

## TRUSTING AUTOMATION IN ARCHITECTURAL AND ENGINEERING WORKFLOWS

*A case study into core wall optimisation for multi-storey buildings*

M.RIZOSKI,  
*University of New South Wales, Sydney, Australia*  
*michael.rizoski@outlook.com*

**Abstract.** Automation has become a useful tool for designers as it removes tedious tasks, yet the architectural and engineering industries still struggle to implement automation into design workflows. This paper attempts to identify barriers to automated workflows and, through a case study of core wall optimisation, develop an application of automation in a pre-existing engineering workflow. Sakr and Johnson (1991) identify the barriers to automation being; lack of understanding of computer technologies, powerful software, but also, trust. As our understanding of computer technologies and increasing powerful software has become available, the barrier that still exists is trust. Interestingly, a 2019 study revealed that 64% of people “Would trust a robot more than their manager” (Oracle and Future Workplace 2019, p. 10). Proving that now is the time to start implementing automation tools. The case study is automating core wall optimisation. The core wall (or shear wall) is used in multi-storey buildings and “...are designed to resist lateral loads by planar wall or core elements.” (Alexander et al. 2984, p. 8). The case study uses a combination of Rhino, Grasshopper, Karamba, and Excel to produce this new workflow and Robot to test the outputs against an industry standard tool. Applying automation to design workflows can have a significant impact on a designer’s ability to be iterative and search for optimal designs that address a range of economic, regulatory, and social and cultural issues. The outcomes of this research will contribute valuable knowledge and understanding of the implication of addressing the issue of trust in automated workflows.

**Keywords.** Automation, Optimisation, Core wall, Trust, Multi-storey buildings, Structural analysis, Computational design

## 1. Introduction

Automation has become a useful tool for designers as it removes tedious tasks, yet the architectural and engineering industries still struggle to implement automation into design workflows. This paper aims to identify barriers to automated workflows and, through a case study of core wall optimisation, develop an application of automation in a pre-existing engineering workflow. Automation in design isn't a novel concept as back in 1991 Sakr and Johnson noted that "Automation as a strategy...has created problems that cannot be resolved unless new approaches are used in the design and implementation of CAD systems." (p. 29). The problems referred to by Sakr and Johnson resulted from a lack of understanding of computer technologies, powerful software, but also, trust. As our understanding of computer technologies and increasingly powerful software has become available, the barrier that still exists is trust. Interestingly, a 2019 study about AI in workplaces revealed that globally 64% of people "Would trust a robot more than their manager." (59% in Aus/NZ) (Oracle and Future Workplace 2019, p. 10). Given that the robot and human relationship is stronger than ever, now is the time to start implementing more of these automation tools into the workforce so that trust can continue to improve.

The case study will explore; how to identify an area to automate, construct a workflow to automate it, and implement it into the design process whilst addressing the issue of trust. The workflow to be automated will be core wall optimisation. The core wall (or shear wall) is used in multi-storey buildings and "...are designed to resist lateral loads by planar wall or core elements." (Alexander et al 1984, p. 8). Currently most engineering consultancies assign the task of building models in software to graduates under supervision as it is laborious and time intensive. Any architectural changes may require the model to be altered or even rebuilt. Even though it is complex I believe these equations can be translated to an automated script.

The workflow created in this case study will use Rhino and Grasshopper to create a parametric script that can produce the initial form of the building and core wall. Karamba and Excel will then be used to perform analysis and run the equations that are required to perform the optimisation. Lastly, Robot will be used to test the outputs from the automated workflow against an industry standard tool (Robot).

This aims to create a workflow for core wall optimisation that eradicates manual and repetitive tasks but is also one that can be validated by Robot to give users confidence in the final output.

Applying automation to design workflows can have a significant impact on a designer's ability to be iterative and search for optimal designs that address a range of economic, regulatory, and social and cultural issues. The outcomes of this research will contribute valuable knowledge and understanding of the implication of addressing the issue of trust in automated workflows.

## 2. Research Aims

Based on the aforementioned issues and opportunities for automation in the AEC industry, this research aims to investigate automated workflows in engineering design processes, and how issues of trust factor as a barrier to implementation. More specifically, this aims to design and create a tool to automate the core wall optimisation workflow.

## 3. Research Question(s)

The question this research project investigates is:

In what ways can automation be implemented in architecture and engineering design workflows for core wall optimisation in multi-storey buildings?

How can verification or trust be built into an automated workflow for core wall optimisation in multi-storey buildings?

## 4. Methodology

The project's research methodology combines tenets of action research and design research to undertake an action design research investigation that engages an industry partner to help define the problem and contribute to its investigation.

Action research (AR) "...is fundamentally a change-oriented approach in which the central assumption is that complex social processes can best be studied by introducing change into these processes and observing their effects. (Baskerville 2001)." (Cole et al. 2005, p. 326). It is a cyclical process that involves, according to Susman and Evered (1978), five stages; "The first phase, diagnosing, is aimed at identifying and defining a problem. The second, action planning, involves considering alternative courses of action for solving the problem. The third, action taking, consists of selecting a course of action. The fourth, evaluating, is aimed at studying consequences of the action. The fifth, specifying learning, completes the loop by identifying general findings." (Cole et al. 2005, p. 327). The action and collaboration are the key parts of action research.

Design Research (DR) "...consists of activities concerned with the construction and evaluation of technology artefacts to meet organisational needs as well as the development of their associated theories." (Cole et al. 2005, p. 326). In this case the focus is the production of the artefacts that can solve the problem. "Designers observe, describe and interpret the world differently that other disciplines such as ethnography, economics, or engineering...we approach these from the key perspective/world view of changing existing situations into desired ones (Simon 1969)" (Findeli 2010,

p. 291). In this project the specific perspective will be from a Computational Design perspective to address problems with core wall optimisation.

Through this I have chosen a combination of both Design and Action Research as the research will be iterative and address an activity within an organisation to address the needs in a real-life context.

#### 4.1. METHOD

The method used to conduct the research is iterative in exploration development using parametric scripting in Grasshopper to create a workflow for core wall optimisation. It also uses the traditional workflow as a guide framing the areas that need to be achieved before, during and after the script.

### 5. Background Research/Literature review

The integration of automated workflows in the architectural and engineering industries is advancing, however is being confronted and constrained by issues of comprehension and trust. I will be conducting a case study to explore this.

The following are key areas of understanding and literature that relate to the research conducted in this paper.

#### 5.1. GENERATIVE DESIGN

Autodesk defines generative design as “A goal-driven approach to design that uses automation to give designers and engineers better insight so they can make faster, more informed design decisions. Your specific design parameters are defined to generate many...potential solutions. (and) with your guidance it arrives at the optimal design.” (Autodesk 2019, p. 2). Autodesk is the main producer of CAD (Computer Aided Design) software for architects and engineers and has been doing so for many years. They have seen the importance of moving towards implementing automation tools in its software. This short paper shows how important automation is to the future of design and therefore needs to be researched further.

A lot of the applications of generative design and the tools so far have been to organise building layouts according to specifications and form finding of buildings to generate interesting and novel forms (Al-Qattan et al. 2017; Anderson et al. 2018; Alfaiate and Leitão 2017). Yet, I still haven't seen a combination of generative design with traditional methods that will allow designers to still produce something to industry standards with all the benefits of generative design.

#### 5.2. AUTOMATION

Automation is a concept of generative design and is defined by Zuboff (1988) as “the implementation of computer technology in a way that

emphasises standardisation, specialisation, and centralisation.” (quoted in Sakr and Johnson 1991, p. 16). Furthermore, “In architectural firms, automation results in the implementation of CAD systems in order to achieve productivity gains by substituting automating procedures for traditional, manual approaches.” (Sakr and Johnson 1991, p. 16). For architects and engineers, the idea of combining their skills with automation has been around for decades, especially with the rise of CAD. Yet there have always been problems with implementing it into design workflows. “Automation as a strategy for implementing information technology has created problems that cannot be resolved unless new approaches are used in the design and implementation of CAD systems.” (Sakr and Johnson 1991, p. 29). The paper also defined the main problems with automation in architecture at the time which were;

1. Over emphasis on structured design processes
2. Under emphasis of the social and organizational aspects of design
3. Over emphasis of explicit vs tacit design knowledge
4. Narrow definition of productivity
5. Marketing and image
6. Under emphasis on the culture of the firm”

(Sakr and Johnson 1991, p. 17)

These problems came down to the structures of architectural and engineering firm workflows as people still tended to work on paper and had just begun implementing computers into the mix. Therefore, if this research process was applied today it could open more possibilities due to the advancements in technology since then.

Scholars such as Nigel Cross have addressed the issue of how machines can design. He delves deeper to discuss the thought process behind getting a machine to design. His conclusion, “I believe that we can learn some important things about the nature of human design cognition from looking at design from the computational perspective.” (Cross 2001, p. 50). These foundational texts provide great insight into design workflows yet still can be added to, due to the changes in modern technology.

### 5.2.1. TRUSTING AUTOMATION

One other barrier to automation other than technology of the time is trust. Hoff and Bashir define trust as “It measures the degree of confidence individuals have in strangers for the degree to which romantic partners

believe in the fidelity of their significant other.” (Hoff and Bashir 2014, p. 409).

In the same way, the relationship between designers and their tools requires a level of trust otherwise design can become open to liabilities in the failure of the technology. Hoff and Bashir also provide some points, “...providing users with ongoing feedback concerning the reliability of automation...” and “...increasing an automated system’s degree of anthropomorphism, transparency, politeness and ease of use.” (Hoff and Bashir 2014, p. 429). Keeping these factors in mind whilst developing an automation tool can have a positive effect on whether it is accepted and therefore, is needed for the tool I will be developing.

A technique of implementing more control in the user was researched by Daniel Davis et al looking into the technique of modular programming. Modular programming being the identification of groups of code that performs a specific task and its inputs and outputs. Davis concludes that “All of these changes are relatively minor, with the benefit in legibility being particularly pertinent in collaborative environments where the model is being shared amongst many people.” (Davis et al. 2011, p. 67). This method can be seen working in projects where people are collaborating in the same environment, yet I would like to see how this can also be implemented to assist in the trust of automation processes where users can see the inputs, processes, and outputs.

### 5.3. OPTIMISATION

Optimisation is defined as “...the analyses’ results for different variations of the design as the functions to optimise.” (Belem and Leita0 2018, p. 550). This will be applied after the generation of a form to find the best possible form based on the criteria given and different standards and code that designers need to follow. Similarly to generative design, the novel ideas that have been appearing in previous years (Das et al. 2016; Bialkowski) still produce complex systems that don’t take into consideration the traditional methods and it is this combination that can attract more designers to apply it to existing design processes.

#### 5.3.1. CORE WALL OPTIMISATION

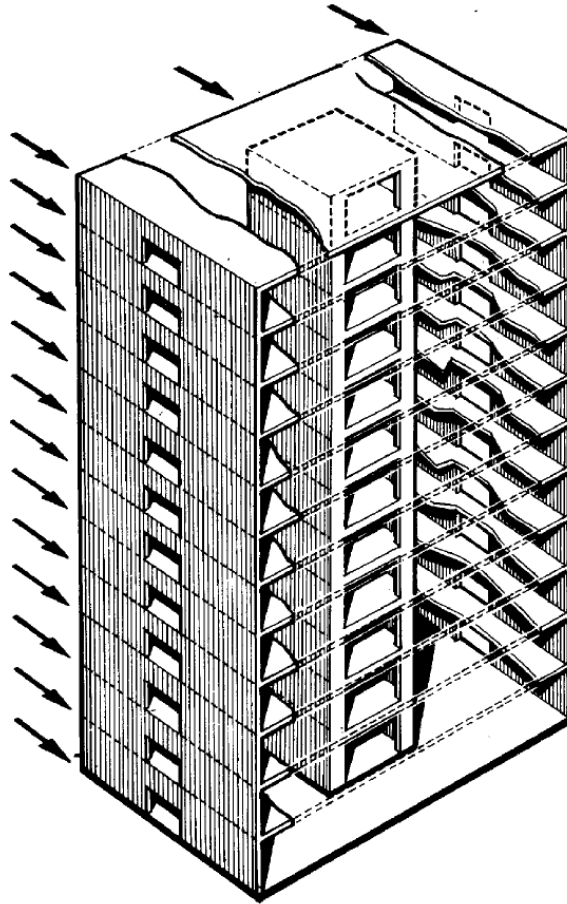
There are very limited resources that relates to both core walls and optimisation but a definition for a core wall, “To survive earthquakes forces in horizontal direction lift core wall gives required lateral strength and are transfer those horizontal forces to next element in the load path beneath them sufficiently. lift core wall provides stability against blast deriving, wind and seismic hazards as a structural system and avoid pounding of adjacent buildings in urban areas and shear failure.” (Varna and Bhavana 2017, p.

27). This is a vital part of high-rise structures and considers various loads to equate its final form. Previous research into trying to optimise core walls has been done but only in comparison to building shapes and heights (Tabassum 2014) (Varna and Bhavana 2017). These studies list the important engineering equations that relate to core walls and applies them to a few building shapes and heights to produce a list of recommendations for these building types. This can assist engineers but only to a limit. I intend to take the equations and processes documented in these papers and automate the process to produce optimal core walls quickly and for any high-rise building that is inputted.

## **6. Case Study**

### **6.1 GAINING INFORMATION**

My lack of knowledge on core walls meant I needed an industry partner to conduct the case study. My industry partner (Mott MacDonald) helped with explanations and resources that were my foundation to understanding what core walls are. One of these resources being “Design of shear wall buildings” (Irwin 1984).



*Figure 1. Core Wall Sketch – Irwin 1984*

Irwin defines shear (core) walls as “structures which are designed to resist lateral loads by planar wall or core elements... Shear walls offer a structurally efficient means of enclosing and utilising space. Their stiffness is such that sway movement under wind load can be minimised.” (Irwin 1984, p. 8). They are generally made from concrete and are created by pouring concrete into formwork and setting it as it continues up the height of the building.





*Figure 2. China Climbing Formwork for Core Wall*

Therefore, most core walls have levels that are grouped in 5 or 6 levels to have the same thickness as this saves time in construction. It goes into further detail about the difference forces that can act on the structure being:

- Dead load
- Imposed load
- Vibration and blast loading
- Wind Loading
- Seismic Loading

The two that were chosen to be analysed in this case study are dead load and wind loading. Dead load being the ability for the building to stand on its own with no other loads (other than gravity) applied to it. Wind loading is addressing the impact of the wind on the building and has an impact in the shape that the shear wall can have.

Next, I needed to understand how the traditional workflow for designing core walls is performed. The insight for this again was given to me by my industry partner and therefore could be different from firm to firm however the same principles still apply.

It starts with the architect handing over the design which can be in the form of a drawing, a 2D model, 3D model, etc. These variants mean that the next step for the engineer is to manually create a digital 3D model to be analysed in an engineering package. The average time it takes to create this

model is three days, which includes applying the materials and loads to the model.

The next step is running preliminary results in the engineering package, Robot, which can take a week, and once a conclusion has been made it is sent back to the architect to review and make changes, which is then sent back. The process of modelling starts over again. This process takes around one week per iteration.

In total this process can vary from a month to two months to complete and the turnaround for each iteration is very slow. The main areas my script will target is the generation of the 3D model and the analysis of the software.

## 6.2 CREATING THE GRASSHOPPER SCRIPT

The workflow is created in Rhino/Grasshopper as it is great at doing parametric design and has a plugin called Karamba that can do FEA (Finite Element Analysis).

The first area we need to address is what inputs are we taking into Rhino/Grasshopper? The script should take inputs that are available to the engineer from the architect to allow them to generate the model. In this case the scripts inputs are:

- Outline of exterior of the building – drawn in Rhino and imported using Human (A plugin for Grasshopper that imports layers from Rhino.)
- Outline of proposed core wall - drawn in Rhino and imported using Human.
- Floor to ceiling heights – specified by a slider
- Wall thickness – controlled by a gene pool, which is a component from Galapagos
- List of levels to group the thickness – controlled by a gene pool, which is a component from Galapagos
- Number of Levels – specified by a slider

Some notes on the inputs:

- The gene pool component is great for iterative design, but the number cannot be manually input which can be tedious for larger numbers and therefore, can be replaced with a panel for specific inputs
- The script also has a section with an MD (Multi-Dimensional) Slider that can be used to move the core within the extents of the building shell. This can be useful for iterative design but if the placement of the core wall is already defined then the original geometry should be used

These inputs are then used to visualise the core wall, building shell and floors for the building. This visualisation step is important as it can be used to visually bug fix the script or communicate the design to others.

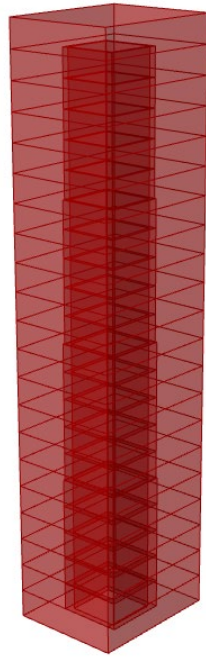


Figure 3. Visualisation of Core and Exterior Walls

### 6.3 PREPARING FOR KARAMBA AND PERFORMING KARAMBA ANALYSIS

For the analysis we need to be able to identify the walls in the x direction and y direction. This is done with a list item component as we know there is going to be four walls for the core.

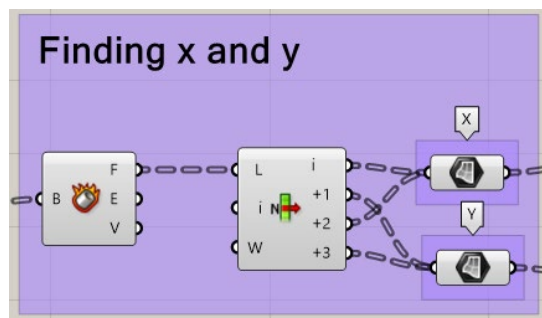


Figure 4. Finding X and Y

The core wall also needs to be turned into a mesh as Karamba only accepts meshes when analysing shells. There are many components that automatically convert Brep to meshes and in this case Mesh Brep was used.

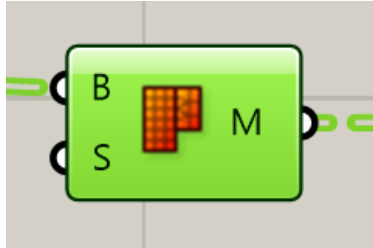


Figure 5. Mesh Brep Component

One last area is the labelling system. This is important as Karamba needs element ids to track where all the materials and cross sections are applied. As the only thing that changes about the model is the thickness at different groups of floors it was decided to label the model by levels.

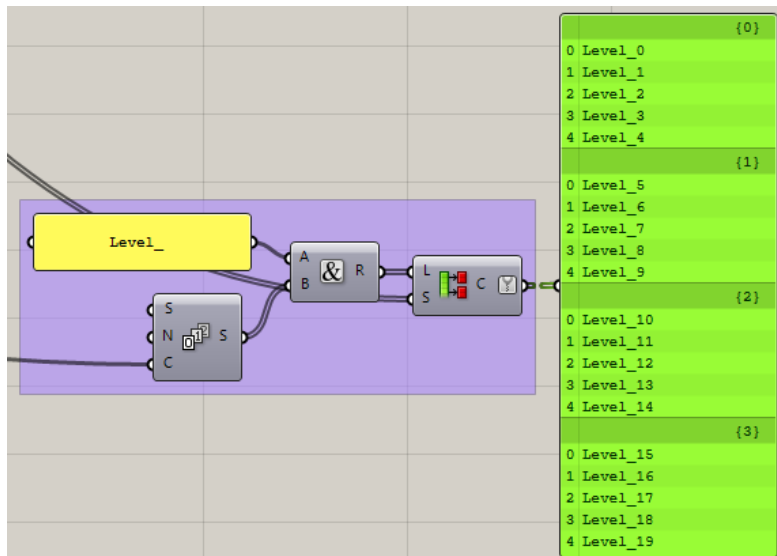


Figure 6. Labelling System

The structure of the data was a big challenge through this process, and we concluded that it should be made according to the structure of the level groupings of thickness.

Karamba has a component called Assemble Model that defines all that you need to perform this analysis. It requires:

- Elements – the geometry to be analysed
- Support – Location of supports
- Load – The loads to be applied to the elements
- Cross section – The type of cross section and its dimensions
- Material – The material to be used for the elements

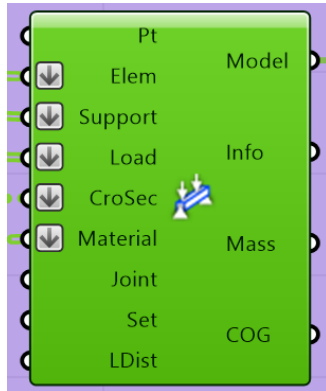


Figure 7. Assemble Model Component

Karamba then has components that allow you to choose how to input the data based on what you are trying to achieve. We are trying to perform shell analysis; therefore, the components need to correspond with this choice.

#### 6.3.1. Elements

Once the Brep is converted to a mesh all it needs to do is be input to the Mesh to Shell component.

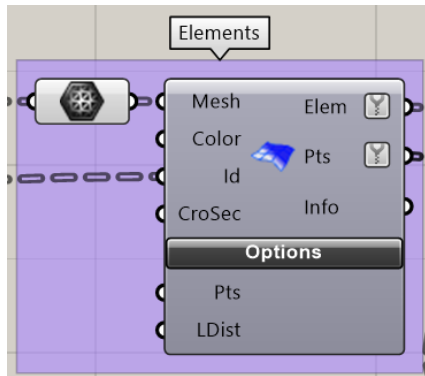


Figure 8. Elements for Karamba

#### 6.3.2. Supports

Support component takes in the point where the support is located on the mesh. Luckily the Mesh to shell component outputs the points of the mesh which we can use to find the points of the mesh at  $Z = 0$  (Ground level).

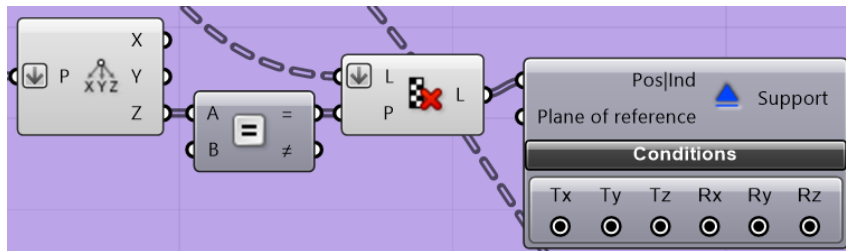


Figure 9. Supports for Karamba

### 6.3.3. Loads

In Karamba you need different load cases for different situations of stress that the building can be under. For this analysis there will be a load case for the X direction and a load case for the Y direction. Karamba has a load component where you can choose what type of load you want with different inputs for each. There is a gravity load that we can apply to both x and y load cases.

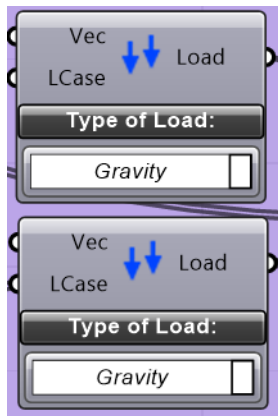


Figure 10. Gravity Loads

For the stresses on the building we need a script that calculates which was given to me by my industry partner. The output of this is then imported into the grasshopper script and then used to apply pressure to the core walls. The script takes the pressure from the excel spreadsheet and puts it into a MeshLoad component (one for each x and y).

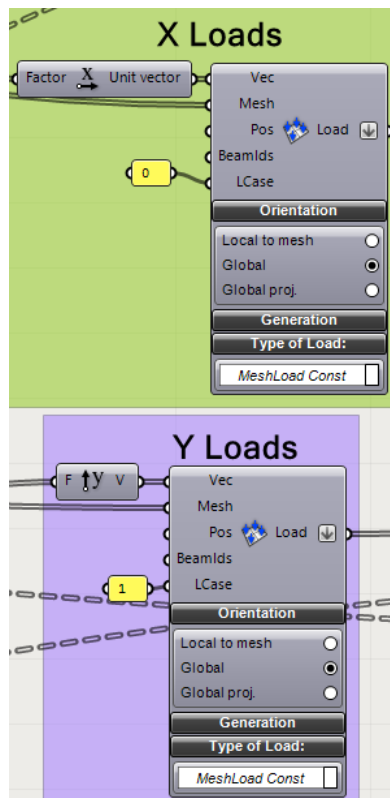


Figure 11. Mesh Load Component for Y Wind Loads

#### 6.3.4. Cross Section

The cross section takes thickness previously defined in the script and applies it to the element ids.



Figure 12. Cross Section Component

### 6.3.5. Material

The material selector component is the easiest to use as you can select a material family (concrete) and name (the grade).

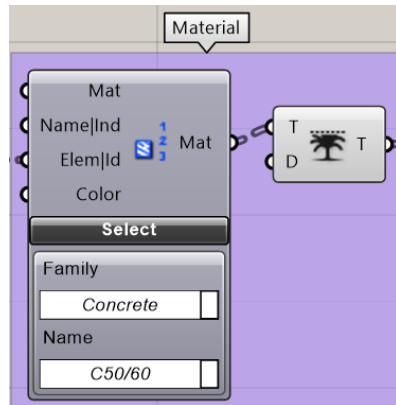


Figure 13. Materials for Karamba

The assembled model is then put into the Analyse component that completes the analysis.

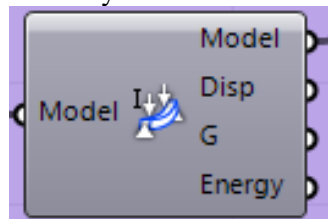


Figure 14. Analyse Component

## 6.4. RUNNING THE ANALYSIS IN ROBOT

When engineers are required to design structural elements, there are many load cases and combinations to consider. Large and complex structures are virtually impossible to analyse manually with accuracy and efficiency, and in today's market too expensive. The program lets the designer assemble a representational model of the real structure, to check critical design parameters.

In the example of a tall building, the use of Robot will allow the designer to check walls, columns, and slabs to ensure that these elements can withstand the code defined load actions. Locating a core wall within a building and sizing the wall group is the most critical design stage for a tall building. At the initial design stage, there will be multiple iterations and changes. One key advantage of using Robot as an analysis tool is that it has a fully open and useable API which allows programs such as Grasshopper to



access the model. This lets us push and pull information from Robot. The grasshopper and Karamba workflow presented aims to allow the engineer to quickly create and apply changes to their model, removing the need to spend hours manually changing geometry.

The geometry can either be manually created in Robot using CAD like tools or imported from a variety of sources including Grasshopper. Once the geometry is in the model, the next step is to define loads and loads cases. Lateral loads are what the building needs to resist to remain upright, these are usually wind or earthquake cases. Wind load calculation can be complicated and requires significant manual work to be applied to a geometry model in Robot. In the example, wind loads are generated from a spreadsheet that calculates the load at every level in accordance with the Australian Standard wind code. These loads are brought into grasshopper and used to automatically populate onto structural elements that are transferred to Robot. In this way, changes to the building geometry do not require the same manual reapplication of the loads every time.

Once loads and load cases are applied, the model can be “run”. This is where the structural software performs internal calculations to determine how the loads and geometry behave in each load case. The software determines forces and stresses that each individual element in the model experience. Once the model has been calculated, the software stores results for each object that allows the designer to interrogate the way each object is performing. Changes are made and the process is iterated and re-checked. Once all the elements are sized to resist all the loads and load cases, the design can be documented.

## 6.4. RESULTS

### 6.4.1. *Karamba*

The analysis in Karamba was run five times to test the script. The first of these five was an initial test and the rest were done after the script was finalised to gain more data.

Test 1: The first test was done alongside the creation of the script and used a random set of values for thickness to test the model. The thicknesses were 726mm, 631mm, 508mm, 388mm in order from base to top.

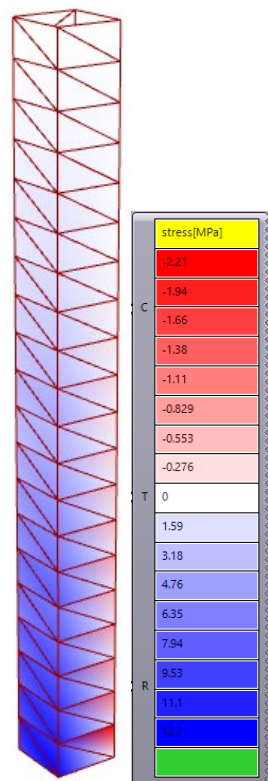


Figure 15. Stress for X, Test 1

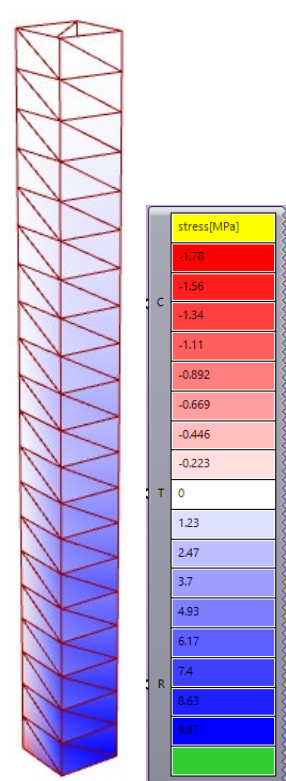


Figure 16. Stress for Y, Test 1

In each figure on the left there is a visualisation of the forces on the model and on the right is a legend for these forces. The blue represents the tension and red the compression.

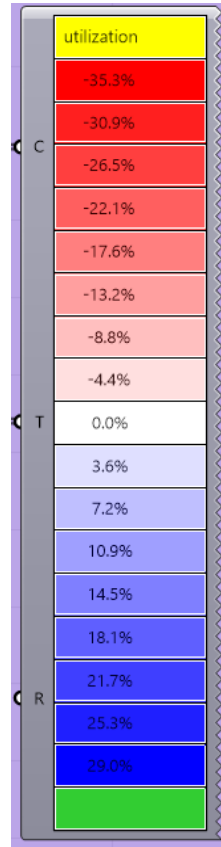


Figure 17. Utilisation for X, Test 1    Figure 18. Utilisation in Y, Test 1

The utilisation shows how much of the total weight the structure can withstand, is being used. Usually a percentage around 80 is desired to give some leniency for extreme stresses.

Tests 2, 3, 4 and 5: For the last 4 tests more reasonable thicknesses were set to test different ranges of thicknesses.

Test 2 = 300mm, 250mm, 200mm, 150mm

Test 3 = 350mm, 300mm, 250mm, 200mm

Test 4 = 400mm, 350mm, 300mm, 250mm

Test 5 = 600mm, 500mm, 450mm, 400mm

The results of these tests are as follows:

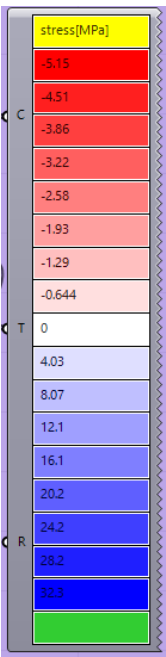


Figure 19. Stress for X, Test 2

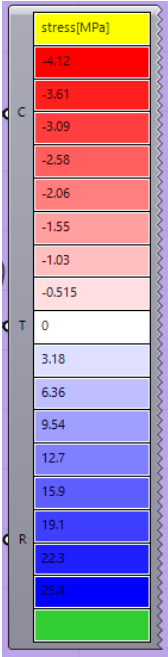


Figure 20. Stress in X for Test 2

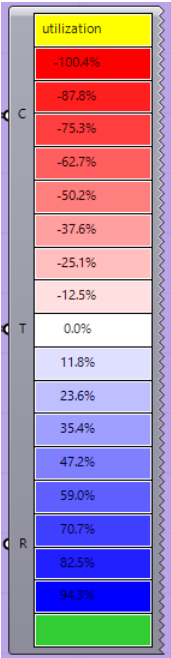


Figure 21. Utilisation for X, Test 2

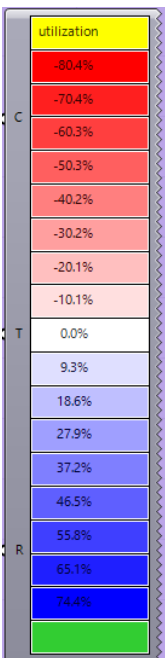


Figure 22. Utilisation for Y, Test 2

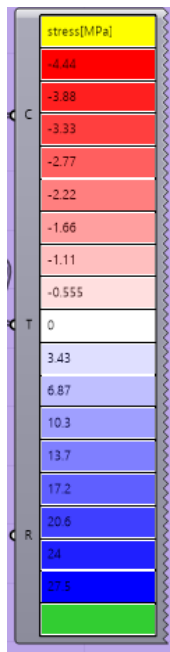


Figure 23. Stress for X, Test 3

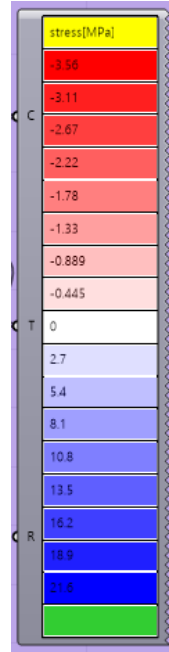


Figure 24. Stress for Y, Test 3

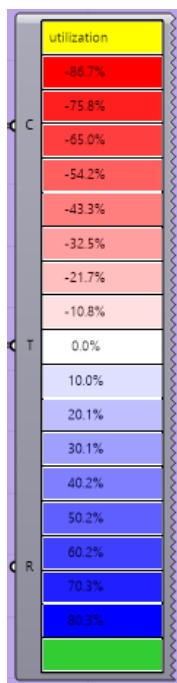


Figure 25. Utilisation for X, Test 3

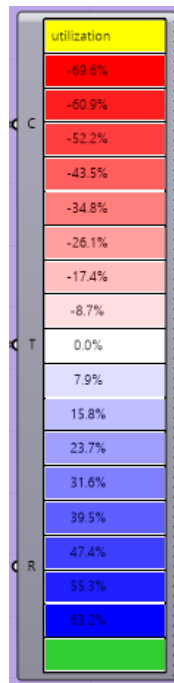


Figure 26. Utilisation for Y, Test 3

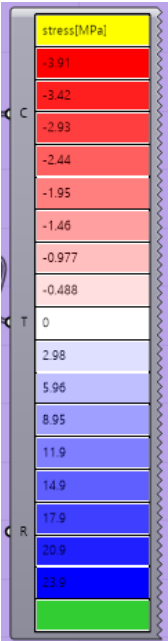


Figure 27. Stress for X, Test 4

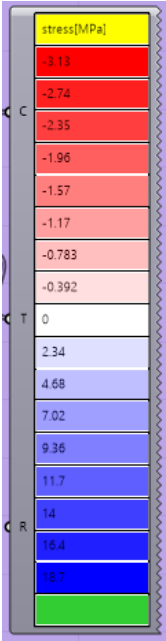


Figure 28. Stress for Y, Test 4

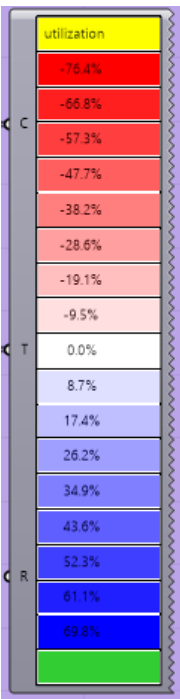


Figure 29. Utilisation for X, Test 4

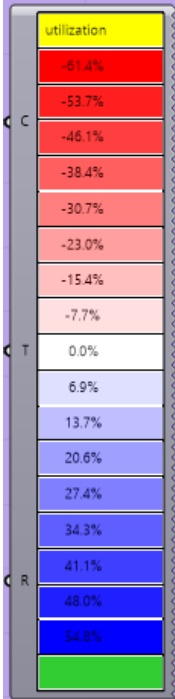


Figure 30. Utilisation for Y, Test 4

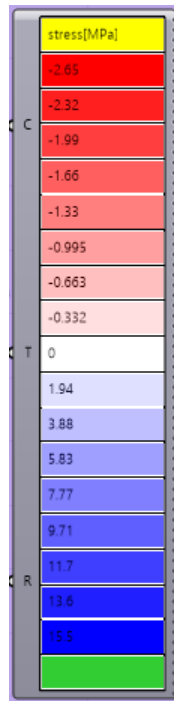


Figure 31. Stress for X, Test 5

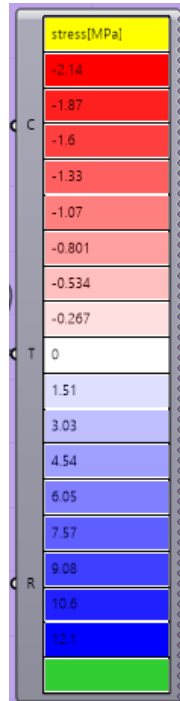


Figure 32. Stress for Y, Test 5

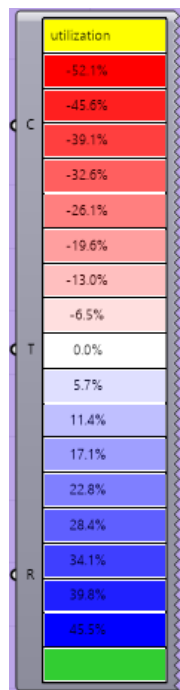


Figure 33. Utilisation for X, Test 5

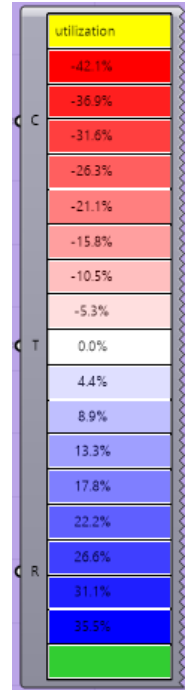
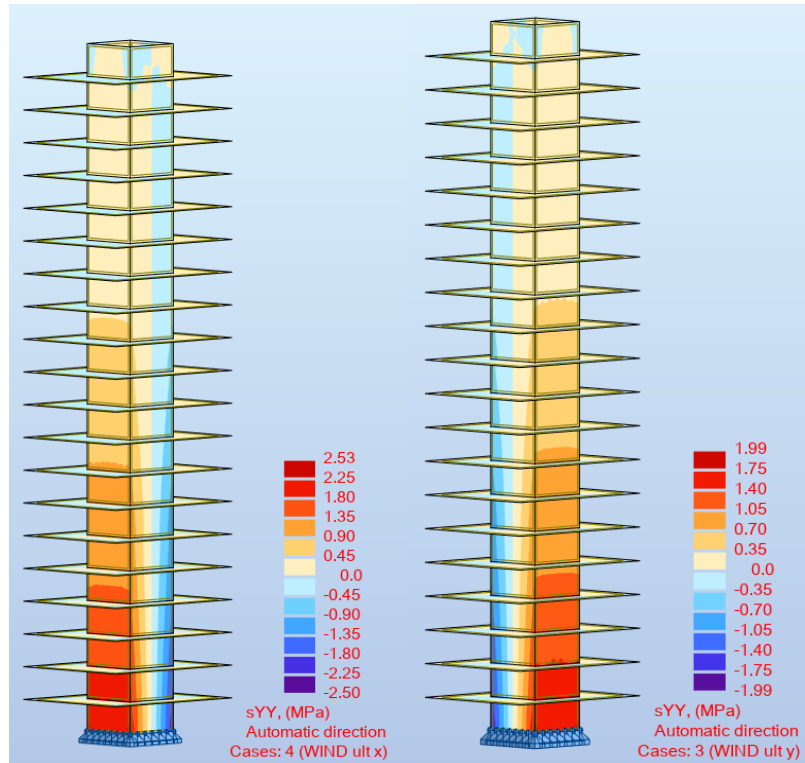


Figure 34. Utilisation for Y, Test 5

### 6.4.2. Robot

The analysis in Robot was performed once for test 1 and produced results in a similar way to Karamba.

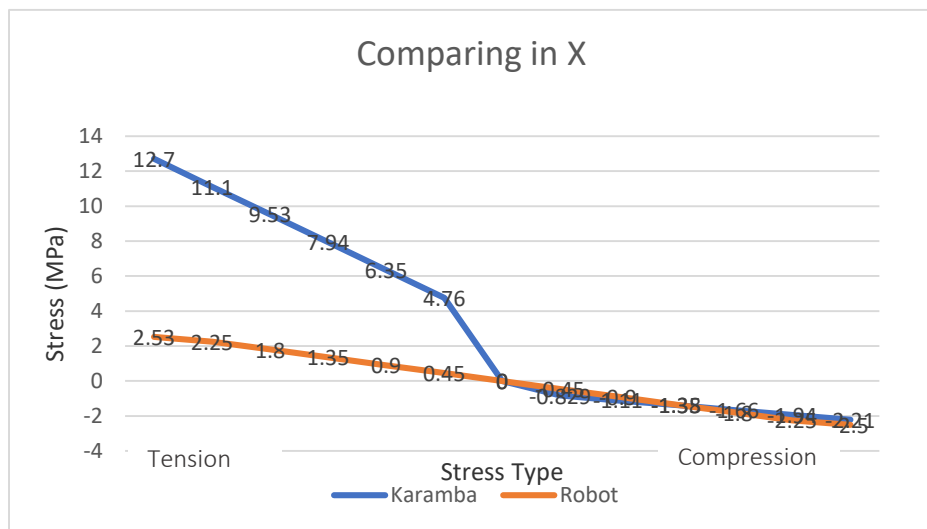
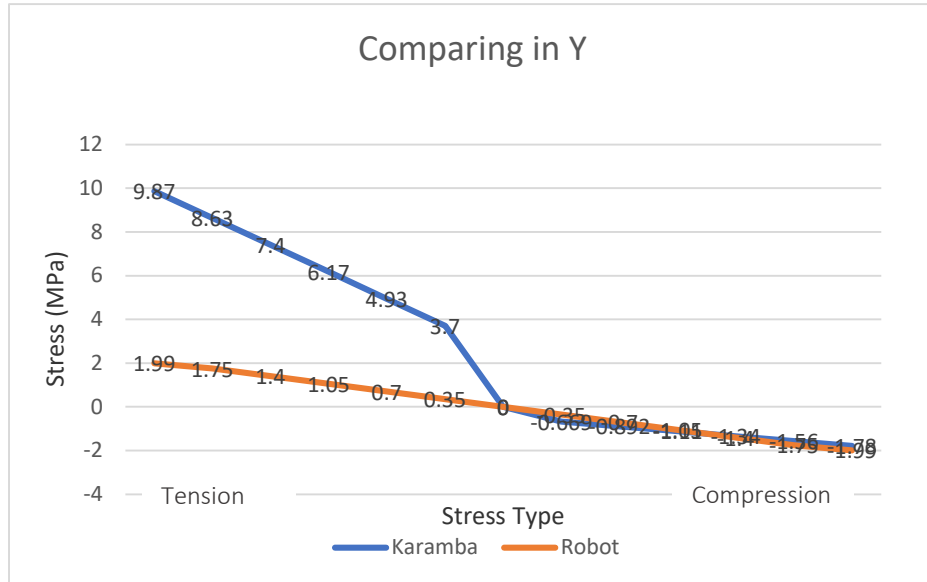


The results show a visualisation of the stresses on the building with a legend that shows the range of stresses in a similar fashion to Karamba. The stresses are in MPa. Due to time we did not get to analyse the other tests in Robot so this test will be used to compare to the Karamba.



#### 6.4.3. Comparing Karamba and Robot

The way we can compare the outputs of Karamba, and Robot is by comparing the legends of the stresses in both x and y to see how close they are. I decided to do this by plotting them on a line graph.



Both graphs show a difference in results for tension but a very close result for the compression. This could be due to a difference in the software's or part of the script missing something.

## 7. Discussion (evaluation and significance)

The research, through the case study, has produced a workflow for core wall optimisation in multi-storey buildings. It has also built in a way to verify the script using an industry standard engineering tool, Robot. Therefore, it has addressed the research questions previously outlined:

‘In what ways can automation be implemented in architecture and engineering design workflows for core wall optimisation in multi-storey buildings?’

‘How can verification or trust be built into an automated workflow for core wall optimisation in multi-storey buildings?’

This workflow addresses issues of trust in the engineering industry, proving that automation can contribute to workflows that struggle with repetitive and manual time-consuming processes. It also proves that both the traditional methods of completing these tasks and the modern solutions can coexist allowing a bridging of understanding as these workflows move into the future. It hasn’t been proven to increase trustworthiness as it hasn’t been tested. It also achieved the aims of the research

‘...to investigate automated workflows in engineering design processes, and how issues of trust factor as a barrier to implementation. More specifically, this aims to design and create a tool to automate a core wall optimisation workflow.’

Yet it has produced a new question being, what rate can we increase the adoption rate of automated workflows?

The limitations of the study include:

- Duration of research did not permit user testing
- Duration of research did not permit further exploration of computational design tools such as evolutionary solvers
- Duration of research did not permit time to apply all the engineering theory that goes into core wall optimisation
- Difficulty and duration of research did not permit applying further types of buildings to be analysed as it is limited to buildings with four exterior walls.
- Duration of research did not permit applying the layout of the core wall, including the locations of the lifts, service areas and emergency exits. The walls in these areas have a slight impact on the stability of the structure but didn’t need to be considered for this project.

Without user testing and the exploration of more complex computational design tools it cannot be concluded that this workflow increases trustworthiness, but it has been designed with previous research’s knowledge in mind which will have increased this rate it only depends by how much.

Future research into this area of study should address user testing of these workflows, applying more complicated computational tools, and combining

it with more engineering theory to see how much of the process can be automated. Also applying it to different weaknesses within the engineering and architectural industry to help increase the adoption rates of automated workflows.

## 8. Conclusion

Automation can be applied to an engineering workflow, such as core wall optimisation, in such a way that can increase trust between technology and its user. This is done by leaving the inner workings of the workflow open and explained to the user to help understand the process. Also designing the process in line with the traditional methods will help to bridge the gap between modern and traditional methods of design.

In this research this has been explored and shown using generative and parametric design within Rhino/Grasshopper and Karamba. Such computational design tools work well with automation and can be adapted to many different scenarios.

It also explored trust using an industry standard Finite Element Analysis software, Robot, to compare the outcomes of Karamba in order to prove it to be correct. Even though the script is limited in its ability to process a range of core wall types and building shapes, it has proven that part of this is possible and has laid valuable groundwork for further research and case studies into areas such as core wall layouts, evolutionary solvers as a tool for optimisation, and further complex engineering equations to be added to the script.

Looking at it broadly, this research and its case study can be used as a framework to identify a weakness in both architectural and engineering workflows and how to apply it into a workflow that may have been performed the same way for a long period of time.

I'd like to see these methods applied throughout firms as their value is not only a time and cost saving device but also its ability to assist designers in decision making for economic, regulatory, and social and cultural issues. The question is how quickly can we develop smart tools that designers can not only implement but trust?

## Acknowledgements

Thank you,

To my industry partner Mott MacDonald, Tony Ridley and Branko Cosic, for your support and guidance through the technical side of this project.

To Nicole, for your support in my writing.

To Hank and Yannis, for your guidance over the course of the project.

To Cristina, for helping with Karamba

To my family and friends, for your encouragement and support.

## References

- Alfaiaite, P, Leitaio, A 2017, 'Luna moth' Simulation and Web-Based Design, vol. 2, pp 511-518
- Al-Qattan, E, Yan, W, Galanter, P 2017, 'Tangible computing for establishing generative algorithms – A case study with cellular automata' Parametric and Generative Design, vol. 2, pp 363-370
- Anderson, C, Bailey, C, Heumann, Davis, D 2018, 'Augmented space planning: Using Procedural generation to automate desk layouts' International Journal of Architectural Computing, vol. 16, no. 2, pp 164-177, DOI:10.1177/1478077118778586
- Autodesk 2019, Demystifying Generative Design – For Architecture, Engineering, and Construction, ebook, accessed 5 October 2019, <https://damassets.autodesk.net/content/dam/autodesk/www/solutions/generative-design/autodesk-aec-generative-design-ebook.pdf>
- Belem, C, Leitaio, A 2018, 'From design to optimised design – An algorithmic-based approach' Simulation, Prediction & Evaluation, vol. 2, pp 549-558
- Bialkowski, S 2018, 'Topology Optimisation Influence on Architectural Design Process - Enhancing Form Finding Routine by tOpos Toolset utilisation' Applications in Construction & Optimisation, vol. 1, pp 139-148
- Cross, N 2001, 'Can a machine design?' Design Issues, vol. 17, pp 44-50, DOI:10.1162/07479360152681083
- Das, S, Zolfagharian, S, Nourbakhsh, M, Haymaker, J 2016, 'Integrated Spatial-Structural Optimization in the Conceptual Design Stage of Project - A tool to generate and optimize design solutions aiding informed decision making for Architects, Engineers and Stakeholders' DESIGN TOOLS Evaluations, vol. 2, pp 117-126
- Davis, D, Burry, J, Burry M 2011, 'Untangling Parametric Schemata : Enhancing Collaboration through Modular Programming' Designing Together, pp 55-68
- Fatima, T 2014, 'Optimisation of Lateral Load-Resisting Systems in Composite High-Rise Buildings', Master thesis, Queensland University of Technology
- He, E, Bertaltee, C, Jones, S, Lyle, L, Meister, J, Schawbel, D 2019, 'From fear to Enthusiasm: artificial intelligence is winning more hearts and minds in the workplace' AI@Work Study, pp 1-18
- Hoff, KA, Bashir, M 2013, 'Trust in Automation: Integrating Empirical Evidence on Factors That Influence Trust' Human Factors, vol. 28, pp 407-434, DOI: 10.1109/MIS.2013.24
- Irwin, AW, Alexander, SJ, Coull, A, Jeary, AR, Michael, D, Pinfold, G, Subedi, NK 1984, Design of shear wall buildings, CIRIA, London
- Made-in-China.com. (2016). [Hot Item] Climbing Formwork for Core Wall and Bridges Concrete. [online] Available at: <https://horizonform.en.made-in-china.com/product/LXwQduCxhEY1/China-Climbing-Formwork-for-Core-Wall-and-Bridges-Concrete.html> [Accessed 1 Dec. 2019].
- Varna, KR, Bhavana, B 2017, 'Optimum location of lift core wall for flat slab and conventional beam system using generated response spectra' International Journal of Advances in Scientific Research and Engineering, vol. 3, no. 9, pp 26-34, DOI: <http://dx.doi.org/10.7324/IJASRE.2017.32514>
- Yasser, HS, Johnson, ER 1991, 'Computer-Aided Architectural Design Strategies: One Size Does Not Fit All' Reality and Virtual Reality, pp 15-31
- Cole, R, Purao, S, Rossi, M, Sein, M 2005, 'Being Proactive: Where Action Research Meets Design Research', International Conference on Information Systems, pp. 324-336
- Findeli, A 2012, 'Searching for design research questions: some conceptual clarifications', Mapping Design Research, pp. 123-134
- Yasser, HS, Johnson, ER 1991, 'Computer-Aided Architectural Design Strategies: One Size Does Not Fit All' Reality and Virtual Reality, pp 15-31