

Contents

Systems overview:	2
Hardware	2
Arm:	2
Gripper:.....	2
Cameras:	2
Base sensors:	2
Software	3
Navigation:.....	3
Arm:	3
Gripper:.....	3
Gripper camera:.....	3
Head camera:.....	3
Waypoint node:	3
Design process:	6
Requirements:	9
1) Object recognition	9
2) General object location	9
3) Navigating to object	10
4) Gripper positioning	10
5) Kinematic model of arm	11
6) Location storage	11
7) User interface	12

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

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 is able to keep track of.

Software

Navigation:

The majority 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 is 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.

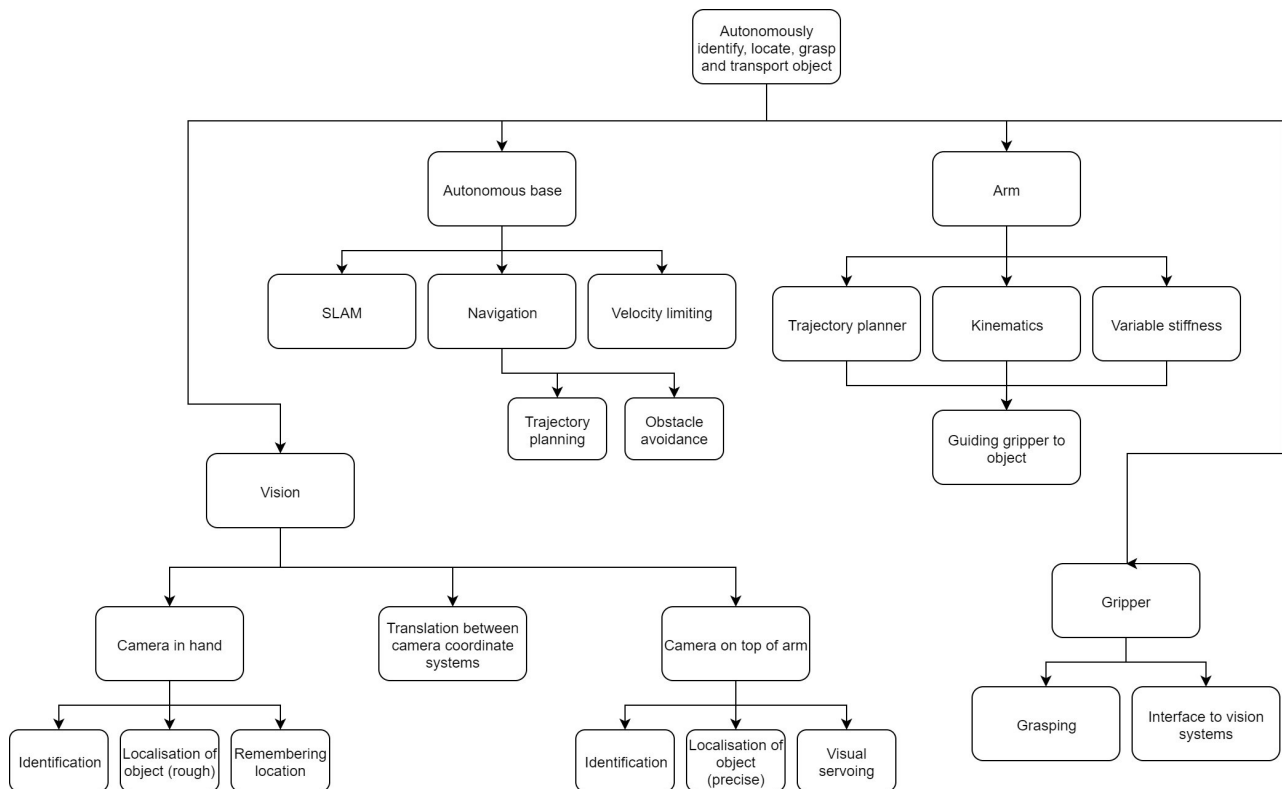
Waypoint node:

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 head towards.



Research, development and testing has to be done on the cascaded vision systems as they have conflicting requirements.

We are still uncertain of the methods we will use for identifying and localising objects with each camera system.

Head camera:

Our options for the head camera are YOLO or fiducials. YOLO is a neural-network based realtime 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.

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.

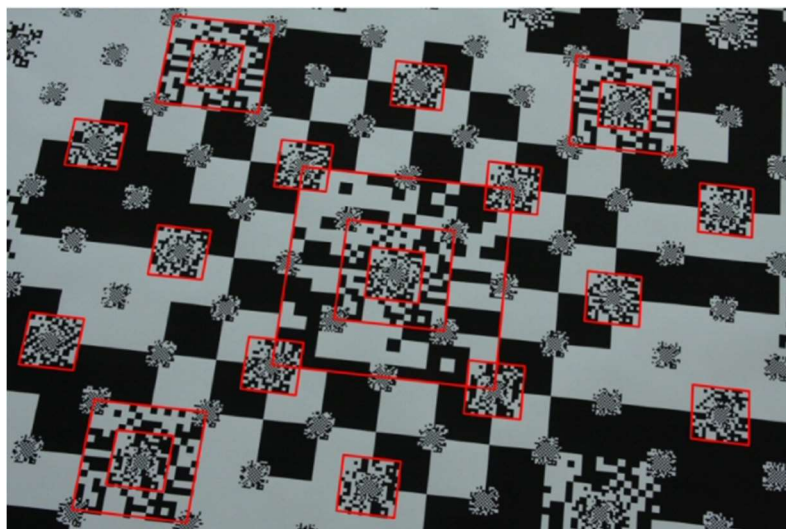
Hand camera:

Fiducials are a better option for the hand camera as YOLO must see the entire object to identify it. As 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 in order 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 as can be seen in image XXXX [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



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.

Bad res/distance:

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 look into 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 reevaluated.

Reasonable res/distance:

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.

[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>

Risk management:

Please see the Risk_management.ods spreadsheet for our risk register, analysis and reduction strategies.

Team organisation:

The team for this project comprises of Tom Queen and Daniel Gregory-Turner. Tasks have been assigned to our respective strengths where possible. By default, electronics work falls to Tom and CAD/mechanical work falls to Dan. We both have experience with ROS (Robot Operating System).

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 phase 2 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 to be undertaken is a performance evaluation of several potential solutions.

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 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 test 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 in order to determine the

accuracy of the system. These tests will be repeated several times to get a measure of repeatability 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 a number of 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.

Requirements:

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 a distance of 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:

In order 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:

In order 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 xy 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

Work Breakdown Structure

