Autonomous Mobile Manipulator

Project Execution Plan

By Tom Queen and Daniel Gregory-Turner

Contents

Abstract

Over the last decade autonomous mobile manipulators (AMMs) have begun to creep onto the market. With applications ranging from helping the elderly and disabled to stacking shelves in a supermarket, the potential market for autonomous mobile manipulators is huge. The field of robotics is advancing rapidly, but with robots like the PR2 retailing at $300,000+ these systems are unobtainable by all but the highest funded robotics research labs. In this project we will develop a prototype autonomous mobile manipulator platform for less than 30 times the price of this commercially available robot, with the aim of teaching the robot to locate, grasp and transport objects.

Project description and objectives

Research Question and objectives

Question

Can a compliant robot arm be integrated into a self-navigating robot platform and programmed to autonomously to identify, locate, grasp and transport objects using data from cascaded camera systems with comparable reliability to existing systems as a function of development time and cost?

Objectives

- Evaluate current methods of autonomous object fetching with respect to functionality and reliability as a function of cost
- Explore methods of camera based object recognition and localisation from cascaded camera systems, and the interaction between these systems
- Investigate and develop a novel approach to grasping a variety of objects
- Build a system capable of autonomously navigating an environment, locating a desired object and transporting the object to a location with comparable reliability to existing systems as a function of development time and cost.

Requirements

Risks to project are out of five.

1) Object recognition

Requirement

The system must be able to autonomously identify desired objects.

Rationale

To enable the system to differentiate between desired and undesired objects.

Objective:

Must(s)	Should(s)	Could(s)
Recognise a single object at a single orientation from 2m	Recognise a single object at multiple orientations from a distance greater than 2m	Recognise multiple objects; differentiate between two similar objects

Test Scheme:

Success rate;

Must(s)	Should(s)	Could(s)
70% true positive, 10% false positive in a single structured environment	Do better	Do better

Risk to the project if not complete:

5

2) General object location

Requirement:

The system must be able to determine the location of a desired object.

Rationale:

This is needed to provide a goal location for the mobile platform to bring the gripper with reach of the object.

Objective:

Must(s)	Should(s)	Could(s)

Provide location of object within a 1m radius	Provide location of object within 30cm radius	Provide location of object within 10cm radius

Test Scheme:

Reliability

Must(s)	Should(s)	Could(s)
20% success rate	50% success rate	70% success rate

Risk to the project if not complete:

5

3) Navigating to object

Requirement:

The system must be able to position itself such that the end effector can be guided to a desired object

Rationale:

To pick up the desired object needs to be within the arms operating range

Objective:

Must(s)	Should(s)	Could(s)
Position the gripper within operating range with 10% success rate	Position the gripper within operating range with 50% success rate	Position the gripper within operating range with 90% success rate

Test Scheme:

Reliability

Must(s)	Should(s)	Could(s)
Have gripper within range of object after 10 attempts; 10% success rate	Have gripper in range after two attempts 50% success rate	Have gripper in range first attempt

Risk to the project if not complete:

5

4) Gripper positioning

Requirement:

The system must be able to guide a grasping mechanism to a desired object

Rationale:

To lift the object the gripper must be positioned within gripping range

Objective:

Must(s)	Should(s)	Could(s)
within 8cm radius	within 5cm radius	within 2cm radius

Test Scheme:

Must(s)	Should(s)	Could(s)
30% success rate	50%	70% success rate

Risk to the project if not complete:

3

5) Kinematic model of arm

Requirements text:

The system must have a mechanism capable of grasping a variety of objects.

Rationale:

To transport the object the system must be able to pick it up.

Objective:

Must(s)	Should(s)	Could(s)
Minimum of two objects from a single orientation	Minimum of two objects from multiple orientations	More than two objects from multiple orientations

Test Scheme:

Measure success rate of grasping an object with enough grip to lift the object.

Must(s)	Should(s)	Could(s)
30% success rate from a single orientation	50% success rate from two orientations	70% success rate from four orientations

Risk to the project if not complete:

5

6) Location storage

Requirements:

The system must be capable storing the location of identified objects

Rationale:

For the system to be able to fetch an object, the system must be able to store a general location of the object

Objective:

Must(s)	Should(s)	Could(s)
Store coordinate of platform when object is identified	Store xy coordinate of object relative to base and xy coordinate of platform when object is identified	Store xy coordinate of object relative to map

Test Scheme:

Does the system successfully store locations?

Must(s)	Should(s)	Could(s)
Succeed	N/A	N/A

Risk to the project if not complete:

2

7) User interface

Requirements:

The system must have a user interface operable by untrained personnel.

Rationale:

Objective:

Must(s)	Should(s)	Could(s)

Some form of user interface	Intuitive GUI	Voice control

Test Scheme:

Can the user readily see;

Must(s)	Should(s)	Could(s)
Untrained user can operate with strict instruction	Untrained user can operate with limited instruction	Untrained user can operate with zero instruction

Risk to the project if not complete:

1

Systems overview

Hardware

The system will comprise of the following key hardware systems:

Base

Most of the work for the self-localising base was completed as part of Toms 3rd year project. The key components of the base are:

- Chassis
- Motors/driver
- LIDAR
- Laptop
- Batteries (power supply)
- Encoders

The power system of the base will be upgraded for the AMM. A 22v to 32v input, 12v output, 8A-10A switch-mode power supply (SMPS) will be purchased to power the arm and any other systems which require mains voltage. All mains voltage equipment we are using runs off a 12v wall plug, so we will discard these plugs and power the systems with the SMPS.

For more information on the base of the robot (including functionality and work done), please look here:

https://github.com/carebare47/dalek/blob/master/Autonomous%20Dalek%20Project%20Report.docx

Arm

The arm we will be using was designed by Martin Stoelen. We will be borrowing it for the duration of the project. The arm uses co-contracting agonist-antagonist pairs of motors to vary the stiffness over several key joints allowing the compliance over these joints to be set.

Gripper

The gripper will be primarily build from 3D printed parts and eight servos. The gripper will have four ‘fingers’, the design of which is inspired by the fin-ray effect gripper. A paper describing the original fin-ray gripper can be found here:
https://www.frontiersin.org/articles/10.3389/frobt.2016.00070/full

Cameras

The robot will use at least two cameras. The first (hand camera), mounted in the gripper, will be used for precise grasping of the object. The second camera (head camera) will be mounted on the highest point of the robot and will serve to identify and store the locations of objects.

Base sensors

The base of the robot is equipped with two incremental wheel encoders and a 2D LIDAR. The encoders are used to limit the velocity of the wheels to prevent the robot moving faster than the SLAM system can keep track of.

Software

Navigation

Most of the navigation system has already been developed. More information on this can be found in the git repository linked above.

Arm

The arm has a kinematic model/solver, but does not come with high-level control algorithms. A node will be written to generate end-effector goals for the arm from both camera systems.

Gripper

A control program will be developed to provide an abstracted interface to control of the grippers servos. Control inputs to this program will include the triggering of sequences of poses and control over individual servo angles. This program will also generate control signals for the LED ring array around the hand camera.

Gripper camera

This program will process fiducials seen by the hand camera and generate signals instructing the arm to align the gripper with the object to be grasped. The program will also trigger the gripper closing when it in position.

Head camera

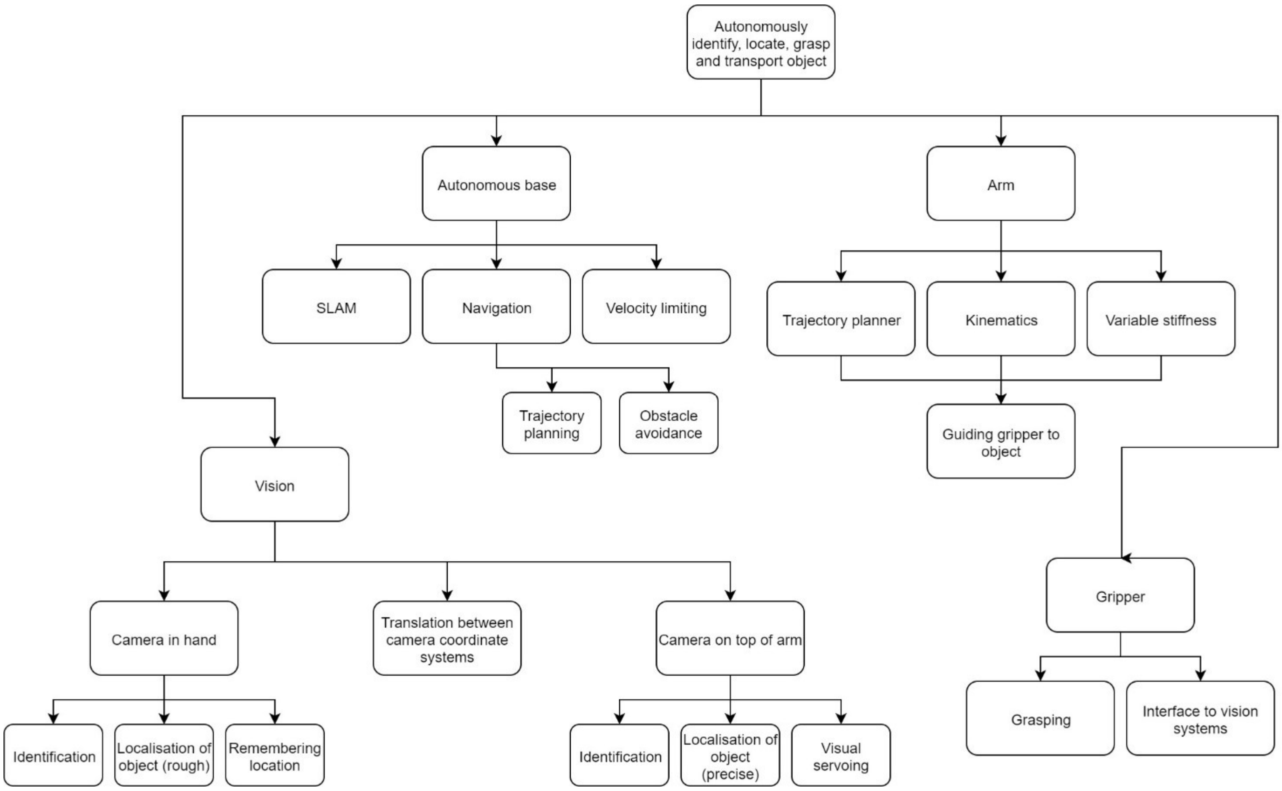
The software for the head camera will detect objects as the robot drives around and send their locations to the waypoint node to be saved. In addition, the head camera program will be used to point the hand camera at the object we wish to pick up.

Waypoint node

The waypoint node will serve as the universal goal coordinate saving and loading system. The robot base can currently save its current location as a waypoint. If the robot moves somewhere else and this waypoint is loaded as a goal for the navigation system, the robot will return to where it was when the waypoint was saved. This node will have the following inputs and outputs:
Node inputs:

- Should it save a waypoint?
- Should it be where I am or somewhere else?
- With respect to which coordinate frame should it save the waypoint?
- What should it label the waypoint?
- Should it load a waypoint?
- Which waypoint should it load?

The output of this node will be goals published for the navigation system to seek out.



Design process

- Problem definition
- Conceptualisation
- Research
- Preliminary design
- Development work
- Final design
- Prototype
- Testing

Problem definition:

Can a compliant robot arm be integrated into a self-navigating robot platform and programmed to autonomously identify, locate, grasp and transport objects using data from cascaded camera systems?

Conceptualisation:

This is the phase of our design process where we discussed potential solutions to producing a mobile manipulator.

Preliminary design:

The concepts generated in conceptualisation phase of the design process were evolved into subsystems. These subsystems are discussed in the Systems Overview.

Research:

Initially our research is going to focus on the exploration of camera-based object recognition and localisation using cascaded camera systems, as well as comparison to existing mobile manipulators.

Development work:

There are currently un-solved subsystems in our preliminary design relating to the cascaded camera systems. Full description of this issue can be found in XXXXXX. The initial primary focus of development work is a performance evaluation of several potential solutions to this problem.

Final design:

Once development work has been complete and subsystems have been tested we can specify our final design and begin integrating subsystems into our prototype.

Prototype:

Will integrate all subsystems.

Testing:

Please see both the testing section and the requirements section for testing criteria. The prototype will be tested against the requirements. If the prototype does not hit these requirements then the design process will return to the development work phase where other solutions will be discussed amongst our team.

Testing

Vision systems

The vision systems in the head and will be tested independently; with the head camera being tested at distances of 1-8m and the in-hand camera being tested between 5-50cm.

Object recognition reliability will be tested by showing the camera an object in a variety of orientations. True positive, false positive and false negative readings will be recorded to determine the reliability of the system. Object localisation will be tested by placing objects at various known distances to the camera, the systems position estimates will be compared to positions measured by us to determine the accuracy of the system. These tests will be repeated several times to get a measure of reliability for the system.

Gripper

The grasping capability of the gripper will be tested without the gripper attached to the arm. The gripper will be closed around several objects of different shapes, sizes and weights with both the objects and the gripper in different orientations. Tests will be conducted multiple times to get a measure of reliability for the gripper, reliability across test cases will form a measure of robustness for the gripper

Goal navigation

The navigation of the autonomous platform will be tested by giving it a goal coordinate and orientation and measuring the accuracy of its final position compared to the goal position. The system will then be given a goal of its original position and a second measure of accuracy will be taken, this will be repeated with and without reset to give a measure of the systems reliability both of a single fetch request and many fetch requests.

Object manipulation

Object manipulation relies on the vision system, gripper and arm all working together. This will be tested with the autonomous base disable. The system will lift an object and place it at a goal location. A measure of accuracy will be taken by measuring the error between the goal position and the final position of the object. A successful transport will be given if the objects final position is below a set error margin. Reliability will be measured as the percentage of successful transportations

Full system test

To test the entire system, the robot will be sent to a location where a desired object is in view. The system must first identify and locate the object using the head cam without human interaction, then position the platform such that the object is visible and reachable by the gripper, re-identify and locate the object with the hand cam, move the gripper close to the object, ensure the object is within grasping range with the in-hand camera and correct if necessary, lift the object and return the object to the platforms starting position. The test will be considered successful if the object successfully grasped and taken to a given location within an allowed error margin. The percentage of successful transportations will form a measure for the reliability of the system.

Development issue with cascaded camera systems

Research, development and testing must be done on the cascaded vision systems as they have conflicting requirements and a solution which serves both systems in tandem has not yet been found. We are still uncertain of the methods we will use for identifying and localising objects with each camera system. The issue arises from the interaction between the two camera systems and is outlined below.

Head camera:

Our options for the head camera are YOLO or fiducials. YOLO is a neural-network based real-time object detection and localisation system. It identifies and localises objects in a camera feed or still picture. If we use YOLO for the head camera we will modify the program found here (https://github.com/pjreddie/darknet/wiki/YOLO:-Real-Time-Object-Detection) to return the CofM of the identified objects. This CofM will then be located in the RGBD image stream from the Kinect, where the depth value of the identified object will be calculated and thus the location of the object will be known. Development has not yet started on this solution and only preliminary research has been done to assess the feasibility of making this system work.

The other option for the head camera is fiducials. If we use this approach, QR codes will be attached to the object we wish to recognise. If a fiducial is seen in a camera feed, the location of the fiducial can be calculated.

Hand camera:

Fiducials are a better option for the hand camera as YOLO must see the entire object to identify it. When the gripper closes in on an object, the objects apparent size will increase until the edges of the object are no longer visible. When this happens, it becomes impossible to calculate the CofM of the object and thus the gripper cannot be properly aligned to grasp the object.

A small QR code on the object could be used to guide the hand camera. The extent to which placing a QR code for the hand camera on the object will obscure the YOLO identification is currently unknown.

The hand camera requires the fiducial to be small to be detectable at close range, whereas the head camera requires the fiducial to be large in order to identify the object at long range. The gripper will have a maximum size of object that it can pick up. It is possible that a fiducial covering an entire face of the largest object that the gripper can grasp may not be large enough to detect at a range of more than 1.5 meters.

Another option to address both systems is fractal fiducials[1] as can be seen in figure 1. This would allow the head camera to see the object far away whilst providing the hand camera with smaller fiducials to track. The data in each sub-fiducial can contain the fiducials coordinate and level within the larger fiducial. Again, only preliminary research has been done into the feasibility of this approach.

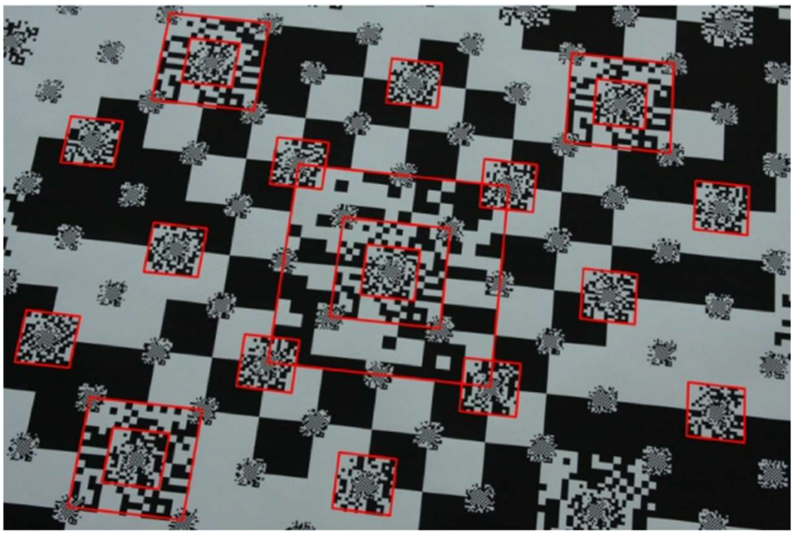


Figure 1 – Fractal Fiducial

Due to the issues outlined above, a high priority in the early stages of our project development is to find out how well we can identify fiducials at a distance. Data on camera resolution, fiducial size and distance to fiducial will be collected for both camera systems and a graph will be generated for both regular and fractal fiducials. From this we will be able to see which approach (or combination of approaches) will be most suitable. The two main outcomes of this experiment are as follows:

If we discover that we can easily detect fiducials at reasonable ranges:
Fractal QR codes and YOLO localisation in 3D will be investigated further to find the relative reliability and feasibility of each approach with emphasis on reducing required development time and finding a robust solution.

If we discover that we cannot easily detect fiducials at reasonable ranges:
If the graph shows that the size of fiducial required for the headcam to detect is larger than the largest object the gripper can pick up, then we will consider higher resolution head cameras. If, with high resolution cameras, it is still not possible to detect objects at a reasonable range, then the scope of the project will have to be quickly re-evaluated.

Project management

Project management philosophy

The largest limiting resource in our project is the amount of time we have available to complete it, and so efficiently utilising the time we have was the biggest focus in creating and refining out project plan and gantt chart. This lead to the decision to take advantage of the critical path method (CPM). To generate the task list for the CPM, a work breakdown structure was produced.

A model of the project was constructed comprising of the following elements:

- A task list of all activities required to complete the project
- The duration of each task
- Dependencies between each task (e.g. the gripper cannot be assembled before it has been 3D printed)
- A project end date and milestones where appropriate

The critical path method of project planning calculates the longest path of planned activities between the start and end of the project and thus determines the shortest possible time to project completion. In figure 2 you can see the critical path of our project as shaded rectangles in the gantt chart. A higher resolution image is available in the 'Autonomous Mobile Manipulator gantt chart.pdf' file included with the submission of this document.

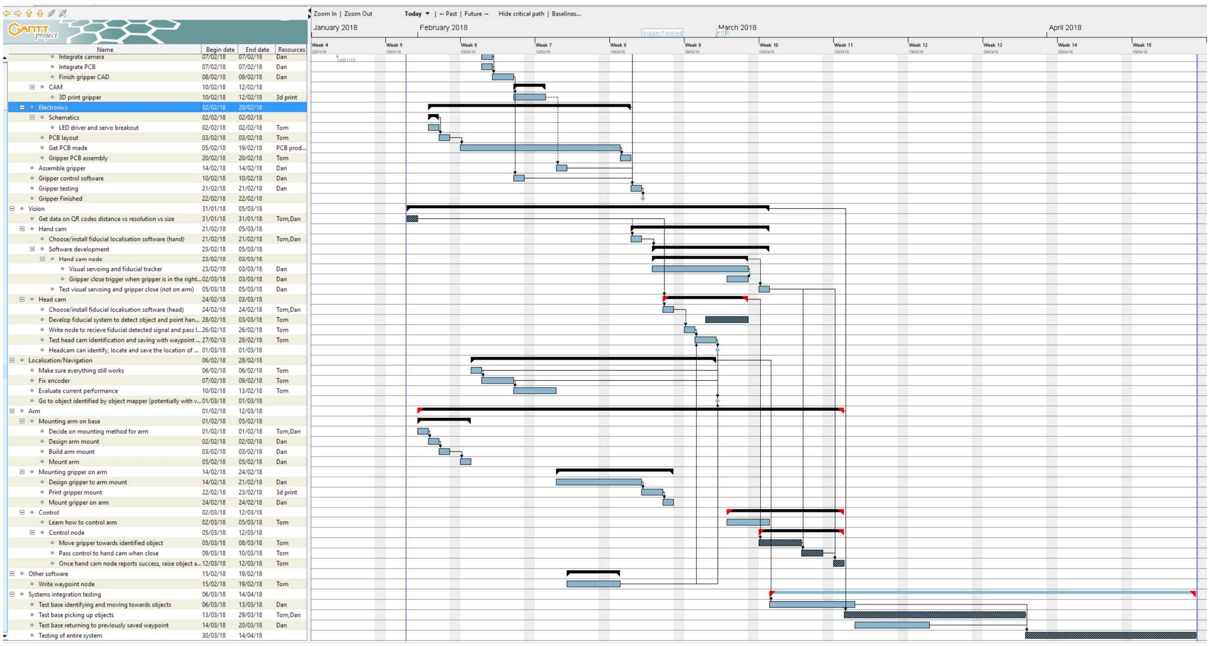


Figure 2

However, we cannot yet implement a fixed CPM methodology due to uncertainty regarding a particular set of subsystems in the robot as mentioned in the 'Development issue with cascaded camera systems' section of this document. For this reason, we have decided to bring elements of Adaptive Project Management (APM) methodologies into our management philosophy.

Traditional project management is based on planning well and planning once, at the start of the project. Managerial roles are rigid and focused on the initial plan, and project control is focused around identifying deviations from this plan and getting things back on track. This planning process can take months to fully design and specify a system before development work has been initiated. The APM methodology differs from traditional project management in that planning is expected to be iterated through the project lifecycle with a more flexible and adaptive managerial approach and plans adjusted if deemed necessary. We have decided to take elements from both approaches as we have not managed to lock-down a final solution and strict time management is essential for the completion of this project.

Currently there is a key re-plan scheduled once development work has identified the solution to our cascaded vision system problem. The current critical path is running under the assumption that fractal fiducials are a feasible solution. This is not yet tested. If fractal fiducials prove to be an unsuitable solution, a re-plan will be initiated and the critical path will be re-evaluated. If further technical hang-ups are encountered (or new approaches discovered), and potential solutions outweigh the time cost of a re-plan, plans may be adapted to account for yet-to-be-identified solutions and the critical path will be evaluated again.

Team organisation and management

The team for this project comprises of Tom Queen and Daniel Gregory-Turner. Tasks and roles have been assigned to our respective strengths where possible. Managerial roles have been assigned to each of us to aid in the management and organisation of the projects. However, all non-trivial elements of the project will be conceptualised by the both of us. Research and development work will often be shared outside of our respective roles. Problem-solving will be initiated by the person who encounters the problem, although the other team member will be called in for a consultation if the problem appears to be non-trivial or is expected to take longer than the time allocated to that task.

Attempts will be made to resolve any disputes via in-depth discussion. Each person will make efforts to completely understand the position of the other. If, after in-depth discussion, a solution to the dispute cannot be found and the dispute relates to how various parts of the system will be approached, developed or structured, or how any technical problems/setbacks will be resolved, then the final say will fall to the manager of the system in question.
Below is a list of roles that have been assigned for this project:

- Project manager
- Electrical manager
- Mechanical manager
- Software manager
- System architect
- Financial manager

Electrical manager

The electrical manager oversees all project elements related to electrical/electronic aspects of our project. Responsibilities of the electrical manager include:

- Regarding electrical systems to be produced by us:
    - Research
    - Specification

    - Design
    - Development (both physically and in simulation)
    - PCB art
    - Utilising (and communication with) the universities in-house PCB production team
    - Selection / acquisition of components
    - Constructing and testing
    - Interfacing with other systems
- Regarding electrical systems to be acquired by us:
    - Specifying product to be purchased
    - Finding a suitable product
    - Buying product
    - Fitting, testing and interfacing the product where appropriate

Mechanical manager

The mechanical manager oversees all project elements relating to mechanical aspects of our project.

Responsibilities of the mechanical manager include:

- Regarding the mechanical systems to be produced by us:
    - Research
    - Specification
    - Design
    - Simulation
    - CAD
    - Contacting and utilising 3rd party contractors or university fabrication facilities
    - 3D printing
    - Construction and testing
- Regarding mechanical systems to be acquired by us:
    - Specifying system/product/component to be acquired
    - Finding suitable system/product/component
    - Purchasing
    - Fitting, testing and interfacing where appropriate

Systems architect

The systems architecture manager oversees all project elements relating to the design and interaction of software subsystems. Responsibilities of this role include:

- High-level design of system architecture
- Identifying and specifying requirements of software subsystems
- Structuring software subsystems
- Specifying requirements of subsystem interfaces
- Testing the interaction between subsystems

Software manager

The software manager will be responsible for the inner workings of all software to be developed by us. Responsibilities of the software manager include:

- Development strategy of software
- Identifying and locating software dependencies
- Software testing
- Architecture of each software subsystem
- Programming
- Organisation of code
- Ensuring smooth interfacing between software modules
- Ensuring the subsystem interface meets specified requirements

Financial manager

The financial manager will be responsible for managing the project budget. Responsibilities include:

- Keeping track of all expenditures on the project budget
- Keeping track of personal expenditures relating to the project but which are not on the project budget
- Approving purchases/use of project budget

Assignment of roles

	Project manager	Electrical manager	Mechanical manager	Software manager	Systems architect	Financial manager
Tom	X	X		X		
Daniel	X		X		X	X

Risk management

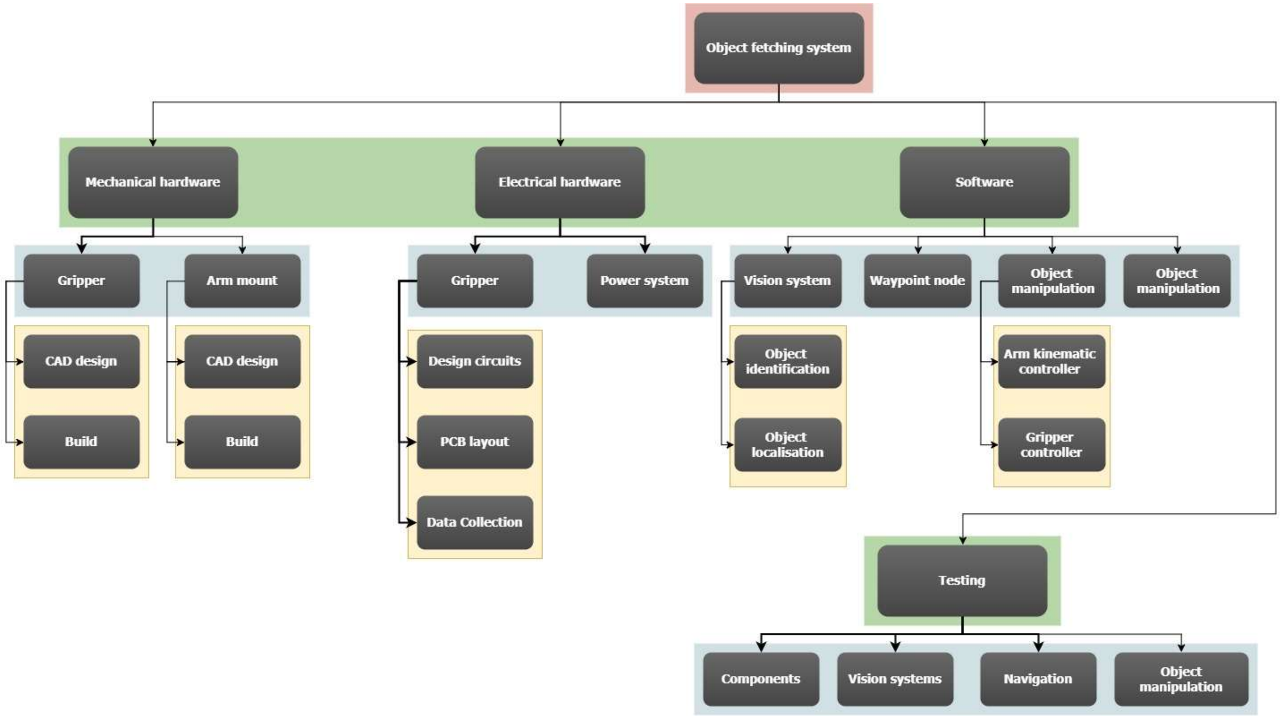
A better view of our risk management plan can be found in the accompanying document Risk_management.ods, but for completeness the contents of this spreadsheet have been included in a table below.

Category	System	Description of Risk	P (1-10)	I (1-10)	Priority (P * I)	Triggers	Response Plan
Technical	Gripper	Servo breaks without prompting	2	5	10		
		Gripper self-destructs breaking servo	3	5	15		
		Gripper self-destructs breaking plastics	3	2	6		
	Kinect				0		
		Power supply voltage issues	2	7	14		
		Runs too slowly on second PC	4	3	12		
		Issues translating kinect pixel numbers to coordinate referenced to any robot	3	8	24		
	Gripper vision				0		
		Fractal QR codes don't work	5	8	40		
		Camera too close, fractal QR code doesn't work	6	8	48		
		Gripper/kinect vision uses up too many computational resources causing issues with kinematic calculations resulting in:					
		Non-smooth trajectories	5	2	10		
		Jumping	4	3	12		
		Collisions with other objects	3	6	18		
		Collisions with self	3	6	18		
		Gripper can't deform to shape of object properly, Can't pick object up	3	6	18		
	Localisation						
		Robot crashes into walls/things resulting in:	6		0		
		Breaks nothing	8	1	8		
		Breaks arm	3	6	18		
		Breaks encoder	3	6	18		
		Robot can't get close enough to object (more than once or twice)	3	7	21		
	Object mapping				0		
		Robot can't see far away objects (1.5m+)	3	7	21		

		Robot falsely identifies objects	2	7	14		
					0		
	Arm mounting				0		
		Arm falls off leading to damage of arm	3	7	21		
		Arm overbalances base	2	9	18		
	Systems integration				0		
		Must and can't stop/pause SLAM without remapping room	4	9	36		
		Object falls out of gripper often	3	7	21		
		Invacare remote finally dies	3	7	21		reserve budget to buy new controller
					0		
Planning		Finished subsystems performance criteria not strict enough to allow for autonomous interaction between subsystems	3	7	21	Unacceptable performance of interaction between systems	Carefully re-evaluate performance criteria of subsystems, potentially suggest alternative subsystems
Human					0		
		Lab is frequently full of people during testing/development, scrambles SLAM	4	8	32	More than two days spent at low productivity due to people walking nearby	
		Getting caught in the details of a subsystem, spending too long trying to fix unexpected issues in a subsystem	7	7	49	Subsystem developent time exceeded	Group meeting, analyse alternative subsystems/different ways of achieving the same thing
		Tom/Dan falls ill for:			0		
		1 week	2	7	14		
		2-4 weeks	2	8	16		
		4-8 weeks	1	9	9		
		8 weeks +	1	10	10		

		CAD could take longer than planned	3	3	9	CAD development time exceeded	Remaining workload redistributed between us
Human / time		Systems integration takes longer than planned (applicable for interaction between every system & subsystem)	7	7	49	Systems integration time exceeded (or looks like it will be)	Reduce scope of project, re-evaluate priority list, divert attention from lower priority tasks, look for potential off-the shelf solutions (unlikely to find any)
		We lose access to the arm from Martin	1	10	10		
Resources		Printers unavailable / break	2	2	4		
Logistics		Long lead times	8	4	32		Specify parts to order early on in project
		Can't return servos if faulty / break	6	6	36		Order more servos if our surplus falls below two
		Can't get PCB made	1	8	8		Point to point wiring
Other		We lose the laptop	2	9	18		
		We lose the arm	2	9	18		

Work Breakdown Structure



Gantt chart

The gantt chart contains a time plan relating each task to the duration it will take with start and end dates for all tasks, as well as the critical project path and resource assignment between team members and facilities. The full gantt chart can be found in the accompanying 'Autonomous Mobile Manipulator gantt chart.pdf' document.

References

[1]A. Herout, M. Zachariáš, M. Dubská and J. Havel, "Fractal marker fields: No more scale limitations for fiduciary markers," *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Atlanta, GA, 2012, pp. 285-286.
doi: 10.1109/ISMAR.2012.6402576
keywords: {augmented reality;cameras;fractals;image motion analysis;FMF;QR-code;camera motion;camera-based augmented reality;data matrix;embedding marker;fiduciary marker;fractal marker field;fractal structure;marker constant density;marker field surface;matrix code;Augmented reality;Calibration;Cameras;Fractals;Image color analysis;Manganese;Reliability},
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6402576&isnumber=6402509