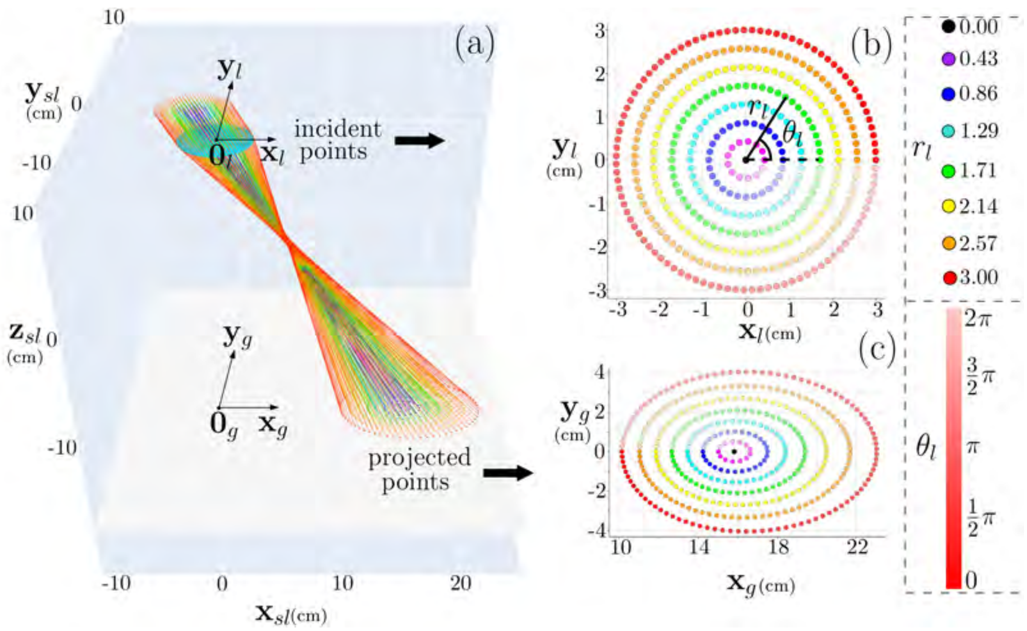


Figure 5.3: Illustration of the geometrical characteristics of the projected points.

of ${}^{o}\mathbf{s}_2$ with the target plane as:

$$ {}^{o}\mathbf{p}_g = {}^{o}\mathbf{p}_{l_1} + \lambda {}^{o}\mathbf{s}_2 \tag{5.6} $$

where λ is an unknown scaling parameter. For a point ${}^{o}\mathbf{p}_{go} = \begin{bmatrix} 0 & 0 & d & 1 \end{bmatrix}$ in the target plane, we can calculate λ and derive the expression of ${}^{o}\mathbf{p}_g$ by solving $({}^{o}\mathbf{p}_g - {}^{o}\mathbf{p}_{go}) \cdot {}^{o}\mathbf{n}_g = 0$. In conclusion, ${}^{o}\mathbf{p}_g$ is coupled with 3 factors:

1. The position of an incident point on the lens, which is denoted by φ_l and θ_l. In the next section, we will discuss a representative group of φ_l and θ_l to simplify the computation process.

2. The direction of the incident ray ${}^{o}s_0$, which is denoted by φ_0 and θ_0. In this study, φ_0 is related to the elevation angle and θ_0 is related to the azimuthal angle of the sun, which varies within a day.
3. The configuration of the lens plane, which is denoted by ${}^{o}\mathbf{n}_g$ and d and can be controlled by the robot motion.

5.2.4 Optical Simulation

To analyse and exploit the coupling between the robot motion, sun orientation variation, and the optical performance of the Fresnel lens, we developed a visualization tool based on the aforementioned derivation with the *Python Plotly* library. An illustration of our ray tracing programme is shown in Figure 5.3, in which the solar incident angles are set to $\varphi_0 = 5\pi/6$, $\theta_0 = 0$, the ground normal is set to ${}^{o}\mathbf{n}_g = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^{\mathsf{T}}$, and the lens surface is set to be parallel to the target plane. To generate a set of uniformly distributed incident points, we pick a series of lengths r_{l_i} that is evenly distributed from 0 to $r_s \mathrm{C}\varphi_{lb}$ (the lens radius); Different values of r_{l_i} are represented by different colors. For each circle of radius r_{l_i}, we select sampling points with an equal arc length; The values of the polar angle θ_l of the selected points are visualized by the color saturation variation. As shown in Figure 5.3 (a), the parallel sun ray first converges then diverges after passing through the lens. Figure 5.3 (b) depicts the incident points on the circular lens surface, and Figure 5.3 (c) depicts the projected points on target plane.

Through the results of the simulation, we observe that for a group of incident points ${}^{o}\mathbf{p}_{l_1}$ that has the same r_{l_i}, their corresponding projected points ${}^{o}\mathbf{p}_{g}$ will form a (distorted) closed contour on the target plane. Furthermore, the hierarchy relationship with respect to r_{l_i} between different groups of ${}^{o}\mathbf{p}_{l_1}$ is preserved after refraction[1]. Concretely, for the group of incident points with the largest r_l (depicted in red in Figure 5.3 b), their corresponding projected points also form the outermost closed contour in Figure 5.3 (c). These neat correspondences allow us to only analysis a representative group of incident points with the same r_{l_i}. Relying on this observation, we select the geometric characteristics (centroid, shape, orientation.) of the **outermost boundary** of the projected light spot in the following sections, as it can be easily extracted from the RGB image feedback.

---

[1] Except for the region that is very close to the focal point.

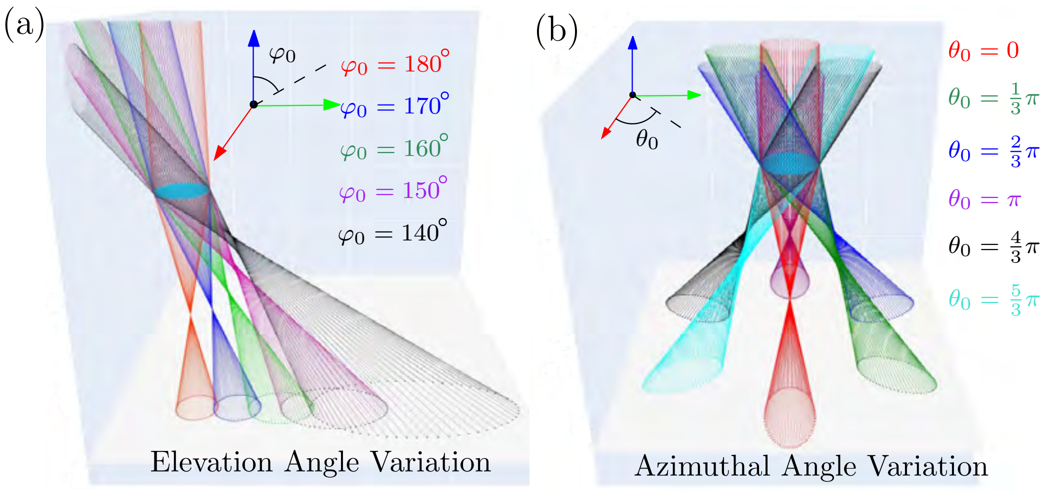


Figure 5.4: The effect of the incident polar angle φ_0 and θ_0 on the light spot while the lens plane is parallel to the ground plane.

5.2.5 Solar Angle Estimation

In the previous literature, the sun orientation is either determined manually or by an electrical compass. To automate the robot solar concentration process, as the system is equipped with an RGB camera to observe the ArUco marker, we propose an image-based method to estimate the sun orientation. Note that the ground projection point ${}^{o}\mathbf{p}_g$ is coupled with the sun orientation and the lens-ground relative configuration. The lens-ground configuration is acquired from the ArUco marker and the robot kinematics. If the relationship between the ${}^{o}\mathbf{p}_g$ and φ_0, θ_0 can be determined with the developed simulation, we can estimate the solar angle with an image feedback.

Consider an RGB image feedback, we segment the image pixels that represent the projected light spot by first transforming it into gray scale, then applying a threshold function to find the region of interest. Our target is to find the corresponding 3D coordinates of these interested image pixels with respect to the optics frame $(xyz)_o$, such that they can be used as a feedback target to the estimation algorithm.

In homogeneous coordinate systems, consider an interested image pixel ${}^{i}\mathbf{p} \in \mathbb{R}^3$, and its corresponding unknown 3D coordinate ${}^{w}\mathbf{p} = \begin{bmatrix} x_w & y_w & z_w & 1 \end{bmatrix}^\intercal$. According to the pinhole camera model [49], the relationship between ${}^{w}\mathbf{p}$ and ${}^{i}\mathbf{p}$ is given by:

$$s\,{}^{i}\mathbf{p} = \mathbf{A} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \end{bmatrix} {}^{w}\mathbf{p} \tag{5.7}$$

where s is an scaling factor, $\mathbf{A}$ is the camera's intrinsic matrix, and $\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \end{bmatrix} \in \mathbb{R}^{3\times 4}$ is the camera's extrinsic matrix $\mathbf{T}_{c_1}^{w}$. Since the projected points are in the same target plane, ${}^{i}\mathbf{p}$ and ${}^{w}\mathbf{p}$ satisfies a homography transformation. Accordingly, we can derive the following equation:

$$\begin{bmatrix} x_w & y_w & a \end{bmatrix}^\mathsf{T} = \left(\mathbf{A}\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix}\right)^{-1} {}^{i}\mathbf{p} \tag{5.8}$$

where a is an scaling factor. Subsequently, we can calculate ${}^{w}\mathbf{p}_g^* \in \mathbb{R}^3$, the position of the interested feedback points with respect to the frame $(xyz)_o$ by the following equation:

$${}^{o}\mathbf{p}_g^* = \mathbf{T}_{c_1}^{w}\mathbf{T}_{l}^{c_1}\mathbf{T}_{o}^{l}\begin{bmatrix} x_w & y_w & 0 & 1 \end{bmatrix}^\mathsf{T} \tag{5.9}$$

As the feedback information from the image is transformed to xyz_o, we can formulate it as a simulation-based servoing problem to estimate the unknown solar orientation. As mentioned in the previous section, we select the outermost boundary as the representative contour. We denote the center of the representative ellipse on the target plane by ${}^{o}\mathbf{c}_g = \begin{bmatrix} x_c & y_c \end{bmatrix}^\mathsf{T}$, and the feature vector is formed as

$$\mathbf{y} = \begin{bmatrix} d_c & \theta_c \end{bmatrix}^\mathsf{T} = \begin{bmatrix} \left(x_c^2 + y_c^2\right)^{\frac{1}{2}} & \tan^{-1}\left(y_c/x_c\right) \end{bmatrix}^\mathsf{T} \tag{5.10}$$

The selection of the feature vector is based on the observation of the developed simulation, and an example is shown in Figure 5.4. When the lens plane is parallel to the target plane, the variation of φ_0 affects

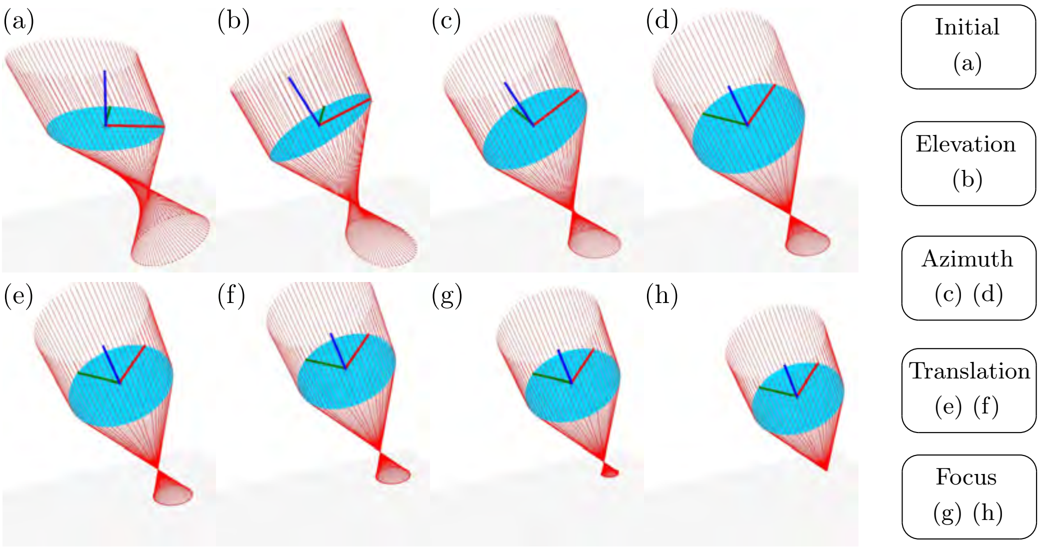


Figure 5.5: The simulation of lens reorientation and focus process while the elevation angle $\varphi_0 = 5/6\pi$ and the azimuthal angle $\theta_0 = 1/6\pi$

d_c, and the variation of θ_0 affects θ_c. We denote the estimation of the solar orientation at the step t by φ_t, θ_t, and its corresponding feature vector acquired from the simulation is denoted by $\mathbf{y}_t = \begin{bmatrix} d_t & \theta_t \end{bmatrix}^\mathsf{T}$. The reason that We select the polar angle of the ellipse instead of its shape or orientation will be discussed in Section 5.3.1. From the equation 5.9, we can compute the feedback target feature vector $\mathbf{y}^*$. We then design the following update rule of the solar angle estimation to asymptotically reduce the error $\mathbf{e} = \mathbf{y}_t - \mathbf{y}^*$:

$$\begin{bmatrix} \Delta\varphi_t & \Delta\theta_t \end{bmatrix}^\mathsf{T} = -\lambda_1 \mathbf{A}_t^+ (\mathbf{y} - \mathbf{y}^*) \tag{5.11}$$

where $\lambda_1 > 0$ is a gain, and the interaction matrix in computed from the simulation as:

$$\mathbf{A}_t = \begin{bmatrix} \dfrac{d_t(\varphi_t + \delta\varphi, \theta_t) - d_t}{\delta\varphi} & \dfrac{d_t(\varphi_t, \theta_t + \delta\theta) - d_t}{\delta\theta} \\ \dfrac{\theta_t(\varphi_t + \delta\varphi, \theta_t) - \theta_t}{\delta\varphi} & \dfrac{\theta_t(\varphi_t, \theta_t + \delta\theta) - \theta_t}{\delta\theta} \end{bmatrix} \tag{5.12}$$

The above interaction matrix can be further simplified. We find that if the RGB image is captured when the lens surface is parallel to the target plane, d_c is coupled with φ_0 independently, and θ_c is coupled with θ_0 independently. In this situation, the $\mathbf{A}_t$ is diagona. Note that we should set the initial guess of $\varphi_l \neq 0$. Otherwise, the initial $\mathbf{A}_t$ will become singular. The estimation algorithm terminates until $|\mathbf{e}|$ is reduced to a small predefined value, and the final estimation of the solar angles are denoted by $\widehat{\varphi}_0$, $\widehat{\theta}_0$.

5.2.6 Lens Configuration

In this section, we introduce the computation of the lens configuration to focus the sun rays to a target point on the target plane. The solar energy power incident to the lens surface is calculated as:

$$w_l = d_s \int_{A_l} \cos\beta \, \mathrm{d}A_l \tag{5.13}$$

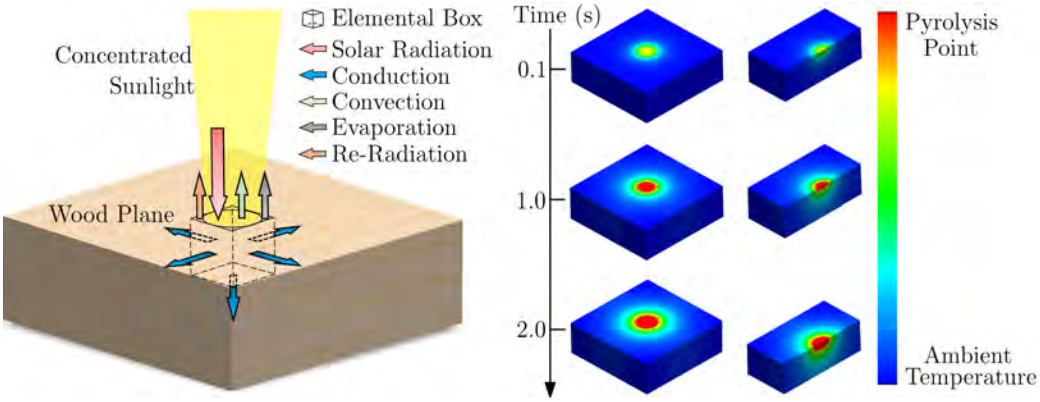


Figure 5.6: A schematic of the heat transfer model. Rmember to change box to block.

where d_s is the solar energy density, A_l is the lens surface, and β is the angle between the target plane normal and the lens surface normal. To maximize the solar energy power collected by the lens, it is straightforward that the lens surface normal should be aligned with the estimated direction of the incident sunlight. Under this constrain, consider a target point ${}^w\mathbf{p}_g^*$ on the target plane, we first transform it to the $(xyz)_o$ frame by ${}^o\mathbf{p}_g^* = \mathbf{T}_l^o\mathbf{T}_{c1}^l\mathbf{T}_w^{c1}\,{}^w\mathbf{p}_g^*$ since the orientation estimated in the previous section is in $(xyz)_o$. The desired position of the lens center is then computed as:

$$ {}^o\mathbf{p}_l^* = {}^o\mathbf{p}_g^* - d_{lp}\,{}^o\mathbf{s}_0 \tag{5.14} $$

where ${}^o\mathbf{s}_0$ is defined by equation 5.2 and the estimated solar orientation $\widehat{\varphi}_0$, $\widehat{\theta}_0$, and d_{lp} is the distance between the lens surface and the target plane. The lens center position in $(xyz)_r$ is computed as ${}^r\mathbf{p}_l^* = \mathbf{T}_l^r \mathbf{T}_o^l {}^o\mathbf{p}_l^*$. The lens is then controlled to reach the target position and orientation according to the robot inverse kinematics. In Figure 5.5, the robot solar concentration tasks is divided into several subprocess including the alignment of the elevation angle, the alignment of the azimuth angle, the translation of the lens to ${}^r\mathbf{p}_l^*$, and the variation of d_{lp}. By reorienting the lens surface towards the direction of the incident sun ray, not only the collected solar power is maximized, the optical concentration performance of the lens is also improved. Therefore in this paper, we always align the lens surface normal with the sun orientation.

Nevertheless, we argue that it is not strictly required to include the reorientation process for robot solar concentration tasks in other scenarios. As shown in Figure 5.5 (a), (b), and (c), the lens also concentrates solar energy when the lens orientation is not well aligned. Relying on the developed simulation model, the robot solar concentration control could be achieved when the robot DOF is limited (0 or 1 controllable rotation angle). Moreover, visiting Figure 5.4 (a), we found that the distance between the lens surface and the optimal focal point (where the cross section of the sun ray array is minimal) is decreasing while the angle difference between the lens surface normal and the solar elevation angle increases. This unique optical characteristic could be useful for scenarios where the working space of the robot is limited.

5.2.7 Heat Transfer Modeling

In this section, we model the heat transfer process induced by the concentrated sunlight. As depicted in Figure 5.6, the sunlight concentrates to an interested region of area A_c. A controlled volume (a cubic with a fixed edge length of d_v) can be extracted for heat transfer modeling. Note that d_v should be selected small enough to satisfy the Biot number criteria [56], which is set to $d_v = 5\,\mathrm{mm}$ in this paper. Note that A_c is coupled with the robot configuration $\mathbf{x}$. Although the direct analytical form of the mapping function $f(\mathbf{x}) = A_c$ is not derived, it can be numerically computed from the developed simulation or from the RGB image feedback.

We denote the solar energy intensity by q_s, which is available if the longitude, latitude, and temporal factors are provided, and the constant that quantifies the energy loss caused by the light transmission through the lens by κ.The heat flux inflow to the interested

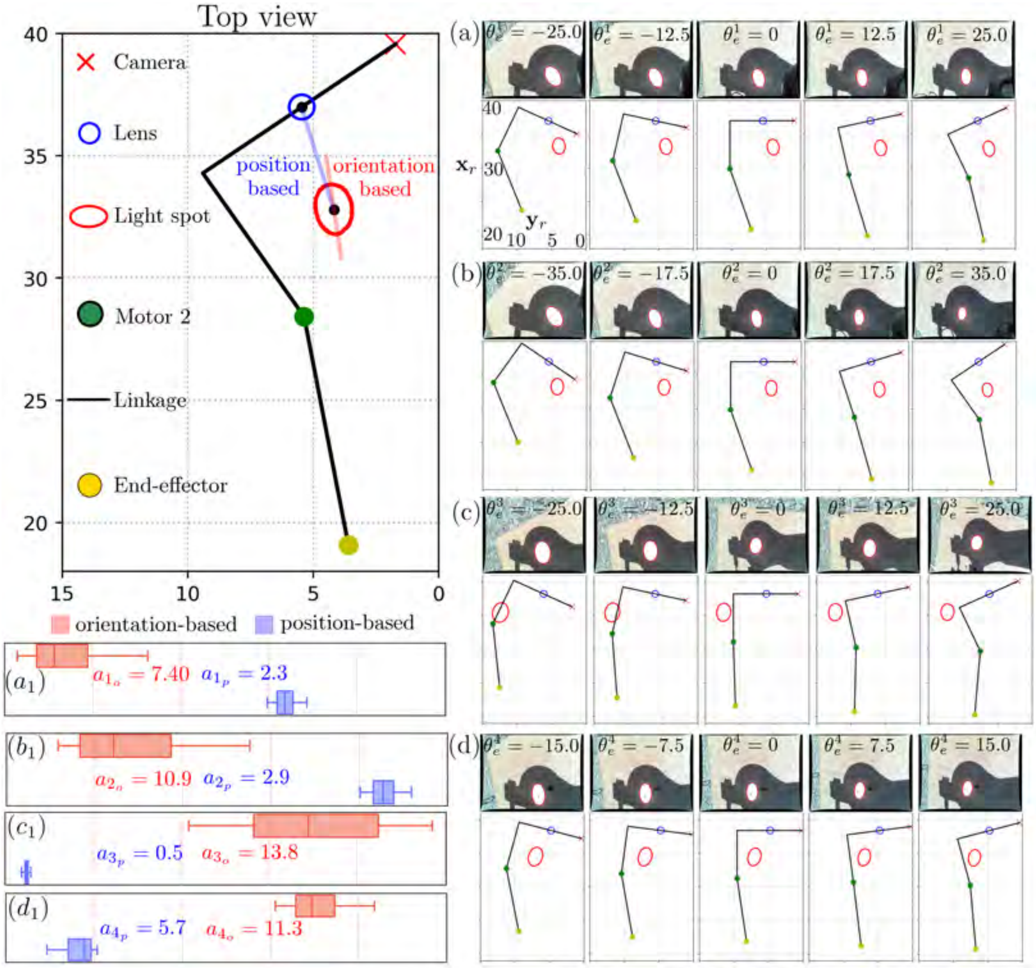


Figure 5.7: Experiments comparing the performance of the solar orientation estimation utilizing different geometric features. The positional-based method outperforms the orientation-based method.

controlled volume's upper surface is computed as:

$$q_{in} = \frac{d_v^2}{A_c(\mathbf{x})} \kappa q_s \tag{5.15}$$

If $d_v^2 > A_c(\mathbf{x})$, then $q_{in} = \kappa q_s$. For the heat flux outflow from the controlled volume, it is composed 3 modes of heat transfer, including heat radiation, convection, and conduction. We denote the temperature of the controlled volume by T_0 and the temporal change of T_0 by $v = \mathrm{d}T_0/\mathrm{d}t$. According to the energy conservation law, the relationship between the net heat flux and v is described by:

$$v = (q_{in} - q_{rad} - q_{conv} - q_{cond}) d_v^2 / mc \tag{5.16}$$

where m, c are the mass and the specific heat of the controlled volume, respectively. Analogous to the formulation in [76], we rewrite the above equation as:

$$v = \lambda_0 g(\mathbf{x}) - \lambda_1 T_0^4 - \lambda_2 T_0 + \lambda_3 \tag{5.17}$$

where $g(\mathbf{x}) = 1/A_c(\mathbf{x})$, and $\lambda_{0,1,2,3}$ are positive time invariant variables if we assume the thermophysical properties of the controlled volume (expect its temperature T_0) remain unchanged, and the prop-

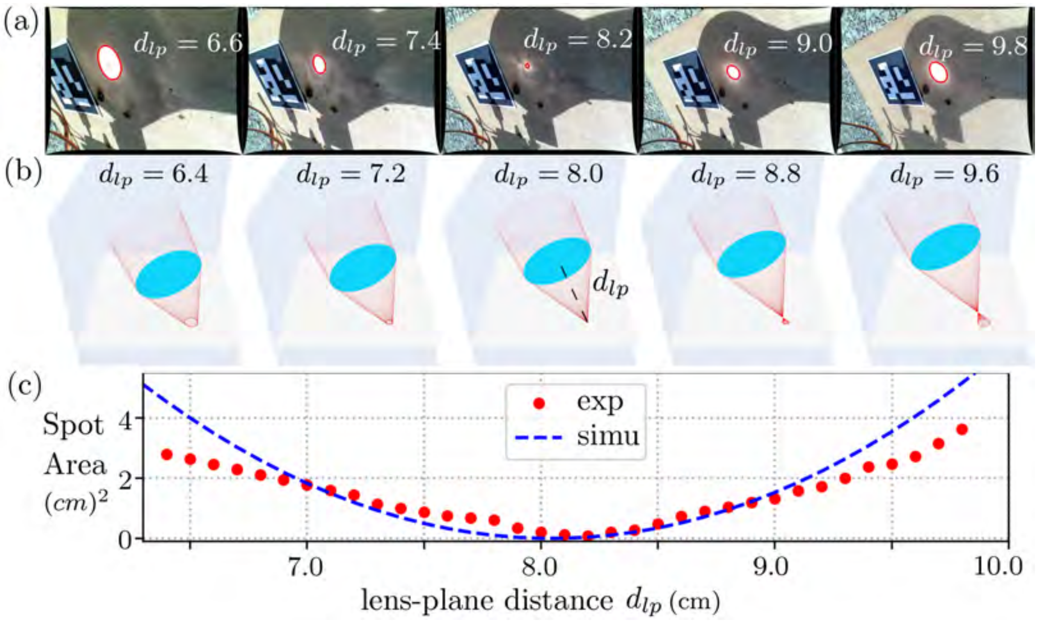


Figure 5.8: Image-based focal length validation.

erties of the environment (e.g., the ambient temperature and the wind speed) are also constants within a short period of time.

5.3 Results

5.3.1 Solar Orientation Estimation Feature Selection

In this section, we compare the performance of the solar orientation estimation methods exploiting different geometric features of the light spot through experiments. We collected in total 400 RGB images in four experiments. For each experiment, we set the lens plane to be parallel to the target plane, and varied its azimuthal angle within a short period of time (less than three minutes). The estimations of solar orientation exploiting different fractures were computed and recorded for each configuration. The purpose of varying the robot configuration is to evaluate the robustness of the estimation algorithm, as in practice the initial robot configuration with respect to the solar orientation is unknown set arbitrarily.

Within a short period of time, the variation of the real sun orientation is negligible. Consequently, the estimations are expected to be the same. Theoretically, the position, orientation and shape of the projected light spot should be invariant to the lens azimuthal angle variation (See Figure 5.4 b again). In practice, note that the movement of the robot end-effector changes the relative configuration between the camera and the projected light spot, which results in a change of the size, orientation, and position of the light spot in the feedback RGB image. If the the robot kinematics and the homography transformation from the ArUco maker are accurate, and the optical performance of the used Fresnel lens is in accord with the simulation, the projected light spot should also be invariant with respect to the robot configuration. Nevertheless, due to the inherent inaccuracy in a robot system, the geometrical features of the pro-

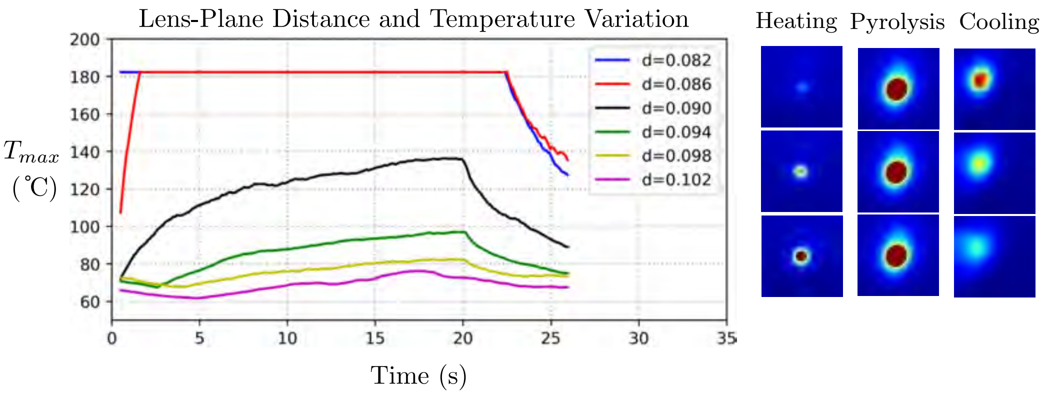


Figure 5.9: Temperature variation for different lens-plane distances.

jected light spot varies with the robot configuration. In conclusion, we select the positional based features (as shown in Equation 5.10) that produces more stable estimation results comparing to the orientation based features through experiments.

Concretely, some representative processes of the four experiments are demonstrated in Figure 5.7 (a) - (d), and the corresponding estimation distributions are shown in Figure 5.7 (a_1) - (d_1). To visualize the configuration variation of the robot during the experiments, we also depict the top view of the robot system and annotate essential components. Note that these illustrative figures are generated referring to the **real** configurations of the robot, and the depicted light spot also comes from the **real** image feedback by transforming image pixels to the points in xyz_r utilizing the method introduced in Section 5.2.5. As shown in 5.7 (a_1) - (d_1), the variance of the position-based method is much smaller than the variance of the orientation-based method.

5.3.2 Focal Length Calibration

The intensity of the concentrated solar energy is related to the projection area of the light spot on the target plane A_c. This coupling is described in 5.15, which is essential for the temperature control in the robot solar concentration tasks. In section 5.2.7, we mentioned that the mapping function between the robot configuration and A_c can be numerically computed or from the RGB image. In this section, we compare the results of A_c computed from simulation and from the image feedback in experiments. We align the orientation of the lens towards the estimated sun orientation, then control the robot to focus the sun on a target point with varying lens-plane distance d_{lp} from 6.4 cm to 9.8 cm with a uniform step of 0.1 cm. As shown in Figure 5.8 (a), the contour of the projected light spot is extracted and depicted in red. The image pixels of the spot is then transformed into points in $(xyz)_r$ coordinate, and the area of the formed contour is computed accordingly. The relationship between A_c computed from experiment and d_{lp} is depicted in red dots in Figure 5.8 (c).

We compute A_c in the simulation with the same sun orientation and d_{lp} variation as in the experiment, and depict the result in Figure 5.8 (c) in blue dashed lined. We claim that coupling between A_c and d_{lp} is similar in the experiment and simulation.

5.3.3 Temperature Variation and Plane-Lens Distance

The thermal camera provides feedback temperature of the heating process. We conduct a series of experiments with varying distance between the lens and the target plane and a fixed heating time. As shown in Figure 5.9, when the distance is close to the focal length, the surface temperature rapidly increases to the charring temperature. Once the wood pyrolysis occurs, the temperature stays at the equilibrium, but the charred area spreads both horizontally and vertically. When the distance gradually increases, the speed of temperature increment and the steady state temperature both decrease. Once the solar concentration period ends (20 seconds), the lens moves upwards by 3 centimeters and the temperature gradually decreases due to heat dissipation and convection. This experiment shows that the temperature of the target point could be effectively regulated by the robot configuration and the duration of the heating process.

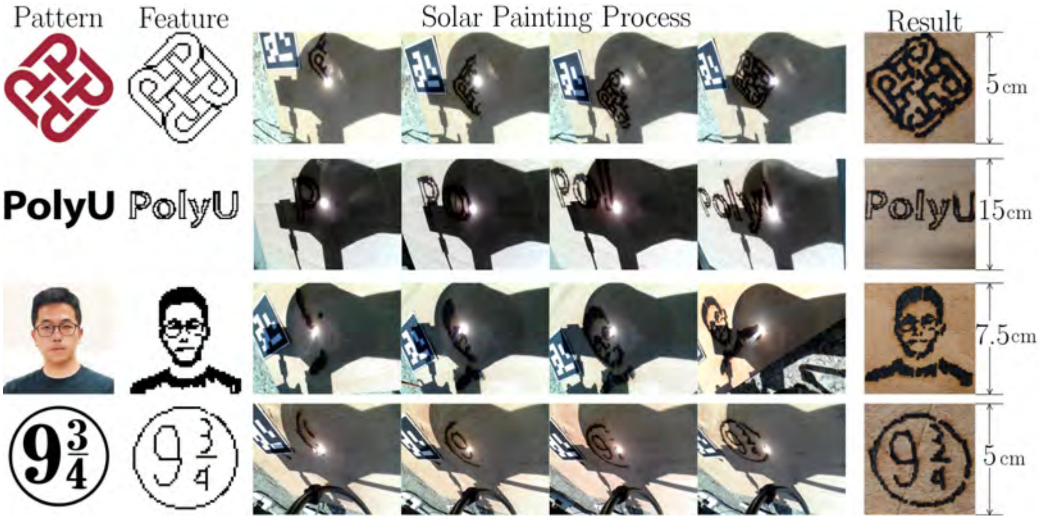


Figure 5.10: Results of the solar painting experiments. The robot path is planned according to the feature points extracted from the target pattern.

5.3.4 Solar Painting Experiments

In this section, we introduce the method of creating a desired pattern on the wood plane with the developed robot system. The target pattern is composed of a series of points, and the positions of which are generated by transforming the feature pixel coordinates extracted from an image using image processing techniques to the world coordinate according to the preset size of the desired pattern. Consider an image that is composed of a_{im} rows and y_{im} columns. The working space on the target plane is a square of length l_w cm. The scaling factor is then computed as $\omega_s = l_w/(\max(a_{im}, y_{im}) \cdot r_p)$, where r_p is the minimum radius of the projected light spot.The target pattern is resized according to the scaling factor ω_s, and the feature points is then extracted. For each image pixel (p_x, p_y) from the scaled image, its corresponding coordinate in the world coordinate is then computed as $(p_x r_p + d_x, p_y r_p + d_y, 0)$, where (d_x, d_y) represents the starting point of the pattern on the target plane. The world coordinate is then grouped according to the x coordinate and ordered according to the y coordinate. We report four experiments with various patterns and different scaling factors to validate the proposed method. As shown in Figure 5.10, the target patterns include logos, letters, and portraits.

Chapter 6

Conclusion and Future Work

In this thesis, we present a through formulation of a new robotic temperature control technique based on heat radiation to automatically regulate the temperature of objects. There are three main aspects to be considered to solve a robot thermal servoing problem: How does the robot perceive the temperature of the target and the environment, what physical model should be developed to describe the dominant heat transfer process, and what control law should be designed to achieve the temperature regulation task.

We first demonstrate the development of a multimodal camera system that consists of a thermal camera and a depth camera. We implemented a customized calibration algorithm to integrate the output from the two cameras, such that the system could provide fused geometric and temperature information, the RGB-D-T point cloud, of its environment in real-time. We then illustrate the mathematical formulation of a fundamental “Fire to hand” radiation-based robot thermal servoing problem in which the target objects are heated by a radiative heat source. Next, a model-based controller and an adaptive controller that online estimates the unknown thermophysical parameters are designed. Then, a series of experiments are reported and evaluated to validate the performance of two controllers. The target feasibility problem is also discussed in depth qualitatively and quantitatively. Finally, we reported a specific “Fire in hand” robot thermal servoing problem, the solar painting robot. The developed system can actively track the sun orientation based on the RGB image feedback, and use a Fresnel lens to concentrate the solar energy to regulate the temperature of a target point. In this experiment, the thermal camera not only responsible for provides the value of the feedback temperature of the target point, but also gives reference information for determining the true position of the focused light spot. We validate that the integration of thermal servoing with other visual servoing is feasible and valuable for the robot system control design.

Yet, the proposed algorithm has some limitations. We mainly consider the effect of heat radiation when analyzing the heat transfer model. In practical situations, heat conduction, convection, and radiation usually coexist and couple with each other. For the cases where the heat conduction is not negligible (e.g., objects contacting non-adiabatic surfaces, objects with large thickness, etc.) or the cases where the heat convection is non-negligible (fluid speed or temperature is high), more sophisticated heat transfer models need to be developed. We also assume the temperature distribution over the heat source and the object is uniform, which is invalid for industrial heating applications where the object is much larger than the heating device.

6.1 Contribution

In this thesis, we present a compact formulation of the radiation-based robotic TS algorithm. The main contributions are summarized as:

- We propose a novel motion control method to automatically regulate the temperature of objects.
- We develop an efficient algorithm to compute radiation-based

thermal interaction matrices in real-time.

- We report detailed experimental studies to validate the proposed theory.

To the best of our knowledge, this is the first time that temperature control task has been formalized as a servoing problem.

6.2 Future Work

For future work, we would like to explore more advanced integration of thermal servoing with visual servoing algorithms. This multimodal perceptual and control capability is essential for developing robust robotic temperature control systems in complex scenarios, such as service tasks in human environments and intelligent industrial manufacturing. Several questions remain unanswered at present. For the cases where heat conduction, convection and radiation are in the same order of magnitude, how to quantitatively evaluate the trade-off between a delicate physical model and the computation time of each feedback control step is critical to the practical implementation of the algorithm. The uniform temperature regulation is commonly required in industrial heat processes. It could be achieved by either utilizing heat sources that output a relative uniform heat flux (low speed hot air) or by designing adaptive algorithms which control multiple small-scale heat sources to complete the task cooperatively. For the solar concentration robot, we believe there is abundant room in investigating its practical application scenarios. It could be integrated with a mobile platform (even a drone) and a reflection system to create a steady and sustainable heat flux at an arbitrary target area in 3D space, which is rather difficult to achieve with traditional static heating systems. We encourage interested readers to work along these open directions.

6.3 Research Output

- L. Hu, D. Navarro-Alarcon, A. Cherubini and M. Li. On Radiation-Based Thermal Servoing: New Models, Controls and Experiments. IEEE Transactions on Robotics, 2021.

- M. Muddassir, D. Gomez, L. Hu, S. Chen and D. Navarro-Alarcon. Paint with the Sun: A Thermal-Vision GuidedRobot to Concentrate Solar Energy into Surfaces, 2021.

- T. Zhang, L. Hu, L. Li and D. Navarro-Alarcon. Towards a

Multispectral RGB-IR-UV-D Vision System —Seeing the Invisible in 3D, IEEE ROBIO, 2021.

- L. Hu, A. Duan, M. Li, A. Cherubini, Q. Zhou, and D. Navarro-Alarcon. Paint with the Sun: A Thermal-Vision GuidedRobot to Concentrate Solar Energy into Surfaces, 2022.

Chapter 7

Appendix

7.1 Online Estimation of the Interaction Vector l

The numerical solvers [62] for definite integrals is used to approximate the interaction matrix in real time. The essence of numerical integration is to divide the complex integrand into small subsections and approximate each subsection with a polynomial which is easy to integrate. We introduce a commonly used numerical approximation method of definite integrals, *Composite Simpson's rule*, which approximates the subsection by quadratic polynomials. The general form of Composite Simpson's rule is as follows:

$$\int_c^d g(x)\,\mathrm{d}x \approx \frac{h}{3}\left[g\left(x_0\right)+2\sum_{j=1}^{n/2-1} g\left(x_{2j}\right)+4\sum_{j=1}^{n/2} g\left(x_{2j-1}\right)+g\left(x_n\right)\right] \tag{7.1}$$

where n is the number of subintervals, $x_j = c + jh$ for $j = 0, 1, ..., \mathrm{n}-1, \mathrm{n}$ with $h = (d-c)/n$; in particular, $x_0 = c$ and $x_n = d$.

7.2 Polynomial Fitting with sliding window

For a single object, we denote its temperature as T_t where $t = t_1, t_2, \cdots, t_n$ and t_n represents the n_{th} sampling time. The size of the sliding window is set to be 10 data samples. When $n < 10$, the estimation is at the initialization stage and the robot will not move. When $n \geq 10$, we denote the sample points in the sliding window by $\mathbf{T}_s = \begin{bmatrix} T_{t_{n-9}} & T_{t_{n-8}} & \cdots & T_{t_n} \end{bmatrix}^\intercal$. We then fit 10 sample points in the sliding window to the polynomial of order 3 as:

$$\widehat{\mathbf{T}}_s = \mathbf{P}\mathbf{c}^{t_n} = \begin{bmatrix} (t_{n-9})^3 & (t_{n-9})^2 & (t_{n-9}) & 1 \\ (t_{n-8})^3 & (t_{n-8})^2 & (t_{n-8}) & 1 \\ \vdots & \vdots & \vdots & \\ (t_n)^3 & (t_n)^2 & (t_n) & 1 \end{bmatrix} \begin{bmatrix} c_3^{t_n} \\ c_2^{t_n} \\ c_1^{t_n} \\ c_0^{t_n} \end{bmatrix} \tag{7.2}$$

where $\mathbf{c}^{t_n}$ is the coefficient vector of the polynomial at sample time t_n. To minimize $\|\mathbf{T}_s - \widehat{\mathbf{T}}_s\|_2$, the coefficients are computed as $\mathbf{c}^{t_n} = \mathbf{P}^{+}\mathbf{T}_s$. Then, the temperature and temperature rate of a single object can be estimated as:

$$T_{t_n} = c_3^{t_n}(t_n)^3 + c_2^{t_n}(t_n)^2 + c_1^{t_n}(t_n) + c_0^{t_n} \tag{7.3}$$

$$v_{t_n} = 3c_3^{t_n}(t_n)^2 + 2c_2^{t_n}(t_n) + c_1^{t_n} \tag{7.4}$$

We follow the same procedure to estimate the temperature and temperature rates for other objects, and $\mathbf{T}_2$ and $\mathbf{v}$ can be estimated accordingly.

Bibliography

[1] M. L. May, “Insect thermoregulation,” *Annual review of entomology*, vol. 24, no. 1, pp. 313–349, 1979.

[2] E. Yıldız, A. M. Başol, and M. P. Mengüç, “Segregated modeling of continuous heat treatment furnaces,” *Journal of Quantitative Spectroscopy and Radiative Transfer*, p. 106 993, 2020.

[3] J. Mlynek, R. Knobloch, and R. Srb, “Optimization of a heat radiation intensity and temperature field on the mould surface.,” in *ECMS*, 2016, pp. 425–431.

[4] M. Sweetland, J. H. Lienhard V, and A. H. Slocum, “A convection/radiation temperature control system for high power density electronic device testing,” *Journal of Electronic Packaging*, vol. 130, no. 3, 2008.

[5] J. De Backer, G. Bolmsjö, and A.-K. Christiansson, “Temperature control of robotic friction stir welding using the thermoelectric effect,” *The International Journal of Advanced Manufacturing Technology*, vol. 70, no. 1-4, pp. 375–383, 2014.

[6] J.-H. Kim and B. Y. Lattimer, “Real-time probabilistic classification of fire and smoke using thermal imagery for intelligent firefighting robot,” *Fire Safety Journal*, vol. 72, pp. 40–49, 2015.

[7] M. Ho and J. P. Desai, “Towards a mri-compatible meso-scale sma-actuated robot using pwm control,” in *2010 3rd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics*, IEEE, 2010, pp. 361–366.

[8] G. Muscato, F. Bonaccorso, L. Cantelli, D. Longo, and C. D. Melita, “Volcanic environments: Robots for exploration and measurement,” *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 40–49, 2012.

[9] A. Cammarata, “Optimized design of a large-workspace 2-dof parallel robot for solar tracking systems,” *Mechanism and machine theory*, vol. 83, pp. 175–186, 2015.

[10] C. P. Bechlioulis, S. Heshmati-Alamdari, G. C. Karras, and K. J. Kyriakopoulos, “Robust image-based visual servoing with prescribed performance under field of view constraints,” *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 1063–1070, 2019.

[11] D. Navarro-Alarcon, H. Yip, Z. Wang, *et al.*, “Automatic 3D manipulation of soft objects by robotic arms with adaptive deformation model,” *IEEE Trans. Robot.*, vol. 32, no. 2, pp. 429–441, 2016.

[12] A. Cherubini and F. Chaumette, “Visual navigation of a mobile robot with laser-based collision avoidance,” *Int. Journal of Robotics Research*, vol. 32, no. 2, pp. 189–205, 2013.

[13] Q. Li, O. Kroemer, Z. Su, F. F. Veiga, M. Kaboli, and H. J. Ritter, "A review of tactile information: Perception and action through touch," *IEEE Transactions on Robotics*, vol. 36, no. 6, pp. 1619–1634, 2020.

[14] A. Magassouba, N. Bertin, and F. Chaumette, "Audio-based robot control from interchannel level difference and absolute sound energy," in *Proc. IEEE Int. Conf. Intelligent Robots and Systems*, 2016, pp. 1992–1999.

[15] F. Rahbar, A. Marjovi, P. Kibleur, and A. Martinoli, "A 3-d bio-inspired odor source localization and its validation in realistic environmental conditions," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2017, pp. 3983–3989.

[16] A. Dena, K. Ahiska, and N. Aouf, "Image based visual servoing for landmine detection using quadrotors," in *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, IEEE, 2020, pp. 527–532.

[17] R. Gade and T. B. Moeslund, "Thermal cameras and applications: A survey," *Machine Vision and Applications*, vol. 25, pp. 245–262, 2014.

[18] F. Chaumette and S. Hutchinson, "Visual servo control. Part I: Basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, 2006.

[19] R. W. Lee, "Thermal tolerances of deep-sea hydrothermal vent animals from the northeast pacific," *The Biological Bulletin*, vol. 205, no. 2, pp. 98–101, 2003.

[20] C.-C. Tsai, R. A. Childers, N. N. Shi, *et al.*, "Physical and behavioral adaptations to prevent overheating of the living wings of butterflies," *Nature communications*, vol. 11, no. 1, pp. 1–14, 2020.

[21] L. Hines, V. Arabagi, and M. Sitti, “Shape memory polymer-based flexure stiffness control in a miniature flapping-wing robot,” *IEEE transactions on robotics*, vol. 28, no. 4, pp. 987–990, 2012.

[22] E. Benli, R. L. Spidalieri, and Y. Motai, “Thermal multisensor fusion for collaborative robotics,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 3784–3795, 2019.

[23] Y. Cao, B. Xu, Z. Ye, *et al.*, “Depth and thermal sensor fusion to enhance 3d thermographic reconstruction,” *Opt. Express*, vol. 26, no. 7, pp. 8179–8193, Apr. 2018.

[24] Y. He and R. Yang, “Eddy current volume heating thermography and phase analysis for imaging characterization of interface delamination in cfrp,” *IEEE Trans. on Industrial Informatics*, vol. 11, no. 6, pp. 1287–1297, 2015.

[25] W. W.-L. Lai, K.-K. Lee, and C.-S. Poon, “Validation of size estimation of debonds in external wall’s composite finishes via passive infrared thermography and a gradient algorithm,” *Construction and Building Materials*, vol. 87, pp. 113–124, 2015, ISSN: 0950-0618.

[26] A. Cherubini and D. Navarro-Alarcon, “Sensor-based control for human-robot collaboration: Fundamentals, challenges and opportunities,” *Front. in Neurorobotics (in press)*, vol. 1, no. 1, pp. 1–21, 2020. [Online]. Available: https://arxiv.org/pdf/2007.02067.

[27] W. Binrui, J. Yinglian, X. Hong, and W. Ling, “Temperature control of pem fuel cell stack application on robot using fuzzy incremental pid,” in *2009 Chinese Control and Decision Conference*, IEEE, 2009, pp. 3293–3297.

[28] M. Muddassir, D. Gomez, L. Hu, S. Chen, and D. Navarro-Alarcon, “Robotics meets cosmetic dermatology: Development of a novel vision-guided system for skin photo-rejuvenation,” *IEEE/ASME Trans. Mechatronics (under review)*, vol. 1, no. 1, pp. 1–15, 2020. [Online]. Available: `https://arxiv.org/pdf/2005.10462`.

[29] D. McKemy, “Temperature sensing across species,” *Eur. J. Physiol.*, vol. 454, pp. 777–791, 2007.

[30] A. Imdoukh, A. Shaker, A. Al-Toukhy, D. Kablaoui, and M. El-Abd, “Semi-autonomous indoor firefighting uav,” in *Int. Conf. on Advanced Robotics*, 2017, pp. 310–315.

[31] M. Fu, W. Weng, W. Chen, and N. Luo, “Review on modeling heat transfer and thermoregulatory responses in human body,” *Journal of Thermal Biology*, vol. 62, pp. 189–200, 2016, Modeling bioheat transfer processes and thermoregulatory responses, ISSN: 0306-4565.

[32] R. Lee, “Thermal tolerances of deep-sea hydrothermal vent animals from the northeast pacific,” *Biological Bulletin*, vol. 205, no. 2, pp. 98–101, 2003.

[33] L. Natale, G. Metta, and G. Sandini, “Development of auditory-evoked reflexes: Visuo-acoustic cues integration in a binocular head,” *Rob. Auton. Syst.*, vol. 39, no. 2, pp. 87–106, 2002.

[34] G. Arechavaleta, J. Laumond, H. Hicheur, and A. Berthoz, “An optimality principle governing human walking,” *IEEE Trans. Robot.*, vol. 24, no. 1, pp. 5–14, 2008.

[35] S. Na, Y. Qiu, A. E. Turgut, *et al.*, “Bio-inspired artificial pheromone system for swarm robotics applications,” *Adaptive Behavior*, pp. 1–21, 2020.

[36] E. Yıldız, A. M. Başol, and M. P. Mengüç, “Segregated modeling of continuous heat treatment furnaces,” *Journal of Quantitative Spectroscopy and Radiative Transfer*, p. 106 993, 2020.

[37] S. H. Han and D. Chang, “Radiative slab heating analysis for various fuel gas compositions in an axial-fired reheating furnace,” *International journal of heat and mass transfer*, vol. 55, no. 15-16, pp. 4029–4036, 2012.

[38] T. Morgado, P. J. Coelho, and P. Talukdar, “Assessment of uniform temperature assumption in zoning on the numerical simulation of a walking beam reheating furnace,” *Applied Thermal Engineering*, vol. 76, pp. 496–508, 2015.

[39] M. Muddassir, D. Gomez, S. Chen, L. Hu, and D. Navarro-Alarcon, “Robotics meets cosmetic dermatology: Development of a novel vision-guided system for skin photo-rejuvenation,” *arXiv preprint arXiv:2005.10462*, 2020.

[40] M. Akuto and E. Iwase, “An origami heat radiation fin for use in a stretchable thermoelectric generator,” *Micromachines*, vol. 11, no. 3, p. 263, 2020.

[41] J.-H. Kim, J. W. Starr, and B. Y. Lattimer, “Firefighting robot stereo infrared vision and radar sensor fusion for imaging through smoke,” *Fire Technology*, vol. 51, no. 4, pp. 823–845, 2015.

[42] M. Evita, A. Zakiyyatuddin, S. Seno, R. Kumalasari, H. Lukado, and M. Djamal, “Development of a robust mobile robot for volcano monitoring application,” in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1572, 2020, p. 012 016.

[43] F. Gravot, A. Haneda, K. Okada, and M. Inaba, “Cooking for humanoid robot, a task that needs symbolic and geometric reasonings,” in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, IEEE, 2006, pp. 462–467.

[44] S. Vidas, P. Moghadam, and M. Bosse, “3d thermal mapping of building interiors using an rgb-d and thermal camera,” in *2013 IEEE International Conference on Robotics and Automation*, IEEE, 2013, pp. 2311–2318.

[45] Z. Zhang, “Camera calibration with one-dimensional objects,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 7, pp. 892–899, 2004.

[46] W. Nakagawa, K. Matsumoto, F. de Sorbier, *et al.*, “Visualization of temperature change using rgb-d camera and thermal camera,” in *European Conference on Computer Vision*, Springer, 2014, pp. 386–400.

[47] D. Borrmann, A. Nüchter, M. Đakulović, *et al.*, “A mobile robot based system for fully automated thermal 3d mapping,” *Advanced Engineering Informatics*, vol. 28, no. 4, pp. 425–440, 2014.

[48] D. Kurz, “Thermal touch: Thermography-enabled everywhere touch interfaces for mobile augmented reality applications,” in *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, IEEE, 2014, pp. 9–16.

[49] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[50] S. Vidas, R. Lakemond, S. Denman, C. Fookes, S. Sridharan, and T. Wark, "A mask-based approach for the geometric calibration of thermal-infrared cameras," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 6, pp. 1625–1635, 2012.

[51] N. Kim, Y. Choi, S. Hwang, K. Park, J. S. Yoon, and I. S. Kweon, "Geometrical calibration of multispectral calibration," in *2015 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, IEEE, 2015, pp. 384–385.

[52] T. Zhang, L. Hu, L. Li, and D. Navarro-Alarcon, "Towards a multispectral rgb-ir-uv-d vision system–seeing the invisible in 3d," *arXiv preprint arXiv:2108.08494*, 2021.

[53] T. L. Bergman, F. P. Incropera, D. P. DeWitt, and A. S. Lavine, *Fundamentals of heat and mass transfer*. John Wiley & Sons, 2011.

[54] M. F. Modest, *Radiative heat transfer*. Academic press, 2013.

[55] I. Lienhard and H. John, *A heat transfer textbook*. phlogiston press, 2005.

[56] T. L. Bergman, F. P. Incropera, D. P. DeWitt, and A. S. Lavine, *Fundamentals of heat and mass transfer*. John Wiley & Sons, 2011.

[57] M. Vujičić, N. Lavery, and S. Brown, "View factor calculation using the monte carlo method and numerical sensitivity," *Communications in numerical methods in Engineering*, vol. 22, no. 3, pp. 197–203, 2006.

[58] J. C. Chai, J. P. Moder, and K. C. Karki, "A procedure for view factor calculation using the finite-volume method," *Numerical Heat Transfer: Part B: Fundamentals*, vol. 40, no. 1, pp. 23–35, 2001.

[59] S. C. Mishra, A. Shukla, and V. Yadav, “View factor calculation in the 2-d geometries using the collapsed dimension method,” *International communications in heat and mass transfer*, vol. 35, no. 5, pp. 630–636, 2008.

[60] E. M. Sparrow, *Radiation heat transfer*. Routledge, 2018.

[61] H. Flanders, “Differentiation under the integral sign,” *The American Mathematical Monthly*, vol. 80, no. 6, pp. 615–627, 1973.

[62] P. Virtanen, R. Gommers, T. E. Oliphant, *et al.*, “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,” *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: `10.1038/s41592-019-0686-2`.

[63] D. Navarro-Alarcon and Y.-H. Liu, “Fourier-based shape servoing: A new feedback method to actively deform soft objects into desired 2-d image contours,” *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 272–279, 2018.

[64] G. P. Tolstov, *Fourier series*. Courier Corporation, 2012.

[65] M. Vidyasagar, *Nonlinear systems analysis*. SIAM, 2002.

[66] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.

[67] P. T. Inc. “Collaborative data science.” (2015), [Online]. Available: `https://plot.ly`.

[68] S. C. Endres, C. Sandrock, and W. W. Focke, “A simplicial homology algorithm for lipschitz optimisation,” *Journal of Global Optimization*, vol. 72, no. 2, pp. 181–217, 2018.

[69] J. Mlynek, R. Knobloch, and R. Srb, “Optimization of a heat radiation intensity and temperature field on the mould surface.,” in *ECMS*, 2016, pp. 425–431.

[70] Z.-Y. Guo, H.-Y. Zhu, and X.-G. Liang, “Entransy—a physical quantity describing heat transfer ability,” *International Journal of Heat and Mass Transfer*, vol. 50, no. 13-14, pp. 2545–2556, 2007.

[71] I. Stamatescu, I. Făgărăşan, G. Stamatescu, N. Arghira, and S. S. Iliescu, “Design and implementation of a solar-tracking algorithm,” *Procedia Engineering*, vol. 69, pp. 500–507, 2014.

[72] P. A. Plonski, P. Tokekar, and V. Isler, “Energy-efficient path planning for solar-powered mobile robots,” *Journal of Field Robotics*, vol. 30, no. 4, pp. 583–601, 2013.

[73] A. M. Andrew, “Multiple view geometry in computer vision,” *Kybernetes*, 2001.

[74] W. Xie, Y. Dai, R. Wang, and K. Sumathy, “Concentrated solar energy applications using fresnel lenses: A review,” *Renewable and Sustainable Energy Reviews*, vol. 15, no. 6, pp. 2588–2606, 2011.

[75] P. D. Lin, *Advanced geometrical optics*. Springer, 2017.

[76] L. Hu, D. Navarro-Alarcon, A. Cherubini, and M. Li, “On radiation-based thermal servoing: New models, controls and experiments,” *IEEE Transactions on Robotics*, 2021.