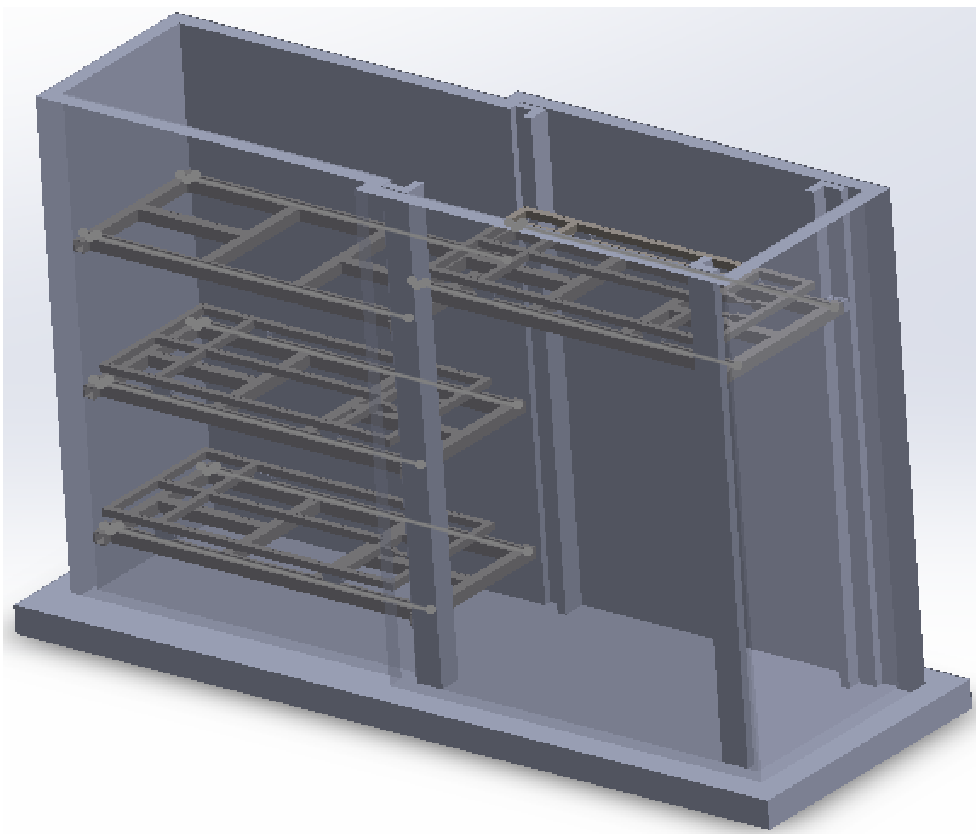
Mechatronic Systems Engineering
School of Engineering Science
SIMON FRASER UNIVERSITY

MSE 320 – Machine Design
- Project Report Part 1-

Roadside Underground Parking System

To the Attention of Dr. Krishna Vijayraghavan



Team 12 – Team Speed-Park

Sohail Sangha 301186636 ______________

Vincent Lin 301166236 ______________

Ehsan Tehran ______________

Sami Namvar 301138889 ______________

Tanvir Sooch 301162489 ______________

Team Speed-Park

8888 University Dr. tsooch@sfu.ca
Burnaby, B C V5A 1S6 (604) 123-4567

October 30, 2014

Dr. Krishna Vijayaraghavan,
Assistant Professor
School of Mechatronic Systems Engineering
Simon Fraser University
250 – 13450 102 Avenue
Surrey, BC V3T 0A3

Dear Dr. Vijayaraghavan,

As per your kind request, please find the attached report titled: *Road Side Underground Parking System Part 1*. This report is intended to provide thorough insight towards the feasibility of such a system by outlining the costs associated with construction and operation, the structural engineering and stress analysis, as well as various other practical constraints applicable to an automated roadside underground parking system.

The report consists of an initial preliminary research of codes and standards to investigate additional design constraints above that stated within our project followed by market research of possible concept designs, summarizing the benefits and drawbacks of the individual systems and areas of possible improvement. An optimal concept system is suggested drawing from the accomplishments of the comparable systems to arrive at the most efficient design with key factors such as safety and reliability, capital cost and overall practicality. Supporting evidence of the design is provided by real-world industry inquiries on items such as cost, component availability, construction process and time requirements for such a system.

Furthermore, this report contains the structural design and analysis of the system to investigate and ensure its capabilities to withstand the applicable stresses through theoretical calculations and Finite-Element-Analysis

For any questions or concerns you may have about the report, please feel free to contact us at (604) 123-4567.

Sincerely,

The Speed-Park Team
tsooch@sfu.ca
(604) 123-4567

Table of Contents

Introduction 4

Objectives and Scope 4

Preliminary Research Findings 5

Applicable Standards and Codes 5

Registered Patents and literature 6

Competitor Products 6

Description of Concept Alternatives 7

Perfect Park 7

Multiparker 710 7

MasterVario S 8

Concept Analysis and Selection 9

Simplified technical and cost analysis 10

Perfect Park 10

Multiparker 710 10

MasterVario S 10

Scoring Matrix and Sensitivity Analysis Results 10

Concept Integration and Recommendation 11

Parts List and Sourcing 12

Design and Stress Analysis 14

Concrete Foundation 14

Parking Stall and Elevator Platforms 14

Translating Platform 15

Conclusions 17

Acknowledgements 17

References 17

Appendix 1 (Detailed Calculations) 18

MATLAB Analysis 21

SolidWorks FEA analysis 22

MATLAB Analysis 24

SolidWorks FEA analysis 28

Appendix 2 (MATLAB CODE) 29

Introduction

Due to the ever increasing population across the globe, ensuring that valuable real estate in metropolises are being utilized at their best and highest value is an ever increasing concern. The promotion of public transportation and other forms of rapidly transporting the masses to ensure society remains productive have battled hard to cut congestion in major cities yet only 12% of Canadians chose this method of transportation for their daily commute [6]. With the large majority of Canadians commuting by personal means it is clear that the parking of motor vehicles is a very area extensive requirement which must be addressed to allocate real estate for more productive means such as additional roadways, walkways, office buildings or residential homes.

This report addresses a solution to this concerning problem, by the way of implementing an optimal underground parking systems and outlining the key performance, cost and safety benefits compared to traditional parking methods, and various other underground parking systems.

Objectives and Scope

The Objective of this report is to exercise and apply engineering fundamentals and knowledge obtained throughout MSE 320 towards the mechanical design and analysis of an underground vehicle parking system. This is accomplished by way of this report as it addresses structural and component design using stress analysis and failure criteria, application of CAD skills in drawing generation and exercising critical thinking towards an innovative yet practical design concepts and prototype.

This report shall focus on the design and analysis of the structural components of an optimal underground parking system by first conducting market research on traditional parking systems and conceptual underground systems and outlining the components which provide the key benefits and drawbacks of their respective system. Compiling the key design features of several parking systems an optimal parking system will be recommended and designed such that it shall withstand and suffice to the safety requirements for the loads it will be subject to. Additional aspects such as capital cost requirements, applicable power drive components, control components, construction methods and sourcing of components will also be addressed with limited depth.

Hard restraints in the design of the underground parking system consist of the following:

1) Surface area capacity of one car
2) No more than one cars height
3) Must be positioned along the direction of road traffic (No parallel/perpendicular parking)
4) Must not block vision and road signs
5) Capability to store a minimum of three vehicles
6) No rotational movement
7) Provide for automatic parking and retrieval in less than 2.5 minutes
8) Auto pay/ticketing

Preliminary Research Findings

Applicable Standards and Codes

The design of our underground parking system must be in compliance to applicable codes, guidelines, and standards of regulatory bodies in order to ensure safety in the construction, operation, maintenance and practicality of the parking system. The governing bodies/legislations which enforce standards and codes within British Columbia are the National and Provincial building codes and WorksafeBC.

The National Building Code of Canada and its subordinate provincial code governs the construction of buildings and structures to ensure public safety, reliability, and high level uniformity across the nation. In particular, for an underground parking structure the building code also requires compliancy to the CSA S413-14 standard, “Parking Structures” [3]. The CSA S413-14 standard outlines design specifications in respect to the construction and maintenance of a parking structure to ensure structural stability and longevity of the structure due to static and dynamic loading as well as environmental conditions [4]. The building codes also integrate the structural design with the regulations to protect public safety such as the requirement to design structures in such a manner that does not obstruct the accessibility of firefighting equipment and personnel to the subject and surrounding buildings.

The system is also subject to the Occupational Health and Safety Regulations of WorksafeBC, which again provides regulatory rules and guidelines to protect the public and those who take part in the construction, operation and maintenance of the parking system. Due to our parking system effectively lifting motor vehicles, our design must comply with ANSI Standard ALCTV-1998, “*American National Standard for Automotive lifts – Safety Requirements for Construction, Testing and Validation*”, ANSI Standard ALOIM-2000, *“American National Standard for Automotive Lifts – Safety requirements for operation, inspection and maintenance”* and OHSR P.12 Tools, Machinery and Equipment [2]. Apart from the aforementioned standards and codes applicable directly to parking structures, the general construction processes such as excavation, shoring and the use of mobile equipment are also to be complaint with the OHSR of WorksafeBC [5].

Standards addressed towards parking vary on locale, and are enforced by municipal by-laws. These requirements are addressed towards practicality of parking within the jurisdiction as expressed by the citizens. All parking structures constructed in the city of Surrey must satisfy the requirements in the chart to the right from the city of Surrey By-Laws 12000, part 5 – B “Dimensions and standards” [14]. The design constraints of the system lead to the dimensional requirements of a parking facility with parallel parking on a one or two way street. Other constraints to keep mindful of would be maximum noise level and the security of stored vehicles.

Types of Uses	Parking Angle in Degrees	Width of *Parking space* [1]		Length of *Parking space* [2]		Width of Aisle[1]		Vertical Clearance	Traffic Direction
		Metres	Feet	Metres	Feet	Metres	Feet		
Parking Facility	90	2.90	9.5	5.5	18	6.1	20.00	2 metres [6 ft. 7 in.], except that this shall be increased to 2.3 metres [7 ft. 6 in.] where *parking spaces* for persons with disabilities are provided as required by Section D of this Part. The required vertical clearance shall be provided over the entire width and length of each required off-street *parking space* and be free of protrusions or encroachments by structural and non-structural elements.	Two-way
	90	2.75	9.0	5.5	18	6.7	22.00		Two-way
	90	2.60	8.5	5.5	18	7.0	23.00		Two-way
	60	2.75	9.0	5.5	18	5.5	18.00		One-way
	45	2.75	9.0	5.5	18	3.9	12.75		One-way
	30	2.75	9.0	5.5	18	3.3	11.00		One-way
	Parallel	2.60	8.5	6.7	22	3.6	12.00		One-way
	Parallel	2.60	8.5	6.7	22	6.0	19.75		Two-way
	Tandem Parking	2.60	8.5	13.4	44	3.6	12.00		One-way
	Tandem Parking	2.60	8.5	13.4	44	6.0	19.75		Two-way
***Ground-oriented Multiple Unit Residential Building*:**									
• Single Garage[3 & 4]	Parallel	3.20	10.5	6.1	20.0	N/A	N/A		N/A
• Double Garage[3 & 4]	Parallel	5.70	18.8	6.1	20.0	N/A	N/A		N/A
	Tandem Parking	3.20	10.5	6.1	20.0	N/A	N/A		N/A
Single Family Dwelling* and *Duplex									
• Single Garage[3]		3.20	10.5	6.1	20.0	N/A	N/A		N/A
• Double Garage[3]		5.70	18.8	6.1	20.0	N/A	N/A		N/A
• Uncovered *Parking Space*		2.75	9.0	6.0	19.7	N/A	N/A		N/A

Registered Patents and literature

With congestion in major cities intensifying by the day, intuitive parking solutions is a fast paced, exciting area of engineering design with an ever increasing number of innovative designs reaching patents. The US patent number 6851921 B2 "Automated parking garage" is a good example of progresses which have been made in the field of vehicular storage. This parking system utilizes a multi-story building and a stacked pallet loading station to automatically transport a vehicle along with the underlying pallet on the loading station to a vehicle storage rack in a storage area [7]. Although this system achieves a high level of convenience it may be improved in areas such as the size requirement of the system and complexity of the design as this system requires translational motion in 4 directions over a large actuation range as well as vertical actuation.

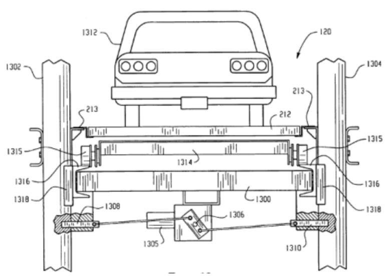


Fig 1: loading bay of US 6851921 B2

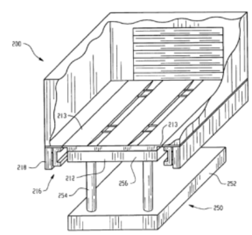


Fig 2: storage bay of US 6851921 B2

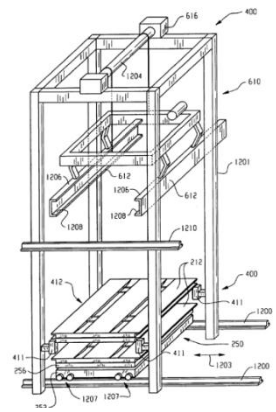


Fig 3: loading bay of US 6851921 B2

As our goal is to improve on the present parking systems we must also ensure our ideas, no matter how novel, do not pose any copyright infringements on products with issued patents. This may be achieved by emphasizing uniqueness and clearly defining the purpose of the overall system along with specifications on individual components. If the optimal design concept appears to share many similarities with issued patents we must acquire legal advice prior to branding it as an idea of our own or contact the owner of said patent and inquire whether the patent is in fact licensed or available for purchase.

Competitor Products

The most prominent competition to an underground parking system such as the imposed product from the design constraints is of course traditional surface parking. Designing a simple road side parking lane would not only consume an equivalent amount of real estate but do so in a manner which does not support the systems goals in reducing congestion and providing high density parking. The comparison between underground parking systems and above ground multi-level parking lots also arrives at a similar conclusion; where the real estate in question is restricted from being utilized in its highest and best value. This may be quantified by analyzing the incremental costs associated with developing office space for lease over and beyond the cost of a multi-level pay parking lot, for which the rate of return on an office building would be several percentages greater. Therefore, the underground parking system clearly is the best option to provide a solution for a growing problem in today's municipalities.

Description of Concept Alternatives

Perfect Park

Perfect Park along with Trevi Group of Italy currently hold exclusive licensing rights for a great solution to achieve high density parking, this system has the capability to provide parking space for 108 cars with minimal space designated at the surface. Perfect Park is suitable for construction of 2 to 9 floors with up to 12 parked cars on each level. Common operation of the Perfect Park usually consists of the upper floors being designated for larger vehicles and the lower levels designed for vehicles of lesser stature. Vehicles to be loaded are initially analyzed by various scanners to calculate the optimal parking spot based on dimensions of the car. Methods used to build the external structure may be chosen from one of the three methods below based on the logistics of the site:

1. Bored secant piles with casing driving
2. Diaphragm wall constructed using the "Trencher" technique
3. Continuous diaphragm wall made of concrete

The exterior diaphragm of the structure requires a minimum thickness of 0.5 m to ensure rigidity and safety [17].Where the internal components of the structure are assembled on-site; Prefabricated parts are joined vertically using steel bars to transmit the vertical and shearing stresses and horizontally by supplementary in-situ concrete. The roof comprises a solid reinforced concrete slab cast in-situ that bonds all the prefabricated elements of the highest floor. The construction cost of this system is between $40,000-45,000per stall with considerations in place for permits and phase I environmental site analyses.

Fig 4: Birds-eye view of construction of Perfect Park

Fig 5: Cut out view of Perfect Park System

Multiparker 710

The Multiparker 710 as shown in Figure 6 is particularly suited to wider structures, providing 2 to 8 parking levels. The capacity depends on the number and arrangement of transfer areas.

Fig 6: Interior of Multiparker 710

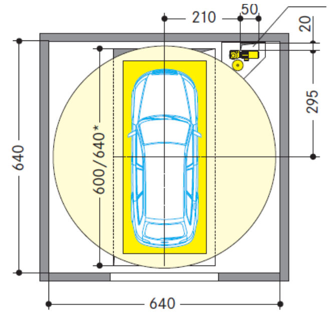


Fig 7: translating area dimensions

In addition to linear actuators, there is an integrated turning device in order to increase user convenience and minimize access time. To achieve that however, some extra space is needed to fully rotate the vehicle as shown in Figure 7. This addition might increase the overall cost, but the Multiparker 710 tends to offer more luxury than some of the other automated parking systems.

A summary of the products data is as follows:

- Max pallet load: 2500kg.
- Max car dimensions (L×W×H): 5.25×2.1×1.9
- Average access time: 115 seconds.

Steel structure is used to serve as the frame-work for the lift system and the pallet. This steel structure is fastened to the floor with metal splay dowels and shored-up sidewise against the external walls. Further features and recommendations are:

- The pallets are designed to prevent water from leaking down on the cars below. A drainage area connected to a pump sump is recommended.
- Noise level should not exceed 30 dB: building must have a sound reduction index of at least 57 dB.
- Ventilation system is required by client to provide continuous exchange of air.

MasterVario S

The MasterVario S, also called the "SmartParker", is a system that focuses on a cost-effective and space-saving design. The fully automated system is a single-row shelf structure with a single lift mechanism and horizontal conveying units to ensure a fast retrieval time. It can be built to provide 2 to 6 parking levels, 1 to 4 vehicles per level; an example is illustrated in Figure 8. MasterVario S can therefore offer a capacity of up to 23 cars, as one empty space is needed for transferring the cars.

Figure 8 : Cut out view of MasterVario S

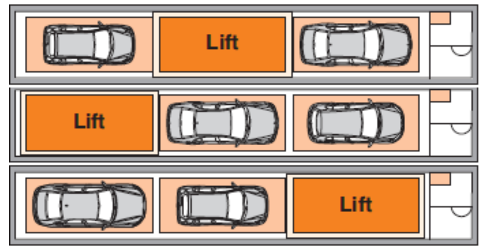


Figure 9: Possible arrangements of MasterVario S

The design is very flexible as the user can choose between 8 different arrangements by having different number of vehicles per floor and by having the lifting elevator in the preferred position. Figure 9 shows three different arrangements of the system with 2 cars per parking floor.

A summary of the product's data are as follows:

- Up to 6 parking levels, 1 to 4 cars per level, for a maximum of 23 cars.
- 3 different vehicle heights (must be uniform).
- Load per parking space: 2000kg or 2600kg.
- Required installation width: <3m.
- Max. car dimensions (L×W×H): 5.3×2.35×2 (in meters)
- Platform dimensions (L×W): 5.4×2.45

The floor plate and walls are made of concrete, while a steel construction supports the pallets and the conveyors. Floor surfaces can be selected by the user, for example topsoil/lawn, sand bedding/pebbles, etc. Some other features include:

- Parking pallets designed to collect water for rain or melting snow.
- A drainage channel with connection to the public sewer or a pump sump.
- Necessary fire protection designed by the architect.
- Sound reduction index of at least 57 dB.
- Ventilation system designed to fulfil safety requirements by reducing humidity to prevent corrosion, and control the temperature to a range of 5-40°C for electrical elements and mechanical parts.

Concept Analysis and Selection

In order to quantify and compare the proposed design concepts, this portion of the report will consist of analysis of the performance of the individual systems in areas such as feasibility and cost; robustness; practicality and aesthetics for comparison in a design matrix.

Simplified technical and cost analysis

Firstly, we shall grade the systems on practical areas such as storage capacity, parking and retrieval time and costs.

Perfect Park

The parking capacity in this product is fixed (on each floor), since it has a circular design. The only variation in capacity is increasing the number of floors which would increase the construction cost; however, the circular structure of the design offers a fast and power efficient way to insert the vehicles in the parking stalls. Based on the capacity and the occupied space, the operational and construction cost of Perfect Park can be up to 3 times less than a traditional parking. However, given the complex design of the structure, a high quality actuation system may be required. As the structure is fairly compact and the elevating system translates only vertically, a limited number of safety concerns need to be addressed.

Multiparker 710

This parking system has variable space based on the suitable system for the client (12 to 40 on each floor). Due to the wide rectangular design of this product, it has the ability to offer high capacity underground parking, however, at the cost of compactness, cost of land, and maintainability. The company claims that the average retrieval time on this product is about 115 seconds, which compared to the size and geometry of the structure, seems to be an average retrieval time. The actuation system is also not very economic as it includes rotational motion as well as lengthy horizontal and vertical translation of the vehicle. In addition, some safety concerns may be raised while the actuation system repeatedly carries the car around the empty space. Putting all the factors into account, this design is evaluated using the scoring matrix in the upcoming section.

MasterVario S

MasterVario S' base structure features a minimalistic design with cars arranged in a single vertical plane. The system can also be fitted with multiple lifts and a 3-D structure instead to increase the storage capacity. The utilization rate of the entire storage volume is high, as only one empty space for car swapping and the space for lift operations are required.

Due to its simple vertical lift and horizontal conveyer design, the retrieval time is expected to be the shortest among the three alternatives. The system is also expected to be safe and reliable because of the minimum number of moving components. Its simple structure also reduces the manufacturing and fabrication costs as well as improves its maintainability. Finally, MasterVario S' platform movement pattern allows it to retrieve cars with relatively low power consumption despite the need of swap operations when there are multiple cars on the same level but also makes it the least aesthetically pleasing of the three.

Scoring Matrix and Sensitivity Analysis Results

In this section we will now evaluate all prospective systems based on concepts of economic feasibility compactness, retrieval time, and aesthetics. Economic feasibility includes machinability, maintainability, and power consumption. Each aspect is assigned a weight based on its relative importance among the factors considered.

Economic feasibility is assigned the heaviest weight, as it is directly related to the amount of investment, operating costs, and profitability of the parking system. Machinability and maintainability refer to the ease of manufacturing and maintenance, respectively. Power consumption is also included as one of the metrics, since utility expenses represent a major portion of the operating costs.

	Perfect Park	**Multiparker 710**	**MasterVario S**	**Weight**
Economic Feasibility:	**5.67**	**5.00**	**7.33**	**45.0%**
Machinability	6.00	5.00	8.00	17.5%
Maintainability	5.00	5.00	6.50	15.0%
Power Consumption	6.00	5.00	6.00	12.5%
Compactness:	**7.00**	**6.00**	**8.00**	**15.0%**
Retrieval Time:	**7.00**	**6.50**	**8.50**	**20.0%**
Safety:	**7.00**	**5.50**	**7.50**	**15.0%**
Aesthetics:	**8.50**	**6.50**	**6.00**	**5.0%**
Weighted Average	**6.48**	**5.60**	**7.45**	

Concept Integration and Recommendation

Through integration of the key features of the respective concepts we arrived at an optimal design based on the defined grading and constraints, which is modeled below.

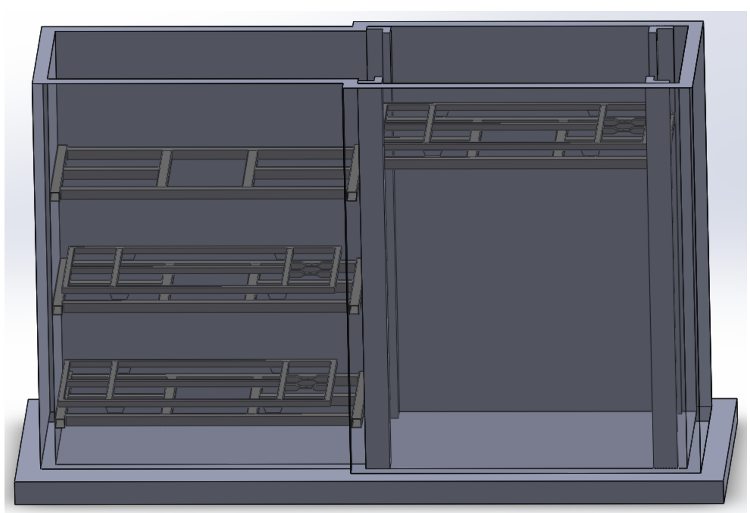


Figure 10: Front view of optimal design proposal

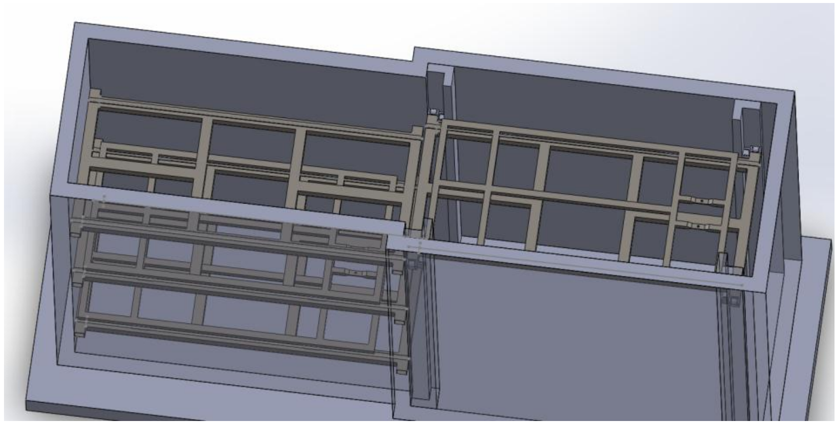


Figure 11: Top view of optimal design proposal

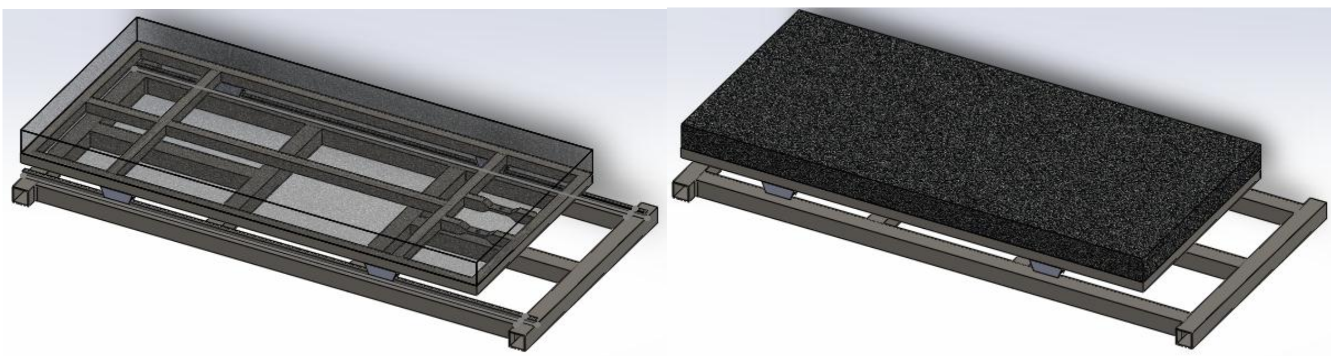


Figure 12: elevator platform and translating platform with asphalt sheet

Parts List and Sourcing

1) **Reinforced Concrete Foundation** – The parking system in its entirety is to be placed with a reinforced concrete foundation which will serve a combination of the structural backbone of the system by providing permanent shoring, a protective barrier from the elements, as well as a vertical load bearing foundation. Construction will be similar if not identical to current commercial practises, at which the site is surveyed and staked followed by excavation under the direction of a P.Eng. The depth of excavation is based on many factors such as climate and soil type with the goal to penetrate deep enough to reach compacted soil which is well confined with our design for a 6m deep system in the region of interest. Construction would proceed with the placement of footings which serve to distribute the weight bearded on the walls onto the surrounding soil and would be the anchoring point of the form work. Once the form work has been assembled to specifications, rebars will be inserted followed by the pouring of concrete mix, letting it settle and removing forms. The foundation would then be completed upon further dampproofing as seen necessary and backfilling of the outer trenches.

   The required materials for the foundation are readily available from a variety of sources globally and do not pose major constraints for sourcing. The budget for the construction of the foundation would be as follows: $2,000 for excavation costs, $20,000-$35,000 for concrete,

concrete pumping, rebars and dampproofing allocating possible a safety figure of $5,000 for miscellaneous charges and permits for possible road closures the total budget for the foundation is $42,000.

2) **Parking stall platforms** – Vehicles are to be supported when stored in parking stalls and whilst being transported on the elevator by platforms comprised of standard 8″x 8″ x ½″ structural steel square beams which cost $1318/20ft [18]. The parking stall platforms are to be fixed directly into the concrete foundation, providing a clamped boundary condition. The construction of both parking and elevator platforms, which follow this product, are to be completed by welding the individual beams at joints and requires a total of 370 ft for the 4 platforms, with consideration to wastage, requiring at total capital of $24,383.

3) **Elevator platform –** The elevator platform performs the role of translating the vehicles between the stalls and surface. Despite variances in the forces exerted onto the elevator platform, the design for the steel platforms has been conducted to satisfy both the elevator platform and parking stall platforms for simplicity. This also acts as a means to reduce costs by allowing bulk orders of beams and an added safety factor in the design for the parking stall platforms.

4) **Innovative translating platform** – In order to translate a vehicle to/from the elevator platform from/to the parking stall platforms, our system will incorporate an innovative moving platform. This platform will also consist of structural steel square bars, but of 5″x 5″ x ½″ dimensions which cost $517.80/20ft for a total budget for the three translating platforms of $5,048.55 [18]. The innovative translating platform is also equipped with internal docking positions to engage the platform with the power drive system in order to translate the platform along with an accompanying vehicle.

5) **Sensors and User interface –** The system will also incorporate basic sensors to monitor the vacancy of parking stalls along with an intuitive user interface perform actions such as displaying the vacancy of the system and to provide a means to process payment and initiate storage/retrieval of the vehicle. Signage along with physical barriers will also ensure loaded vehicles are within the dimensional capabilities of the system, which have been designed to incorporate vehicles as tall and wide as a Hummer H2 and as long as a Cadillac Escalade ESV.

6) **Actuators/Power Drive System –** the actuation mechanism will move underneath the translating platform on a separate set of rails. The System will be composed of two parts, first an actuation trolley which will be actuated along the length of the elevator platform using steel cables, second will be a set of vertical hydraulic jacks which engage with the docking sites on the translating platform. The actuation mechanism will be local to the elevator platform, but will be able to exceed slightly on to the stationary platform for the placement or extraction of the moving platform.

Design and Stress Analysis

For detailed calculations please refer to Appendix 1, while the main results are summarized in the following sections:

Concrete Foundation

The concrete foundation must be capable of withstanding the forces exerted upon it by the soil, any surface forces from vehicles as well as its own weight. For simplicity, forces at applied at the pavement above the concrete walls are assumed to directly translate onto the foundation and not disperse amongst the asphalt or surrounding soil. From the nature of the system the greatest purely compressive stress on the concrete walls would be a result of a vehicle being placed with two tires directly over a concrete wall. The average front engine vehicle is assumed to have a weight distribution of 60% over the front axle; therefore the maximum compressive stress is calculated as follows

$$\sigma = \frac{F}{A} = \frac{\big((car\ mass * 60\%) + \rho wld\big) * g}{wl} = \frac{\big((4000 * .6) + 2800 * 0.3 * 3.6 * 6\big) * 9.81}{0.3 * 3.6} \leq \frac{\sigma_{comp}}{N} = \frac{28 * 10^6}{4}$$

Where the mass of a car has been substituted for that of a Hummer H2, ρ is an upper bound for the density of concrete and σ_{comp} is the compressive strength of concrete; this relation checks through well within the safety factor of 4 with a maximum of 306kpa applied compressive stress[1][8][9] .

The design for underground concrete foundations subjected to translational forces such as the pressure from exterior soil along with the moments which are produced as a result peer out of the scope of this report, and analysis for this topic will rely strictly on literature and professional consultation along with reference to standards and codes such as the Building codes of Canada, ACI 318-14, ACI 301-10, CSA-G30.18-M92, ISO 15673:2005 and CSA O121. From the multitude of references an acceptable design would consist of 20-30cm thick concrete walls, with size 20M steel reinforcement bars with diameters of approximately 25mm placed in a grid like fashion with 20cm separations both horizontally and vertically[10][11][12][13][15][16]. Although these dimensions have been adjusted by factors for possible shock forces such as earthquakes and the physical wear of the materials we shall continue with a width of 30cm for our foundation for an added safety factor as the incremental cost of cement required is a very miniscule drawback to the added safety and rigidity of the system.

Parking Stall and Elevator Platforms

The bare-bone structure of the elevator platform is shown in Figure 13. The maximum allowable stress on the beams given the applied fluctuating load was calculated to be σ=27.57MPa (refer to Appendix) for a square tubing steel structure (A 992, machined). A series of stress analysis was first done using MATLAB by manually calculating the maximum stress on each beam due to forces on each beam. Afterwards, finite element analysis in SolidWorks was conducted to verify the value of stresses at the concentration points, mainly where the beams are sharply welded together. The size of the structural beams were first initialized by trial and error using MATLAB, with the steel tube dimensions as the input, and the maximum stress on each beam as output. A summary of three runs at the platform configuration is as follows:

- First Run: 5×5×1/2 inch square tubes with the bare-bone structure as in Figure x1.
  Result: stress at the sharp corners (54MPa) exceeds maximum allowable stress (27.6MPa).
- Second Run: 5×5×1/2 inch square tubes with added support beams across the long side.
  Result: Slight improvement, maximum allowable stress still exceeded at sharp corners.
- Third Run: 8×8×1/2 inch square tubes with previously added support beams.
  Result: maximum stress at the sharp corners (24.5MPa) is below the maximum allowable stress.

Using the configuration explained in the third run, without considering the points of stress concentration, the maximum stress due to forces on beam (1) and (3) were calculated using MATLAB:

- Max. Stress on Beam (1): 1.99Mpa
- Max. Stress on Beam (3): 3.71MPa

As it can be seen, the sharp interconnection of beams significantly amplify the stress on the structure under fluctuating loads. The larger square tube structures were chosen to ensure safety under almost any circumstance.

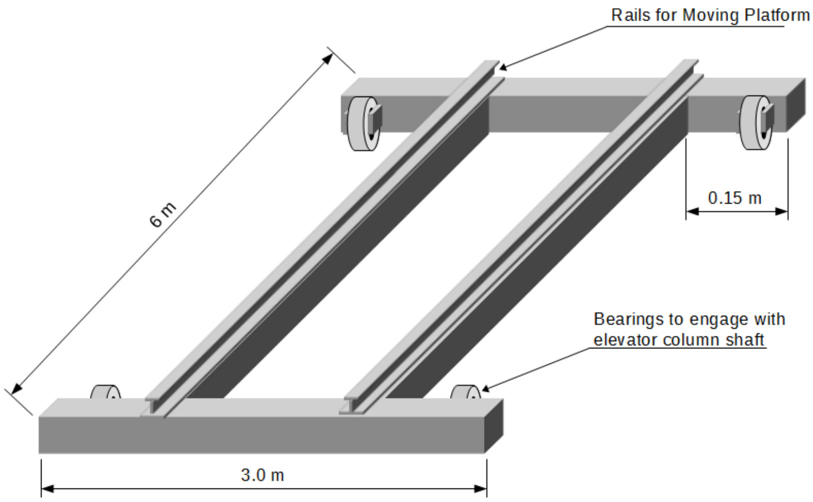


Figure 13: Bare-bone Structure of the Elevator Platform.

Translating Platform

The bare-bone structure of the moving platform including the dimensions of the beams, the resting position of the car wheels, and the actuator docking site is shown in Figure 14.

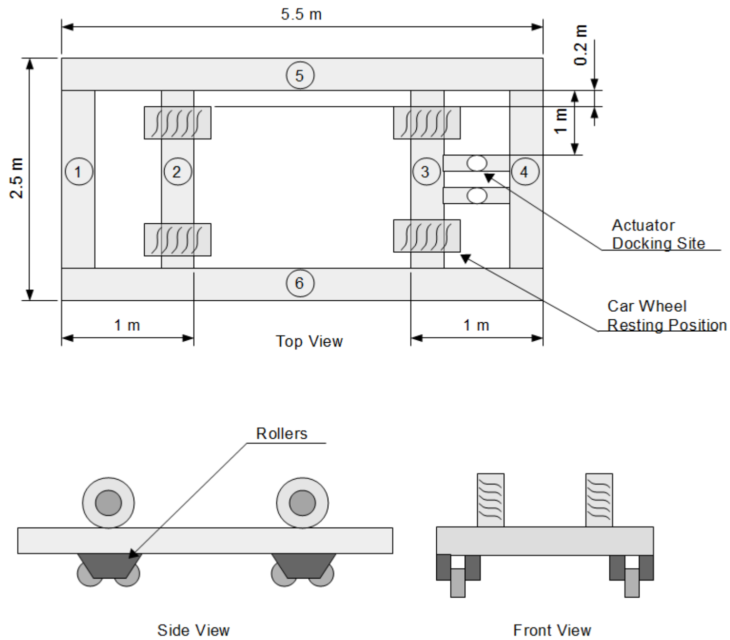


Figure 14: Multi-view Drawings of the Moving Platform

A pair of pins (steel AISI 1137) withstand the reversed and repeated horizontal load as the actuator translates the moving platform; the structure of the support is shown in Figure 15. Referring to Appendix 1, the diameter of the pin and the width of the thin beam were calculated to be 4 cm and 10 cm, respectively.

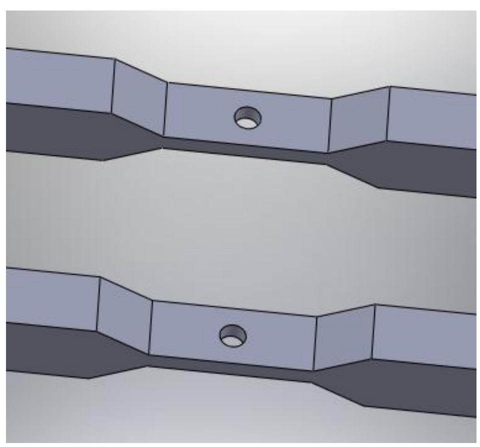


Figure 15: Structure of the actuator docking site

The stress analysis on the moving platform was first conducted using 5×5×1/2 inch square tubes as structural steel. The FEA results from SolidWorks proved that the smaller beam size is sufficient to withstand both the maximum stress vertically (18.52MPa) and horizontally (1.79MPa). Ignoring the stress concentration factor at sharp corners, the net stress from the combination of bending and torsional moments on each individual beam was calculated using MATLAB:

- Bending Stress on Beam (1): 0.40MPa

- Bending Stress on Beam (2): 0.50MPa
- Net Stress on Beam (3): 2.39MPa
- Net Stress on Beam (4): 2.19MPa
- Net Stress on Beam (5): 1.08 MPa

Conclusions

To conclude, given the design constraints, our design is more suitable for a single-lane roadside parking system than the alternatives. Its design nature allows for high scalability and flexibility in arrangement and does so in a cost-effective manner. The structural stability of the system enables it to withstand extreme loading conditions with a higher safety factor, ensuring both safety and robustness.

With this system in place, the problem of insufficient parking solutions may be drastically reduced, resulting in higher land utilization rates and decreasing the severity of congestion.

Acknowledgements

- Abbas Namvar –Civil Engineer, MASc, P.Eng
- Idin Khoshnavaz – UBC Civil Engineering Student
- Ravinder Sooch – General Contractor
- Doug Haid – Perfect Park USA

References

[1] "2005 Hummer dimensions" – URL: http://www.motortrend.com/cars/2005/hummer/h2/specifications/exterior.html
[2] "OHSR P.12" – URL: http://www2.worksafebc.com/publications/OHSRegulation/Part12.asp?ReportID=18474
[3] "Building Code Overview" – URL: http://www.mah.gov.on.ca/Page10806.aspx
[4] "Parking Structures 413-07" - URL: http://shop.csa.ca/en/canada/structures/s413-07-r2012/invt/27005102007
[5] "WorkSafeBC OHSR" - URL:
http://www2.worksafebc.com/publications/OHSRegulation/Home.asp?_ga=1.180446756.2069115581.1415085690
[6] "Commuting to Work" – URL:http://www12.statcan.gc.ca/nhs-enm/2011/as-sa/99-012-x/99-012-x2011003_1-eng.cfm
[7] "Automated Parking System Patent" – URL: http://www.google.com/patents/US6851921
[8] "Concrete in Practice" – URL:http://www.nrmca.org/aboutconcrete/cips/35p.pdf
[9] "Steel Reinforced Concrete" – URL: http://www.matbase.com/material-categories/natural-and-synthetic-composites/natural-material-matrix-composites/concrete/material_properties_of_steel-reinforced-concrete.html#properties
[10] "Building Code for Structural Concrete" – URL: https://law.resource.org/pub/us/cfr/ibr/001/aci.318.1995.pdf
[11] "Concrete and Underground Structures and Works" – URL:
http://www.winnipeg.ca/finance/findata/matmgt/std_const_spec/current/Docs/CW2160.pdf
[12] "Reinforced Concrete Design" – URL: http://faculty-legacy.arch.tamu.edu/anichols/index_files/courses/arch331/NS22-1cncrtdesign.pdf
[13] "Cast in Place Concrete" – URL:
http://books.google.ca/books?id=zsuVMWGDW2QC&pg=PA11&source=gbs_toc_r&cad=3#v=onepage&q&f=false
[14] " Surrey Zoning By-Laws" – URL: http://www.surrey.ca/bylawsandcouncillibrary/BYL_Zoning_12000.pdf
[16] " Specifications for Structural Concrete" – URL: http://www.concrete.org/Store/ProductDetail.aspx?ItemID=30110
[17] "PerfectPark System Overview" – URL: http://www.perfectparkusa.com/system-overview.html
[18] "Steel Square Tube" – URL: http://www.metalsdepot.com/index.php

Appendix 1 (Detailed Calculations)

Actuator Pin

Assuming a very heavy car (Hummer with a mass of 2600kg) fully accelerating off the platform, the stress maximum stress at the hole (σ_{nom}) due to the horizontal force (L) can be calculated as follows:

$$a = \frac{(60-0)\,\frac{miles}{hr}}{9.5\ s} = 2.82\,\frac{m}{s^2}$$

$$L = ma = (2600kg).\left(2.82\,\frac{m}{s^2}\right) = 7341N$$

$$\sigma_{nom} = \frac{F}{A_{rect}} = \frac{L}{(w-d).t}$$

Similarly, the shear stress on the pin actuating the platform is:

$$\tau = \frac{F}{A_{pin}} = \frac{7341}{A_{pin}}$$

Assuming steel AISI 1137 is used for pin, the ultimate strength S_u=1083MPa (oil-quenched at 400°C). The shear strength can be calculated as:

$$S_n = (0.6)(1083MPa) = 649.8MPa$$

Letting C_m=1, C_{st}=0.8. C_R=0.8, C_s≈0.8, and S_n=250MPa (machined), the actual endurance strength is:

$$S_n' = (250MPa)(0.8)(0.8)(0.8) = 128MPa$$

As the pin is under reversed and repeated shear stress, the mean stress σ_m=0, thus the area of the pin can be calculated by evaluating the stress amplitude (σ_a= σ) with a safety factor of N=3:

$$\frac{\sigma}{128} < \frac{1}{3} \rightarrow \sigma = 42.67MPa$$
$$A_{pin} > 1.72 \times 10^{-2}\ m^2$$
$$D_{pin} > 1.48\ cm$$

The above results stands if the entire load is being taken by a single pin with no stress concentration. For our design, a standardized diameter of 4 cm is chosen, and by assuming a stress concentration of k=4, the standard width length of the beam is evaluated as:

$$\frac{d}{w} \approx 0.3 \rightarrow w = 10cm$$

Beams

Square tubing structure, as shown in figure xx, is to be used as the beams in the platforms. The material of choice is A 992 structural steel, offering an ultimate strength of S_u=450MPa and resulting in an endurance strength of Sn=220MPa. The actual endurance strength is then calculated by setting C_m=C_{st}=1, C_R=0.81, and C_S=0.82:

$$\begin{aligned} S_n' &= S_n.C_m.C_{st}.C_R.C_s \\ &= (220)(1)(1)(0.81)(0.82) \\ &= 146.1MPa \end{aligned}$$

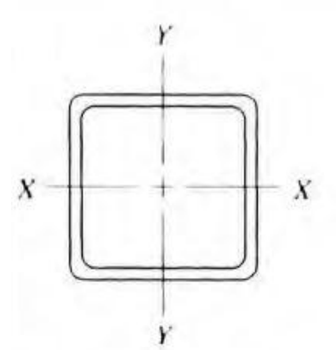


Figure xx: Cross-section of structural steel used as beams in the platforms.

The minimum stress on the platforms is zero for the case when the moving platform is mounted off the stationary elevator; therefore, the fluctuating stress on the beams will have an equal mean and amplitude. The maximum allowable stress using a safety factor of N=4 is then:

$$\sigma_a = \sigma_m = \sigma$$

$$\frac{\sigma_a}{S_n'} + \frac{\sigma_m}{S_u} = \frac{1}{N}$$

$$\sigma = \left(\frac{1}{146.1} + \frac{1}{450}\right)^{-1} \left(\frac{1}{4}\right) = 27.57MPa$$

Elevator Platform

The Loading pattern on the elevator platform is shown in the following figure:

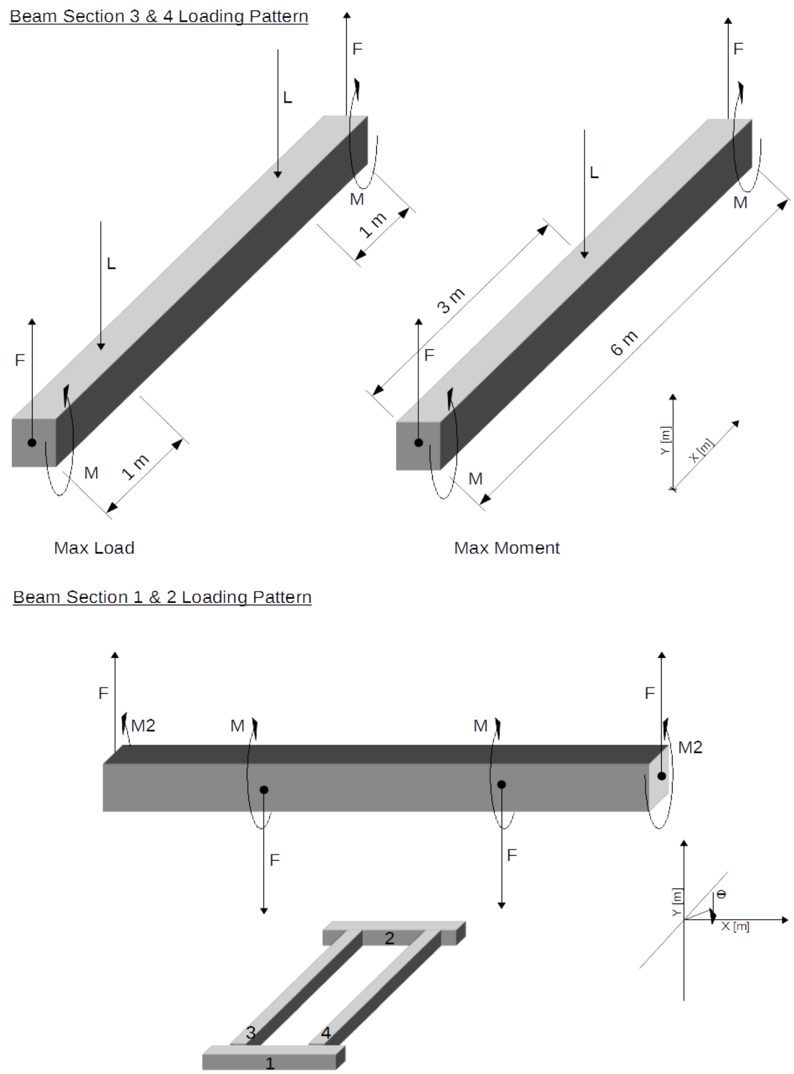


Figure A1: Horizontal and Vertical Loadings on the Elevator Platform.

MATLAB Analysis

This section gives an overview of the stress calculation on each beam using MATLAB; the source code can be found in appendix 2. The following variables were using in the stress analysis:

$M_x(X)$, is the moment distribution in beam 3 & 4.
$Y(X)$, is the deflection of beam 3 & 4 in vertical direction.
M, is the moment applied on beam 3 & 4, due to twisting of beam 1 & 2
$M_1(X)$, is the moment distribution in beam 1 & 2 in the vertical direction.
$Y_1(X)$, is the deflection of beam 1 & 2 in vertical direction.

θ, is the angle of twist at the intersection of beam 1 with beam 4.
b, is the width of rectangular sections
l_6, is the distance from end point of beam 1 to its intersection with beam 4.
Therefore building our system equations:

$$\frac{d^2Y}{dX^2} = \frac{M_x}{EI}$$

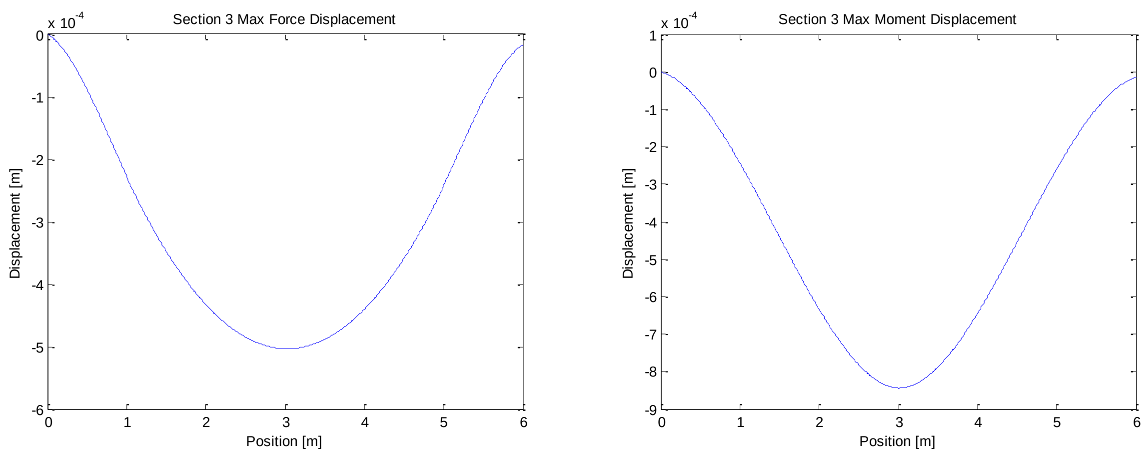
$$\frac{dY(0)}{dX} = \tan(-\theta)$$

$$\theta = \frac{(M + F.b/2)l_6}{GK}$$

We can solve the above equations to obtain the unknown M, and further calculate the stresses using applicable bending and torsional stress formulas. Finally for bending of beam 1 & 2 in vertical direction:

$$\frac{d^2Y_1}{dX_1^2} = \frac{M_1}{EI}$$

The following figures illustrate the deflection along beam 3

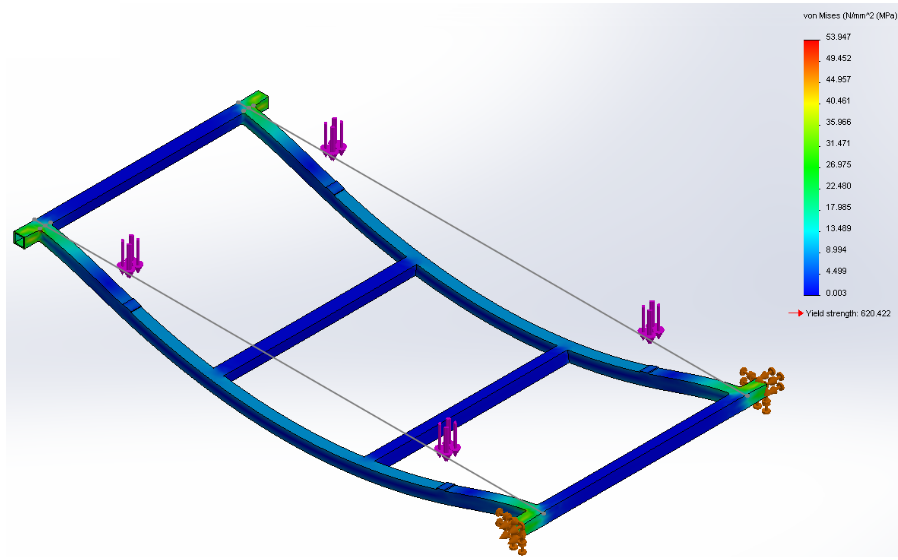


(a) (b)

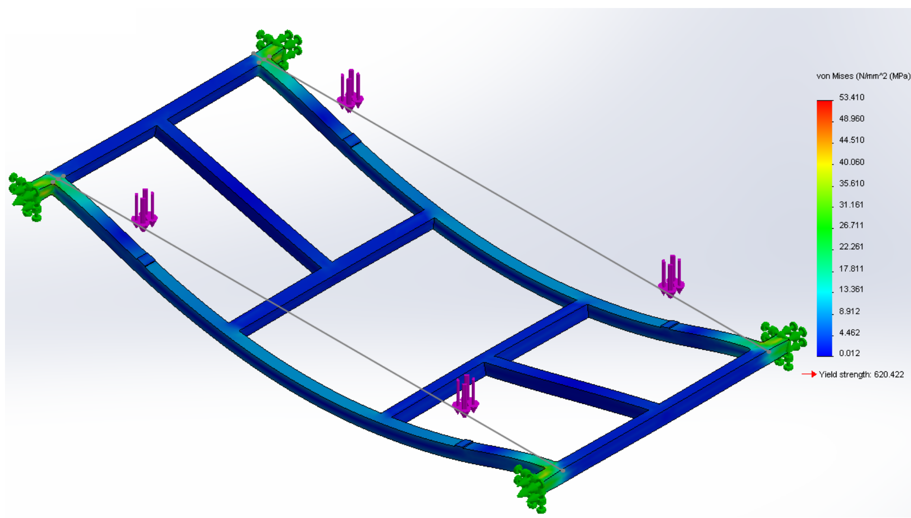
Figure A2: Vertical displacement (dy) across each beam in the elevator platform.

SolidWorks FEA analysis

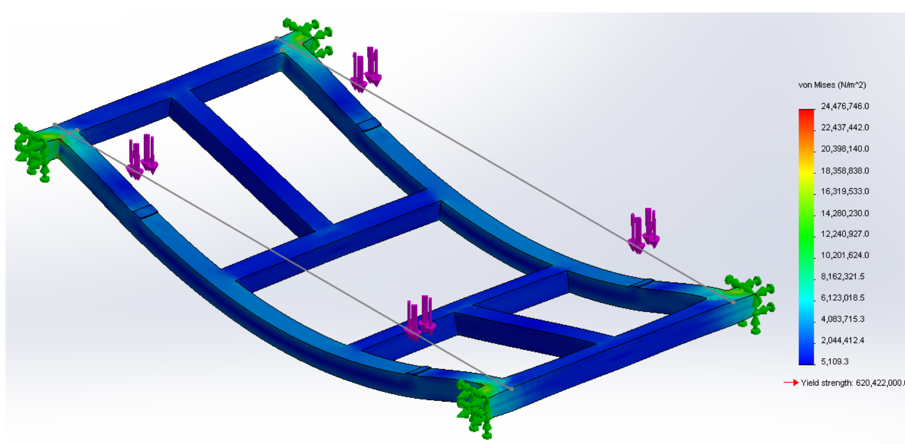
Figure A3 shows the stress distribution of three runs attempted in order to obtain the required beam structure and sizes for the elevator platform to withstand the maximum allowable stress (27.57MPa). First, square tubes of 5×5×1/2 inch was used as the beams, with the bear-bone structure shown in Figure A3(a). As it can be seen, the maximum stress on the structure, occurred at the area of stress concentration, is much higher than the maximum allowable stress. Secondly, two supporting beams were added as shown in Figure A3(b), keeping the square tube dimensions the same. A slight improvement was observed; however, the point of stress concentration still exceeds the maximum allowable stress due to sharp perpendicular connection of the beams. As the last attempt, the size of the square tubing was increased to 8×8×1/2 inch as shown in Figure A3(c). With this configuration, the elevator platform seems to withstand the maximum stress at the sharp corners (24.5MPa).



(a)



(b)



(c)

Figure A3: Three different runs to achieve allowable stress. (a)bare-bone. (b)Added support. (c)increased beam size.

Translating Platform

Following Figure shows the horizontal and vertical forces on the moving platform:

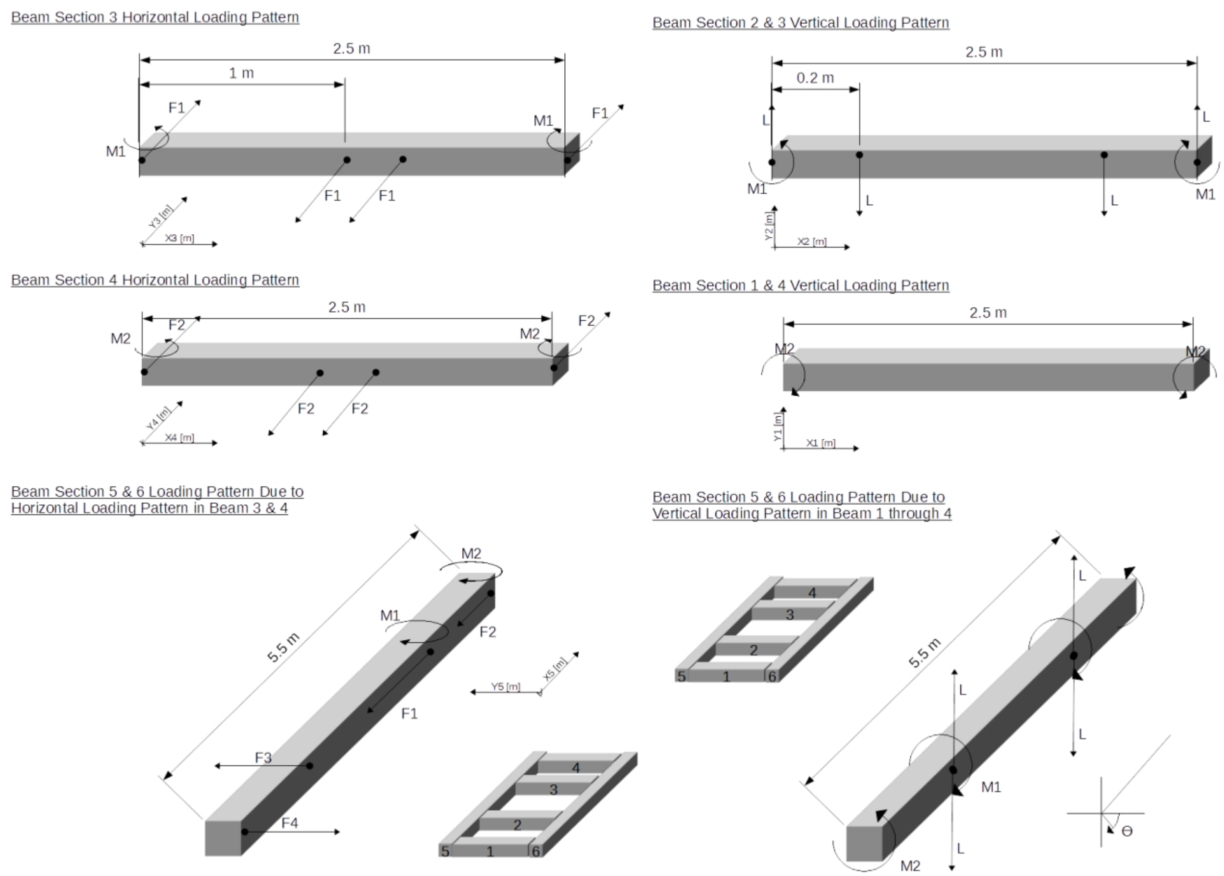


Figure A4: Horizontal (left) and Vertical (right) Loadings on the Moving Platform.

MATLAB Analysis

Considering that vertical loading from the car weight will be symmetric, the following variables are used in the stress analysis:

$M(X_2)$, Moment distribution in beam 2 & 3
M_1, Moment applied on beams 2 & 3 due to twisting of beam 5 & 6
$Y_2(X_2)$, Deflection of beams 2 & 3 in vertical direction
M_2, Moment applied on beams 1 & 4 due to twisting of beam 5 & 6
$Y_1(X_1)$, Deflection of beams 1 & 4 in vertical direction
θ_1, angle of twist in beam 5, at its intersection with beam 1
θ_2, angle of twist in beam 5, at its intersection with beam 1
l_4, length between intersection point between beam 1 & 5, and 1 & 2

Therefore, building the system equations:

$$M_1 = M_2$$

$$\frac{d^2Y_1}{dX_1^2} = \frac{M_2}{EI}, \qquad \frac{d^2Y_2}{dX_2^2} = \frac{M}{EI}$$

$$\frac{dY_1(0)}{dX_1} = \tan(-\theta_1), \qquad \frac{dY_2(0)}{dX_2} = \tan(-\theta_2)$$

$$\theta_2 - \theta_1 = \frac{M_1 l_4}{GK}$$

Also due to symmetric loading:

$$\frac{dY_1(0)}{dX_1} = \frac{dY_1(l)}{dX_1}, \qquad \frac{dY_2(0)}{dX_2} = \frac{dY_2(l)}{dX_2}$$

Where *l* is length of beam 1 through 4. The above set of equations are solved to calculate the unknown value of moment M_1, once known, the stresses in different parts of the structure can be calculated using bending and torsional stress formulas, as may be applicable. Considering the actuation force from delivered by the docking site, and assuming that the stabilizing force is provided by the top surface of the beams 5 & 6, furthermore the actuation dock is considered non-deformable for this case.

$M_{x3}(X_3)$, Moment distribution in beam 3
M_1, Moment applied on beams 3 due to twisting of beam 5 & 6
F_1, force applied on beam 3 by docking members
$Y_3(X_3)$, Deflection of beam 3 in horizontal direction
$M_{x4}(X_4)$, Moment distribution in beam 3
M_2, Moment applied on beam 4 due to twisting of beam 5 & 6
F_2, force applied on beam 4 by docking members
$Y_4(X_4)$, Deflection of beam 4 in horizontal direction
$M_{x5}(X_5)$, Moment distribution in beam 3
F_3, force applied by compression of beam 2 on beams 5 & 6
F_4, force applied by tensioning of beam 1 on beams 5 & 6
$Y_5(X_5)$, Deflection of beam 5 in horizontal plane

Before we build our equations we can keep in mind that loadings in beams 3 & 4 are symmetric, further for each relationship we build, a linearized function about an operating point can be calculated. Building our system equations:

$$\frac{dY_3(0)}{dX_3} = \frac{dY_3(l)}{dX_3}, \qquad \frac{dY_4(0)}{dX_4} = \frac{dY_4(l)}{dX_4}$$

$$\frac{d^2Y_3}{dX_3^2} = \frac{M_{x3}}{EI}, \qquad \frac{d^2Y_4}{dX_4^2} = \frac{M_{x4}}{EI}, \qquad \frac{d^2Y_5}{dX_5^2} = \frac{M_{x5}}{EI}$$

$$F_3 = F_4 = \frac{M_1 + M_2}{l_4}$$

$$F_1 + F_3 = F, [Eq\ 1]$$

where *F* is the force exerted by actuation system on beam 3. By considering that the only unknowns are $M_1 and\ F_1$, the rate of change of deflection of beam 3, at its intersection with beam 5 can be described as:

$$\frac{dY_3(0)}{dX_3} = a_1 + a_2.M_1 + a_3.F_1$$

Further, the displacement at the midsections can be described as:

$$Y_3\left(\frac{l}{2}\right) = b_1 + b_2.M_1 + b_3.F_1$$
$$Y_4\left(\frac{l}{2}\right) = b_1 + b_2.M_2 + b_3.F_2$$

For beam 4, by considering that the only unknowns are M_2 and F_2, the rate of change of deflection of beam 4, at its intersection with beam 5 can be described as:

$$\frac{dY_4(0)}{dX_4} = a_1 + a_2.M_2 + a_3.F_2$$

The coefficients among equations for beam 3 & 4 are the same as they dictate the response of the beam, and hence are dependent only on its structure. For beam 5, the only unknowns are M_1 and M_2, thus the rates of change of deflection at intersections with beams 3 & 4 can be described respectively as:

$$\frac{dY_5(l_5)}{dX_5} = c_1 + c_2.M_1 + c_3.M_2$$

$$\frac{dY_5(l_6)}{dX_5} = d_1 + d_2.M_1 + d_3.M_2$$

Hence defining the relations between the above equations:

$$\frac{dY_3(0)}{dX_3} = \frac{dY_5(l_5)}{dX_5}, [Eq\ 2]$$

$$\frac{dY_4(0)}{dX_4} = \frac{dY_5(l_6)}{dX_5}, [Eq\ 3]$$

$$Y_3\left(\frac{l}{2}\right) = Y_4\left(\frac{l}{2}\right), [Eq\ 4]$$

Thus by solving [Eq. 1] through [Eq. 4] as a system of linear equations we can compute the four unknowns M_1, M_2, F_1, F_2. Once these loadings are known, respective bending stress equations can be used to compute the stress for individual members. Figure A5 illustrates the vertical displacement along each beam (left) and also the rate of change of displacement (right) with respect to horizontal position.

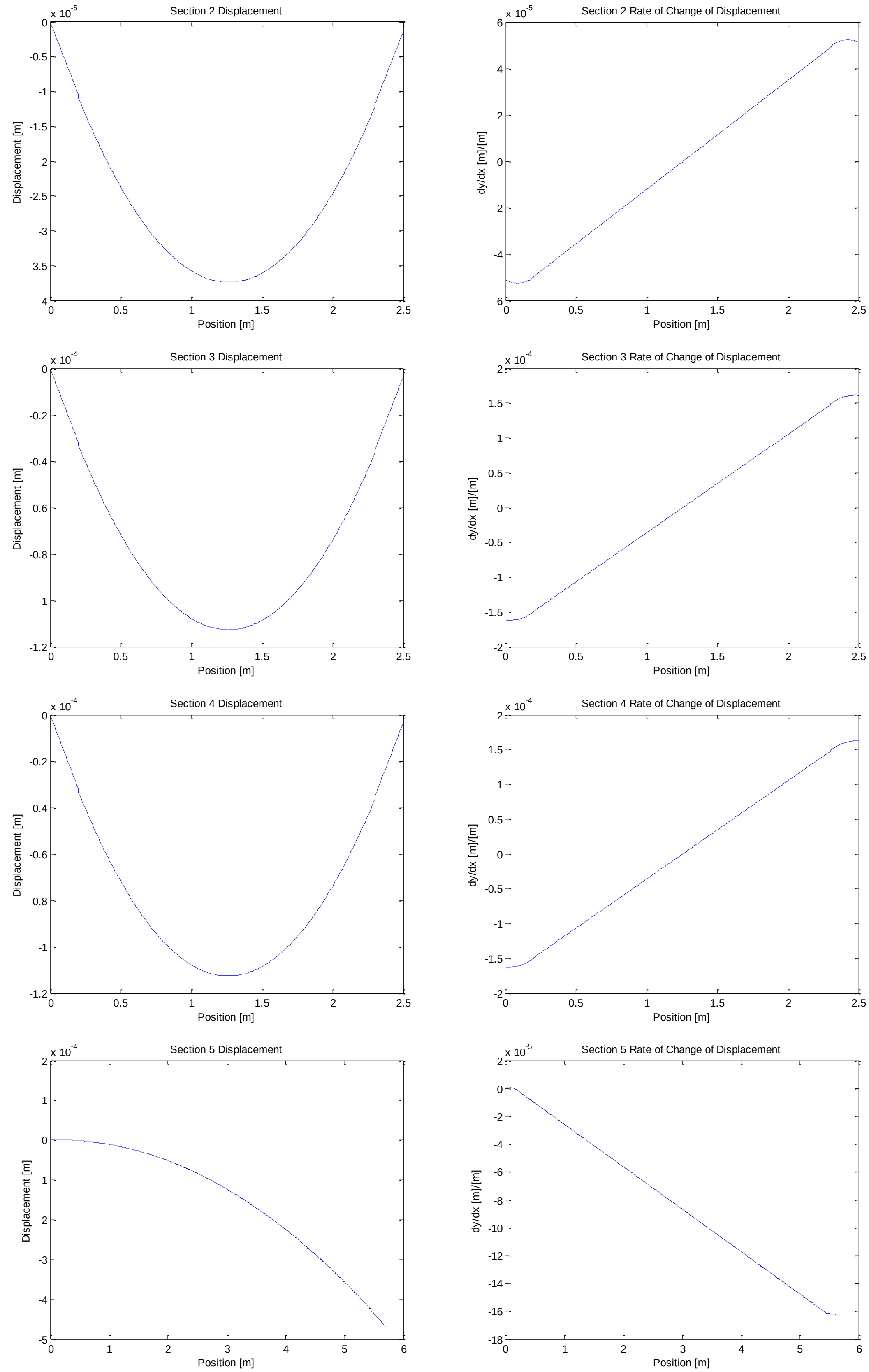
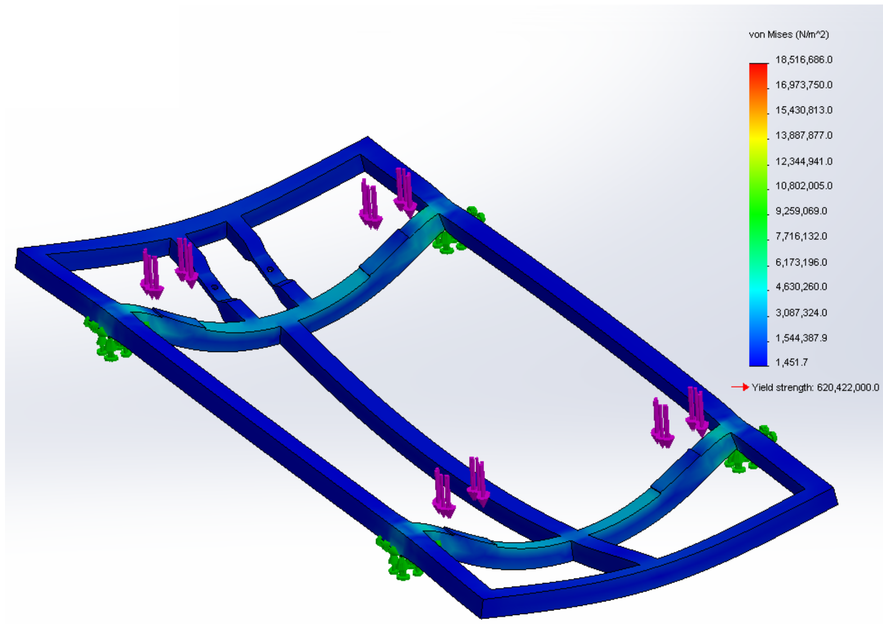


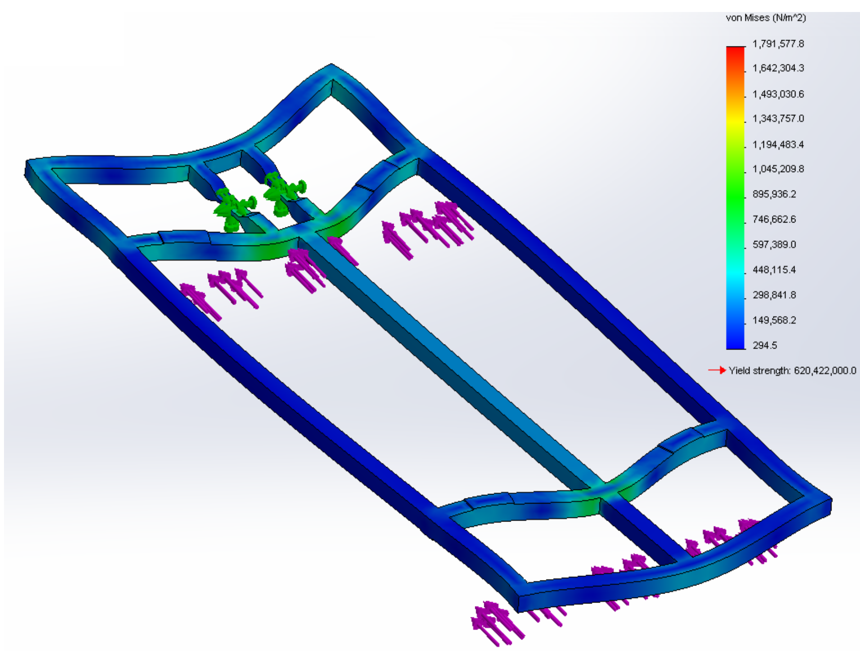
Figure A6: Vertical displacement (right) and the rate of change of displacement (left) across beams in the moving platform.

SolidWorks FEA analysis

The initial square tubing dimensions (5×5×1/2 inch) was first tested to see if the strength of the structure is sufficient given the loadings. Through finite element analysis shown in Figure A7, the maximum pressure on the beam at the concentration point is well below the maximum allowable stress both vertically (fig. A7a) and horizontally (fig. A7b) using the smaller size of the square tubing structures.



(a)



(b)

Figure A7: (a)Vertical and (b) Horizontal Stress Analysis of The Translating Platform.

Appendix 2 (MATLAB CODE)

Moving Platform

```
function movingPlatform()
   clc;
   clear all;
   close all;

   %I beam variables
   IBeam_E = 200*10^9; %Pa

   %Rectagular beam variables
   Rect_E = 200*10^9; %Pa
   Rect_G = 79.3*10^9; %Pa
   Rect_a = 0.127; %m
   IBeam_h = Rect_a; %m
   Rect_b = 0.127; %m
   Rect_t = 0.0127; %m

   %Load and chassis dimensions
   L = 1000; %Nm load exerted by eah wheel
   l_1 = 2.5; %m total length of I beam and shorter rectangular section
   l_2 = 0.2; %m length from closest longer rectangular section and the loading point on I
beam
   l_4 = 0.25; %m length from shorter rectangular section to I beam
   l_5 = 5.7; %m length of section 5

   %Tolerance for finding the beam deflection solution
   Deflect_diff_tolerance = 0.1*10^-3;

   %force exerted by one of the platform actuation points
   Push_Force = 4000; %N

   %Auto configured
   Rect_Area = Rect_a*Rect_b - (Rect_a - 2*Rect_t)*(Rect_b - 2*Rect_t);
   Rect_Q = 2*Rect_t*(Rect_a - Rect_t)*(Rect_b - Rect_t);
   Rect_K = 2*Rect_t*((Rect_a - Rect_t)^2)*((Rect_b - Rect_t)^2)/(Rect_a + Rect_b -
2*Rect_t);
   Rect_Ia = Rect_b*(Rect_a^3)/12 - (Rect_b - 2*Rect_t)*((Rect_a - 2*Rect_t)^3)/12;
   IBeam_I = Rect_Ia;
   Rect_Ib = Rect_a*(Rect_b^3)/12 - (Rect_a - 2*Rect_t)*((Rect_b - 2*Rect_t)^3)/12;
   F = L;
   l_3 = l_1 - l_2; %m

   %Iterating through different moment forces to calculate the solution
   found_hit = 0;
   ranges_IBeam = [];
   ranges_Rect = [];
   k = 1;
   for M_range = 0:10:l_1*L/2 - 10
      [ranges_IBeam(k), disp_diff, disp] = beamLoading_Type1(M_range, l_1, l_2, l_3, L,
IBeam_E, IBeam_I, 0, 'Section 2');
      ranges_Rect(k) = 2*tan(atan(M_range*l_1/(2*Rect_E*Rect_Ia)) +
M_range*l_4/(Rect_G*Rect_K));

      %simulataniously searching for solution within required tolerance
      if(abs(ranges_IBeam(k) - ranges_Rect(k)) <= Deflect_diff_tolerance)
         if(found_hit == 0)
            found_hit = k;
         elseif(abs(ranges_IBeam(k) - ranges_Rect(k)) < abs(ranges_IBeam(found_hit) -
ranges_Rect(found_hit)))
            found_hit = k;
         end
      end
      k = k + 1;
   end

   figure();
   M_range = 0:10:l_1*L/2 - 10;
   plot(M_range, ranges_IBeam);
   hold on;
   plot(M_range, ranges_Rect, 'r');

   if(found_hit == 0)
      display('Intersection not found!');
      return
   end

   %found a solution, doing stress calculations
   plot(M_range(found_hit), ranges_Rect(found_hit), 'o');
   beamLoading_Type1(M_range(found_hit), l_1, l_2, l_3, L, IBeam_E, IBeam_I, 1, 'Section
2');

   Section2BendingStress = ((l_1 - l_3)*L - M_range(found_hit))*(IBeam_h/2)/IBeam_I;
   Section1BendingStress = M_range(found_hit)*(Rect_a/2)/Rect_Ia;
   Section5TorionalStress = M_range(found_hit)/Rect_Q;

   display(Section2BendingStress);
   display(Section1BendingStress);
   display(Section5TorionalStress);

   %stage two for stress due to linear actuation

   %calculating the response of section 5 to M1 and M2, so as to derive a
   %polynomial function describing the rate of change of deflection at
   %interested points in terms of M1 and M2, see beamDiffRate_p1 and beamDiffRate_p2
   rateDefl_1 = [];
   rateDefl_2 = [];
   mRange_row = 1;
   mRange_col = 1;
   xRange = [];
   yRange = [];
   M_range = 10:100:1000;
   rateDelf_11 = [];
   rateDelf_12 = [];
   rateDelf_21 = [];
   rateDelf_22 = [];
   for M_range2 = 10:100:1000
      mRange_col = 1;
      xRange = [xRange, M_range2.*ones(size(M_range))];
      yRange = [yRange, M_range];
      rateDelf_12 = [];
      rateDelf_22 = [];
      for M_range3 = 10:100:1000
         [rateDelf_12(mRange_col), rateDelf_22(mRange_col)] =
beamLoading_Type2(M_range2, M_range3, l_4, l_5, Rect_E, Rect_Ia, 0, 'Section 5');
         mRange_col = mRange_col + 1;
      end
      rateDelf_11 = [rateDelf_11, rateDelf_12];
      rateDelf_21 = [rateDelf_21, rateDelf_22];
      mRange_row = mRange_row + 1;
   end

   beamDiffRate_p1 = fit([xRange', yRange'], rateDelf_11', 'poly11');
   eq1 = coeffvalues(beamDiffRate_p1);
   beamDiffRate_p2 = fit([xRange', yRange'], rateDelf_21', 'poly11');
   eq2 = coeffvalues(beamDiffRate_p2);

   %calculating the response of Section 3 and 4, so as to derive the rate
   %of change of deflection and dispalcement in terms of M1, F1, or M2,
   %F2, see beamDiffRate_M_F and beamMidDisp_M_F
   rateDefl_beam1 = [];
   rateDefl_beam2 = [];
   midDefl_beam1 = [];
   midDefl_beam2 = [];
   mRange_row = 1;
   xRange = [];
   yRange = [];
   M_range = 0:100:1000;
```

```
    for M_range2 = 0:100:1000
        xRange = [xRange, M_range2.*ones(size(M_range))];
        yRange = [yRange, 0:800:8000];
        mRange_col = 1;
        midDefl_beam2 = [];
        rateDefl_beam2 = [];
        for L_range = 0:800:8000
            [a, b, c] = beamLoading_Type1(M_range2, l_1, l_2, l_3, L_range, IBeam_E, IBeam_I, 0,
'Section 4');
            rateDefl_beam2(mRange_col) = b(1);
            sizeC = size(c);
            midDefl_beam2(mRange_col) = c(round(sizeC(2)/2));
            mRange_col = mRange_col + 1;
        end
        rateDefl_beam1 = [rateDefl_beam1, rateDefl_beam2];
        midDefl_beam1 = [midDefl_beam1, midDefl_beam2];
        mRange_row = mRange_row + 1;
    end

    beamDiffRate_M_F = fit([xRange', yRange'], rateDefl_beam1', 'poly11');
    eq3 = coeffvalues(beamDiffRate_M_F);
    beamMidDisp_M_F = fit([xRange', yRange'], midDefl_beam1', 'poly11');
    eq4 = coeffvalues(beamMidDisp_M_F);

    %solving linear set of equations to solve for M1, M2, F1, F2
    b = [eq1(1) - eq3(1); eq2(1) - eq3(1); 0; Push_Force];
    A = [eq3(2) - eq1(2), -eq1(3), eq3(3), 0; -eq2(2), eq3(2) - eq2(3), 0, eq3(3); eq4(2), -eq4(2),
eq4(3), -eq4(3); 0, 0, 1, 1];
    sol = A\b;
    M1 = sol(1);
    M2 = sol(2);
    F1 = sol(3);
    F2 = sol(4);

    %display the moment diagrams and deflection
    beamLoading_Type2(M1, M2, l_4, l_5, Rect_E, Rect_Ia, 1, 'Section 5');
    beamLoading_Type1(M1, l_1, l_2, l_3, F1, IBeam_E, IBeam_I, 1, 'Section 3');
    beamLoading_Type1(M2, l_1, l_2, l_3, F2, IBeam_E, IBeam_I, 1, 'Section 4');

    %calculating the stress duw to actuation
    Section3BendingStress = ((l_1 - l_3)*F1 - M1)*(IBeam_h/2)/IBeam_I;
    Section4BendingStress = ((l_1 - l_3)*F2 - M2)*(IBeam_h/2)/IBeam_I;
    Section5BendingStress = (M1 + M2)*(Rect_b/2)/Rect_Ib;
    Section5CompressiveStress = (F1 + F2)/Rect_Area;

    display(Section3BendingStress);
    display(Section4BendingStress);
    display(Section5BendingStress);
    display(Section5CompressiveStress);

    %calculating net stress due to actuation and weight
    Section3NetStress = Section2BendingStress + Section3BendingStress;
    Section4NetStress = Section4BendingStress + Section1BendingStress;
    Section5NetStress = (Section5BendingStress + Section5CompressiveStress)/2 +
sqrt(((Section5BendingStress + Section5CompressiveStress)/2)^2 + Section5TorionalStress);

    display(Section3NetStress);
    display(Section4NetStress);
    display(Section5NetStress);

    display('Done');
end

function movingPlatform()
    clc;
    clear all;
    close all;

    %I beam variables
    IBeam_E = 200*10^9; %Pa

    %Rectagular beam variables
    Rect_E = 200*10^9; %Pa
    Rect_G = 79.3*10^9; %Pa
    Rect_a = 0.127; %m
    IBeam_h = Rect_a; %m
    Rect_b = 0.127; %m
    Rect_t = 0.0127; %m

    %Load and chassis dimensions
    L = 1000; %Nm load exerted by eah wheel
    l_1 = 2.5; %m total length of I beam and shorter rectangular section
    l_2 = 0.2; %m length from closest longer rectangular section and the loading point on I
beam
    l_4 = 0.25; %m length from shorter rectangular section to I beam
    l_5 = 5.7; %m length of section 5

    %Tolerance for finding the beam deflection solution
    Deflect_diff_tolerance = 0.1*10^-3;

    %force exerted by one of the platform actuation points
    Push_Force = 4000; %N

    %Auto configured
    Rect_Area = Rect_a*Rect_b - (Rect_a - 2*Rect_t)*(Rect_b - 2*Rect_t);
    Rect_Q = 2*Rect_t*(Rect_a - Rect_t)*(Rect_b - Rect_t);
    Rect_K = 2*Rect_t*((Rect_a - Rect_t)^2)*((Rect_b - Rect_t)^2)/(Rect_a + Rect_b -
2*Rect_t);
    Rect_Ia = Rect_b*(Rect_a^3)/12 - (Rect_b - 2*Rect_t)*((Rect_a - 2*Rect_t)^3)/12;
    IBeam_I = Rect_Ia;
    Rect_Ib = Rect_a*(Rect_b^3)/12 - (Rect_a - 2*Rect_t)*((Rect_b - 2*Rect_t)^3)/12;
    F = L;
    l_3 = l_1 - l_2; %m

    %Iterating through different moment forces to calculate the solution
    found_hit = 0;
    ranges_IBeam = [];
    ranges_Rect = [];
    k = 1;
    for M_range = 0:10:l_1*L/2 - 10
        [ranges_IBeam(k), disp_diff, disp] = beamLoading_Type1(M_range, l_1, l_2, l_3, L,
IBeam_E, IBeam_I, 0, 'Section 2');
        ranges_Rect(k) = 2*tan(atan(M_range*l_1/(2*Rect_E*Rect_Ia)) +
M_range*l_4/(Rect_G*Rect_K));

        %simulataniously searching for solution within required tolerance
        if(abs(ranges_IBeam(k) - ranges_Rect(k)) <= Deflect_diff_tolerance)
            if(found_hit == 0)
                found_hit = k;
            elseif(abs(ranges_IBeam(k) - ranges_Rect(k)) < abs(ranges_IBeam(found_hit) -
ranges_Rect(found_hit)))
                found_hit = k;
            end
        end
        k = k + 1;
    end

    figure();
    M_range = 0:10:l_1*L/2 - 10;
    plot(M_range, ranges_IBeam);
    hold on;
    plot(M_range, ranges_Rect, 'r');

    if(found_hit == 0)
        display('Intersection not found!');
        return
    end

    %found a solution, doing stress calculations
    plot(M_range(found_hit), ranges_Rect(found_hit), 'o');
    beamLoading_Type1(M_range(found_hit), l_1, l_2, l_3, L, IBeam_E, IBeam_I, 1, 'Section
2');

    Section2BendingStress = ((l_1 - l_3)*L - M_range(found_hit))*(IBeam_h/2)/IBeam_I;
    Section1BendingStress = M_range(found_hit)*(Rect_a/2)/Rect_Ia;
    Section5TorionalStress = M_range(found_hit)/Rect_Q;

    display(Section2BendingStress);
    display(Section1BendingStress);
    display(Section5TorionalStress);

    %stage two for stress due to linear actuation

    %calculating the response of section 5 to M1 and M2, so as to derive a
    %polynomial function describing the rate of change of deflection at
    %interested points in terms of M1 and M2, see beamDiffRate_p1 and beamDiffRate_p2
```

```
    rateDefl_1 = [];
    rateDefl_2 = [];
    mRange_row = 1;
    mRange_col = 1;
    xRange = [];
    yRange = [];
    M_range = 10:100:1000;
    rateDelf_11 = [];
    rateDelf_12 = [];
    rateDelf_21 = [];
    rateDelf_22 = [];
    for M_range2 = 10:100:1000
      mRange_col = 1;
      xRange = [xRange, M_range2.*ones(size(M_range))];
      yRange = [yRange, M_range];
      rateDelf_12 = [];
      rateDelf_22 = [];
      for M_range3 = 10:100:1000
        [rateDelf_12(mRange_col), rateDelf_22(mRange_col)] =
  beamLoading_Type2(M_range2, M_range3, l_4, l_5, Rect_E, Rect_Ia, 0, 'Section 5');
        mRange_col = mRange_col + 1;
      end
      rateDelf_11 = [rateDelf_11, rateDelf_12];
      rateDelf_21 = [rateDelf_21, rateDelf_22];
      mRange_row = mRange_row + 1;
    end

    beamDiffRate_p1 = fit([xRange', yRange'], rateDelf_11', 'poly11');
    eq1 = coeffvalues(beamDiffRate_p1);
    beamDiffRate_p2 = fit([xRange', yRange'], rateDelf_21', 'poly11');
    eq2 = coeffvalues(beamDiffRate_p2);

    %calculating the response of Section 3 and 4, so as to derive the rate
    %of change of deflection and dispalcement in terms of M1, F1, or M2,
    %F2, see beamDiffRate_M_F and beamMidDisp_M_F
    rateDefl_beam1 = [];
    rateDefl_beam2 = [];
    midDefl_beam1 = [];
    midDefl_beam2 = [];
    mRange_row = 1;
    xRange = [];
    yRange = [];
    M_range = 0:100:1000;
    for M_range2 = 0:100:1000
      xRange = [xRange, M_range2.*ones(size(M_range))];
      yRange = [yRange, 0:800:8000];
      mRange_col = 1;
      midDefl_beam2 = [];
      rateDefl_beam2 = [];
      for L_range = 0:800:8000
        [a, b, c] = beamLoading_Type1(M_range2, l_1, l_2, l_3, L_range, IBeam_E, IBeam_I, 0,
  'Section 4');
        rateDefl_beam2(mRange_col) = b(1);
        sizeC = size(c);
        midDefl_beam2(mRange_col) = c(round(sizeC(2)/2));
        mRange_col = mRange_col + 1;
      end
      rateDefl_beam1 = [rateDefl_beam1, rateDefl_beam2];
      midDefl_beam1 = [midDefl_beam1, midDefl_beam2];
      mRange_row = mRange_row + 1;
    end

    beamDiffRate_M_F = fit([xRange', yRange'], rateDefl_beam1', 'poly11');
    eq3 = coeffvalues(beamDiffRate_M_F);
    beamMidDisp_M_F = fit([xRange', yRange'], midDefl_beam1', 'poly11');
    eq4 = coeffvalues(beamMidDisp_M_F);

    %solving linear set of equations to solve for M1, M2, F1, F2
    b = [eq1(1) - eq3(1); eq2(1) - eq3(1); 0; Push_Force];
    A = [eq3(2) - eq1(2), -eq1(3), eq3(3), 0; -eq2(2), eq3(2) - eq2(2), 0, eq3(3); eq4(2), -eq4(2),
  eq4(3), -eq4(3); 0, 0, 1, 1];
    sol = A\b;
    M1 = sol(1);
    M2 = sol(2);
    F1 = sol(3);
    F2 = sol(4);

    %display the moment diagrams and deflection
    beamLoading_Type2(M1, M2, l_4, l_5, Rect_E, Rect_Ia, 1, 'Section 5');
    beamLoading_Type1(M1, l_1, l_2, l_3, F1, IBeam_E, IBeam_I, 1, 'Section 3');
    beamLoading_Type1(M2, l_1, l_2, l_3, F2, IBeam_E, IBeam_I, 1, 'Section 4');

    %calculating the stress duw to actuation
    Section3BendingStress = ((l_1 - l_3)*F1 - M1)*(IBeam_h/2)/IBeam_I;
    Section4BendingStress = ((l_1 - l_3)*F2 - M2)*(IBeam_h/2)/IBeam_I;
    Section5BendingStress = (M1 + M2)*(Rect_b/2)/Rect_Ib;
    Section5CompressiveStress = (F1 + F2)/Rect_Area;

    display(Section3BendingStress);
    display(Section4BendingStress);
    display(Section5BendingStress);
    display(Section5CompressiveStress);

    %calculating net stress due to actuation and weight
    Section3NetStress = Section2BendingStress + Section3BendingStress;
    Section4NetStress = Section4BendingStress + Section1BendingStress;
    Section5NetStress = (Section5BendingStress + Section5CompressiveStress)/2 +
  sqrt(((Section5BendingStress + Section5CompressiveStress)/2)^2 + Section5TorionalStress);

    display(Section3NetStress);
    display(Section4NetStress);
    display(Section5NetStress);

    display('Done');
  end
```

Square Beam

```
function IBeam()
  clc;
  clear all;
  close all;

  %I beam variables
  IBeam_E = 200*10^9; %Pa
  IBeam_h = 0.206; %m
  IBeam_I = 1.998*10^-5;

  %Rectagular beam variables
  Rect_E = 200*10^9; %Pa
  Rect_G = 79.3*10^9; %Pa
  Rect_a = 0.206; %m
  Rect_b = 0.15; %m
  Rect_t = 0.02; %m

  %Load and chassis dimensions
  L = 1000; %Nm load exerted by eah wheel
  l_1 = 10; %m total length of I beam and shorter rectangular section
  l_2 = 2; %m length from closest longer rectangular section and the loading point on I beam
  l_4 = 1; %m length from shorter rectangular section to I beam

  %Tolerance for finding the beam deflection solution
  Deflect_diff_tolerance = 0.04*10^-3;

  %Auto configured
  Rect_Q = 2*Rect_t*(Rect_a - Rect_t)*(Rect_b - Rect_t);
  Rect_K = 2*Rect_t*((Rect_a - Rect_t)^2)*((Rect_b - Rect_t)^2)/(Rect_a + Rect_b - 2*Rect_t);
  Rect_I = Rect_b*(Rect_a^3)/12 - (Rect_b - 2*Rect_t)*((Rect_a - 2*Rect_t)^3)/12;
  F = L;
  l_3 = l_1 - l_2; %m

  %Iterating through different moment forces to calculate the solution
  found_hit = 0;
  ranges_IBeam = [];
  ranges_Rect = [];
  k = 1;
```

```
    for M_range = 0:10:l_1*F/2 - 10
        ranges_IBeam(k) = IBeam_defl(M_range, 0);
        ranges_Rect(k) = 2*tan(atan(M_range*l_1/(2*Rect_E*Rect_I)) +
M_range*l_4/(Rect_G*Rect_K));
        if(abs(ranges_IBeam(k) - ranges_Rect(k)) <= Deflect_diff_tolerance)
            if(found_hit == 0)
                found_hit = k;
            elseif(abs(ranges_IBeam(k) - ranges_Rect(k)) < abs(ranges_IBeam(found_hit) -
ranges_Rect(found_hit)))
                found_hit = k;
            end
        end
        k = k + 1;
    end

    M_range = 0:10:l_1*F/2 - 10;
    plot(M_range, ranges_IBeam);
    hold on;
    plot(M_range, ranges_Rect, 'r');

    if(found_hit == 0)
        display('Intersection not found!');
        return
    end

    %found a solution, doing stress calculations
    plot(M_range(found_hit), ranges_Rect(found_hit), 'o');
    IBeam_defl(M_range(found_hit), 1);

    IBeam_bendingStress = ((l_1 - l_2)*L/2 + M_range(found_hit))*(IBeam_h/2)/IBeam_I;
    Rect_bendingStress = M_range(found_hit)*(Rect_a/2)/Rect_I;
    Rect_torsionalStress = M_range(found_hit)/Rect_Q;

    display(IBeam_bendingStress);
    display(Rect_bendingStress);
    display(Rect_torsionalStress);

    %function calculates the range of rate of deflection from one end of I
    %beam to the other
    %args-
    %M Moment applied at the ends of the beam by rectangular bars
    %dispFigure 1 to display the moment diagram and deflection in the beam
    function [range] = IBeam_defl(M, dispFigure)

        pos_interval = 0.01;
        x_1 = 0:pos_interval:l_2;
        x_2 = l_2:pos_interval:l_3;
        size_x2 = size(x_2);
        x_3 = l_3:pos_interval:l_1;
        %building the moment distribution through the beam
        M_x_1 = -(M + (l_2 - x_1).*L + (l_3 - x_1).*L - (l_1 - x_1).*F);
        M_x_2 = -(M + (l_3 - x_2).*L - (l_1 - x_2).*F);
        M_x_3 = -M + (l_1 - x_3).*F;

        M_x = [M_x_1, M_x_2, M_x_3];
        x = [x_1, x_2, x_3];

        size_M = size(M_x);
        M_integ_1(size_M(2)) = 0;
        M_integ_1(1) = 0;
        M_integ_2(size_M(2)) = 0;
        M_integ_2(1) = 0;
        %first integral to find the rate of deflection for the beam
        for i = 1:(size_M(2) - 1)
            M_integ_1(i+1) = (pos_interval*M_x(i))/(IBeam_E*IBeam_I) + M_integ_1(i);
        end

        %readjusting as the rate oif deflection has to be symmetric and
        %inverted about the midpoint of the point, as this is the
        %methamatical requirement for symmetric loading.
        M_integ_1 = M_integ_1 - (M_integ_1(end) - M_integ_1(1))/2;

        %calculating the beam deflection
        for i = 1:(size_M(2) - 1)
            M_integ_2(i+1) = pos_interval*M_integ_1(i) + M_integ_2(i);
        end

        if(dispFigure)
            figure();
            plot(x, M_x);
            figure();
            plot(x, M_integ_1, 'r');
            figure();
            plot(x, M_integ_2, 'g');
        end

        range = M_integ_1(end)*2;
    end

end
```

Elevator Platform

```
function elevatorPlatform()
    clc;
    clear all;
    close all;

    %Rectagular beam variables
    Rect_E = 200*10^9; %Pa
    Rect_G = 79.3*10^9; %Pa
    Rect_a = 0.2032; %m
    Rect_b = 0.2032; %m
    Rect_t = 0.0127; %m

    %Load and chassis dimensions
    L = 2600*(9.81 + 2)/4;
    l_1 = 0.15;
    l_2 = 2.8;
    l_3 = 1;
    l_4 = 6;

    %Auto configured
    Rect_Area = Rect_a*Rect_b - (Rect_a - 2*Rect_t)*(Rect_b - 2*Rect_t);
    Rect_Q = 2*Rect_t*(Rect_a - Rect_t)*(Rect_b - Rect_t);
    Rect_K = 2*Rect_t*((Rect_a - Rect_t)^2)*((Rect_b - Rect_t)^2)/(Rect_a + Rect_b - 2*Rect_t);
    Rect_Ia = Rect_b*(Rect_a^3)/12 - (Rect_b - 2*Rect_t)*(Rect_a - 2*Rect_t)^3)/12;
    Rect_Ib = Rect_a*(Rect_b^3)/12 - (Rect_a - 2*Rect_t)*((Rect_b - 2*Rect_t)^3)/12;

    Deflect_diff_tolerance = 0.04*10^-3;

    %running through max force config

    diffRange = [];
    numStored = 1;
    foundhit = 0;
    M_range = 0:10:l_4*L/2 - 10;
    for M = M_range
        [diffRange(numStored), deflDiff, delf] = beamLoading_Type1(M, l_4, l_3, l_4 - l_3, L, Rect_E,
Rect_Ia, 0, 'Section 3');
        diffRange_Section1(numStored) = tan((M + L*Rect_b/2)*l_1/(Rect_G*Rect_K));

        if(abs(diffRange(numStored) - diffRange_Section1(numStored)) < Deflect_diff_tolerance)
            if(foundhit == 0)
                foundhit = numStored;
            else
                if(abs(diffRange(numStored) - diffRange_Section1(numStored)) <
abs(diffRange(foundhit) - diffRange_Section1(foundhit)))
                    foundhit = numStored;
                end
            end
        end
```

```
    numStored = numStored + 1;
  end

  figure();
  plot(M_range, diffRange_Section1, 'r');
  hold on;
  plot(M_range, diffRange);

  if(foundhit == 0)
    display('Intersection not found!');
    return
  end

  plot(M_range(foundhit), diffRange(foundhit), 'o');
  beamLoading_Type1(M_range(foundhit), l_4, l_3, l_4 - l_3, L, Rect_E, Rect_Ia, 1, 'Section 3
Max Force');

  Section3BendingStress = ((l_3)*L - M_range(foundhit))*(Rect_a/2)/Rect_Ia;
  Section1TorsionalStress = M_range(foundhit)/Rect_K;
  Section1BendingStress = L*l_1*(Rect_a/2)/Rect_Ia;

  Section3MaxForceNetStress = Section3BendingStress;
  Seciton1MaxForceNetStress = (Section1BendingStress/2) + sqrt((Section1BendingStress/2)^2
+ Section1TorsionalStress);

  display(Section3MaxForceNetStress);
  display(Seciton1MaxForceNetStress);

  %running through max moment config

  diffRange = [];
  numStored = 1;
  foundhit = 0;
  M_range = 0:10:l_4*L/2 - 10;
  for M = M_range
    [diffRange(numStored), deflDiff, delf] = beamLoading_Type1(M, l_4, l_4/2, l_4/2, L/2,
Rect_E, Rect_Ia, 0, 'Section 3');
    diffRange_Section1(numStored) = tan((M + L*Rect_b/4)*l_1/(Rect_G*Rect_K));
    if(abs(diffRange(numStored) - diffRange_Section1(numStored)) < Deflect_diff_tolerance)
      if(foundhit == 0)
        foundhit = numStored;
      else
        if(abs(diffRange(numStored) - diffRange_Section1(numStored)) <
abs(diffRange(foundhit) - diffRange_Section1(foundhit)))
          foundhit = numStored;
        end
      end
    end

    numStored = numStored + 1;
  end

  figure();
  plot(M_range, diffRange_Section1, 'r');
  hold on;
  plot(M_range, diffRange);

   if(foundhit == 0)
    display('Intersection not found!');
    return
   end

   plot(M_range(foundhit), diffRange(foundhit), 'o');
   beamLoading_Type1(M_range(foundhit), l_4, l_4/2, l_4/2, L/2, Rect_E, Rect_Ia, 1, 'Section 3
Max Moment');

   Section3BendingStress = ((l_3)*L - M_range(foundhit))*(Rect_a/2)/Rect_Ia;
   Section1TorsionalStress = M_range(foundhit)/Rect_K;
   Section1BendingStress = (L)*l_1*(Rect_a/2)/Rect_Ia;

   Section3MaxForceNetStress = Section3BendingStress;
   Seciton1MaxForceNetStress = (Section1BendingStress/2) + sqrt((Section1BendingStress/2)^2
+ Section1TorsionalStress);

   display(Section3MaxForceNetStress);
   display(Seciton1MaxForceNetStress);
end
```

Bi-Section

```
function x = bisection(a, b, threshold, n)
  %clc;
  %clear all;
  close all;

  if b < a
    tmp = b;
    b = a;
    a = tmp;
  end
  if n < 1
    n = 1;
  elseif n > 1000
    n = 1000;
  end
  fprintf('Running with:\n[a = %d][b = %d][threshold = %d][n = %d]\n', a, b, threshold, n);
  if b == a
    x = a;
    return;
  end

  ya = evaluateFunc(a);
  yb = evaluateFunc(b);
  x = (a+b)/2;
  y = evaluateFunc(x);
  figure;
  scatter([a, x, b], [ya, y, yb], 25, 'b', '.');
  hold on;
  ymin = min([ya, y, yb]); ymax = max([ya, y, yb]);
  axis([a-(b-a)/8, b+(b-a)/8, ymin-(ymax-ymin)/8, ymax+(ymax-ymin)/8]);

  i = 0;
  while (b-a)/2 > threshold
    i=i+1;
    hold all;
    if sign(ya)*sign(y) < 0
      b = x;
      yb = evaluateFunc(b);
      x = (a+b)/2;
      y = evaluateFunc(x);
      scatter([x, b], [y, yb], 25, 'b', '.');
    elseif sign(y)*sign(yb) < 0
      a = x;
      ya = evaluateFunc(a);
      x = (a+b)/2;
      y = evaluateFunc(x);
      scatter([a, x], [ya, y], 25, 'b', '.');
    elseif ya == 0
      x = a; % x = a is the solution
      break;
    elseif yb == 0
      x = b; % x = b is the solution
      break;
    else
      break; % No solution or x is the solution
    end
    hold off;
    drawnow;

    if i >= n
      break;
    end
  end

  fprintf('Finished at:\n[i = %d][f(x) = %d (error bound = %d)]\n', i, y, (b-a)/2);
end
```

```
function y = evaluateFunc(x)
    y = (127.11/(2*(0.42*x-4*x^2))+sqrt((127.11/(2*(0.42*x-4*x^2)))^2 + (669.53/(2*x*(0.105-x)^2))^2))/2;
    y = y*(1/(146.124*10^6) + 1/(450*10^6)) - 1/5;
end
```

Initial Beam loading

```
%calculates the deflction of a beam under the given condition
%     l2
%<----------->
% M,L       L                    L          M,L
% ---------------------------------------------------
% ---------------------------------------------------
%<------------------------------------>
%             l3
%<---------------------------------------------------->
%                  l1
%dispFigure, 1 to display the moment distribution, and deflection diagram
%name, string name to put in the figures opened by above option

function [range, M_integ_1, M_integ_2] = beamLoading_Type1(M, l_1, l_2, l_3, L, IBeam_E, IBeam_I, dispFigure, name)

    pos_interval = 0.01;
    x_1 = 0:pos_interval:l_2;
    x_2 = l_2:pos_interval:l_3;
    size_x2 = size(x_2);
    x_3 = l_3:pos_interval:l_1;
    %building the moment distribution through the beam
    M_x_1 = -(M + (l_2 - x_1).*L + (l_3 - x_1).*L - (l_1 - x_1).*L);
    M_x_2 = -(M + (l_3 - x_2).*L - (l_1 - x_2).*L);
    M_x_3 = -M + (l_1 - x_3).*L;

    M_x = [M_x_1, M_x_2, M_x_3];
    x = [x_1, x_2, x_3];

    M_integ_1 = beamDeflDiff(M_x, pos_interval, IBeam_E, IBeam_I);

    %readjusting as the rate oif deflection has to be symmetric and
    %inverted about the midpoint of the point, as this is the
    %methamatical requirement for symmetric loading.
    M_integ_1 = M_integ_1 - (M_integ_1(end) - M_integ_1(1))/2;

    %calculating the beam deflection
    M_integ_2 = beamDefl(M_integ_1, pos_interval);

    if(dispFigure)
        figure();
        plot(x, M_x);
        hold on;
        title([name ' Moment Distribution']);
        xlabel('Position [m]');
        ylabel('Moment [Nm]');

        figure();
        plot(x, M_integ_1);
        hold on;
        title([name ' Rate of Change of Displacement']);
        xlabel('Position [m]');
        ylabel('dy/dx [m]/[m]');

        figure();
        plot(x, M_integ_2);
        hold on;
        title([name ' Displacement']);
        xlabel('Position [m]');
        ylabel('Displacement [m]');
    end

    range = M_integ_1(end)*2;
end
```

Secondary Beam Loading

```
%calculates the deflction of a beam under the given condition
%    l4                          l4
%<---------→                ←---------→
% F4        F3                  M1          M2
% ---------------------------------------------------
% ---------------------------------------------------
%<-------------------------------------------------→
%                  l5
%dispFigure, 1 to display the moment distribution, and deflection diagram
%name, string name to put in the figures opened by above option


function [rd_1, rd_2] = beamLoading_Type2(M1, M2, l_4, l_5, Rect_E, Rect_I, dispFigure, name)
    F_3 = (M1 + M2) / l_4;

    pos_interval = 0.01;
    x1_dist = 0:pos_interval:l_4;
    x2_dist = l_4:pos_interval:l_5 – l_4;
    x3_dist = l_5 – l_4:pos_interval:l_5;

    sizeX1 = size(x1_dist);
    M_x1 = -(M1 + M2) + F_3.*(l_4 – x1_dist);
    sizeX2 = size(x2_dist);
    M_x2 = -(M1 + M2).*ones(sizeX2);
    sizeX3 = size(x3_dist);
    M_x3 = -M2.*ones(sizeX3);

    x_dist = [x1_dist, x2_dist, x3_dist];
    M_dist = [M_x1, M_x2, M_x3];
    M_integ_1 = beamDeflDiff(M_dist, pos_interval, Rect_E, Rect_I);
    M_integ_2 = beamDefl(M_integ_1, pos_interval);

    D = -M_integ_2(1);
    C = -(M_integ_2(sizeX1(2)) + D)/l_4;

    M_integ_1 = M_integ_1 + C;
    M_integ_2 = M_integ_2 + C.*x_dist + D;

    if(dispFigure)
        figure();
        plot(x_dist, M_dist);
        hold on;
        title([name ‘ Moment Distribution’]);
        xlabel(‘Position [m]’);
        ylabel(‘Moment [Nm]’);

        figure();
        plot(x_dist, M_integ_1);
        hold on;
        title([name ‘ Rate of Change of Displacement’]);
        xlabel(‘Position [m]’);
        ylabel(‘dy/dx [m]/[m]’);

        figure();
        plot(x_dist, M_integ_2);
        hold on;
        title([name ‘ Displacement’]);
        xlabel(‘Position [m]’);
```

```
    ylabel('Displacement [m]');
  end

  hold on;
```

```
        rd_1 = M_integ_1(sizeX1(2) + sizeX2(2));
        rd_2 = M_integ_1(end);
    end
```

Beam Deflection

```
%computes the rate of change of deflection
%arguments
%M_x, moment distribution [Nm]
%pos_interval, delta x being used to represent M_x [m]
%E, modulus of elasticity [Pa]
%I, Inertia against the bending of beam [m^4]

function [M_integ_1] = beamDeflDiff(M_x, pos_interval, E, I)
  size_M = size(M_x);
  M_integ_1(size_M(2)) = 0;
  M_integ_1(1) = 0;
  %first integral to find the rate of deflection for the beam
  for i = 1:(size_M(2) - 1)
    M_integ_1(i+1) = (pos_interval*M_x(i))/(E*I) + M_integ_1(i);
  end
end
%computes the deflection of beam
%arguments
%M_integ_1, rate of change of deflection, computed by beamDeflDiff(...)
%pos_interval, pos_interval passed to beamDeflDiff(...)

function [M_integ_2] = beamDefl(M_integ_1, pos_interval)
  size_M = size(M_integ_1);
  M_integ_2(size_M(2)) = 0;
  M_integ_2(1) = 0;
  for i = 1:(size_M(2) - 1)
    M_integ_2(i+1) = pos_interval*M_integ_1(i) + M_integ_2(i);
  end
end
```