Joan Domènech Masferrer

Design and implementation of a control algorithm for the Adaptive Solar Facade

Master’s Thesis

ITA – Architecture and Building Systems
Swiss Federal Institute of Technology (ETH) Zurich

Examiner:
Prof. Dr. Arno Schlüter

Supervisors:
Prageeth Jayathissa, Stefan Caranovic

Zürich, March 18, 2018
Abstract

The Adaptive Solar Facade (ASF) comprises the benefits of both dynamic shading and Building Integrated Photovoltaics to assess the building energy demand and improve the inhabitant experience. Unfortunately, the adaptive control system to be implemented in the real prototypes is yet to be developed. Moreover, the current computation times are still excessively high for their proper integration in a physical facade. This thesis presents a control algorithm that has been successfully implemented in a prototype using an Arduino board and a Raspberry Pi. The complex behavior of the facade is resembled using Python scripts, which strategically divide the algorithm in subprocesses. By organizing appropriately the execution of the scripts, the current computational period is reduced 200 times. The present control algorithm supports any arbitrary configuration of the facade, and is capable to process whether a direct command of a module, or the computation of the ASF optimal position with respect to the user requirements. The implementation of the control system takes advantage of the integrated Digital Motion Processors of the Inertial Measurement Units, which relieve the system from processing raw data. The performance of the algorithm is compared with the former simulation framework. The results show a precise interpretation of the hourly changes of the weather excepting some eventual large errors related with the diffuse irradiation. The sensitivity analyses of the sensors demonstrate their suitability for the control process, showing deviations of 0.05° in rest position and maximum drifts of 1.4° when abrupt movements are performed. In a worst case scenario, the complete movement of one panel would take 20 seconds. For steady-state purposes, this timing is acceptable, but for features with high interaction with the inhabitant, a parallel control of the panels must be introduced.
Acknowledgements

I would like to thank Dr. Prageeth Jayathissa and Stefan Caranovic for putting their full confidence on me. Especially, I would like to express my gratitude to Prageeth for his nice coaching and continuous feedback, and Stefan for the enormous help facing the hardware problems of the facade. I would also like to thank Pr. Dr. Arno Schlüter for giving me the opportunity to work in such exciting thesis, and Bratislav Svetozarevic for giving me a helping hand facing the low-level control of the actuators. I would also like express my sincerest gratitude for the warm welcome that the whole chair has given to me, and I would also like to say thank you to the persons who have been accompanying me everyday and making me smile: Linus, Laura, Viviana, Justin, Thomas, Carlota, Mariana and Victor. Last but not least, all this work would have never been possible with the unconditional support and love of my parents, without which I would not have made it this far.
Contents

List of Acronyms .................................................. v
List of Figures ......................................................... x
Nomenclature ............................................................. xi
List of Tables ............................................................ xii

1 Introduction .......................................................... 1
  1.1 Motivation and Literature Review ................................. 1
    1.1.1 Adaptive systems as a new integrated building solution ...... 1
    1.1.2 The Adaptive Solar Facade .................................... 2
  1.2 Overview of the control problem .................................. 5
  1.3 Problem Statement ............................................... 8
  1.4 Objectives of Research ......................................... 8
  1.5 Thesis Outline ................................................... 9

2 Methodology ......................................................... 10
  2.1 Design of the control strategy .................................. 10
  2.2 Simulation framework enhancement ............................... 13
    2.2.1 Sun position .................................................. 14
    2.2.2 Shading pattern .............................................. 17
    2.2.3 Radiation ..................................................... 21
    2.2.4 PV supply ................................................... 28
    2.2.5 RC building simulator ....................................... 30
  2.3 Main control algorithm .......................................... 31
  2.4 Implementation to prototypes ................................... 34
    2.4.1 The control scheme .......................................... 35
    2.4.2 Low-level control algorithm of the prototype ............... 39

3 Results ............................................................. 42
  3.1 Simulation Framework review .................................... 42
    3.1.1 Sun position validation ..................................... 42
    3.1.2 Shading pattern and radiation ................................ 43
    3.1.3 PV Supply, RC building simulator and main control algorithm .... 52
    3.1.4 Simulation performance ..................................... 55
  3.2 Performance on prototypes ...................................... 55
## List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASF</td>
<td>Adaptive Solar Facade</td>
</tr>
<tr>
<td>BIPV</td>
<td>Building Integrated Photovoltaics</td>
</tr>
<tr>
<td>PV</td>
<td>Photovoltaics</td>
</tr>
<tr>
<td>CIGS</td>
<td>Copper Indium Gallium Selenide</td>
</tr>
<tr>
<td>SRA</td>
<td>Soft Robotic Actuator</td>
</tr>
<tr>
<td>RC</td>
<td>Resistor-Capacitor</td>
</tr>
<tr>
<td>EPW</td>
<td>EnergyPlus weather</td>
</tr>
<tr>
<td>LST</td>
<td>Local Solar Time</td>
</tr>
<tr>
<td>LT</td>
<td>Local Time</td>
</tr>
<tr>
<td>ICM</td>
<td>Integrated Control Module</td>
</tr>
<tr>
<td>IMU</td>
<td>Inertial Measurement Unit</td>
</tr>
<tr>
<td>I²C</td>
<td>Inter-integrated Circuit</td>
</tr>
<tr>
<td>GPIO</td>
<td>General Purpose Input Output</td>
</tr>
<tr>
<td>COP</td>
<td>Coefficient of Performance</td>
</tr>
<tr>
<td>DMP</td>
<td>Digital Motion Processor&lt;sup&gt;TM&lt;/sup&gt;</td>
</tr>
<tr>
<td>FIFO</td>
<td>First-in first-out</td>
</tr>
</tbody>
</table>
List of Figures

1.1 ASF prototype in the House of Natural Resources [1] . . . . . . . . . . . . 3
1.2 ASF actuation system and fixation [2] ............................................. 4
1.3 SRA pneumatic actuation [1]. Three symmetrically distributed chambers are allocated between two discs. The disc on the top acts as the holding unit for the PV panels, while the one at the bottom is set as the structural support. When a chamber inflates, it expands causing the upper disc to rotate. ................................................................. 4
1.4 Potential features of the ASF. From the comfort regime side, the facade must be fast-responsive and address inhabitants’ commands precisely. Regarding the autonomous regime side, the inhabitant interaction is less influential, but the processing time of complex features must be enough efficient to not slow the responsiveness too much. ........................................ 5
1.5 ASF clustering example for multi-purpose features [1]. ....................... 6
1.6 ASF control problem overview. Inside the red frame, the scope of the thesis is placed. An algorithm to command ASF configurations taking into account customer’s needs, weather and building data, and the dynamic model of the SRA is about to be developed. The implementation of this algorithm to a microprocessor in cooperation with a microcontroller is also covered. The sensing elements that close the feedback loop in the regulating unit will also be explored. ................................................................. 7
2.1 Overview of the usage of subprocesses in every ASF potential feature . . . 11
2.2 Generalist input-output diagram of the control algorithm. The 5 subprocesses mainly require parameters that define the ASF geometry, the building characteristics, and the weather. There’s also a relation of dependency between some subprocesses. The main control algorithm will take in account the user input, execute the necessary subprocesses, and command the position of the panels taking in account the sensing data. . 12
2.3 ASF simulation framework for optimal net building energy consumption. The framework definition has been conveniently adapted with the control strategy by differentiating the 5 subprocesses. ................................. 14
2.4 Comparison between sun angle conventions. Instead of setting the azimuth origin clockwise due North, the proposed sun-facade convention defines its origin perpendicular to the facade normal. Azimuth values clockwise until \(180^\circ\) are positive while negative values are counterclockwise. The relative definition of the sun also applies with the panels.

2.5 Main positions for and ASF panel. The geometrical definition is done with four vertices. The movement is simulated defining two axes of rotation. The pitch axis is fixed in space while the yaw axis depends of the pitch rotation.

2.6 Main boolean operations of Pyclipper [3]. To extract the mutual shading, a boolean difference will be used. To compute the global shadow lying on the window, a boolean intersection will be executed.

2.7 Shading pattern script output: single-cluster example for 18 modules.

2.8 Shading pattern script output: 18-cluster example for 18 modules.

2.9 Sun-ASF panel angle convention: south-facade case example. The dashed lines define the panel position following the sun-observer standard convention from 2.4a. For consistency, the panel orientation is defined following the sun-facade convention from 2.4b.

2.10 First step for sky view factor approach: panel tilt vs wall effect. \(\beta\) are the angles to be evaluated in Liu-Jordan equation. The least obtained value from the formula, hence the most restricting, is kept.

2.11 Second step for the sky view factor approach: azimuth effect. When the panel azimuth value exceeds a maximum value defined by its distance with the wall edge, then the interaction with the wall is evaluated with Liu-Jordan equation adapted for a half hemisphere.

2.12 Third step for sky view factor approach: surrounding panels effect. The involved part of the sky dome is sliced in five parts. Depending on the azimuth position of the panel and the spacing between panels, every part will have a defined weight. Notice that the closer a sight interception occurs, the higher effect it has on the sky view factor.

2.13 Experimental power losses for CIGS panels. A power loss because of self-shading can be defined with the percentages of longitudinal and lateral mutual shading, i.e. the width and the height of the shade divided by the respective panel side measurement [4] (edited).

2.14 5R1C model for a single zone office space [5]

2.15 General panel identification strategy for the ASF. The relative position of a panel within the facade can be quickly defined with its row and column numbers, counted from the upper-left part of the ASF.
2.16 Work-flow example of the main control algorithm. The configuration of the facade is set sequentially by establishing a priority between the several kinds of inputs from the user, which are, in order: direct control of panels, comfort features and autonomous features. In case some panels are strictly set to a position directly, the following comfort features will be restricted. Likewise, the autonomous features are applied to the rest of panels which have not been yet set.

2.17 ASF prototype for the GlassTec Trade Fair

2.18 General diagram of the control scheme. The Raspberry Pi is connected with the Arduino Board via USB serial communication. The Raspberry Pi is the unit responsible to compute the simulations and send the commands to the Arduino. The Arduino is responsible of the sensor readings and the control of the triad of chambers of every panel via I2C protocol, as well as the direct opening and closing of the valves for pressure inlet and outlet.

2.19 I2C bus data protocol with ICMs. Example of command decodification for an ASF of 8 modules.

2.20 I2C bus data protocol standard model. The communication is established between a master device and only one slave device at a time. The serial data line carries the the messages following the emisor-receptor structure, pulling the data from a precise register address. The serial clock line synchronizes all the systems and the frequency of data transmission.

2.21 2-angles-3-pressures mapping strategy for the SRA. Every chamber individual inflation or deflation defines a main direction of actuation regarding the pitch and yaw rotation. The triad of directions define 3 regions. The set of pitch and yaw rotations within every region can only be achieved when the chamber with the same number is totally deflated. Note that a positive pitch increases the panel tilt, while a positive yaw rotates it clockwise.

3.1 Sun position comparison: Script vs SunCalc.org.

3.2 Qualitative comparison between Grasshopper shading pattern and script output. Single-cluster example. Panel neutral position: $\theta = 0^\circ$, $\psi = 0^\circ$.

3.3 Qualitative comparison between Grasshopper shading pattern and script output. Single-cluster example. Panel neutral position: $\theta = 0^\circ$, $\psi = 0^\circ$.

3.4 Qualitative comparison between Grasshopper shading pattern and script output. Single-cluster example. Panel netrual position: $\theta = 0^\circ$, $\psi = 0^\circ$.

3.5 Qualitative comparison between Grasshopper shading pattern and script output. Single-cluster example. Panel maximum degree of actuation: $\theta = 60^\circ$, $\psi = 40^\circ$. Sun position: $\alpha_s = 37^\circ$, $\gamma_s^{ASF} = 72^\circ$. 
3.6 Qualitative comparison between Grasshopper shading pattern and script output. Single-cluster example. Panel maximum degree of actuation: $\theta = 60^\circ$, $\psi = -40^\circ$. Sun position: $\alpha_s = 46^\circ$, $\gamma_{sASF} = 57^\circ$.

3.7 Grasshopper simple model for radiation comparison. 54 modules separated 300 mm from the glazing surface. South-oriented. Single-cluster case.

3.8 Incident Direct Irradiation on facade during a summer day: script vs Grasshopper model comparison. The images show the incident irradiation along the day for every extreme combination of panel pitch $\theta$ and the panel yaw $\psi$. Note: open position is $\theta = -30^\circ$, $\psi = 0^\circ$. Closed position is $\theta = 60^\circ$, $\psi = 0^\circ$.

3.9 Incident Diffuse Irradiation on facade during a summer day: script vs Grasshopper model comparison. The images show the incident irradiation along the day for every extreme combination of panel pitch $\theta$ and panel yaw $\psi$. Note: open position is $\theta = -30^\circ$, $\psi = 0^\circ$. Closed position is $\theta = 60^\circ$, $\psi = 0^\circ$.

3.10 Incident Direct Irradiation on the window behind the facade during a winter day: script vs Grasshopper model comparison. The images show the incident irradiation along the day for every extreme combination of panel pitch $\theta$ and panel yaw $\psi$. Note: open position is $\theta = -30^\circ$, $\psi = 0^\circ$. Closed position is $\theta = 60^\circ$, $\psi = 0^\circ$.

3.11 Incident Direct Irradiation on facade panels during a winter day: script vs Grasshopper model comparison. The images show the incident irradiation along the day for 6 combinations of panel pitch $\theta$ and panel yaw $\psi$. Note: open position is $\theta = -30^\circ$, $\psi = 0^\circ$. Closed position is $\theta = 60^\circ$, $\psi = 0^\circ$.

3.12 Incident Diffuse Irradiation on facade during a winter day: script vs Grasshopper model comparison. The images show the incident irradiation along the day for every extreme combination of panel pitch $\theta$ and panel yaw $\psi$. Note: open position is $\theta = -30^\circ$, $\psi = 0^\circ$. Closed position is $\theta = 60^\circ$, $\psi = 0^\circ$.

3.13 Incident Direct Irradiation on the window behind the facade during a winter day: script vs Grasshopper model comparison. The images show the incident irradiation along the day for every extreme combination of panel pitch $\theta$ and panel yaw $\psi$. Note: open position is $\theta = -30^\circ$, $\psi = 0^\circ$. Closed position is $\theta = 60^\circ$, $\psi = 0^\circ$.

3.14 ASF optimal configurations in terms of building net energy consumption for a winter day. Single-cluster example.

3.15 ASF optimal configurations in terms of building net energy consumption for a summer day. Single-cluster example.

3.16 Samples per second read from the Arduino board during 10 minutes.

3.17 Stability analysis for the pitch and yaw rotations during one hour in rest position.
3.18 Drift analysis for the pitch and yaw rotations during one hour. The panels are eventually moved manually to check the robustness of the sensors . . . 57
3.19 First run procedure: calibration of regions with experimental values. The red lines represent the boundaries of regions one and two proposed from the 2-angles-3-pressures model assuming a perfectly symmetrical SRA. . . 58
3.20 Experimental regulation of one panel . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59
A.1 General overview of the folder structure of the GitHub repository. . . . . 66
Nomenclature

\( C \)  Cooling demand of the building behind the facade
\( H \)  Heating demand of the building behind the facade
\( L \)  Lighting demand of the building behind the facade
\( PV \)  Electricity production of the ASF
\( h \)  Half diagonal of the squared PV panel
\( w \)  Hour angle
\( \delta \)  Declination angle
\( \alpha_s \)  Sun altitude
\( \gamma_s \)  Sun azimuth (following standard sun-observer convention)
\( \gamma_{ASF} \)  Panel azimuth relative to the facade (following sun-facade convention)
\( \gamma_{ASF}^\alpha \)  Sun azimuth relative to the facade (following sun-facade convention)
\( \gamma_{N,ASF} \)  Azimuth of the normal of the ASF holding surface (following sun-observer convention)
\( \theta \)  Panel pitch rotation
\( \theta_i \)  Angle of incidence
\( \theta_z \)  Sun zenith
\( \beta \)  Panel tilt/slope/elevation
\( \psi \)  Panel yaw rotation
\( \phi \)  Geographical latitude
\( \phi_s \)  Sun vector
\( \lambda \)  Geographical longitude
\( G_{tot} \)  Total incident irradiation
\( G_{dir} \)  Direct incident irradiation
\( G_{dif} \)  Diffuse incident irradiation
\( G_{ref} \)  Ground-reflected incident irradiation
\( G_{dni} \)  Direct beam irradiation
\( G_{iso} \)  Diffuse isotropic irradiation
\( G_{cir} \)  Circumsolar diffuse irradiation
\( G_{dif,h} \)  Horizontal diffuse irradiation
\( G_{glo,h} \)  Global horizontal diffuse irradiation
\( \varepsilon \)  Sky clearness
\( \Delta \)  Sky brightness
List of Tables

2.1 Perez experimental sky diffuse irradiance model coefficients [6] . . . . . . 27

3.1 Case study parameters for the House of Natural Resources . . . . . . . . 53
3.2 Execution time of the main control algorithm features. Comparison of performance between computer and Raspberry Pi. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
Introduction

1.1 Motivation and Literature Review

1.1.1 Adaptive systems as a new integrated building solution

The building sector is responsible for as much as 30% of the final energy consumption when residential and services sub-sectors are considered [7, 8]. Furthermore, the global greenhouse gases emissions from buildings are estimated to be up to 25% worldwide [9]. Since the global population is still expected to increase up to 9.7 billion in 2050 [10], addressing our efforts to improve significantly the performance of existing and new buildings is a key strategy to reduce the environmental footprint of the sector in the future.

New integrated building technologies range from envelopes and insulation to lighting, heating, and cooling systems. Among them, Building Integrated Photovoltaics are gaining presence in the construction of new buildings as a principal or auxiliary source of electricity. In addition, developments regarding efficiency and costs of flexible thin-film photovoltaic technologies have brought new design possibilities [11, 12]. The improved aesthetics, the lower cost, and the wider distribution place the BIPV as one of the main future research segments in photovoltaics [13].

On the other hand, the dynamic shading systems have also gained popularity because of their capacity to control direct and indirect radiation into the building. Furthermore, Rossi et al. [14] pointed that their integration with adaptive distributed robotics in architectural design has the potential to improve building energy performance while simultaneously increasing occupant comfort. The introduction of robotic features changes completely the conception of the building from a rigid structure to a dynamic system with the ability to adapt itself to the changing environments.

Rossi et al. proposed the combination of both dynamic shading and BIPV technologies to assess the building energy demand and improve the inhabitant experience. It was firstly approached by Jayathissa et al. [15] and it supposed the start of the Adaptive Solar Facade project [1]. The ASF is a promising new integrated building solution that comprises the benefits from the dynamic shading systems and the BIPV, it is currently being developed within the A/S group at ETH Zürich.
1.1.2 The Adaptive Solar Facade

The Adaptive Solar Facade is a modular, lightweight photovoltaic shading system composed of CIGS panels, that can be easily installed on any surface of new or existing buildings. The modules can be independently actuated in both altitude and azimuth orientation, enabling the system to react to internal user needs and external weather conditions (see Figures 1.1 and 1.2).

The actuation system is fully pneumatic, and the orientation of every panel can be controlled with a two-axis soft robotic actuator, developed by Svetozarevic et al. [2]. By inflating or deflating the three cylindrical chambers symmetrically distributed around the center of the SRA, the CIGS panel orientation can be set with two degrees of freedom (see Figure 1.3). Svetozarevic et al. modeled the dynamic behavior of these elements and developed a low-level control algorithm based on a 2-angles-to-3-pressures mapping strategy. Their findings show the suitability of the SRA for solar tracking applications and a viable component for dynamic building facades, measuring power gains of 36% with respect to static panels and a predicted power gain of 73% if the azimuth range could be expanded from ±20° to ±45°.

The promising results of the analyses led the ASF project to the implementation of its first prototype at the House of Natural Resources at ETH Zürich in 2015. Case studies show that the utilization of the ASF in optimal control conditions, can compensate for 37% of the annual energy demand of the office behind the facade, and compared to an equivalent static BIPV shading system, the adaptive facade reduces the building energy demand by 20% and increases PV electricity generation by 7% [16].

In parallel to that, a methodology for simulating the electricity generation of the facade combined with the energy consumption analysis of the building behind it, was conducted within the parametric Rhino [17]/Grasshopper [18] and DIVA [19]/EnergyPlus [20] environments, respectively. From that point, it became possible to determine the optimum hourly configuration of the PV panels taking into account the electricity generated and the overall energy balance of the building [21].

The next big step of the ASF project required the facade to evolve from a research product to a potential commercial system. To accomplish that, the whole design methodology was redefined in order to achieve a coherent and reliable product. Caranovic [22] proposed a customized parametric design environment, which for a given series of parameters, the structural stability of the facade and the feasibility of the system could be checked.

Another new feature to be highlighted from the redesign of the facade is the structure. A double curved, cross-hatched pipe network was proposed as the new frame [23]. In addition, the structure would be capable to cleanly integrate all electricity cables, control cables, and pneumatic tubes which are required for the function actuators, while simultaneously acting as a mounting point for photovoltaic modules. In order to integrate the full wiring system into the pipe network, Caranovic proposed a tree-like layout based on a series of main lines, which branch out at each actuator.

Finally, the communication protocol of the proposed routing strategy is the I2C. With a dedicated electronic module, the ICM, the bus lines can connect up to 8 panels.
in parallel and control the sensor readings as well as the inflation or deflation of every triad of chambers from every SRA.

With regard to building net energy consumption, a simulation framework for the ASF which takes into account partial shading between modules was developed [24]. The framework integrated high-resolution radiance and PV models with a resistance-capacitance model of the building [5] to properly simulate the solar gains through the window. With that, it was possible to define the appropriate facade configuration for controlling solar heat gains and natural lighting, while simultaneously generating electricity on site, for each hourly time step. The framework reports a 20% - 80% net energy saving potential compared to an equivalent static PV shading system, depending on the efficiency of the heating and cooling system. In some cases, the ASF can almost compensate for the entire energy demand of the building space behind it [25].

Nonetheless, one bottleneck in that framework is the time required to compute the optimum set of angles. A computation of 49 configurations takes approximately 20 minutes, mainly due to the slow computational speed of the radiation analysis, as a result, the angle resolution was limited to 15°. A faster radiation model or an alternative control strategy is required to assess the optimum angles at a finer resolution and a better timing.

The ASF project places its second prototype in 2017 at NEST; the future living and working lab on the Empa-Eawag campus in Dübendorf, Switzerland. Nevertheless, both prototypes neither have the ability to process and command specific user inputs nor can self-move to their optimal configurations in terms of building energy efficiency. Besides, the occupant interaction with the ASF and its comfort parameters are still to be explored in the control scope.

![Figure 1.1: ASF prototype in the House of Natural Resources](image)

Figure 1.1: ASF prototype in the House of Natural Resources [1]
Figure 1.2: ASF actuation system and fixation [2]

Figure 1.3: SRA pneumatic actuation [1]. Three symmetrically distributed chambers are allocated between two discs. The disc on the top acts as the holding unit for the PV panels, while the one at the bottom is set as the structural support. When a chamber inflates, it expands causing the upper disc to rotate.
1.2 Overview of the control problem

Before facing how to control the ASF, the first question one should ask oneself is: “What is the purpose of the system?”. The ASF is conceived as a balance between occupant comfort and building energy efficiency. This means that it must be fast-responsive to user interaction and match their requirements, but at the same time, it has to be capable of self-moving to energy efficient configurations. In Figure 1.4, some potential features for the ASF in terms of occupant comfort and energy efficiency are shown.

From the comfort perspective, the system which monitors the facade has to guarantee to the occupant to have a full control over the facade; if that feature is not covered, the ASF experience could become a distinctively uncomfortable one. Furthermore, since the facade control system contemplates to integrate processing, some more built-in features can be introduced in the comfort point of view. For example, the facade could behave as an intelligent Venetian blind by combining glare control with sun tracking. Similarly, the solar gains through the window behind the ASF could be minimized or maximized in case the occupant wants to warm-up or cool-down the room as fast as possible. Finally, in case the inhabitants wanted to see through the window, the system would have to agilely set to an open position the panels which intercept the sight of the occupants.

![Figure 1.4: Potential features of the ASF. From the comfort regime side, the facade must be fast-responsive and address inhabitants’ commands precisely. Regarding the autonomous regime side, the inhabitant interaction is less influential, but the processing time of complex features must be enough efficient to not slow the responsiveness too much.](image)

Other potential applications for the ASF come to mind when user interaction is less intense. As an example, if occupants do not feel influenced by the facade, then a self-regulating system that monitors the modules to their most optimal configurations concerning building’s net energy consumption might be introduced. Even if the building...
is expected to be uninhabited for a while, a self-adjustment to maximize the electrical energy production could be another interesting feature.

Therefore, two different regimes of regulation can be defined: comfort regime and autonomous regime. The comfort regime is high-demanding in terms of actuation and responsiveness, mainly because of user interaction. Meanwhile, the autonomous regime requires more processing resources to compute optimal configurations, but is not as demanding in terms of responsiveness as the previous one since the comfort parameters take less priority. Notice that the ASF is capable to move every panel individually, so both regimes could be run simultaneously. Nonetheless, this quality makes necessary to group the several modules of the facade in arrays or clusters and establish a hierarchy between them prioritizing occupant comfort (see Figure 1.5).

![Figure 1.5: ASF clustering example for multi-purpose features [1].](image)

Based on the previous perception of the nature of the system, in Figure 1.6 an overview of the control problem is presented as well as the scope of the thesis with a red rectangle.

The overview shows on the one hand, that the facade requires of 3 different inputs to effectively act as an adaptive system. First, a dynamic model of the SRA and its integration in a low-level control algorithm is needed for the proper movement of every individual CIGS panel. Second, a user interface must be developed, so the inhabitants can manually control the ASF or execute some of the features from Figure 1.4. Third, a specific data set containing weather information and details from the building and the ASF has to be introduced, so the simulation framework can effectively estimate the interaction of the facade with the building behind it. None of these previous elements are a goal of the thesis itself, however, their integration within the control algorithm is one of the first steps of the thesis.

The system responsible to monitor the facade has to process the user requirements in order to calculate and command the appropriate ASF configurations. These tasks can
Figure 1.6: ASF control problem overview. Inside the red frame, the scope of the thesis is placed. An algorithm to command ASF configurations taking into account customer’s needs, weather and building data, and the dynamic model of the SRA is about to be developed. The implementation of this algorithm to a microprocessor in cooperation with a microcontroller is also covered. The sensing elements that close the feedback loop in the regulating unit will also be explored.

be high-demanding in terms of processing, and a microcontroller could easily become overwhelmed, hence, in the present work, the main control algorithm will be implemented in a Raspberry Pi, while an Arduino Board will act as the regulating unit of the soft robotic actuators. As a result, with that strategy, the active regulation is decoupled from the high-demanding processing tasks. This is of great importance because the resources consumed while processing highly complex algorithms will not degrade the performance of the actuating system. Therefore, high-responsiveness is not severely affected and safety procedures for SRAs are not disturbed. The justification of this decision will be assessed more extensively in Section 2.4.

The overview points on the other hand, that a feedback loop is necessary in order to properly command the configurations to the facade. The integration of the communication protocol and the sensors with the Arduino Board is also covered in this thesis.

Finally, an outlook leaves the possibility to implement a feedback loop to the processing unit, so advanced control techniques such as machine learning can be implemented.
1.3 Problem Statement

The adaptive control system of the ASF is yet to be designed and implemented in the physical prototypes, therefore, the vast majority of the potential features exposed in Section 1.2 are currently not available. The impossibility of the prototypes to self-move wisely, impedes a proper analysis of the interaction between the ASF and the inhabitants. Besides, the real-time performance of the prototypes as adaptive systems can’t be studied in depth either.

In addition to that, the control of the facade as an adaptive system is challenging since the computational times with the current simulation framework are large. The current state of the art of the ASF simulation framework mentioned in Section 1.1.2, requires from one side a strategy to assess the optimum angles at a finer resolution and a better timing, so new models and approaches have to be explored. Moreover, the ASF prototypes are expected to be actuated by tiny microcontrollers and kept open-source, therefore, software with a high demand of resources or simply incompatible must be replaced (e.g. with Python [26] scripts).

In conclusion, it is necessary to develop and implement a control algorithm capable to process user inputs and monitor the ASF with the aim to match the inhabitants’ requirements. In order to achieve that, the control algorithm must integrate an enhanced version of the simulation framework in terms of resolution, efficiency, and timing.

1.4 Objectives of Research

The following goals should be achieved within this master thesis:

- Define a control scheme for the physical ASF and develop its main control algorithm.
- Develop fast, scalable, readable, and high-resolution scripts within the ASF simulation framework.
- Check the performance of the simulation framework with case studies and sensitivity analyses.
- Implement low-level actuator control techniques to real microcontrollers.
- Implement the control algorithm and the simulation framework to real microprocessors.
- Implement the control scheme to a physical prototype.
- Evaluate the performance of the prototype.
1.5 Thesis Outline

The present thesis aims to become one of the first milestones for the control challenge of the ASF project, which is very wide.

With respect to the methodology of the project, in Section 2.1, the control strategy is introduced. Subsequently, in Section 2.2, the simulation framework is presented and its subprocesses are extensively reviewed and enhanced with new approaches. Afterwards, in Section 2.3, the main control algorithm is defined. Finally, in Section 2.4, all the previous work is implemented in a physical prototype. The control scheme is presented along with the communication protocol responsible to monitor the ASF.

With regard to the results, in Section 3.1, the performance of the subprocesses of the enhanced simulation framework is checked with simulations from other software and case studies. In Section 3.2, the overall performance of the prototype is evaluated with sensitivity analyses.
Methodology

The methodology applied within this thesis is presented as following: first, in Section 2.1, the control strategy is introduced. In Section 2.2, the current simulation framework is presented and every of its subprocesses is reviewed and enhanced one by one. In Section 2.3, the main control algorithm that monitors the previous subprocesses is defined. Finally, in Section 2.4, the work from the previous sections is implemented to a physical prototype.

2.1 Design of the control strategy

An efficient main control algorithm that monitors the ASF must take into account all the possible scenarios and act accordingly. As introduced in Section 1.2, the range of applications the ASF offers is very wide, notwithstanding, the complexity of the processes is also clearly diverse. As an example, the manual control of the facade would not require further information than the desired orientation and the sensing parameters to close the feedback loop of the regulating unit. Alternatively, the self-regulation process that commands an optimal facade configuration in terms of net building energy consumption would require more inputs (e.g. weather data, building heat transfer parameters, etc.) and more complex tasks. An appropriate main control algorithm must take into account these facts, therefore, it is of great importance to clearly discern which tasks and resources are involved for every possible case. To achieve that, the control procedure has been separated in subprocesses.

In Figure 2.1, the relationship between the potential features of the ASF and the subprocesses involved to accomplish them is shown. If the performance of the mentioned subprocesses shows a reasonable precision and processing time, then the efficiency of the system will rely on the main control algorithm itself, which is responsible to organize the execution of the subprocesses in order to find the appropriate facade configuration. Notice that with this strategy, the tasks can be easily identified and even excluded in case they do not apply to achieve the user goals.
The subprocesses within the main control algorithm are 5: *sun position, shading pattern, radiation, PV supply*, and *RC building simulator*. *Sun position* outputs the relative location of the sun with respect to the facade by determining its azimuth and altitude for a given hour of the year. *Shading pattern* computes the shadow geometry that lies on the window because of the ASF, as well as the mutual shading performed between the modules. *Radiation* estimates the incident irradiation on the facade and the solar and lighting gains through the window behind it. *PV supply* simulates the electricity production of the CIGS panels taking into account the power losses because of their self-shading, their estimated temperature, and their efficiency. Likewise, the *RC building simulator* reproduces the thermal behavior of the room behind the facade taking into account the solar gains through the window and the heating systems. The lighting intensity is also estimated in this last step.

The main control algorithm will be responsible to establish the clusters of the ASF, organize the execution of the 5 mentioned subprocesses, and run optimization procedures with the aim to command the appropriate configuration of the facade taking in account the user inputs. To illustrate the connection between the subprocesses and the main control algorithm, an input-output diagram in its most generalist fashion is presented in Figure 2.2.
Figure 2.2: Generalist input-output diagram of the control algorithm. The 5 subprocesses mainly require parameters that define the ASF geometry, the building characteristics, and the weather. There’s also a relation of dependency between some subprocesses. The main control algorithm will take into account the user input, execute the necessary subprocesses, and command the position of the panels taking into account the sensing data.

The font of the inputs can be classified into 5 types: facade parameters, weather parameters, building parameters, sensing parameters and user inputs.

The facade parameters are basic features that define the ASF geometry and specifications. The geometry can be specified by introducing the exact position of the panels and their normal vectors in a 3D space or parametrically. The nature of the cross-hatched pipe network that holds the panels makes the parametric construction of the facade straightforward by only defining its curvature, the spacing between modules and the total amount of them expressed in rows and columns. The technical data of the CIGS panels, especially their size and their power losses because of disturbances, are also included in the facade parameters.

The weather parameters are mainly supplied from a weather station. For every hour of the year and a precise location, the algorithm collects as an input the solar irradiation components and the outer temperature.

The building parameters consist essentially of geometrical and heating coefficients that ease to simulate the thermal behavior of the building at behind the facade. The room temperature and some specifications from the heating systems are also collected to consider the overall energy consumption. Last but not least, the window geometry is
precisely defined by its width and height.

The sensing parameters are mostly associated with the low-level control scheme. The closed-loop control requires a constant awareness of the panel orientation as well as the pressure level in the SRA chambers.

Finally, the user input is the most-conditioning variable introduced to the system. The user input directly depends on the inhabitant and it can be used whether to select an ASF feature (e.g. glare control, max PV generation...) or to insert a precise panel position that overrides the current feature working on that particular module.

These 5 kinds of inputs are directly or indirectly supplied to the several subprocesses, and as a result, a facade configuration is sent to the low-level control scheme, which commands the pneumatic systems to inflate or deflate the corresponding chambers. Notice that most of the inputs are static, which means that they will only have to be introduced once. Nonetheless, the readings from the sensors and the weather parameters will change dynamically during the time, and the system must always be aware of the user inputs.

2.2 Simulation framework enhancement

Using the same naming of subprocesses from Section 2.1, the simulation framework developed in the work of Schmidli [21] and Luzzato [24], is presented as a diagram of subprocesses in Figure 2.1. The framework is mainly designed to compute the optimal facade configuration that minimizes the overall net energy consumption of the building. The computation is fundamentally done by processing all the possible ASF configurations and searching for the minimum value according to the expression:

\[
\min (C + H + L - PV)
\] (2.1)

Where \(C\) and \(H\) are respectively the cooling and heating demand of the room, \(L\) is the lighting power demand and \(PV\) represents the electrical supply from the modules of the facade.

Notice that the present framework matches one of the potential applications for the ASF mentioned in Figure 1.4; the optimal net energy consumption, and makes use of all the subprocesses presented in Section 2.1, therefore, the new improvements for the simulation framework will influence directly the implementation of the main control algorithm to prototypes.
As mentioned in the problem statement, some of the previous work might not be useful for the present control system, whether for incompatibility reasons or for improvable performance. In the following parts, every subprocess will be reviewed and enhanced.

2.2.1 Sun position

As a standard, sun-observer angles are defined as in Figure 2.4a. The solar azimuth ($\gamma_s$) ranges from 0° to 360°, clockwise due North, and the solar altitude ($\alpha_s$) ranges from -90° to 90°. The proposed script computes the position of the sun following the procedure introduced by Honsberg and Bowden [27].

When defining the sun position, it is crucial to pay attention to the use of Local Solar Time vs. Local Time. Local Time ($LT$) usually varies from Local Solar Time ($LST$) because of the eccentricity of the Earth’s orbit, and because of human adjustments such as time zones. In addition, Switzerland is one of the several countries that sets daylight saving time during summer, and this must be taken into account. In this solar script, only $LST$ will be used, the relationship between the time expressions can be defined as:

$$
LST = \begin{cases} 
LT + TC & \text{if Standard time} \\
LT + TC - 60 \text{ min} & \text{if Daylight saving time}
\end{cases}
$$

(2.2)

The net time correction factor ($TC$) accounts for the variation in minutes of the $LST$ due to the longitude variations within the given time zone:

$$
TC = 4 \frac{\text{min}}{\text{min}} \cdot (\lambda - LST M) + E_t
$$

(2.3)
Where $\lambda$ is the geographical longitude, $LSTM$ is the Local Standard Time Meridian (equation 2.4), and $E_t$ is an empirical equation [28] that corrects for the eccentricity of the Earth’s orbit and the Earth’s axial tilt called equation of time (equation 2.5).

$$LSTM = \frac{360^\circ}{24h} \cdot \Delta h_{GMT}$$ \hspace{1cm} (2.4)

$$E_t = 229.2 \cdot 0.000075 + 229.2 \cdot (0.001868 \cdot \cos B - 0.032077 \cdot \sin B) - 229.2 \cdot (0.014615 \cdot \cos 2B + 0.04089 \cdot \sin 2B)$$ \hspace{1cm} (2.5)

$$B = (d - 1) \cdot \frac{360^\circ}{365}$$ \hspace{1cm} (2.6)

$\Delta h_{GMT}$ is the time zone difference in hours between the time zone of study and the Prime Meridian, and $d$ is the day of the year, counting from the 1st of January.

Once the Local Solar Time is known, the script proceeds to convert it into a sun angle, this is done with a parameter called hour angle ($\omega$); the apparent displacement of the sun away from solar noon. Note that a solar day is 24 hours long, and Solar noon is always used locally as the center of time, therefore, before Solar noon time is counted backwards and after Solar noon it is counted forward. Thus, the sign of the hour angle is determined by occurring either before noon (negative) or after noon (positive):

$$\omega = \frac{360^\circ}{24h} \cdot (LST - 12)$$ \hspace{1cm} (2.7)

$$\omega = \begin{cases} 
0^\circ \text{ to } -180^\circ & \text{if before noon (morning)} \\
0^\circ \text{ to } 180^\circ & \text{if after noon (evening)}
\end{cases}$$

To take into account the Earth’s tilt, the angle of declination $\delta$ is calculated. The angle is considered to be $23.45^\circ$ during summer solstice:

$$\delta = 23.45^\circ \cdot \cos \left( \frac{360^\circ}{365} \cdot (d - 81) \right)$$ \hspace{1cm} (2.8)

Once the hour angle ($\omega$) and the declination angle ($\delta$) are known, and the geographical latitude ($\phi$) is introduced, then the altitude ($\alpha_s$) and azimuth ($\gamma_s$) values of the sun can be finally calculated:

$$\alpha_s = \arcsin \left( \cos \phi \cdot \cos \delta \cdot \cos \omega + \sin \phi \cdot \sin \delta \right)$$ \hspace{1cm} (2.9)

$$\gamma_s = \begin{cases} 
\arccos \left( \frac{\sin \delta \cdot \cos \phi - \cos \delta \cdot \sin \phi \cdot \cos \omega}{\cos \alpha_s} \right) & \text{if } \omega \leq 0 \\
360^\circ - \arccos \left( \frac{\sin \delta \cdot \cos \phi - \cos \delta \cdot \sin \phi \cdot \cos \omega}{\cos \alpha_s} \right) & \text{if } \omega > 0
\end{cases}$$ \hspace{1cm} (2.10)
Nonetheless, the standard angle description from Figure 2.4a is not the most convenient for the simulation framework, especially regarding the azimuth range (see Equation 2.10). A more advantageous strategy (see Figure 2.4b) could be to redefine the azimuth angle as:

\[
\gamma_{ASF}^s = \begin{cases} 
360^\circ - \left| \gamma_{s}^{obs} - \gamma_{NASF}^{obs} \right| & \text{if } (\gamma_{s}^{obs} - \gamma_{NASF}^{obs}) \leq -180^\circ \\
\gamma_{s}^{obs} - \gamma_{NASF}^{obs} & \text{otherwise}
\end{cases}
\]

(2.11)

Where \(\gamma_{NASF}^{obs}\) is the azimuth position of the normal to the ASF plane, and \(\gamma_{s}^{obs}\) is the sun azimuth, both according to sun-observer standard convention.

Where \(\gamma_{NASF}^{obs}\) is the azimuth position of the normal to the ASF plane, and \(\gamma_{s}^{obs}\) is the sun azimuth, both according to sun-observer standard convention.

\[\gamma_{ASF}^s = \begin{cases} 
360^\circ - \left| \gamma_{s}^{obs} - \gamma_{NASF}^{obs} \right| & \text{if } (\gamma_{s}^{obs} - \gamma_{NASF}^{obs}) \leq -180^\circ \\
\gamma_{s}^{obs} - \gamma_{NASF}^{obs} & \text{otherwise}
\end{cases}
\]

With that, the sun position is redefined relative to the normal of the facade, and a procedure for the subsequent processes that is independent to the orientation of the surface which holds the ASF can be introduced. As an example, whenever the azimuth component surpasses the \([-90^\circ, 90^\circ]\) range, then no direct sunlight will reach the facade, so the ASF calculations with respect to direct solar gains might be deactivated. In addition, the positive and negative values of the sun azimuth give a rapid understanding of the direction of the sun rays (right or left, respectively). Notice that, for consistency, the orientation of the panels must be defined relative to the facade as well.
2.2.2 Shading pattern

The shadow caused by the ASF is of great importance. In addition to glare control, the shadow directly affects the solar gains through the window. Moreover, the modules of the ASF can perform shading between themselves, causing decisive electric power losses depending on the shape and the location of the dark areas.

Even though the number of modules and their individual level of actuation can increase the complexity of the problem potentially, the shading pattern is still a geometric problem combined with vector algebra. Previous work introduced the calculation of the shadow using Ladybug [21], yet there is no need to make use of high-demanding software to compute the shadow. A geometrical approach for mutual shading was introduced with Python scripting by checking the enclosing of the shadow vertices with their immediate surrounds [29], however, the performance of the script can be severely improved, and some big simplifications concerning mutual shading can’t be accepted when complex shading patterns come into sight.

In the present thesis, a new high-resolution script for arbitrary shading patterns has been developed based on Vatti’s clipping algorithm. First, a 3-dimensional representation of every module is defined by its 4 vertices. This is done by simulating the SRA movement with respect to a generic PV panel in neutral position (pitch: 0°, yaw: 0°, panel elevation: 30°), with its centerpoint placed in the origin. The designation of the axes of actuation and the naming of vertices is illustrated in Figure 2.5.

![Figure 2.5](image)

Figure 2.5: Main positions for and ASF panel. The geometrical definition is done with four vertices. The movement is simulated defining two axes of rotation. The pitch axis is fixed in space while the yaw axis depends on the pitch rotation.
Notice that the pitch and yaw rotations are strongly related with the final elevation and azimuth of the panels, but they are not certainly the same. The relationship between the final orientation of the panels and the pitch and yaw values introduces an inverse kinematic problem that arises the computational complexity of the process. It will not be covered in this work.

Taking as a basis the kinematic model introduced in Figure 2.5, observe that the natural movement of the SRA places the pitch axis in a fixed location, while the yaw axis is dependent on the pitch value and is offset by 45°. If we define an arbitrary axis vector with two aligned points, then the rotation of a point \( P_i \) by \( \theta \) around an arbitrary axis defined by \( \mathbf{v} \) is:

\[
\mathbf{v} = \| P_2 - P_1 \| \quad (2.12)
\]

\[
Q_i = M_{\text{rot}}(\mathbf{v}, \theta) \cdot (P_i - P_1) + P_1 \quad (2.13)
\]

Where the rotation matrix \( M_{\text{rot}}(\mathbf{v}, \theta) \) is:

\[
\begin{bmatrix}
  v_x^2 \cdot (1 - \cos \theta) + \cos \theta & v_x \cdot v_y \cdot (1 - \cos \theta) - v_z \cdot \sin \theta & v_x \cdot v_z \cdot (1 - \cos \theta) + v_y \cdot \sin \theta \\
  v_x \cdot v_y \cdot (1 - \cos \theta) + v_z \cdot \sin \theta & v_y^2 \cdot (1 - \cos \theta) + \cos \theta & v_y \cdot v_z \cdot (1 - \cos \theta) - v_x \cdot \sin \theta \\
  v_x \cdot v_z \cdot (1 - \cos \theta) - v_y \cdot \sin \theta & v_y \cdot v_z \cdot (1 - \cos \theta) + v_x \cdot \sin \theta & v_z^2 \cdot (1 - \cos \theta) + \cos \theta
\end{bmatrix}
\]

Hence, it is possible to define the yaw axis as:

\[
\mathbf{v}_\psi = M_{\text{rot}}(\hat{\mathbf{i}}, \theta + \pi/4) \cdot \hat{\mathbf{k}} \quad (2.14)
\]

Note that the yaw axis direction is defined in order to match the clockwise and counterclockwise criteria for the sun azimuth position from Section 2.2.1.

Taking the previous geometric operation as a reference, if we define the four vertices of a generic PV panel in neutral position (elevation of 30°), with its centerpoint at the origin, and naming \( h \) as half the diagonal of the squared PV panel:

\[
(S_0, S_1, S_2, S_3) = \begin{pmatrix}
  -h & 0 & h & 0 \\
  0 & h \cdot \sin(\pi/6) & 0 & -h \cdot \sin(\pi/6) \\
  0 & -h \cdot \cos(\pi/6) & 0 & h \cdot \cos(\pi/6)
\end{pmatrix}
\]

Then the general expression to compute the position of the \( S_i \) vertices of a generic PV panel is:

\[
S_{i,\text{rot}} = M_{\text{rot}}(\mathbf{v}_\psi, \psi) \cdot [(M_{\text{rot}}(\hat{\mathbf{i}}, \theta) \cdot (S_i - P_{1p})) + P_{1p}] \quad (2.15)
\]

Where \( P_{1p} \) is the pitch centerpoint of rotation (in millimeters):

\[
P_{1p} = (0 \ 49 \ -29)^T
\]

In conclusion, the movement of the modules is resembled by applying the pitch rotation (\( \theta \)) to a generic panel in neutral position, and later, the yaw rotation (\( \psi \)). Once the generic position is computed, then the physical position of the corresponding panel is calculated whether introducing the offset with respect its actual 3D position or using
a parametric constructor taking into account the cross-hatched structure of the facade. Afterwards, the projection of the 3D geometry as a shadow on the wall is computed.

For a given sun position according to the sun-facade convention (see Section 2.2.1), the sun vector is calculated as:

\[
\phi_s = (\cos (\gamma^{ASF} + \pi/2) \cdot \cos \alpha_s \sin \alpha_s \sin \gamma^{ASF} \cdot \cos \alpha_s)
\]

The parametric equation of \(\phi_s\) is:

\[
\frac{x - x_0}{\phi_x} = \frac{y - y_0}{\phi_y} = \frac{z - z_0}{\phi_z} = t
\]  

(2.16)

For convenience, the wall is placed at \(z = 0\), hence, the \(t\) parameter that performs the projection of the vertex \(S_i\) is:

\[
t = -\frac{S_{iz}}{\phi_z}
\]

(2.17)

Thus, the 2-dimensional projection of the geometry on the wall is:

\[
P_{ix} = S_{ix} + \phi_x \cdot t
\]

\[
P_{iy} = S_{iy} + \phi_y \cdot t
\]

(2.18)

This procedure is executed on all the panels, creating a gross shadow on a generic surface. If the gross shadows overlap themselves, this means that the projection of some shadows are not actually reaching the window, hence, mutual shading between panels is being performed.

To quantify the mutual shading of every panel, the intersection between the 2-dimensional gross shadows is checked using Pyclipper [30]; a Cython wrapper for the C++ translation of the Angus Johnson’s Clipper library [3]. Pyclipper allows clipping of any number of arbitrarily shaped subject polygons by any number of arbitrarily shaped clip polygons.

Figure 2.6: Main boolean operations of Pyclipper [3]. To extract the mutual shading, a boolean difference will be used. To compute the global shadow lying on the window, a boolean intersection will be executed.
The library supports finding the intersections (regions of overlap) of subject and clip polygons and other boolean clipping operations: difference, where the clipping polygons remove overlapping regions from the subject; union, where clipping returns the regions covered by either subject or clip polygons, and; XOR, where clipping returns the union geometry except the regions covered by both subject and clip polygons (see Figure 2.6).

The script considers separately every panel as the subject shadow once and checks the intersection between the surrounding shades, defined as clips. In case both regions overlap, then the difference boolean operation is executed, and the overlapping area is extracted from the subject shadow. Taking into consideration the provenance of the sun rays, the mutual shading area is only subtracted to the panels that receive the self-shading, therefore, the resulting shapes accurately define the net shadow area on the wall. Finally, the resulting shading pattern is allocated with respect to the window frame, and the shade lying outside its limits is extracted with a boolean intersection between the window and the subject shadow.

In Figures 2.8, an example of the clipping procedure for a single-cluster ASF is shown. Observe that the plotted results output the shading shapes in millimeters for every single module, magnifying the even the smallest geometries.

Figure 2.7: Shading pattern script output: single-cluster example for 18 modules.
Since the execution of the code is implemented processing every panel one by one, the script is capable to compute arbitrary configurations of the ASF with n-clusters. Nonetheless, the mutual shading shapes become more complex, and the quantification of the mutual-shading shapes becomes more challenging, this part is assessed in more detail in Section 2.2.4.

![ ASF configuration represented in 2D (a) ASF configuration in 2D ](image1)

![ ASF shadow on wall Sun Alt & Az: (80, -22) (b) Gross shadow on generic surface ](image2)

![ ASF shadow on wall, Mutual shading extracted Sun Alt & Az: (80, -22) (c) Mutual shading extraction ](image3)

![ Shadow on window frame Sun Alt & Az: (80, -22) (d) Resulting net shadow on window ](image4)

Figure 2.8: Shading pattern script output: 18-cluster example for 18 modules

### 2.2.3 Radiation

If real-time weather data is supplied to the facade, especially the direct normal irradiation and its diffuse horizontal value, then it is possible to estimate the incident irradiation on the modules and the surface behind them. Likewise, it is possible to estimate the lighting flux when the direct and diffuse horizontal luminosity is supplied.

As in the shading pattern, previous work made use of Grasshopper plug-ins such as Ladybug and DIVA coupled with an electric model of the panels [21, 24], but again, this software is not compatible with the new control scheme. A new approach was also introduced with Python by calculating the solar irradiation with the three-component model [29, 31, 32].

21
The present script also considers the irradiation on each module as the sum of direct beam ($G_{dir}$), diffuse sky ($G_{dif}$), and diffuse reflected irradiation ($G_{ref}$):

$$G_{tot} = G_{dir} + G_{dif} + G_{ref}$$  \hspace{1cm} (2.19)

**Direct beam irradiation**

Direct beam irradiation is calculated as the product of the direct normal irradiation ($G_{dni}$) and the cosine of the angle of incidence ($\theta_i$):

$$G_{dir} = G_{dni} \cdot \cos \theta_i$$  \hspace{1cm} (2.20)

The angle of incidence is the angle between the vector perpendicular to the module plane, called the normal of the plane, and the projection of the sun central beam to the panel surface. It is calculated as:

$$\cos \theta_i = \frac{\cos \alpha_s \cdot \cos (\gamma^A_{SF} - \gamma^C_{ASF}) \cdot \sin \beta + \sin \alpha_s \cdot \cos \beta}{\sin \alpha_s}$$  \hspace{1cm} (2.21)

Where $\alpha_s$ and $\gamma^A_{SF}$ are the solar altitude and azimuth respectively, $\beta$ is the slope or tilt of the panel and $\gamma^C_{ASF}$ is the cell azimuth angle. Whenever the angle of incidence exceeds 90° (so the cosine reaches a negative value) the panel is turning its back to the sun, thus, it is shading itself.

Figure 2.9: Sun-ASF panel angle convention: south-facade case example. The dashed lines define the panel position following the sun-observer standard convention from 2.4a. For consistency, the panel orientation is defined following the sun-facade convention from 2.4b.
Diffuse sky irradiation

The diffuse sky irradiation is approached introducing Perez incident sky diffuse model [33, 34, 6]. In this model, the diffuse irradiation is divided in 3 components: the diffuse isotropic irradiation ($G_{iso}$), the circumsolar diffuse irradiation ($G_{cir}$) and the horizon brightening ($G_{hb}$).

$$G_{dif} = G_{iso} + G_{cir} + G_{hb} \tag{2.22}$$

Diffuse isotropic irradiation For the particular case of the ASF, the approach proposed by Perez for the isotropic component is not appropriate, since it only considers the panel tilt angle and the complementary value of the circumsolar brightening coefficient ($F_1$):

$$G_{iso} = G_{dif,h}(1 - F_1) \cdot \frac{1 + \cos\beta}{2} \tag{2.23}$$

The isotropic component considers the diffuse radiation to be distributed uniformly all over the sky dome, therefore, the percentage of irradiation gathered mainly depends on the geometrical configuration of the panel. This relationship is comprised within the concept of the sky view factor. The sky view factor is a dimensionless parameter which ranges between zero and one, that represents the fraction of the visible sky on a hemisphere which lays centered over the analyzed location. In fact, the last product term in Equation 2.23, is the sky view factor definition from Liu-Jordan [35]:

$$SV_{LJ} = \frac{1 + \cos\beta}{2} \tag{2.24}$$

However, the PV modules of the facade have two degrees of freedom, the surface behind the facade covers an important fraction of the visible sky and the sky view factor of every individual panel is influenced by its surrounding modules. Previous studies in urban sites show that the sky view factor can be approached by graphically evaluating the visible fraction of the sky from images taken by eye-fish cameras placed on the panels [36], for the case of the facade, however, it is not feasible to set a camera on every module. In order to contemplate these effects without a graphical engine, Hofer et al. approximated the sky view factor value by placing hemispherical meshes on every module. By projecting vectors from the respective centerpoints to the sky, if any external geometries where intercepted by the vectors, then that part of the sky was not visible. In other words, the sight of the sky was discretized.

Having this last idea in mind, the sky view factor is approached with 3 steps: wall vs tilt effect, panel azimuth effect, and surrounding panels effect. The two first steps consist of a successive application of the Liu-Jordan equation (Equation 2.24). First, it calculates the sky view factors obtained by just considering the height between the panel and the roof, and compares this value with the one obtained by just considering the panel tilt (see Figure 2.10). The minimum value is kept.
\[
SV_1 = \min\left(\frac{1 + \cos \beta}{2}, \frac{1 + \cos \beta_{wall}}{2}\right)
\] (2.25)

If the azimuth value is high enough to establish an interaction between the side part of the panel and the wall (see Figure: 2.11), then from the lowest value obtained from the first step, the complementary value of the sky view factor just considering the panel azimuth is subtracted. In that case, the Liu-Jordan expression is divided by two since the azimuth effect is not hemispherical but half-hemispherical:

\[
SV_2 = SV_1 - 0.5 \cdot \left(1 - \frac{1 + \cos \Delta \gamma^{ASF}}{2}\right) \quad \text{if } \Delta \gamma^{ASF}_{c-VF_{max}} > 0
\] (2.26)

\[
\Delta \gamma^{ASF}_{c-VF_{max}} = \text{sign}(\Delta \gamma^{ASF}_c) \cdot (\gamma^{ASF}_{c_{VF_{max}}} - \Delta \gamma^{ASF}_c)
\] (2.27)

Where \(\Delta \gamma^{ASF}_{c-VF_{max}}\) is the difference between the current module azimuth (\(\gamma^{ASF}_c\)) and the maximum angle that neglects the azimuth effect (\(\gamma^{ASF}_{c_{VF_{max}}}\)). Notice that the maximum angle is twice defined for positive and negative panel azimuth cases.
Finally, the surrounding panels effect is considered by slicing the particular sky dome of every panel in 5 zones (see Figure 2.12). The ASF geometry is projected along the normal vectors of every part, and with the same clipping strategy as in the shading pattern, the intersection between geometries is evaluated in order to set a relative value of the geometric interaction. Depending on the orientation of the panel and the spacing between the panels, the zones will have more or less influence to the final result.
Figure 2.12: Third step for sky view factor approach: surrounding panels effect. The involved part of the sky dome is sliced in five parts. Depending on the azimuth position of the panel and the spacing between panels, every part will have a defined weight. Notice that the closer a sight interception occurs, the higher effect it has on the sky view factor.

The resulting value of intersection is subtracted as a separate part of the sky view factor, since it only interferes a particular range of the sky dome:

$$SV_3 = SV_2 - \sum_{i=1}^{5} \lambda_i(\gamma cASF) \cdot \left(1 - \frac{1 + \cos(\arctan(h/\Delta c))}{2}\right) \cdot \%SV_3 \quad (2.28)$$

Where $\lambda_i$ is the weight of the analyzed sky portion $i$, and is dependent to the azimuth position of the panel, $h$ is the diagonal of the squared panel, $\Delta c$ is the average spacing between panels, and $\%SV_3$ is the percentage of overlap using the clipping algorithm.

**Diffuse circumsolar irradiation**  The circumsolar component represents the diffuse radiation coming from the vicinity of the sun. It is calculated following the expression:

$$G_{cir} = G_{dif,h} \cdot F_1 \cdot a/b \quad (2.29)$$

Where the parameters $a$ and $b$ are dependent to the angle of incidence ($\theta_i$), and describe the view of the sky from the perspective of the surface:

$$a = \max(0, \cos \theta_i)$$
$$b = \max(\cos 85^\circ, \cos \theta_i) \quad (2.30)$$

$F_1$ is an empirical coefficient which is function of the sky clearness $\varepsilon$, the sky brightness $\Delta$, and the sun zenith in radians ($\theta_z$).

$$F_1 = \max[0, f_{11}(\varepsilon) + \Delta f_{12}(\varepsilon) + \theta_z f_{13}(\varepsilon)] \quad (2.31)$$
The sky clearness is calculated as in Equation 2.32, with $\kappa = 1.041$. Meanwhile, the sky brightness assumes an extraterrestrial irradiance value ($I_0$) of 1367$\, W/m^2$, and it is calculated as in Equation 2.33

$$
\varepsilon = \frac{G_{dif,h} + G_{dni}}{G_{dif,h}} + \kappa \cdot \theta_z^3
$$

(2.32)

$$
\Delta = \frac{G_{dif,h} \cdot AM_o}{I_0}
$$

(2.33)

Finally, the sky brightness depends on the absolute optical air mass:

$$
AM_o = (\cos b + 0.15 \cdot (93.9^\circ - \theta_z)^{-1.253})^{-1}
$$

(2.34)

### Table 2.1: Perez experimental sky diffuse irradiance model coefficients [6]

<table>
<thead>
<tr>
<th>$\varepsilon$</th>
<th>$f_{11}$</th>
<th>$f_{12}$</th>
<th>$f_{13}$</th>
<th>$f_{21}$</th>
<th>$f_{22}$</th>
<th>$f_{23}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\leq 1.065$</td>
<td>-0.0083117</td>
<td>0.5877285</td>
<td>-0.0620636</td>
<td>-0.0596012</td>
<td>0.0721249</td>
<td>-0.0220216</td>
</tr>
<tr>
<td>$1.23$</td>
<td>0.1299457</td>
<td>0.6825954</td>
<td>-0.1513752</td>
<td>-0.0189325</td>
<td>0.065965</td>
<td>-0.0288748</td>
</tr>
<tr>
<td>$1.5$</td>
<td>0.3296958</td>
<td>0.4868735</td>
<td>-0.2210958</td>
<td>0.055414</td>
<td>-0.0639588</td>
<td>-0.0260542</td>
</tr>
<tr>
<td>$1.95$</td>
<td>0.5682053</td>
<td>0.1874525</td>
<td>-0.295129</td>
<td>0.1088631</td>
<td>-0.1519229</td>
<td>-0.0139754</td>
</tr>
<tr>
<td>$2.8$</td>
<td>0.873028</td>
<td>-0.3920403</td>
<td>-0.3616149</td>
<td>0.2255647</td>
<td>-0.4620442</td>
<td>0.0012448</td>
</tr>
<tr>
<td>$4.5$</td>
<td>1.1326077</td>
<td>-1.2367284</td>
<td>-0.4118494</td>
<td>0.2877813</td>
<td>-0.8230357</td>
<td>0.0558651</td>
</tr>
<tr>
<td>$6.2$</td>
<td>1.0601591</td>
<td>-1.5999137</td>
<td>-0.3589221</td>
<td>0.2642124</td>
<td>-1.127234</td>
<td>0.1310694</td>
</tr>
<tr>
<td>$&gt; 6.2$</td>
<td>0.677747</td>
<td>-0.3272588</td>
<td>-0.2504286</td>
<td>0.1561313</td>
<td>-1.3765031</td>
<td>0.2506212</td>
</tr>
</tbody>
</table>

#### Horizon brightening

The horizon brightening is calculated as in Equation 2.35. For clear skies, the value of the horizon brightening decreases as the sun is placed away from the horizon. For overcast skies, the horizon brightening can achieve negative values.

$$
G_{hb} = G_{dif,h} \cdot F_2 \cdot \sin \beta
$$

(2.35)

Where $\beta$ is the panel tilt, and $F_2$ is an empirical coefficient likewise $F_1$:

$$
F_2 = f_{21}(\varepsilon) + \Delta f_{22}(\varepsilon) + \theta_z f_{23}(\varepsilon)
$$

(2.36)

#### Diffuse reflected irradiation

Diffuse reflected radiation is the product of the fraction of ground light irradiance reflected toward the module, the albedo of reflecting objects ($\rho$) and global irradiance on the horizontal plane ($G_{glo,h}$). It is assumed, that the sum of the fraction of light reflected from the ground to the module and the sky view factor for this module equals one. Note
that currently only a single albedo value is used, not distinguishing between different reflecting objects:

$$G_{ref} = \rho \cdot G_{glo,h} \cdot (1 - SV)$$  \hspace{1cm} (2.37)

The global horizontal irradiation is function of the sun altitude ($\alpha_s$):

$$G_{glo,h} = G_{dni} \cdot \sin \alpha_s + G_{dif,h}$$  \hspace{1cm} (2.38)

**Application on the window behind the facade**

For the case of the glazing surface the same model using the three-components is used. Notwithstanding, the calculations concerning the sky view factors are less intense. On the one hand, the position of the window is fixed to 90°, setting the Liu-Jordan equations to a constant value of 0.5. On the other hand, the visible sky from the window is assumed to be directly related with the panels, therefore, a perpendicular projection of the panels is executed on the window, and again, the clipping algorithm is used to quantify the percentage of visible sky that the panels are intercepting. Nonetheless, irrespective of the configuration of the facade, the ASF structure itself will always intercept a fraction of the visible sky. The geometrical evaluation of the sky view factor has been experimentally reduced to 0.4 with the help of the simulation tool in Section 3.1.2. This last value would correspond for the case of a complete open position of the ASF.

For the calculation of the illuminance on the window, a parallelism is carried out using the same exact procedure but with direct and horizontal diffuse components of the sunlight.

**2.2.4 PV supply**

To set its most appropriate configuration, the facade requires being aware of the potential electricity it is capable to generate. Once the global incident irradiation on the facade is calculated, there are some more factors to take in account in order to estimate the energy supply of the panels: the mutual shading, the temperature power losses and the panel efficiency.

**Mutual shading power loss**

In order to evaluate the location and the shape of the dark areas caused by mutual shading between modules and how they affect the power supply, Hofer et al. [4] extensively reviewed and experimentally quantified the power losses of the CIGS panels when a rectangular shading area was located on the modules (see Figure 2.13). Taking advantage of the high-resolution definition of the shade on the panels, the script evaluates the area of the region and applies the corresponding experimental power loss.
Figure 2.13: Experimental power losses for CIGS panels. A power loss because of self-shading can be defined with the percentages of longitudinal and lateral mutual shading, i.e. the width and the height of the shade divided by the respective panel side measurement [4] (edited).

As mentioned in Section 2.2.2, the shading shapes on the panels can become really complex as the number of clusters increase, and it can be really challenging to relate them with a rectangular power loss. From Figure 2.13, one can observe that the longitudinal shading percentage affects dramatically the power loss, while the lateral shading is kind of proportional after 20% of longitudinal shading. This makes sense, since the electric layout of the modules connects the by-pass diodes in the longitudinal direction, and the sub-cells are distributed laterally.

In the present simulation framework, from the shading pattern subprocess, the precise shaded area percentage ($%_{sh}$), and its maximum longitudinal width percentage ($%_{long}$) following the direction of the vector defined by $S_0$ and $S_1$, are extracted. Afterwards, the lateral shading value is calculated following the expression:

$$%_{lat} = \min \left( \frac{%_{sh}}{%_{long}}, 1 \right)$$  \hspace{1cm} (2.39)

Notice that the approach magnifies the effect of the longitudinal shading since the lateral shading is corrected to not exceed 100%.

Since the dark areas are directly related to the direct beam radiation of the sun, the experimental power loss values are applied to the incident direct irradiation on the facade. By that strategy, the electric model of the CIGS panels is resembled when mutual shading applies, because the shape of the dark area is penalized emphasizing the influence of the longitudinal direction of the shadow.
**Temperature power loss**

The dependence of the electric supply on temperature is introduced with the product of the temperature coefficient of the CIGS panel (set as $-0.35\%/\degree C$) and the estimated temperature of the cell, calculated as:

$$T_c = T_{air} + \left( \frac{T_0^c - T_{air}^0}{S_0} \right) \cdot S_c \quad (2.40)$$

Where $T_{air}$ is the ambient temperature in Celsius, $S$ is the cell insulation $W/m^2$, and $T_0^c$ is the temperature of the cell at reference insulation $S_0 = 800W/m^2$ and reference air temperature $T_{air}^0 = 20^\circ C$. The value of $T_0^c$ was estimated using thermal images of the solar facade and typical values given in [37] to be $38^\circ C$.

**Panel efficiency**

Finally, the CIGS panel efficiency ($\eta_c$), which is set at 16%, is implemented in the energy supply estimation. The general expression for the estimation of the electricity production in Wh of a panel ($E_{cell}$) would be:

$$E_{cell} = [G_{dir} \cdot \eta_{sh}(\%_{long}, \%_{lat}) + G_{dif} + G_{ref}] \cdot \eta_{temp} \cdot (T_c - 25) \cdot \eta_c \cdot A_c \quad (2.41)$$

Where $G_{dir}$, $G_{dif}$, and $G_{ref}$ are the three irradiation components from the radiation subprocess in $Wh/m^2$, $\eta_{sh}$ is the power loss due to mutual shading, $\eta_t$ is the temperature power loss, and $A_c$ is the area of the squared panel in $m^2$.

**2.2.5 RC building simulator**

The RC building simulator is the implementation within the simulation framework of the work previously conducted by Jayathissa et al. [5] and will be briefly introduced here for completeness. The model is based on the 5R1C model, an electrical analogy corresponding to the equivalent thermal physics [38, 39, 40], and is based on the ISO 13790 standard [41]. Figure 2.14 shows the setup of the model which consists of one internal thermal capacitance and five thermal resistances.
The model assumes that only one surface is in contact with the external environment (normally this is the case of the south-facing window surface), and all other surfaces of the room are in contact with other thermal zones of the building. In addition, it is considered that they have the same room temperature and therefore the surfaces are modeled as adiabatic.

The temperature of the thermal mass in the room is denoted by \( T_m \), the differential equation for the circuit in Figure 2.14 is given by:

\[
C_m \cdot \frac{dT_m}{dt} + T_m \cdot (H_{tr3} + H_e) = \phi_{mtot}
\]  

(2.42)

Where \( C_m \) is the thermal capacitance of the room, \( T_m \) is the thermal mass temperature, \( H_{tr3} \) and \( H_e \) are equivalent thermal conductances, and \( \phi_{mtot} \) is the equivalent thermal heat flux based on the solar heat gains, internal heat gains, external air temperature and the thermal conductance of the building elements.

The integration of the RC building simulator with the simulation framework is done by multiplying the solar and illuminance gains by the respective transmittances of the window, to finally include them in the \( \phi_{mtot} \) parameter.

### 2.3 Main control algorithm

The main control algorithm has the responsibility to manage the execution of the subprocesses from Section 2.2 with the aim to define an appropriate configuration of the ASF that matches the user requirements. In other words, the goal of the algorithm is to output a set of angles for the panels of the facade. Since the ASF is a modular system, it is convenient to propose an identification of the panels that is intuitive and easily scalable. The proposed identification of the modules is presented in Figure 2.15.
Figure 2.15: General panel identification strategy for the ASF. The relative position of a panel within the facade can be quickly defined with its row and column numbers, counted from the upper-left part of the ASF.

The cross-hatched structure of the ASF, enables to identify a particular panel by precising its row and column number. In addition, notice that the panels are physically placed only when both row and column numbers are even or odd, this eases even more the parametric construction of the geometry in case it is desired. Following the procedure of the shading pattern calculation in Section 2.2.2, the physical position of the $S_i$ vertex of the PV at row $a$ and column $b$ using a parametric constructor, can be calculated as:

$$S_{i}^{a,b} = S_{i,rot}^{a,b} + \left( \frac{b \cdot \Delta_c}{2}, -\frac{a \cdot \Delta_c}{2}, \Delta_{glass}^{a,b} \right) \quad \{ a \mod 2 = b \mod 2 \} \quad (2.43)$$

Where $S_{i,rot}^{a,b}$ is the position of the vertex after applying the pitch and yaw rotation, $\Delta_c$ is the Cartesian distance between panels of the same row or column, and $\Delta_{glass}^{a,b}$ is the distance between the glazing surface and the panel. For a highly-precise definition of the position, keep in mind that the physical orientation offset from the neutral position must be also applied to every particular panel.

The main algorithm will mainly consider two different kinds of inputs: an ASF feature or a desired configuration of the facade. Notice that both inputs can be sent combined, making necessary the definition of clusters. The main algorithm establishes a hierarchy putting the highest priority on the configurations directly sent by the user, then the
comfort regime features, and finally the autonomous regime features. An example of the priority establishment is shown in Figure 2.16.

Figure 2.16: Work-flow example of the main control algorithm. The configuration of the facade is set sequentially by establishing a priority between the several kinds of inputs from the user, which are, in order: direct control of panels, comfort features and autonomous features. In case some panels are strictly set to a position directly, the following comfort features will be restricted. Likewise, the autonomous features are applied to the rest of panels which have not been yet set.

Essentially, without any direct command or a comfort feature, the main control algorithm will be solving an optimization problem considering every possible configuration of the facade, i.e. n-clusters. Depending of the main interest of the inhabitant, the algorithm will establish a proper weighting in the following formula:

\[ \text{minimise } (\lambda_C \cdot C + \lambda_H \cdot H + \lambda_L \cdot L - \lambda_{PV} \cdot PV) \]  

(2.44)

Depending on the feature, some weights could actually be set to zero, relieving the execution of some subprocesses on the main program.
2.4 Implementation to prototypes

In the present thesis, the control algorithm will be implemented in the GlassTec prototype placed in the Chair of Architecture and Building Systems at ETH Zürich (see Figure: 2.17). The GlassTec prototype comprises a total of 8 CIGS panels in a cross-hatched pipe structure of width 1.3 m and height 2.05 m.

As introduced in Section 1.2, the algorithm will be implemented with an Arduino Uno board plus a Raspberry Pi 3 board. This decision has been taken with the aim to decouple the direct control of the SRAs from the demanding processing tasks, but there are many other reasons that will be introduced in this chapter.

When real-time control capabilities are considered, one parameter of extreme importance when dealing with dynamic components is the timing scope. Some actions of the system like the inflation of chambers are hard timed, which means that if the response time is not within the expected range, the chambers could explode. Some other tasks are soft timed, like the command of ASF configurations, when if the time response is not respected just the user experience is degraded.

Even though the Raspberry Pi processor runs at higher frequencies than Arduino, its operating system and the subsequent higher priority tasks may conduct the application to not notice that a GPIO pin has changed state. Furthermore, the Raspberry Pi is selected as the processing unit, if it had to control the actuators, then parallel programming would have to be implemented, reducing, even more, the accuracy of the time scope. These limitations could also become critical if high-performance of the actuators is desired in future steps, and so makes the Raspberry Pi a not recommendable unit for real-time control. All that, lead to the selection of a microcontroller to perform the hard timed tasks and leave the Raspberry Pi for the soft timed tasks.

Nonetheless, it is still possible to deactivate some of the main features of the Raspberry Pi and change the interruption hierarchy, however, that strategy would lead to an inefficient use of the board. Notice that there is still another big step in the control scheme; the user interface (see Figure 1.6), and the Raspberry Pi would be a well-suited candidate to implement this functionality as well.
In the following sections, the control scheme, the hardware, the communication protocol and the direct control of the actuators will be reviewed in detail.

### 2.4.1 The control scheme

In the control scheme we can find 4 units: the Arduino Board, the Raspberry Pi, the ASF pneumatic system and the various ICMs to control the readings and the triad of chambers of every SRA. A big picture of the control scheme is presented in Figure 2.18.

Both boards are connected via USB serial communication. The Arduino Board acts as the regulating unit: it reads the sensor data, sends it to the Raspberry Pi, and commands the opening and closing of the SRA chambers as well as the pneumatic valves of the pressure supply. The Raspberry Pi will be responsible to compute the
appropriate facade configuration and to transmit the necessary command to correct the
current angles of the modules to the Arduino Board.

The sensor reading and the direct control of the SRA chambers is done establishing
an I²C communication between the Arduino Board and the ICMs. With respect to the
valves that control the pressure inlet and outlet of the facade, they are directly controlled
with 3 digital pins of the Arduino Board.

Figure 2.18: General diagram of the control scheme. The Raspberry Pi is connected
with the Arduino Board via USB serial communication. The Raspberry Pi is the unit
responsible to compute the simulations and send the commands to the Arduino. The
Arduino is responsible of the sensor readings and the control of the triad of chambers of
every panel via I²C protocol, as well as the direct opening and closing of the valves for
pressure inlet and outlet.

Data acquisition and chamber control; the I²C protocol

As proposed in Caranovic’s work [22], the communication strategy between the modules
and the ASF consists of a 32-bit communication integrating the I²C protocol for sensors.
Thanks to the ICMs units, it is possible to establish a communication with a total of 32
peripherals. In the present case, these 32 peripherals will be the sensor and the triad of
valves of every on of the eight modules of the GlassTec prototype (8 +24). The ICMs
are the responsible to decode the messages in order to regulate the data acquisition from
the sensors to the Arduino board and the proper opening and closing of the chambers
(see Figure 2.19).
Regarding the first eight bits of the message, every ICM will change the IMU address code accordingly. The IMUs have two addresses: 0x68 and 0x69. 0x68 is used as a virtual off-state, while 0x69 is used as the sensor address for data reading. Therefore, depending on the 8 most significant bits from the 32 string pattern, the shift register of the IMU will switch these addresses by changing its output to low (0V) or high (5V). Note that only one IMU at a time can be set active because of the emisor-receptor nature of the protocol, shown in Figure 2.20.
The sensing units of the facade are the MPU-9250. The MPU-9250 is a 9-axis MotionTracking device that combines a 3-axis gyroscope, 3-axis accelerometer, 3-axis magnetometer and a Digital Motion Processor™ (DMP), which directly provides complete 9-axis MotionFusion™ output.

When a reading procedure is initiated by the Arduino Board, the sensor data can be collected as raw or directly extracted from the DMP unit. If data is collected as raw, then some motion processing algorithms must be run inside the Arduino. If we expect accurate results and low latency, then the algorithm should be run at a high rate (often 200 Hz), however, the Arduino Board could become easily overwhelmed as the number of ICMs increases. Moreover, the performance of the system would still have to deal with the computation of the dynamic behavior of the components, explained in Section 2.4.2.

In order to offload both timing requirements and processing power from the main processor, the present data acquisition is performed by extracting complete 9-axis data from the FIFO buffers of the DMP units. It is necessary, however, to find an efficient way to loop all the ICMs and properly extract the values. In the present thesis, the library developed by Jeff Rowberg, i2cdevlib [42], has been used in cooperation with an adapted version to the MPU-9250 of the library MPU-6050, which can be found at the
same repository.

With that set, the microcontroller must cycle fast enough through all the 8 IMUs in order to avoid the FIFO buffer of the DMP units to overflow. The performance of the system is then directly related to the clock frequency of the \( I^2C \) protocol, which is governed by the Arduino.

For the case of the GlassTec prototype, the clock frequency has been set to 100kHz. Nonetheless, the inherent high capacitance of the system makes the internal pull-ups of the Arduino Board (between 20-50 kΩ) not suitable for this sampling rate. To solve that, external pull-up resistors have been added to the system, their value has been defined within the range proposed in the following formulas:

\[
R_p \in \begin{cases} 
R_{p_{\text{min}}} = \frac{V_{cc} - V_{0l}}{I_{0l}} \\
R_{p_{\text{max}}} = \frac{t_r}{C_b}
\end{cases}
\]  

Where \( V_{cc} \) is the nominal current, which for the Arduino Board is 5V, \( V_{0l} \) is the maximum voltage for logical low, set as 0.4V, \( I_{0l} \) is the maximum current for logical low, set as 3mA, \( t_r \) is the maximum rise time, which for 100kHz is 100 nanoseconds, and \( C_b \) is the global bus capacitance. Since the Arduino and every IMU introduce 10 pF to the system, the bus capacitance is estimated to be 90 pF considering all peripherals. Therefore, the value of the pull-up resistors is comprised within 1.5 kΩ and 11.1 kΩ. A mid-value of 4.7kΩ has been chosen for the pull-up resistors.

### Serial communication between boards

The serial communication between the boards is bi-directional. The Arduino Board sends the pitch and yaw values from the sensors to the Raspberry Pi. The Raspberry Pi calculates the appropriate command and sends the 32-bit message to the Arduino. Additionally, one more byte is sent for the control of the pressure inlet system.

The USB serial communication is set with a baud-rate of 2 Mbps. Notice that the Arduino Board is the unit responsible to set the clock synchronization of the whole system. The Arduino sends the sensor data every 100ms using its timer interruptions, therefore, the whole process becomes dependent to this time scope. The accuracy of the timing interruptions is governed by the clock in the Arduino board, which is driven by a crystal resonator. The crystal has an accuracy of approximately +/- 50 parts per million (ppm). For the case of 100ms, the deviation of the time scope would be 5\( \mu \)s.

#### 2.4.2 Low-level control algorithm of the prototype

To appropriately command the inflation and deflation of the triad of chambers inside every SRA, the present control strategy is based on the work of Svetozarevic et al. [2], which has been adapted to the nature of the GlassTec prototype.

Svetozarevic states that, when a single chamber is inflated, it will expand creating pressure on the top and the bottom disks (see Figure 1.3), hence causing the top platform
to rotate around the the remaining part that contains the other two chambers. Following that statement, in Figure 2.21, the application of this concept in the SRA of the GlassTec prototype is presented.

Figure 2.21: 2-angles-3-pressures mapping strategy for the SRA. Every chamber individual inflation or deflation defines a main direction of actuation regarding the pitch and yaw rotation. The triad of directions define 3 regions. The set of pitch and yaw rotations within every region can only be achieved when the chamber with the same number is totally deflated. Note that a positive pitch increases the panel tilt, while a positive yaw rotates it clockwise \[2\], edited.

Following the panel movement simulation from Figure 2.5, the low-level control is based on the yaw and pitch rotations transmitted when the chambers change of pressure. Taking the origin of the picture as the rest position of the panel i.e. all the chambers are deflated \((\text{yaw} = 0^\circ, \text{pitch} = 0^\circ)\), when chamber 3 is inflated or deflated, the panel
will rotate around the pitch axis decreasing or increasing the panel tilt, respectively. On the other hand, when chambers 1 or 2 change their pressure, the panel will rotate around the pitch and yaw axis, because their expansion is not aligned with the yaw axis of rotation. These characteristic directions of movement when a single chamber is being pressurized are called main directions of actuation. Likewise, by combining the simultaneous expansion or the deflation of two chambers, the main directions of movement can be extended to a total of 6.

The three main directions of actuation define a total of three zones, introducing the interesting concept of regions of actuation. As an example, it seems obvious that, when we want to decrease the panel tilt from the a rest position (i.e. introduce a negative pitch), the chamber number 3 should always be deflated, since any pressure inside it would increase the slope of the panel. A similar effect but combining both yaw and pitch rotations would happen with the other two chambers. Therefore, it is convenient to define a range of zones where these condition must be respected, and these are precisely the zones delimited by the main directions of actuation. As a result, the 3 zones have been named accordingly to the number of the pertinent deflated chamber.

From that point, the process of commanding a configuration to a panel starts with identifying the region of the desired position, then deflate the chamber of the region, and finally combine the two other main directions of actuation wisely to reach the pitch and yaw values as soon as possible.

Nonetheless, it is crucial to pay attention at the pressure level of the chambers and their inherent behavior. On the one hand, it is necessary to be aware of the pressure status of the chamber. In the present work the chambers are classified as deflated, pressurized or saturated. When a chamber is saturated, the system is informed that the chamber has reached its elastic limit. On the other hand, the present dynamic approach considers the distribution of chambers ideally symmetric, and the nonlinearities of the elastic components as well as their hysteretical behavior is not covered.

The danger of these last misconceptions can lead a desired position to be is assigned to a wrong region. The whole control system would then get stuck since the physical position needs a chamber of another region to be deflated and not the one assigned. To avoid this, a first run procedure of calibration is implemented, and some safety procedures are also covered.

In the first run procedure, the system inflates every chamber separately in order to adapt the main directions of actuation from the ideal conception of Figure 2.21 to a one closer to the physical prototype. The issue of wrongly assigning a region to a desired position is, however, still present, especially in the domains close to the boundaries of the regions. In that special case, the system will change the region assignment. Of course, this is only implemented as long as the desired position is physically feasible.
Results

3.1 Simulation Framework review

In the present section, the performance of the several subprocesses from the simulation framework will be checked. For the sun position, a comparison of results will be done with respect to a dedicated source. For the shading pattern, a qualitative validation will be made by comparing the geometrical results with a Grasshopper model combined with LadyBug. For the quantitative evaluation, the shading pattern plus the radiation subprocess will be compared with the simulation results of a Grasshopper model. For the PV supply and the Main Control algorithms, the results will be compared with a previous case study.

3.1.1 Sun position validation

For the proper comparison of the sun position calculation, the results have been contrasted with SunCalc.org [43], a web tool that integrates the sun path computation with a minute resolution and Google Maps. The angle mismatch obtained for the geolocation with latitude 47.48° and longitude 8.536° (Zürich, Kloten) for every hour of 2017 is presented in Figure 3.1.
The maximum deviations for the altitude angles are at around 1.4°, placing the average at 0.622°. Meanwhile, the maximum azimuth mismatch is placed at approximately 0.9°, having a mean of 0.504°. Interestingly, the largest errors are detected during the daylight savings time period, however, the deviations are still too small to consider an hourly offset between the algorithms. For the purpose of this project, the accuracy of the results is considered admissible.

### 3.1.2 Shading pattern and radiation

#### Qualitative analysis of the shading pattern

Since the shading pattern has a strong effect on the final output of the control algorithm, its shape has been firstly inspected with a visual comparison between the shading geometries generated using a simple model with GrassHopper and LadyBug. In the following pictures, it can be seen that the shading pattern output resembles with high accuracy the geometries from the software. The correct overlap of the geometries taking into account the position of the sun and the location of the panels can also be checked.
Figure 3.2: Qualitative comparison between Grasshopper shading pattern and script output. Single-cluster example. Panel neutral position: $\theta = 0^\circ$, $\psi = 0^\circ$. Sun position: $\alpha_s = 47^\circ$, $\gamma_s^{ASF} = -54^\circ$.

Figure 3.3: Qualitative comparison between Grasshopper shading pattern and script output. Single-cluster example. Panel neutral position: $\theta = 0^\circ$, $\psi = 0^\circ$. Sun position: $\alpha_s = 27^\circ$, $\gamma_s^{ASF} = -85^\circ$. 
Figure 3.4: Qualitative comparison between Grasshopper shading pattern and script output. Single-cluster example. Panel neutral position: $\theta = 0^\circ$, $\psi = 0^\circ$. Sun position: $\alpha_s = 39^\circ$, $\gamma_{ASF}^s = 69^\circ$.

Figure 3.5: Qualitative comparison between Grasshopper shading pattern and script output. Single-cluster example. Panel maximum degree of actuation: $\theta = 60^\circ$, $\psi = -40^\circ$. Sun position: $\alpha_s = 37^\circ$, $\gamma_{ASF}^s = 72^\circ$. 

45
The shading pattern has been developed with a degree of resolution of 1 mm, and from a qualitative point of view, it shows suitable for glare control operations. Nonetheless, a quantitative evaluation is proposed in the next section.

**Quantitative analysis of shading pattern and radiation**

The results obtained with the combination of the shading pattern script plus the radiation script have been compared with a simple Grasshopper model combined with LadyBug (see Figure 3.7).
The validation procedure is carried out by analyzing the outputs from the software and the script during one day of summer and one day of winter. In addition, a total of six different configurations are examined for every case. The positions correspond to the combination of the maximum pitch and yaw rotations with respect to the neutral position (see Figure 2.5). The studied models execute the configurations as a single cluster.

For coherency, the incident direct irradiation on the panels is separately studied from the diffuse component. This step is important, since the shading pattern mainly affects the direct component of the irradiation. It also helps to discern the high complexities from the diffuse calculation shown in Section 2.2.3. The total incident irradiation on the window is also examined.

Summer day analysis: direct irradiation on ASF In Figure 3.8, the global incident irradiation on the facade for every extreme configuration during a summer day is compared with the Grasshopper model. Because of the higher altitude of the sun, the facade is more susceptible to perform mutual shading, however, the results resemble the hourly changes with high precision, finding eventually large errors during midday (19%).

Figure 3.8: Incident Direct Irradiation on facade during a summer day: script vs Grasshopper model comparison. The images show the incident irradiation along the day for every extreme combination of panel pitch $\theta$ and the panel yaw $\psi$. Note: open position is $\theta = -30^\circ$, $\psi = 0^\circ$. Closed position is $\theta = 60^\circ$, $\psi = 0^\circ$. 

47
Summer day analysis: diffuse irradiation on ASF  The same procedure is applied for the incident diffuse irradiation as shown in Figure 3.9. The complexity of the events is high, however, the script still resembles the shape of the simulation results quite well. The largest errors are placed when the circumsolar effect achieves its maximum value. These errors are certainly magnified because of the high number of modules and some slight misalignment between the definition of the hour of the day from every source, and so the sun position.

![Graphs showing incident diffuse irradiation comparison](image)

(a) $\theta = -30^\circ$, $\psi = 0^\circ$  
(b) $\theta = -30^\circ$, $\psi = 40^\circ$  
(c) $\theta = -30^\circ$, $\psi = -40^\circ$  
(d) $\theta = 60^\circ$, $\psi = 0^\circ$  
(e) $\theta = 60^\circ$, $\psi = 40^\circ$  
(f) $\theta = 60^\circ$, $\psi = -40^\circ$

Figure 3.9: Incident Diffuse Irradiation on facade during a summer day: script vs Grasshopper model comparison. The images show the incident irradiation along the day for every extreme combination of panel pitch $\theta$ and panel yaw $\psi$. Note: open position is $\theta = -30^\circ$, $\psi = 0^\circ$. Closed position is $\theta = 60^\circ$, $\psi = 0^\circ$.

Summer day analysis: Irradiation on window  Finally, the incident irradiation on the window during a summer day is shown in Figure 3.10. The script shows a fair accuracy, placing the major errors in midday (20%).
Figure 3.10: Incident Direct Irradiation on the window behind the facade during a winter day: script vs Grasshopper model comparison. The images show the incident irradiation along the day for every extreme combination of panel pitch $\theta$ and panel yaw $\psi$. Note: open position is $\theta = -30^\circ$, $\psi = 0^\circ$. Closed position is $\theta = 60^\circ$, $\psi = 0^\circ$.

Winter day analysis: direct irradiation on ASF Following the same procedure as the summer day, in Figure 3.11, the incident direct irradiation on the ASF for every extreme configuration of the modules during a winter day is compared with the Grasshopper model. The results show generally a good precision, finding the largest errors (15%) during midday.
Figure 3.11: Incident Direct Irradiation on facade panels during a winter day: script vs Grasshopper model comparison. The images show the incident irradiation along the day for 6 combinations of panel pitch $\theta$ and panel yaw $\psi$. Note: open position is $\theta = -30^\circ$, $\psi = 0^\circ$. Closed position is $\theta = 60^\circ$, $\psi = 0^\circ$.

**Winter day analysis: diffuse irradiation on ASF** Likewise, in Figure 3.12, the same comparison is done for the diffuse component of the incident irradiation on the ASF. The errors, even though their magnitude is small compared to the direct component of the irradiation, are eventually large. These errors coincide when the circumsolar diffuse component become dominant.
Figure 3.12: Incident Diffuse Irradiation on facade during a winter day: script vs Grasshopper model comparison. The images show the incident irradiation along the day for every extreme combination of panel pitch $\theta$ and panel yaw $\psi$. Note: open position is $\theta = -30^\circ$, $\psi = 0^\circ$. Closed position is $\theta = 60^\circ$, $\psi = 0^\circ$.

Winter day analysis: Window In Figure 3.13, the comparison is applied in the context of the window. The results show an impressive good precision, justifying the less complexity of the sky view factor calculations. An eventual large error is found in Figure 3.13e, when the horizon brightening component becomes dominant.
Figure 3.13: Incident Direct Irradiation on the window behind the facade during a winter day: script vs Grasshopper model comparison. The images show the incident irradiation along the day for every extreme combination of panel pitch $\theta$ and panel yaw $\psi$. Note: open position is $\theta = -30^\circ$, $\psi = 0^\circ$. Closed position is $\theta = 60^\circ$, $\psi = 0^\circ$.

3.1.3 PV Supply, RC building simulator and main control algorithm

Since the previous Grasshopper model does not include the experimental power losses of the panels and the subsequent building simulation, the final subprocesses of the new framework have been compared with a case study conducted in Schmidli and Luzzato’s work.
## Parameter

### Building Dimensions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Width</td>
<td>4.9</td>
<td>m</td>
</tr>
<tr>
<td>Depth</td>
<td>7</td>
<td>m</td>
</tr>
<tr>
<td>Height</td>
<td>3.1</td>
<td>m</td>
</tr>
<tr>
<td>Horizontal Glazing</td>
<td>91.83</td>
<td>%</td>
</tr>
<tr>
<td>Vertical Glazing</td>
<td>96.77</td>
<td>%</td>
</tr>
</tbody>
</table>

### Building System

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heating COP</td>
<td>3</td>
<td>–</td>
</tr>
<tr>
<td>Cooling COP</td>
<td>3</td>
<td>–</td>
</tr>
<tr>
<td>Lighting Load</td>
<td>11.74</td>
<td>W/m²</td>
</tr>
<tr>
<td>Infiltration Rate</td>
<td>1.5</td>
<td>1/h</td>
</tr>
<tr>
<td>Fresh Air</td>
<td>0.016</td>
<td>m³/s per person</td>
</tr>
<tr>
<td>Equipment</td>
<td>4</td>
<td>W/m²</td>
</tr>
<tr>
<td>Heating Setpoint</td>
<td>22</td>
<td>°C</td>
</tr>
<tr>
<td>Cooling Setpoint</td>
<td>26</td>
<td>°C</td>
</tr>
<tr>
<td>Lighting Control</td>
<td>300</td>
<td>lux</td>
</tr>
<tr>
<td>R-Value of Exterior Wall</td>
<td>3.61</td>
<td>K · m²/W</td>
</tr>
<tr>
<td>R-Value of Window</td>
<td>0.585</td>
<td>K · m²/W</td>
</tr>
<tr>
<td>Solar Heat Gain Coefficient of Window</td>
<td>0.691</td>
<td>–</td>
</tr>
<tr>
<td>Visible Light Transmittance of Window</td>
<td>0.744</td>
<td>–</td>
</tr>
<tr>
<td>Occupancy</td>
<td>Weekdays from 8:00-18:00</td>
<td></td>
</tr>
</tbody>
</table>

### Façade Dimensions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Panel Size</td>
<td>400</td>
<td>mm</td>
</tr>
<tr>
<td>Panel Spacing</td>
<td>500</td>
<td>mm</td>
</tr>
<tr>
<td>Panel Offset</td>
<td>400</td>
<td>mm</td>
</tr>
<tr>
<td>Number of Panels</td>
<td>50</td>
<td>–</td>
</tr>
<tr>
<td>Number of rows</td>
<td>9</td>
<td>–</td>
</tr>
<tr>
<td>Number of columns</td>
<td>11</td>
<td>–</td>
</tr>
</tbody>
</table>

### Location Assumptions

- Weather File: Zurich-Kloten, Switzerland 2013
- Orientation: South

---

Table 3.1: Case study parameters for the House of Natural Resources
For the optimal configurations during winter, both scripts output a really similar set of angles. Since in the former framework estimated a higher heating demand, the electrical production is less intense. The obtained results for a winter day are shown in Figure 3.15.

Figure 3.14: ASF optimal configurations in terms of building net energy consumption for a winter day. Single-cluster example

Figure 3.15: ASF optimal configurations in terms of building net energy consumption for a summer day. Single-cluster example
3.1.4 Simulation performance

The execution time of the new simulation framework and its subprocesses running a Intel Core i7 processor with 3.40 Ghz is shown in Table 3.2. For a proper comparison with the former simulation environment, the execution times have been studied with only 49 feasible configurations. The literature shows that the previous simulation environment took 20 minutes to complete the optimization algorithm for building’s net energy consumption.

<table>
<thead>
<tr>
<th>ASF feature</th>
<th>Intel Core i7</th>
<th>Raspberry Pi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature Control</td>
<td>5.83</td>
<td>10.73</td>
</tr>
<tr>
<td>Electric Supply</td>
<td>5.8</td>
<td>10.73</td>
</tr>
<tr>
<td>Net Energy Consumption</td>
<td>6.02</td>
<td>11.14</td>
</tr>
</tbody>
</table>

Table 3.2: Execution time of the main control algorithm features. Comparison of performance between computer and Raspberry Pi.

The results show, for the same time scope of 20 minutes, that the algorithm is capable to compute the estimated net energy consumption to up to 9,800 different configurations of the facade. Because of its lower processing capabilities, the Raspberry Pi almost doubles the time to execute the algorithms.

3.2 Performance on prototypes

3.2.1 Data acquisition

The present scheme outputs an average value of 88 samples per second, eventually throwing 1 or 2 corrupted samples. After 4 hours of running, it has been proved that the system presents a stable lower bound of 86 samples per second. In Figure 3.16, the stability of the readings during 10 minutes is shown.
Figure 3.16: Samples per second read from the Arduino board during 10 minutes.

Once the sampling rate reaches a reliable value, then the quality of the sensed data is checked. In Figure 3.17, the stability of the sensor readings when the modules are in rest position is shown.

![Data acquisition stability graph](image)

Figure 3.17: Stability analysis for the pitch and yaw rotations during one hour in rest position.
Both images show a maximum deviation range of 0.05°. The stability of the sensed data is also checked when an abrupt movement is performed on the modules.

![Drift analysis of the pitch](image1.png) ![Drift analysis of the yaw](image2.png)

(a) Drift of the pitch angle  
(b) Drift of the yaw angle

Figure 3.18: Drift analysis for the pitch and yaw rotations during one hour. The panels are eventually moved manually to check the robustness of the sensors

The maximum drift is placed at around 1.5°. Therefore, this value will act as the boundary of precision. Finally, the imperfections of the chambers as well as their nonlinearities are resembled with a first run procedure when the ASF switches on. By inflating the chambers number one and two separately, the region boundaries are defined:
Figure 3.19: First run procedure: calibration of regions with experimental values. The red lines represent the boundaries of regions one and two proposed from the 2-angles-3-pressures model assuming a perfectly symmetrical SRA.

3.2.2 Usage of processing units

The sketch uploaded in the Arduino board occupies a total of 14720 bytes, which is approximately the 45% of the memory space for programs. The global variables inside it use 638 bytes (31%) of the dynamic memory, leaving 1410 bytes for the local variables.

The regulating unit is then, clearly not overwhelmed, and some further processing of the data to for example attenuate the effect of the wind can still be implemented. The python scripts occupy approximately 150 kilobytes.

3.2.3 Responsiveness of the system

The implementation of the low-level control has been successfully implemented for the individual monitoring of the modules. The complete inflation of one chamber would take approximately 10 seconds, and depending the change of position, the value can be increased to 20 seconds. Therefore, the positioning of 8 modules would take 2.5 minutes in a worst case scenario. This responsiveness can be acceptable when autonomous features of the facade are pursued, but when user interaction is intense, parallel control must be introduced.
In Figure 3.20 one can observe how well does the low-level algorithm regulate the position of the panel to a desired one.
Discussion

In the present section, the enhancement of the simulation framework and the performance of the prototypes will be discussed.

4.1 Enhancement of the simulation framework

The development of the new simulation framework eliminates the restricting timing issues and makes the implementation of a simulation environment to a physical prototype feasible.

Regarding the precision of the framework, with respect to the geometrical part, i.e. the direct components of the irradiation and the influence of the shading pattern, the accuracy of the framework with respect to other simulation tools is effectively high.

With respect to the diffuse components of the irradiation, even though their magnitude is generally lower than the direct component, their accuracy is questionable and some more work should be done in that direction. Nonetheless, the hourly changes are still correctly modeled by the scripts.

Despite the mismatches between the framework and the simulations, the fact is that the simulation framework is simply an approach, and is determined to sacrifice some precision in exchange of a drastic reduction of the computational time. From that point of view, the framework succeeds in that aspect reducing the computing period of 49 configurations from 20 minutes to 6 seconds.

Furthermore, even if the framework matched the simulations with high precision, there’s still no direct contrast with real data from the prototypes. Therefore, before digging into depth with the diffuse mismatches, a sensitivity analysis should be done with the physical prototypes of the facade.

4.2 Performance of the prototypes

With respect to the sensors, the new control scheme introduces the Arduino board as the master of the $I^2C$ protocol. The synchronization of the bus shows a stable value of 86 samples per second, which is really reasonable for a precise calculation of the position of 8 modules. For a larger number of panels, it is possible that the Arduino
Uno processing power might not be enough. In that case, there are alternative boards like Teensy, which runs at higher frequencies and its IDE is still compatible with the already developed Arduino sketch. On the other hand, a larger number of modules will increase proportionally the $I^2C$ bus capacitance. Generally, a value higher than 400 pF is not recommended. This effect has to be taken into account when choosing the pull-up resistors and the clock frequency.

Regarding the data acquisition strategy, the DMP units from the sensors relieve the system from processing the raw data from the IMUs, and it’s been shown that their deviations of 1.5° are completely admissible within the scope of the present thesis.

With respect to the responsiveness of the system, the needing of twenty seconds to perform a complete movement of a panel (worst case scenario) is a matter of discuss. Essentially, this constraint would only be acceptable as a steady-state feature, since it corrects the position of a single panel as fast as possible and does not influence with any other. When user interaction is not really intense, as in the case of the autonomous features, the constraint becomes dependent to the number of panels. The larger the number of modules, the more time needed to implement a new configuration. Finally, if a comfort feature is desired from the inhabitant, let’s say for instance that it wants to see outside (view control), making the user await for minutes is certainly inadmissible. This makes necessary to implement parallel control on the modules. Nonetheless, the implementation of this technique is not simple, since there’s only one pneumatic source for all the panels, hence the the inflation and the deflation of the chambers have to be synced.

4.3 Limitations of study

The present thesis has implemented the control scheme to 8 modules, which is the maximum supported with the 32-bit communication protocol. If the number of modules is greater, for example 16, the protocol would have to be extended to 64-bit and some more complexities would have to be dealt with. With respect to the low-level control, the parallel control of panels has only been introduced as a preset function for rapid opening and closing of clusters.

The simulation framework solves the optimization problems by brute-force, this is by looping through all possible configurations of the facade. An enhanced optimization procedure could decrease the computational time drastically.

The glare control feature has not been explored in depth. Although the shading pattern calculation greatly reduces the complexity of the feature, there’s still a high interaction with the occupant to be studied, and the GlassTec prototype is not a proper example to implement that application.
Conclusion

The present thesis has successfully designed and implemented an adaptive control strategy to the physical ASF prototypes using an Arduino board and Raspberry Pi. Within it, an enhanced simulation framework which reduces dramatically the computational time has also been developed.

The control algorithm is able to process three different kinds of inputs from the inhabitant, ordered by priority: direct command of a module configuration, selection of a comfort feature, selection of an autonomous feature. The control scheme, implemented with an Arduino Uno board and a Raspberry Pi, establishes the several facade clusters and computes the appropriate configuration which matches the user requirements.

The computation of the appropriate configuration is implemented with the enhanced simulation framework, which mainly contains the following five subprocesses:

- Sun position: Calculation of the sun position relative to the facade with a maximum deviation of 1.4°.

- Shading pattern: Calculation of the shadow lying on the window behind the facade and the mutual shading between panels with 1 mm resolution.

- Radiation: Estimation of the incident solar irradiation on the panels as well as the solar gains and the illuminance through the window. The direct components of the irradiation show a maximum deviation of 20% with respect to simulations with LadyBug.

- PV supply: Estimation of the electricity production of the facade taking into account experimental power losses due to mutual shading.

- RC Building simulator: Integration of a resistance-capacitance simulator of the building behind the facade.

Main control algorithm: Interpretation of the user inputs, establishment of clusters and efficient execution of the previous subprocesses.

In doing so, this thesis demonstrates:

- The complete design of a standalone complex radiation which include aspects of solar self shading that can operate on a Linux/UNIX operating system.
• Compared with the former simulation framework, the computation time of 49 ASF configurations has been reduced from 20 minutes to 6 seconds.

The implementation of the control system takes advantage of the integrated Digital Motion Processors of the Inertial Measurement Units, which relieve the system from processing raw data. The stability analysis of the sensing data demonstrates the suitability of the sensors after showing a deviation range of 0.05° in rest position and maximum drifts of 1.4° when abrupt movements are performed.

In worst case scenario, the complete movement of one panel takes 20 seconds. For steady-state purposes, this timing could be acceptable, but for features which user interaction is important, parallel control must be introduced.

This thesis ultimately sets the knowledge base, and software to bring the ASF research to an experimental level on the physical prototypes.
Outlook

The continuation of the present work would require to explore in depth the possibility to control in parallel various panels. The responsiveness of the system is crucial when the inhabitant interaction with the facade is high. As seen in the present thesis, the current low-level control algorithm is only suitable for steady-state configurations and some features where the user interaction is less intense.

On the other hand, the present thesis encourages to sense the ASF in order to define how accurate is the simulation framework, and how well helps the facade to properly adapt its shape to the changing conditions.

The inclusion of the Digital Motion Processors of the sensing units as the source for closing the feedback loop has decreased notably the stress on the microcontroller. The present project would also recommend to explore in more depth the capabilities of these units in a larger ASF prototype.

Finally, a long-term vision of the project proposes the implementation of machine-learning techniques to the facade, so it can finally adapt to the user requirements without even needing them to execute commands.
Appendices
A.1 GitHub repository

All the work presented in this Master’s Thesis is available at the ASF Control repository in GitHub: https://github.com/architecture-building-systems/ASF_Control. A lot of effort has been dedicated with the aim to easily transfer the content of this project.

The repository organizes all the content of the thesis in two main folders: the GlassTec implementation and the ASF simulation framework.

Figure A.1: General overview of the folder structure of the GitHub repository.

Inside the GlassTec implementation folder, you can find the scripts implemented in the Raspberry Pi and the Arduino Board that monitor and regulate the ASF.
Inside the ASF simulation framework folder, you will find the 5 subprocesses organized in 5 folders. For every script, a replica with the prefix `dev` has been uploaded in order to ease the understanding of the code. The replicated versions are not optimized but count with a high amount of commentaries and plots. Inside every process you will also find unittest files to check the proper functionality of the codes.
Bibliography


Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Design and implementation of a control algorithm for the Adaptive Solar Facade

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s): Domènech Masferrer

First name(s): Joan

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date
Zürich, 18/03/2018

Signature(s)

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.