KESSLER
Preliminary Design Review

ASEN 4018 Fall 2017

Abdiel Agramonte-Moreno, Glenda Alvarenga, Thanh Cong Bui, Christopher Choate, Lauren Darling, Sergey Derevyanko, Cassidy Hawthorne, Abigail Johnson, Nick Thurmes, Jannine Vela, Taylor Way

Agenda

Kinesthetic
Engineered
Solution to
Space
Litter &
Exhausted
Resources

- Project Description
  - Project Definition
  - Baseline Design
- Evidence of Feasibility
  - Critical Project Feasibility Elements
  - First Level Feasibility Analysis
    - CPE 1
    - CPE 2
    - CPE 3
- Status & Future Work

Project Description

Glenda Alvarenga (Project Manager) &
Jannine Vela (Systems Engineer)

Project Definition | Baseline Design | CPE Overview | CPE 1 Feasibility | CPE 2 Feasibility | CPE 3 Feasibility | Status & Future Work

Project Definition

Project Motivation

Amount of orbital debris is set to triple by 2030 (More than 500,000 in orbit today). Consists of:

- Pieces of satellite components
- Satellites at EOL
- Malfunctioning satellites

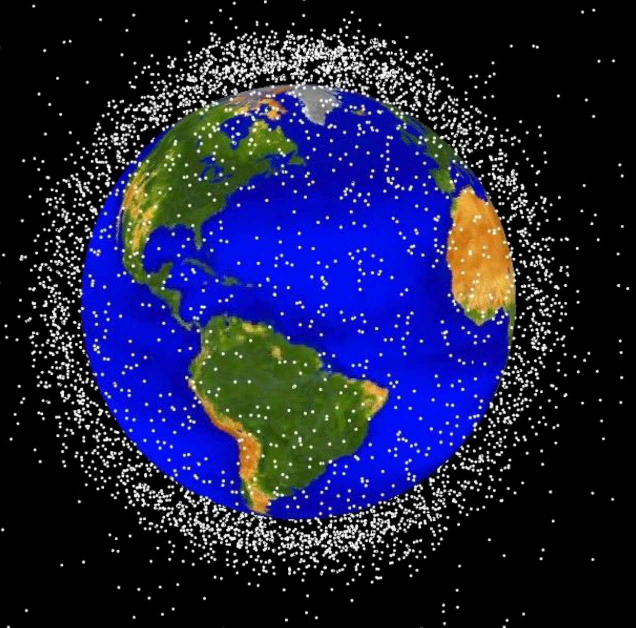


Fig. 1 Space Debris 2013 Model [1]

Sierra Nevada Corporation:

- Grappling feature recognition with an RGB sensor
- Autonomously capture feature with robotic manipulator arm

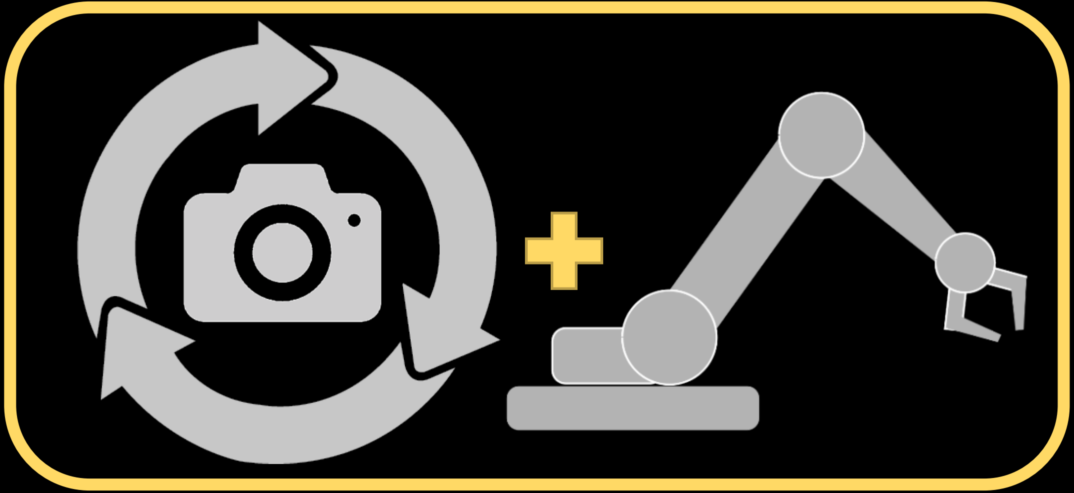
Fig. 2 SNC Developed OrbComm G2 Assets [2]

Project Definition | Baseline Design | CPE Overview | CPE 1 Feasibility | CPE 2 Feasibility | CPE 3 Feasibility | Status & Future Work

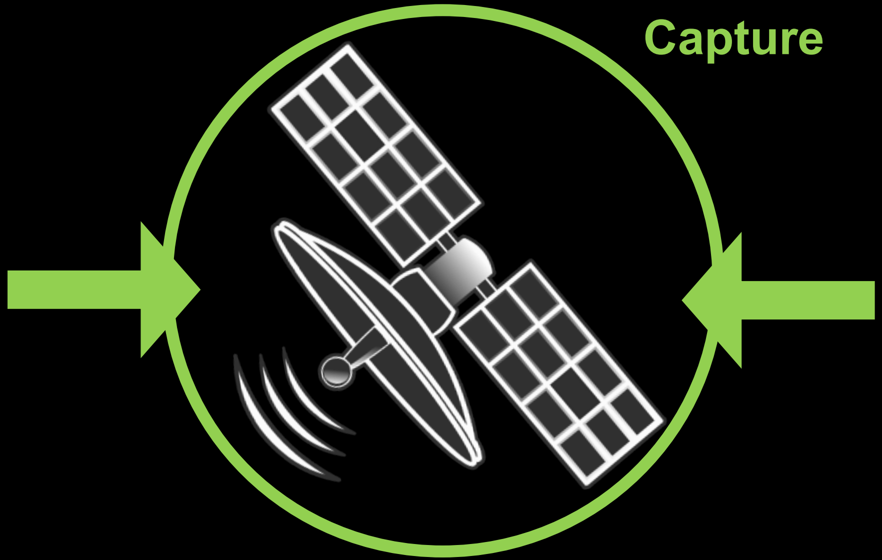
Project Definition

Project Statement

KESSLER



Satellite Capture



The KESSLER Project will design a system that utilizes visual processing and a robotic arm to autonomously capture space debris. This project will be developed using heritage hardware and software from the CASCADE capstone project.

Project Definition

- The simulated target satellite is modeled after the Iridium satellite series.
- Model will be 30% scale
- Features are:
  - Solar Panel Joints
  - Bus Structure Support
  - Antenna
- Features on Iridium are commonly found on other satellites as well.

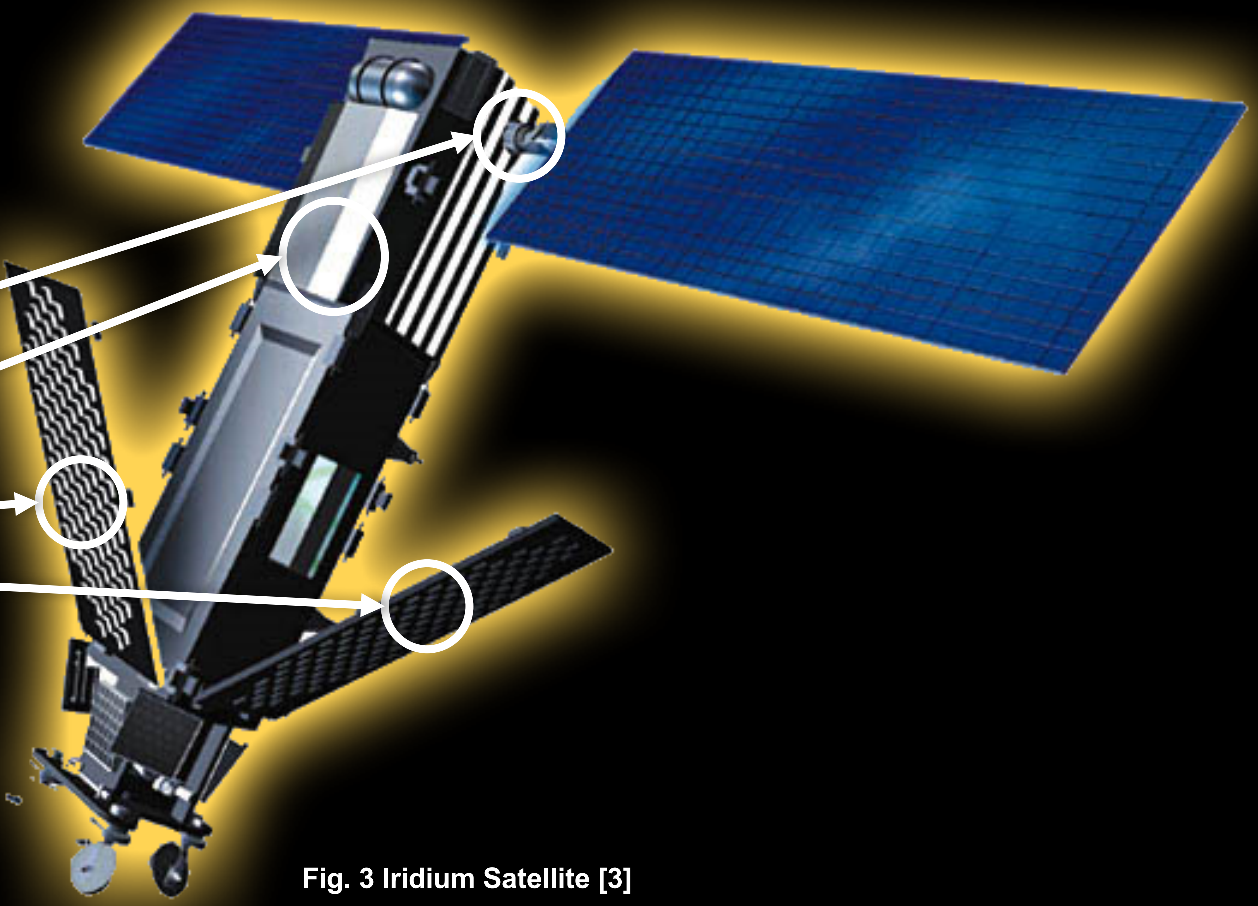
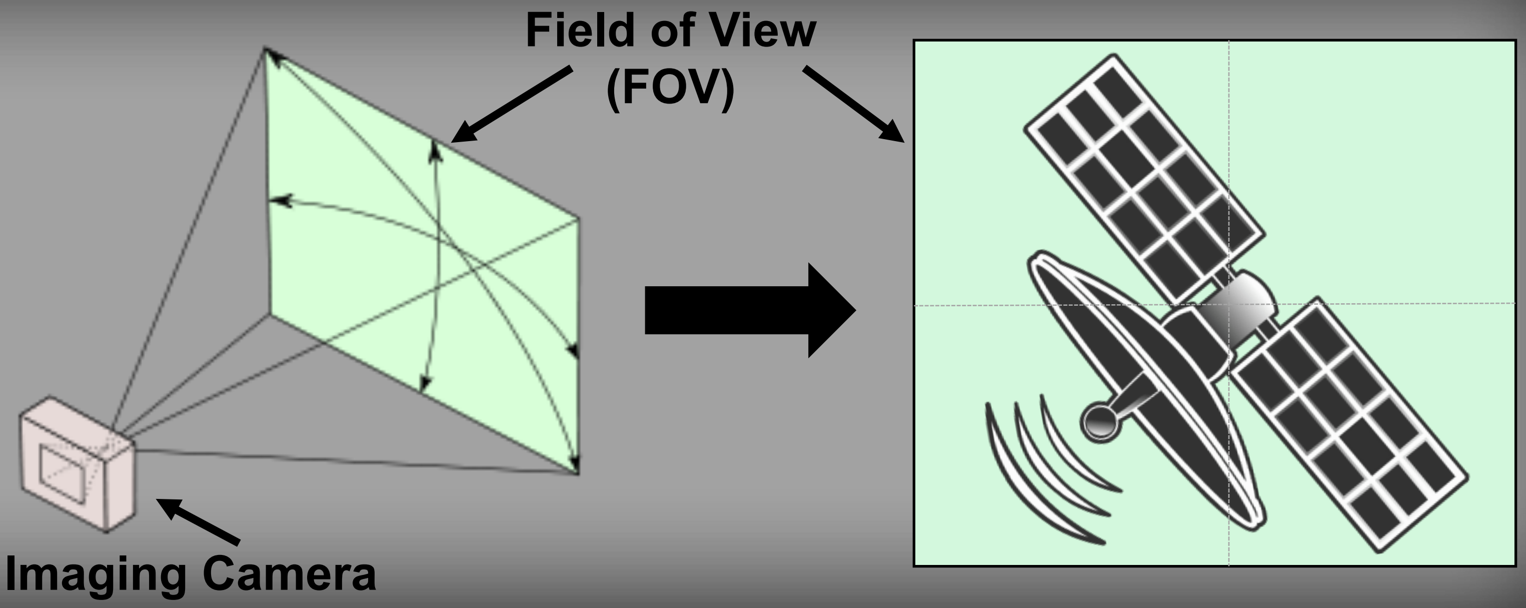


Fig. 3 Iridium Satellite [3]

Project Definition

Project Objectives

1. Take visual data confirming the target object is within FOV.



Project Definition | Baseline Design | CPE Overview | CPE 1 Feasibility | CPE 2 Feasibility | CPE 3 Feasibility | Status & Future Work

Project Definition

Project Objectives

2. Identify pre-defined grappling feature.

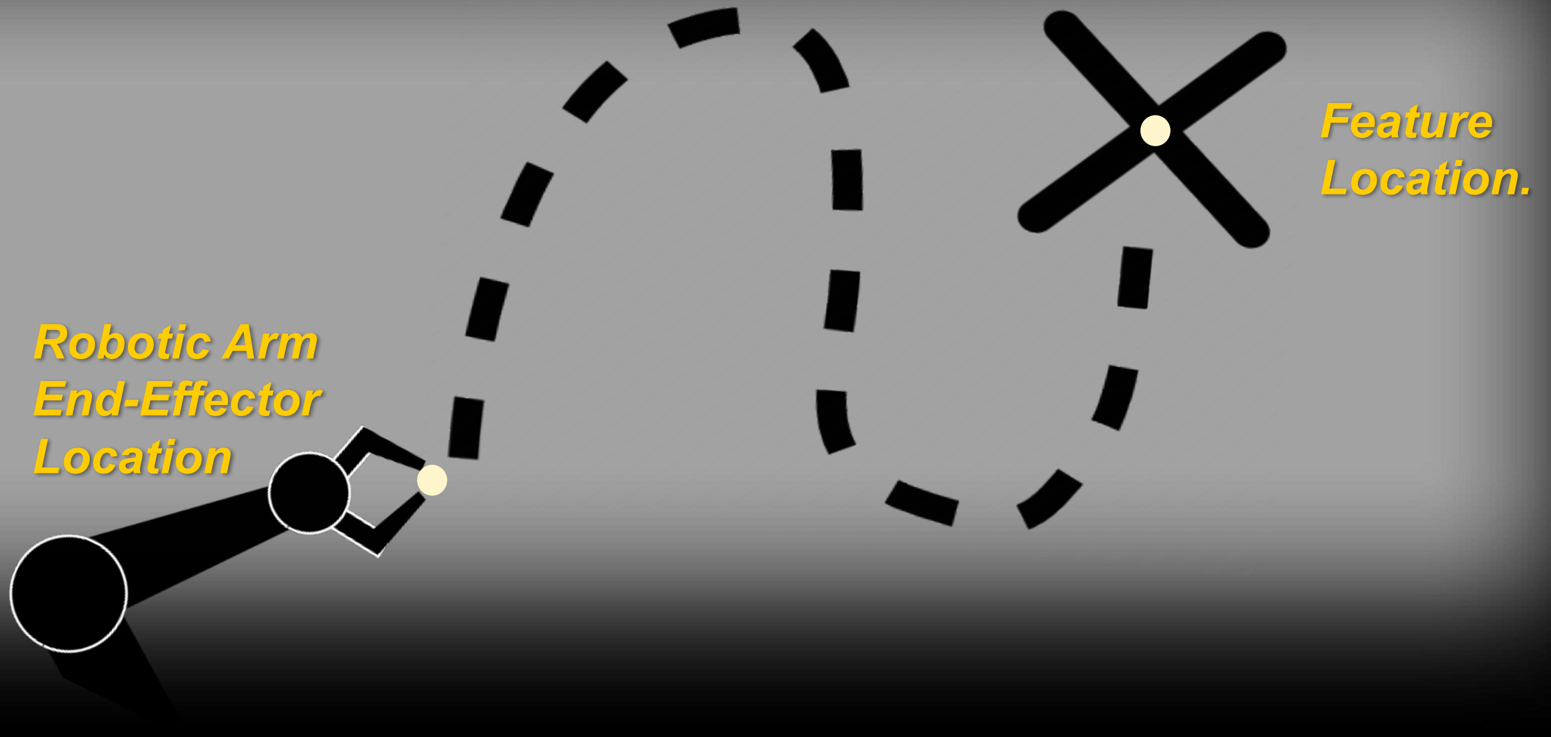
Is feature any of the following?

- **Antenna**
- **Solar Panel Joint**
- **Bus Support Structure**

Project Definition

Project Objectives

3. Determine prediction path to feature location.

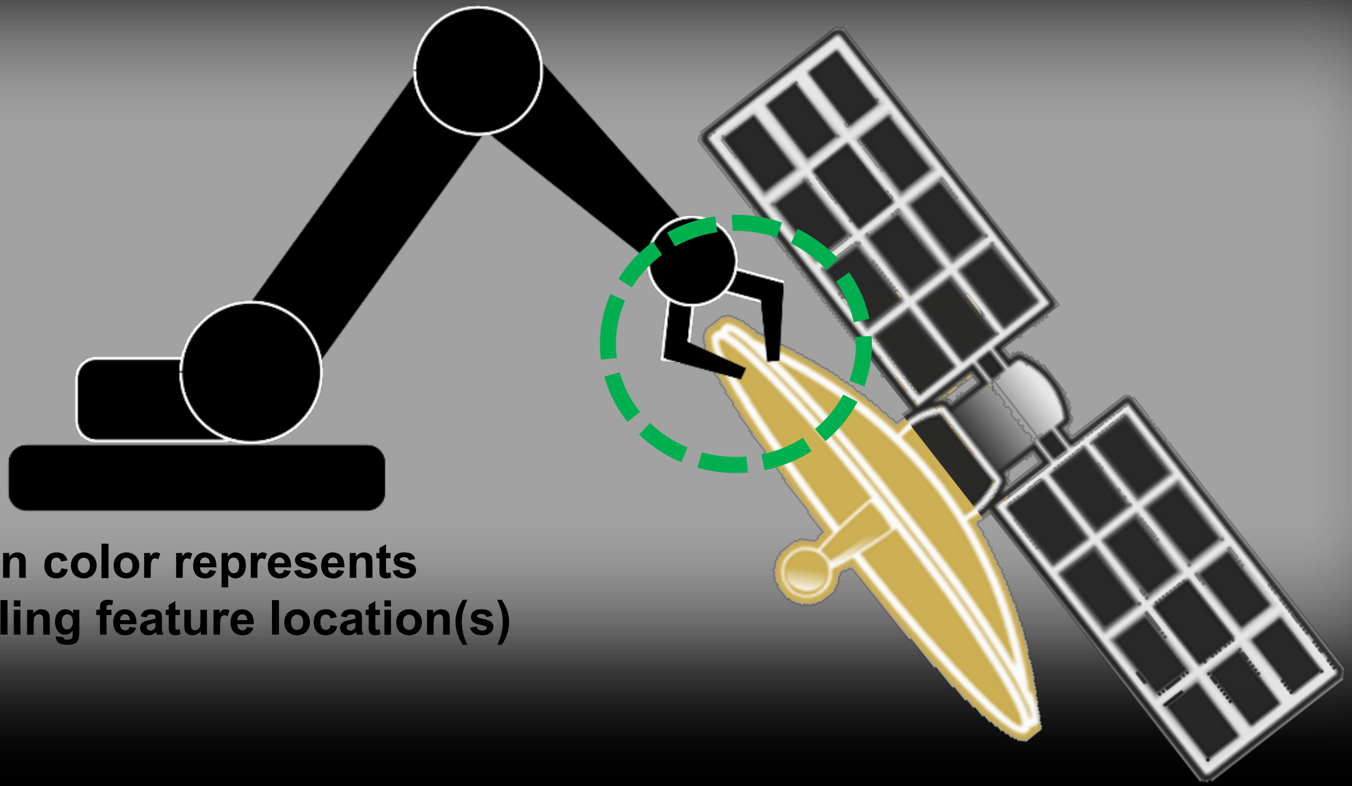


Project Definition | Baseline Design | CPE Overview | CPE 1 Feasibility | CPE 2 Feasibility | CPE 3 Feasibility | Status & Future Work

Project Definition

<u>*Project Objectives*</u>

4. Autonomously capture the feature via robotic arm

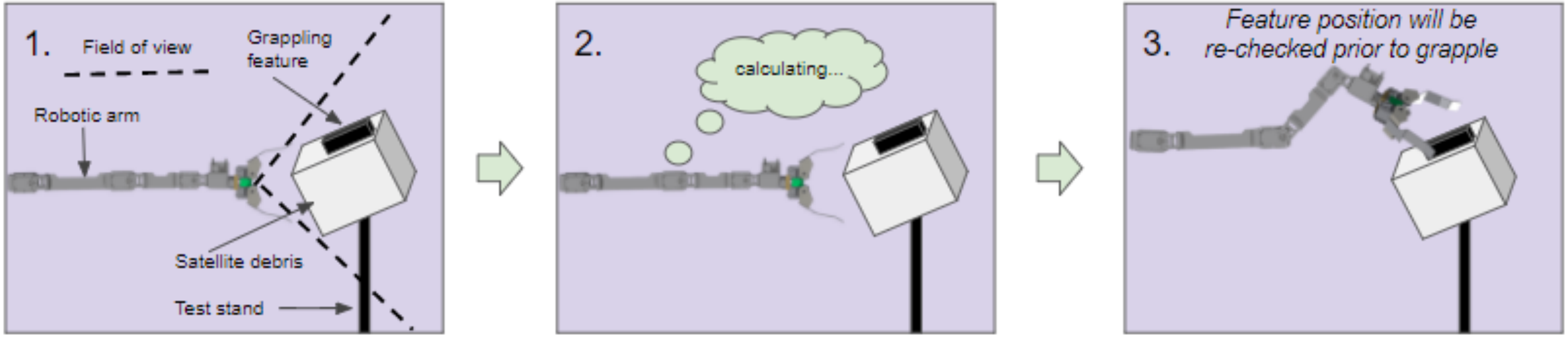


Golden color represents grappling feature location(s)

Test Facility

KESSLER Concept of Operations

Case 1

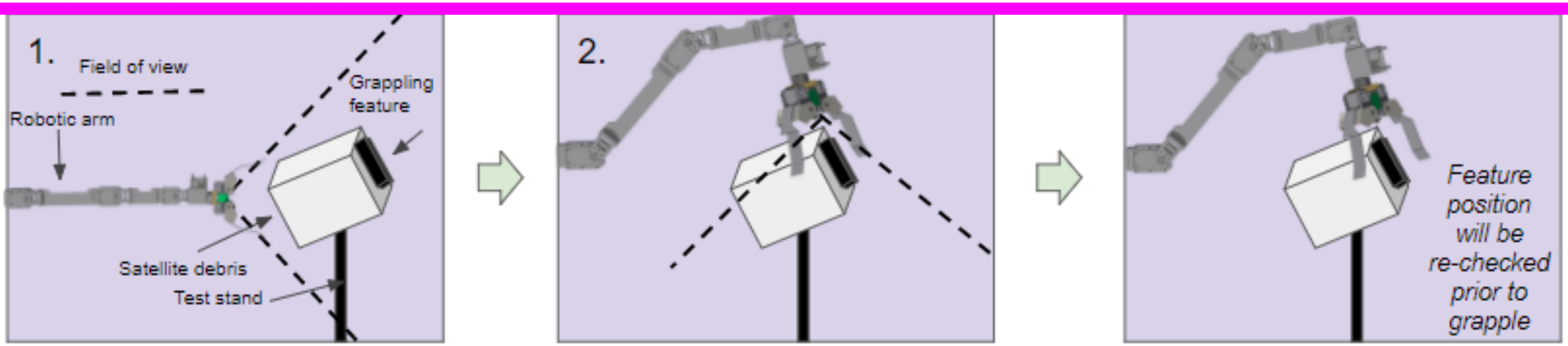


Robotic arm uses visual system to search for grappling feature on a satellite with unknown orientation.

The feature is found in FOV, arm calculates how to grab it.

Arm moves around satellite and rotates as necessary to grab feature.

Case 2



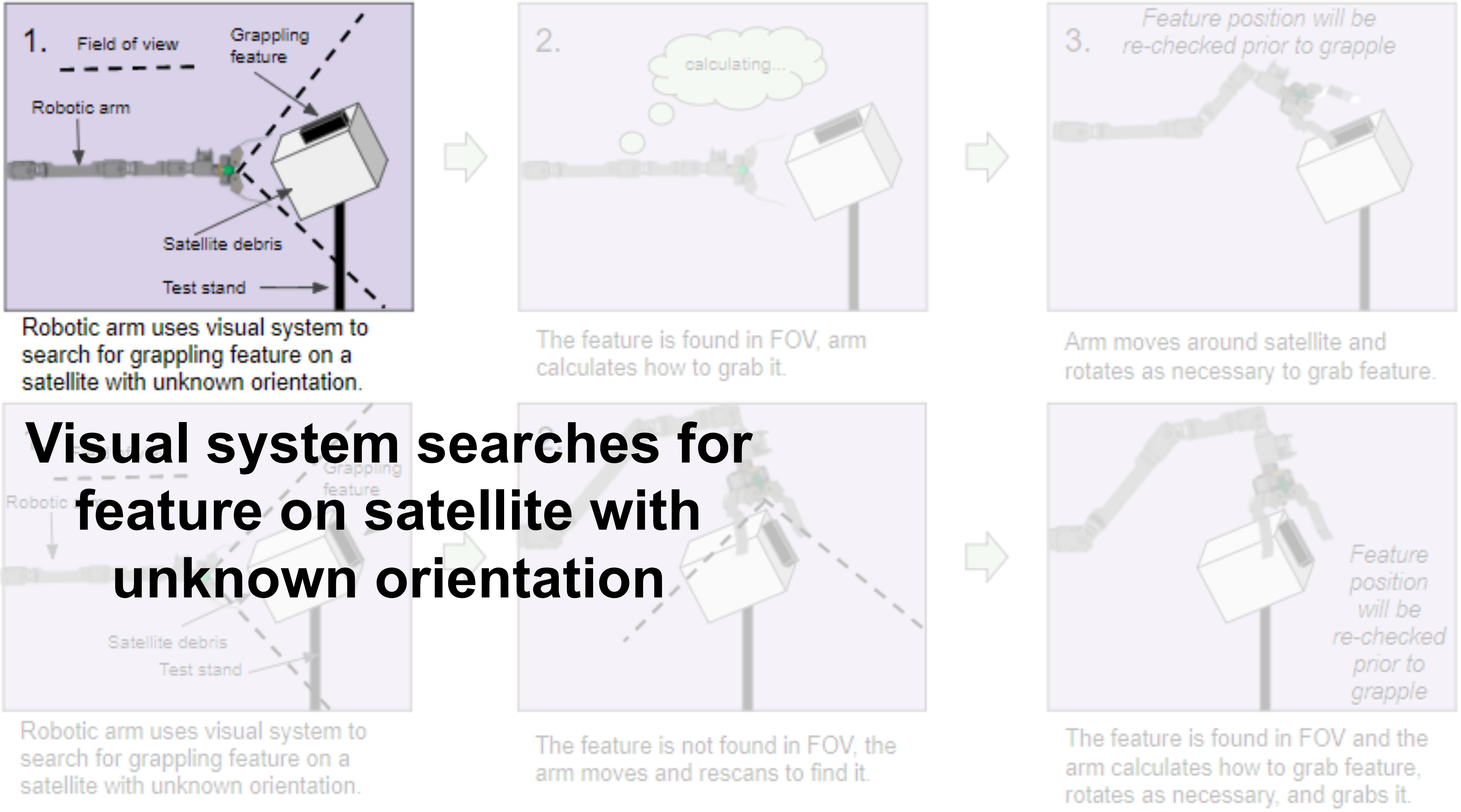
Robotic arm uses visual system to search for grappling feature on a satellite with unknown orientation.

The feature is not found in FOV, the arm moves and rescans to find it.

The feature is found in FOV and the arm calculates how to grab feature, rotates as necessary, and grabs it.

KESSLER Concept of Operations

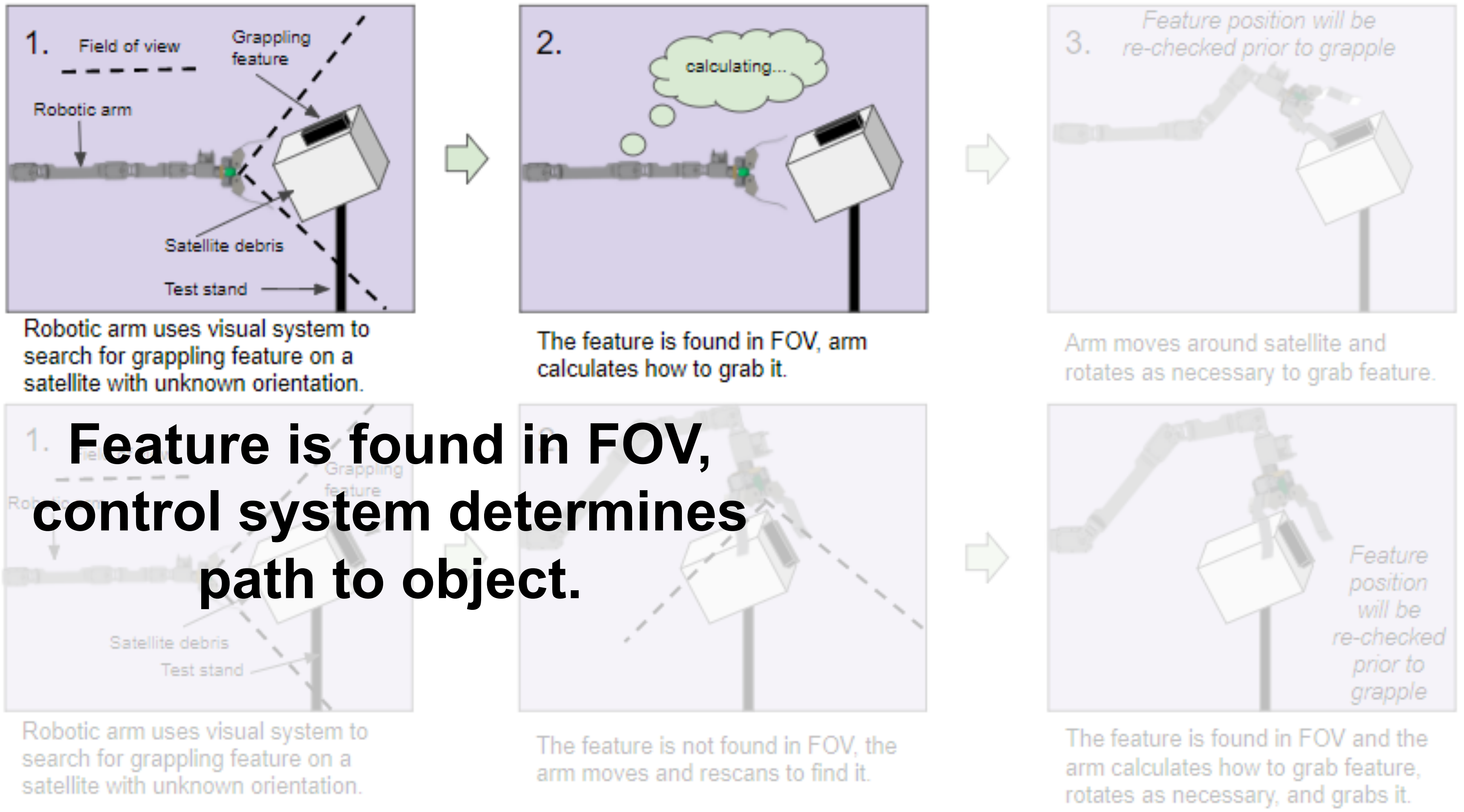
Test Facility



Visual system searches for feature on satellite with unknown orientation

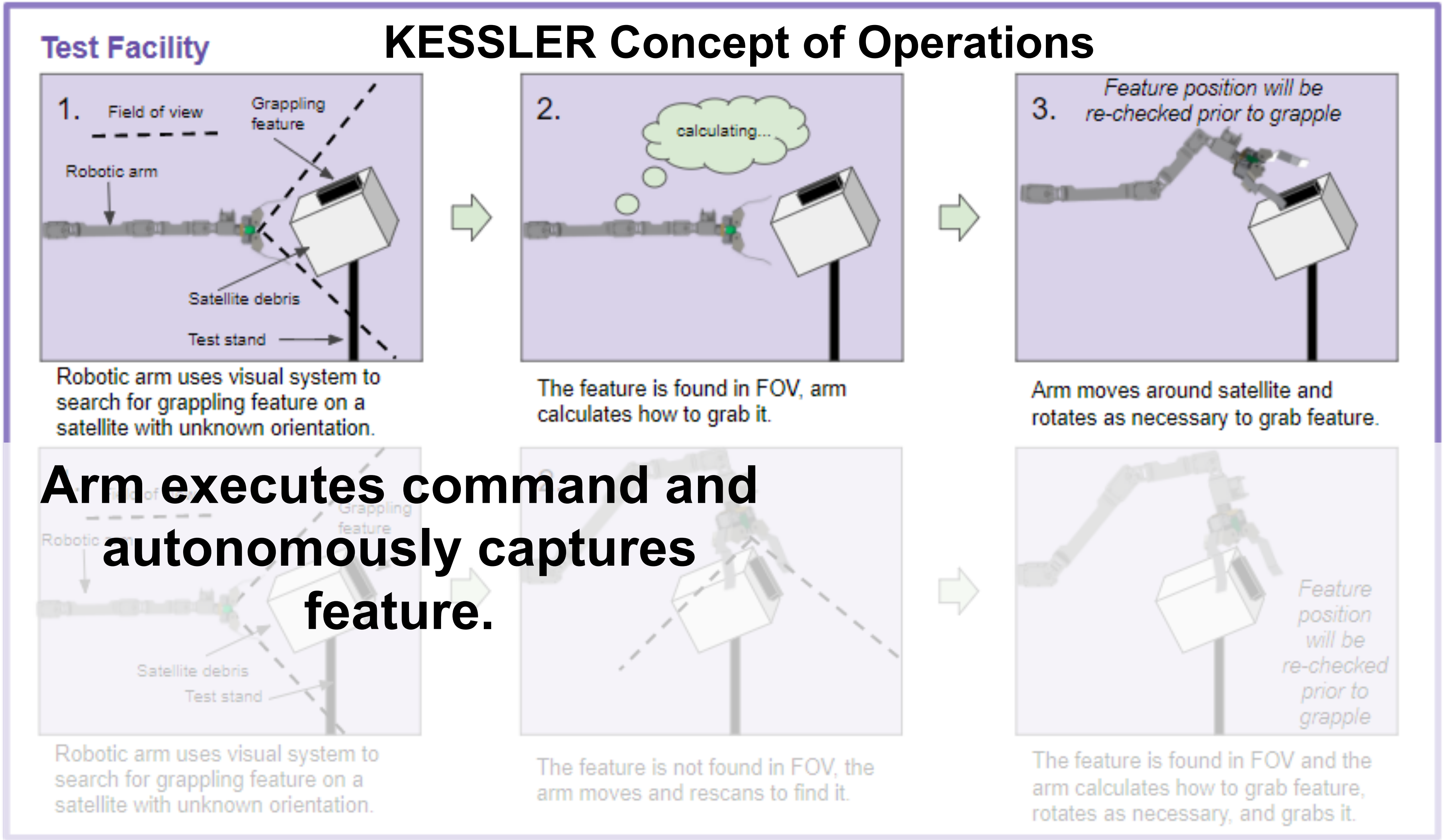
KESSLER Concept of Operations

Test Facility



Feature is found in FOV, control system determines path to object.

KESSLER Concept of Operations

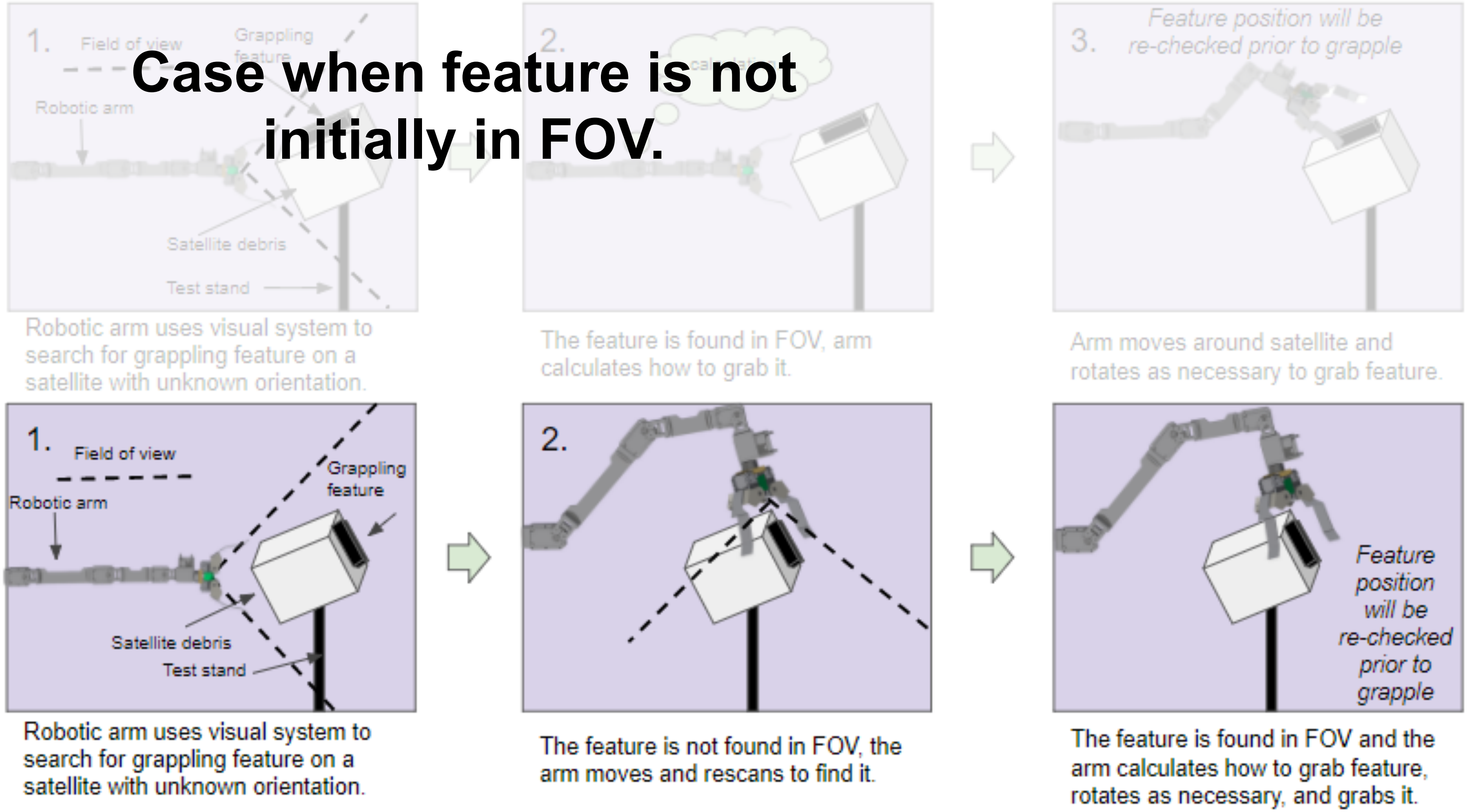


Arm executes command and autonomously captures feature.

KESSLER Concept of Operations

Test Facility

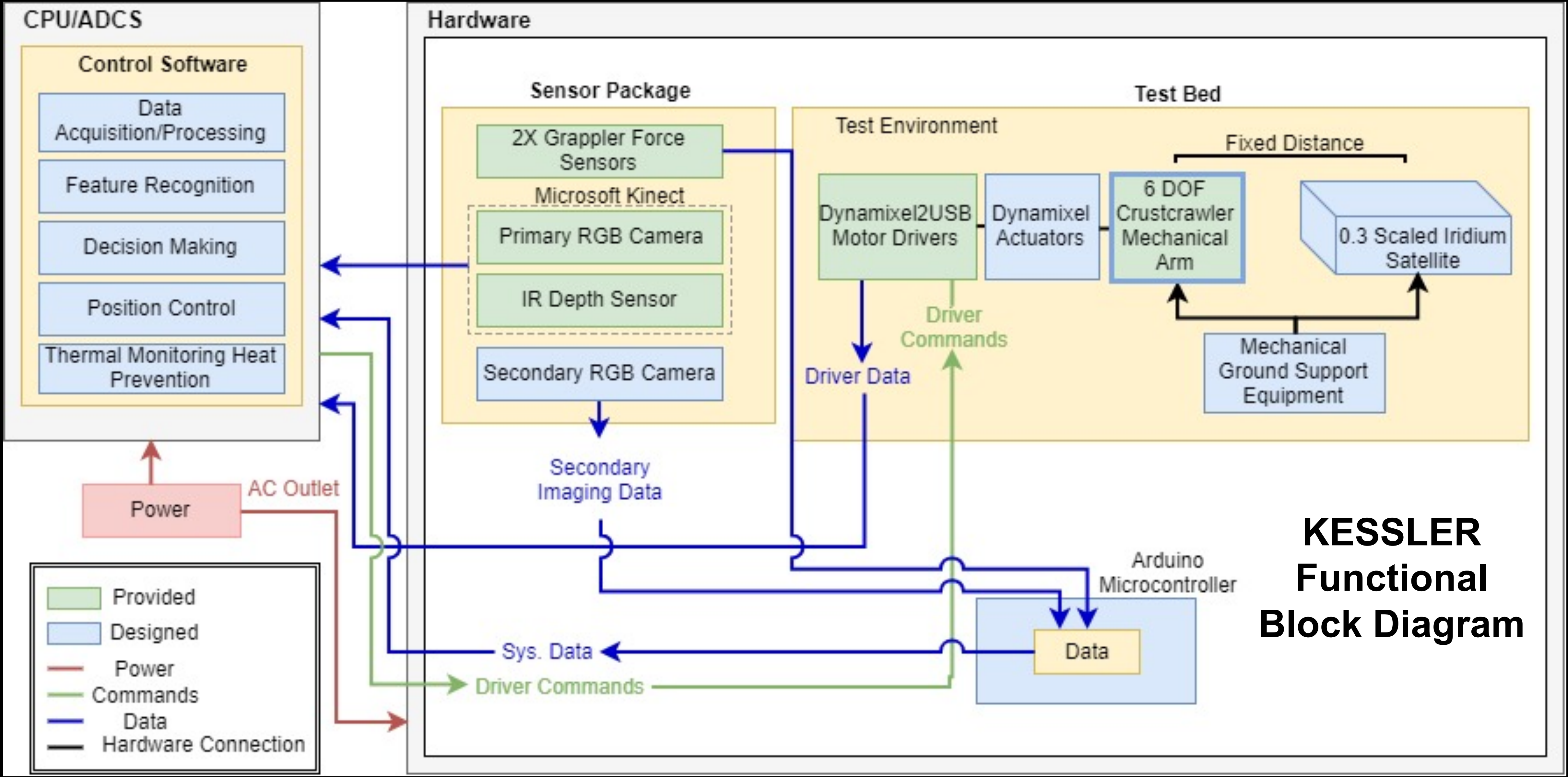
Case when feature is not initially in FOV.



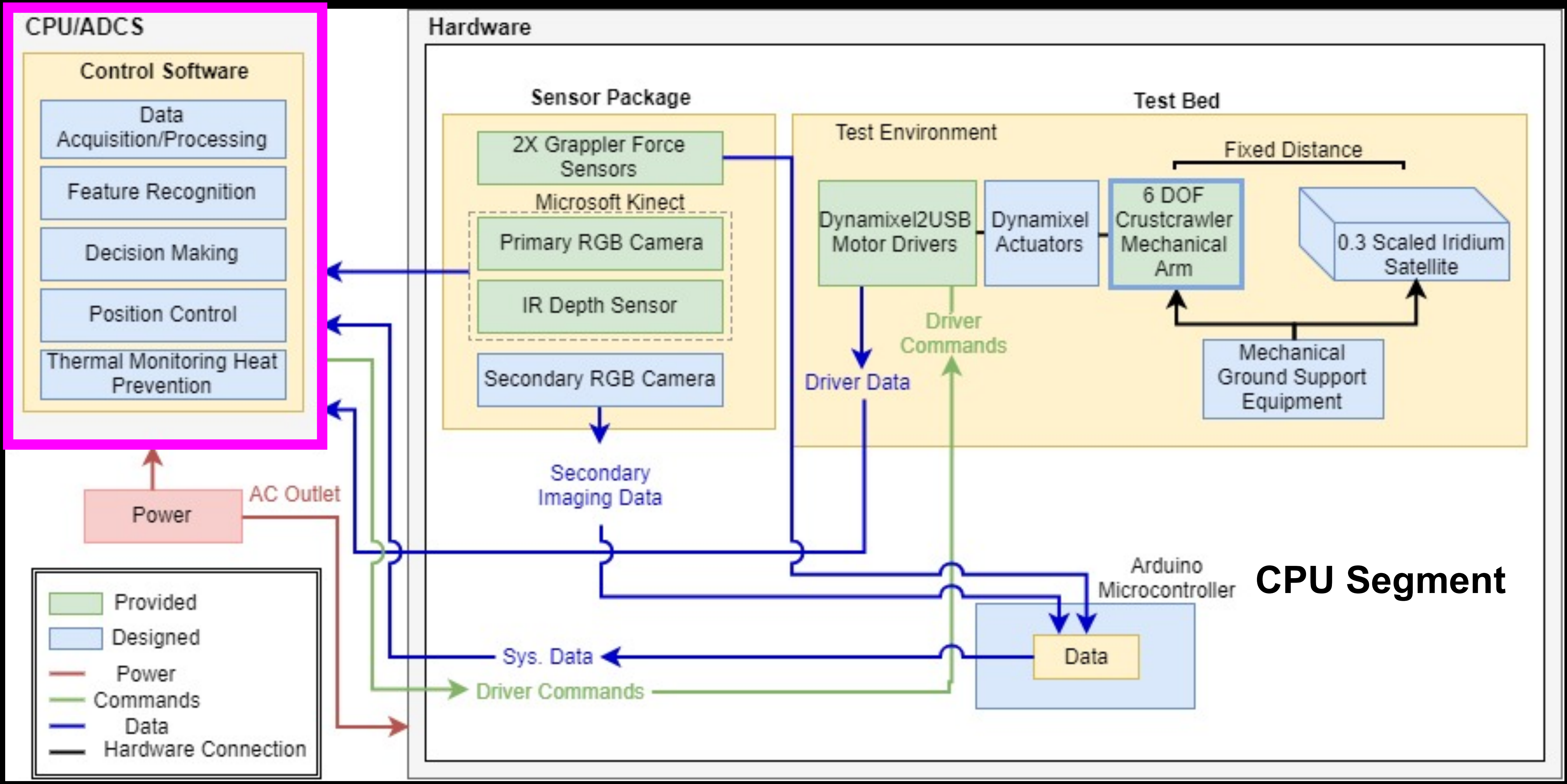
Project Definition

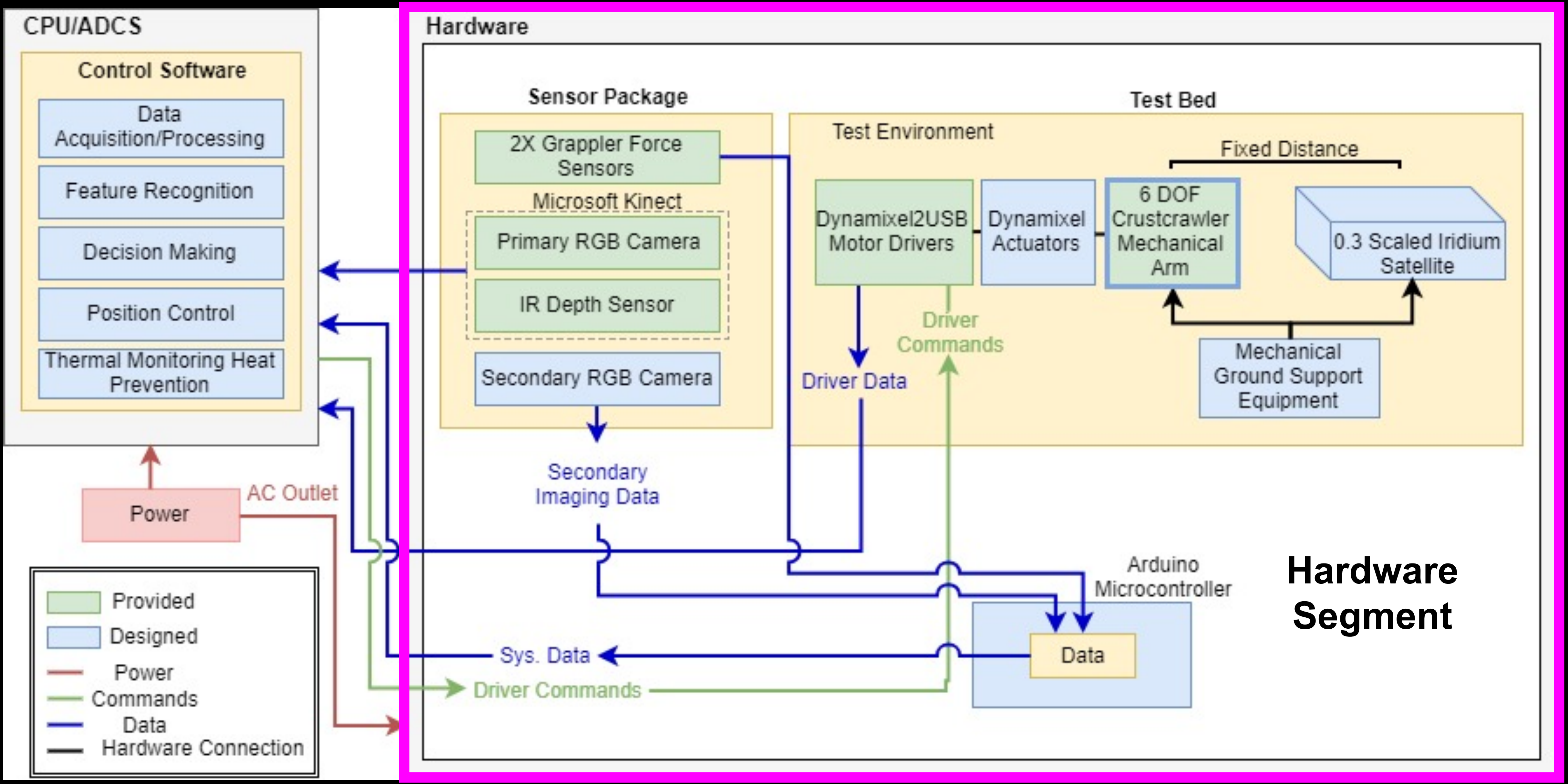
Functional Requirements

Req. ID	Requirement	Verification Method
F1	The **visual processing algorithm** shall identify the surface of a satellite in the primary camera's (RGB) field of view (FOV) and within the robotic arm's reach.	Imaging Analysis & Visual Inspection
F2	Control **algorithm** shall **define a path** to the location of a grappling feature.	Path Simulation (Experimental vs. Theoretical Location)
F3	Robotic arm shall **autonomously navigate and secure** at least one preselected grappling feature on the satellite.	Demonstration/Test
F4	The KESSLER system shall have a **total mission** time no greater than **53 minutes.**	Timing Analysis
F5	KESSLER shall **execute** a total of **3 end to end process operations** and succeed at least twice within the total mission time.	Demonstration/Test

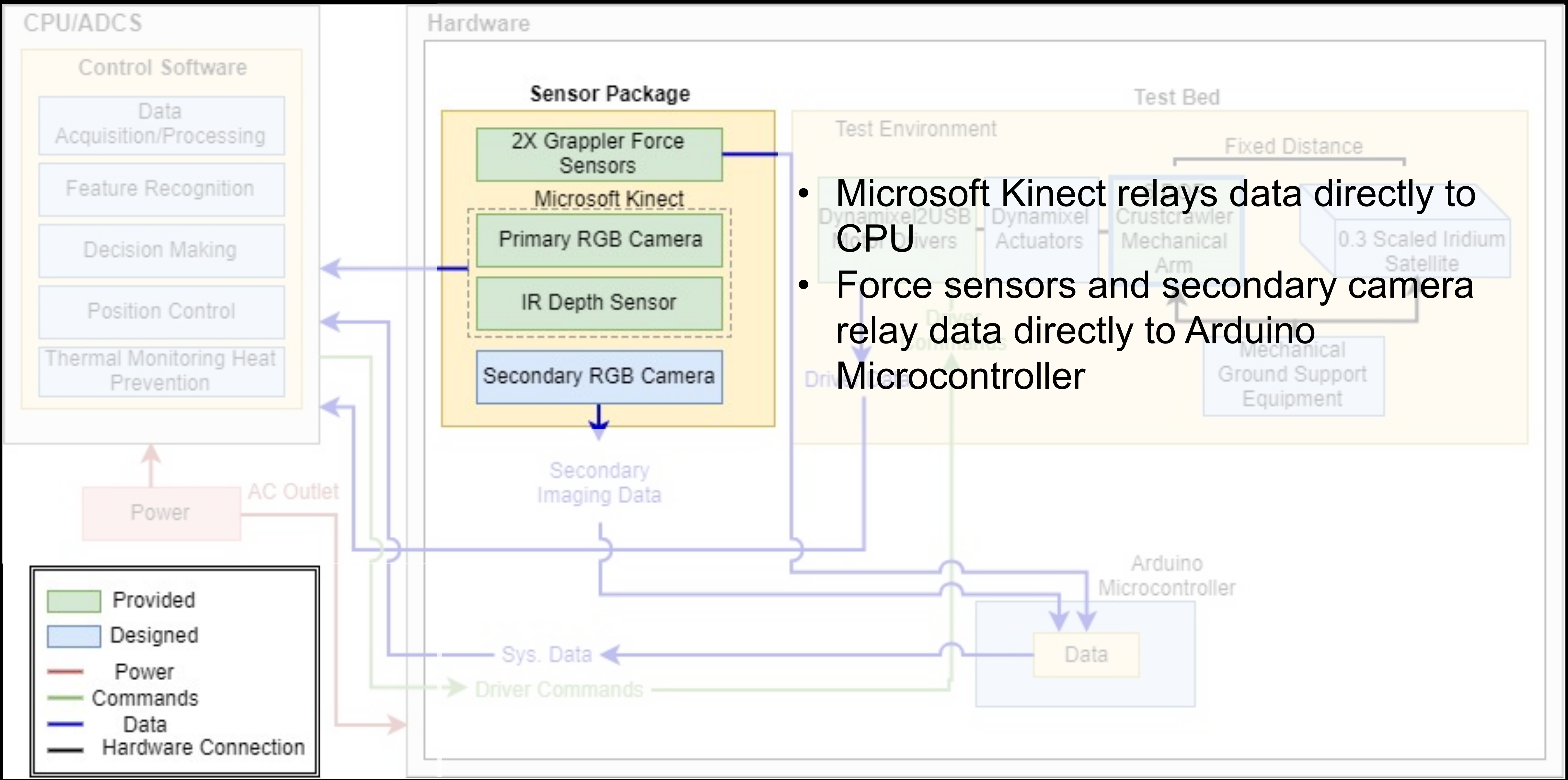


KESSLER Functional Block Diagram

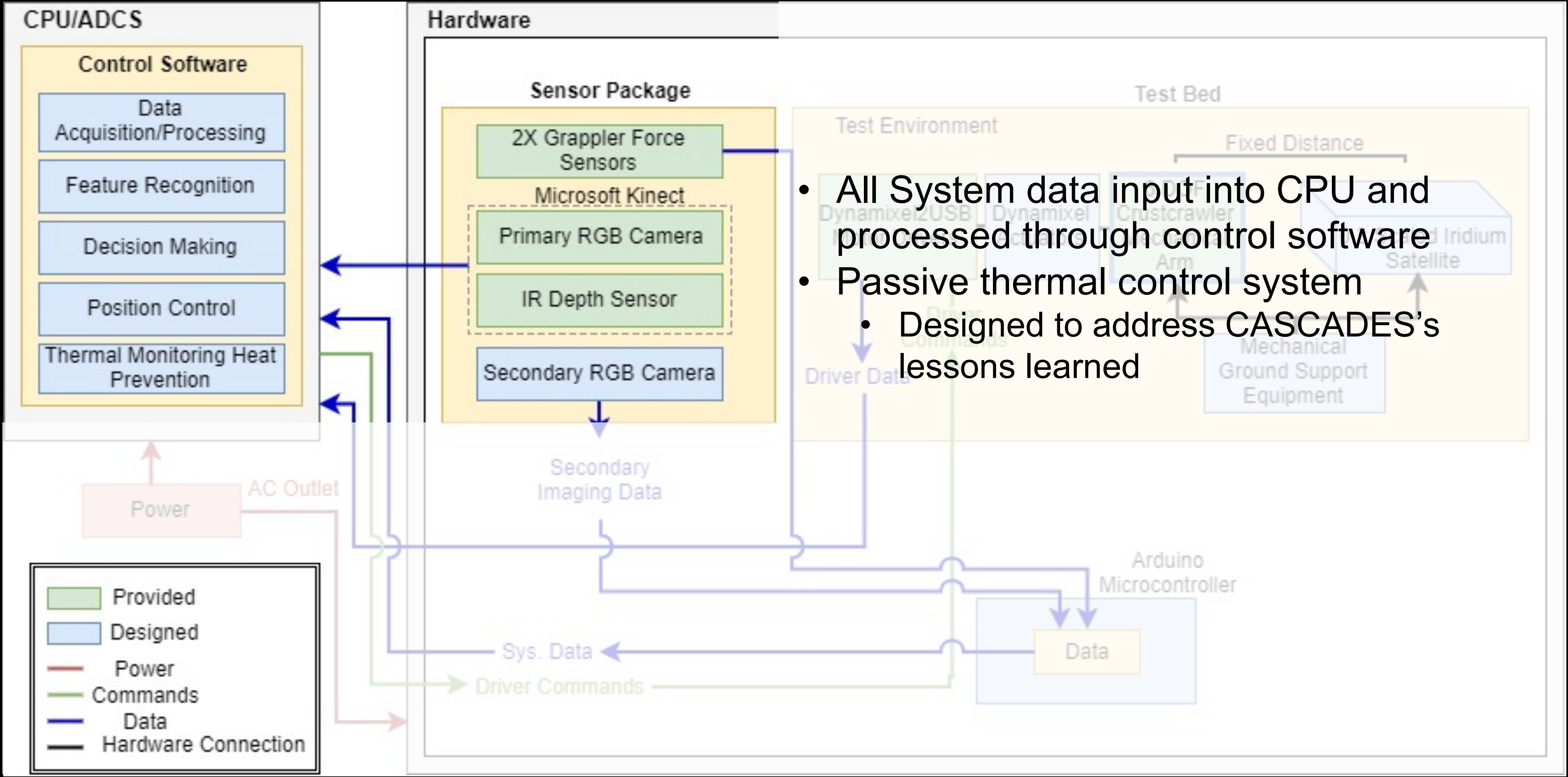

CPU/ADCS
Control Software
Data Acquisition/Processing
Feature Recognition
Decision Making
Position Control
Thermal Monitoring Heat Prevention
Hardware
Sensor Package
2X Grappler Force Sensors
Microsoft Kinect
Primary RGB Camera
IR Depth Sensor
Secondary RGB Camera
Secondary Imaging Data
Test Bed
Test Environment
Fixed Distance
Dynamixel2USB Motor Drivers
Dynamixel Actuators
6 DOF Crustcrawler Mechanical Arm
0.3 Scaled Iridium Satellite
Driver Commands
Driver Data
Mechanical Ground Support Equipment
Power
AC Outlet
Arduino Microcontroller
Data
Sys. Data
Driver Commands
Provided
Designed
Power
Commands
Data
Hardware Connection
CPU Segment



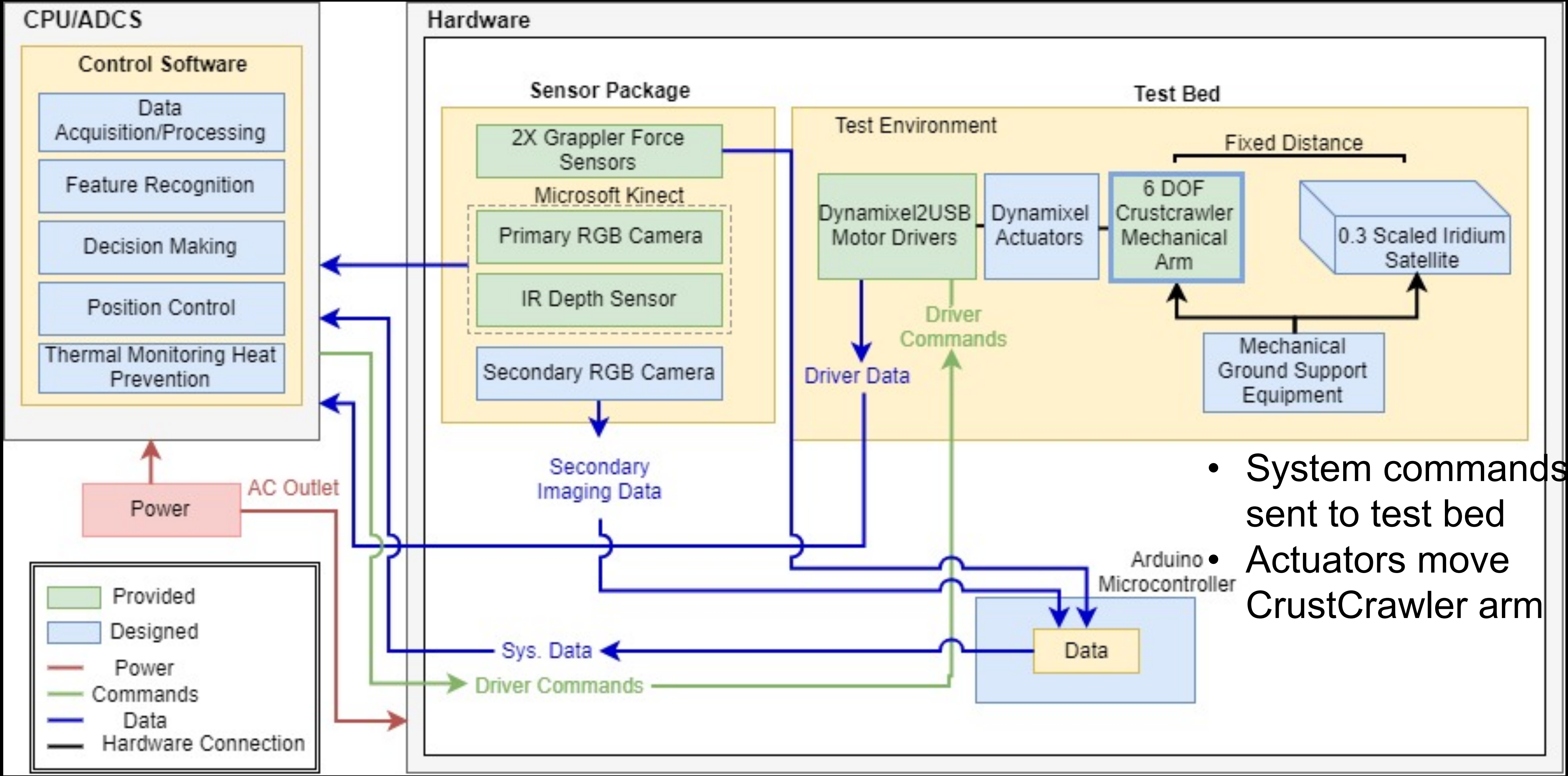
Hardware Segment



- Microsoft Kinect relays data directly to CPU
- Force sensors and secondary camera relay data directly to Arduino Microcontroller

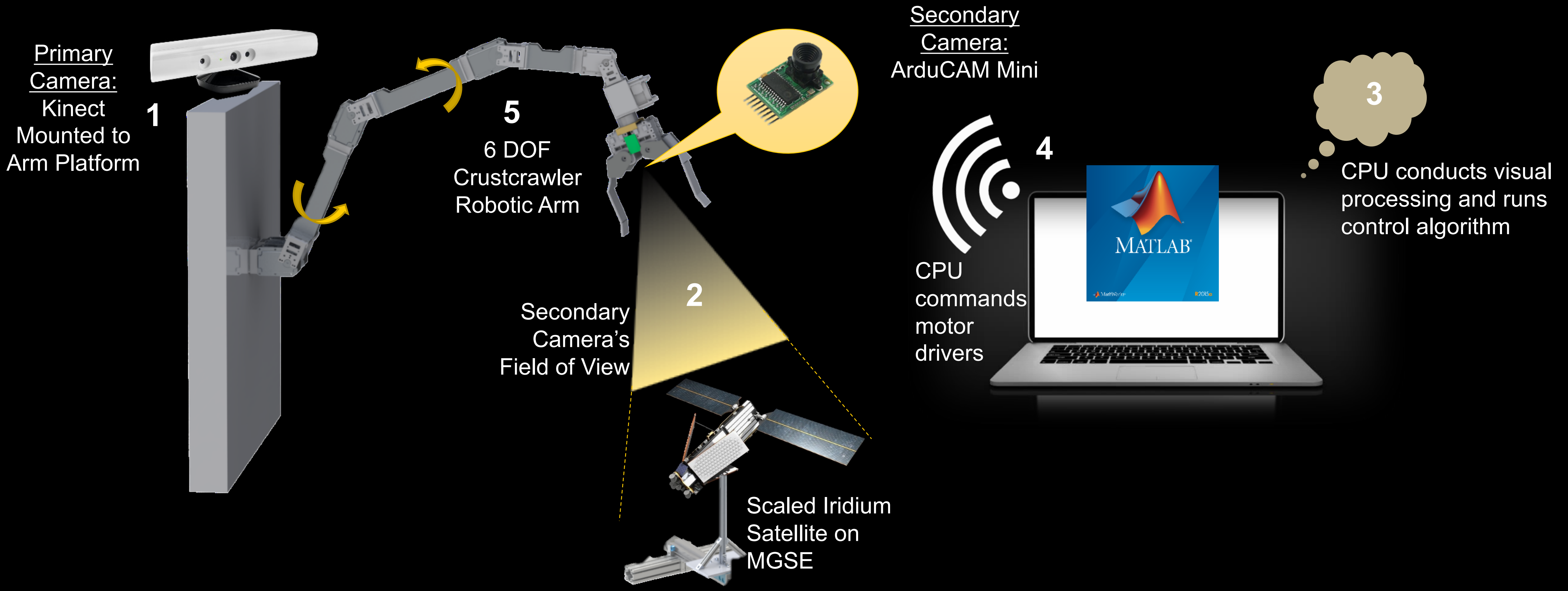


- All System data input into CPU and processed through control software
- Passive thermal control system
  - Designed to address CASCADES's lessons learned



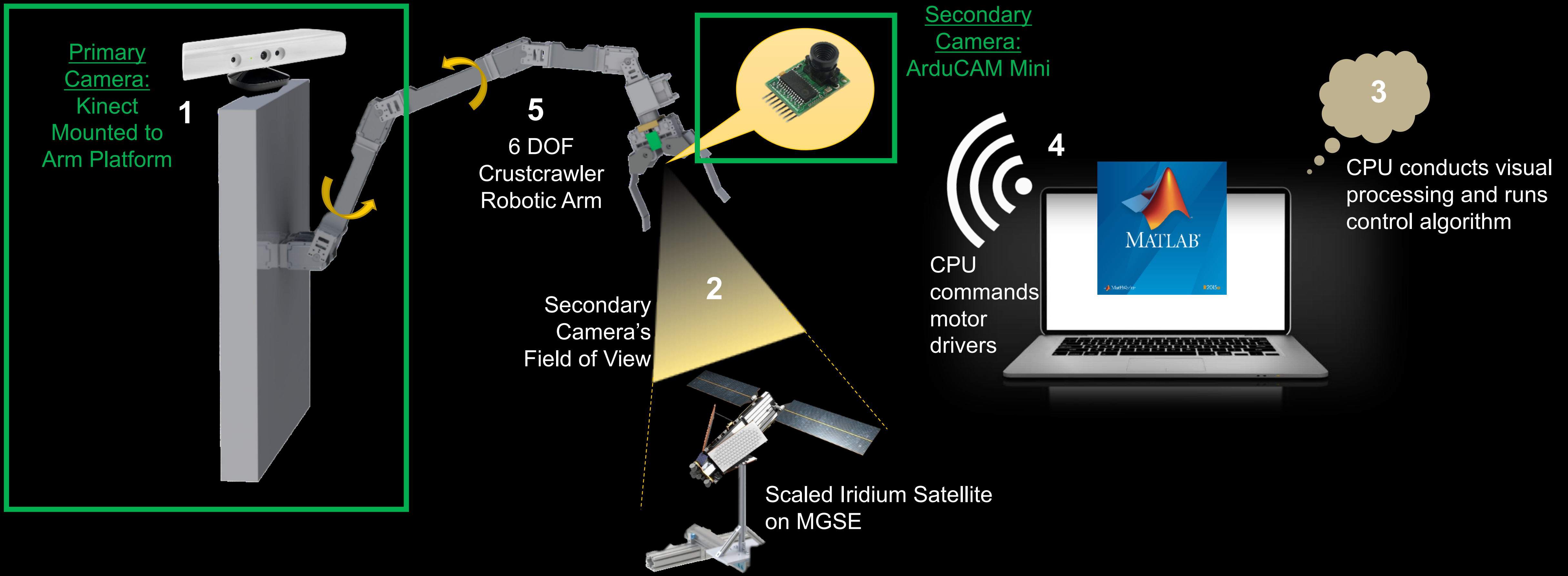
- System commands sent to test bed
- Actuators move CrustCrawler arm

Baseline Design Overview



KESSLER Primary Components & Functionality

Baseline Design Overview



1. Visual system searches for grappling feature on satellite

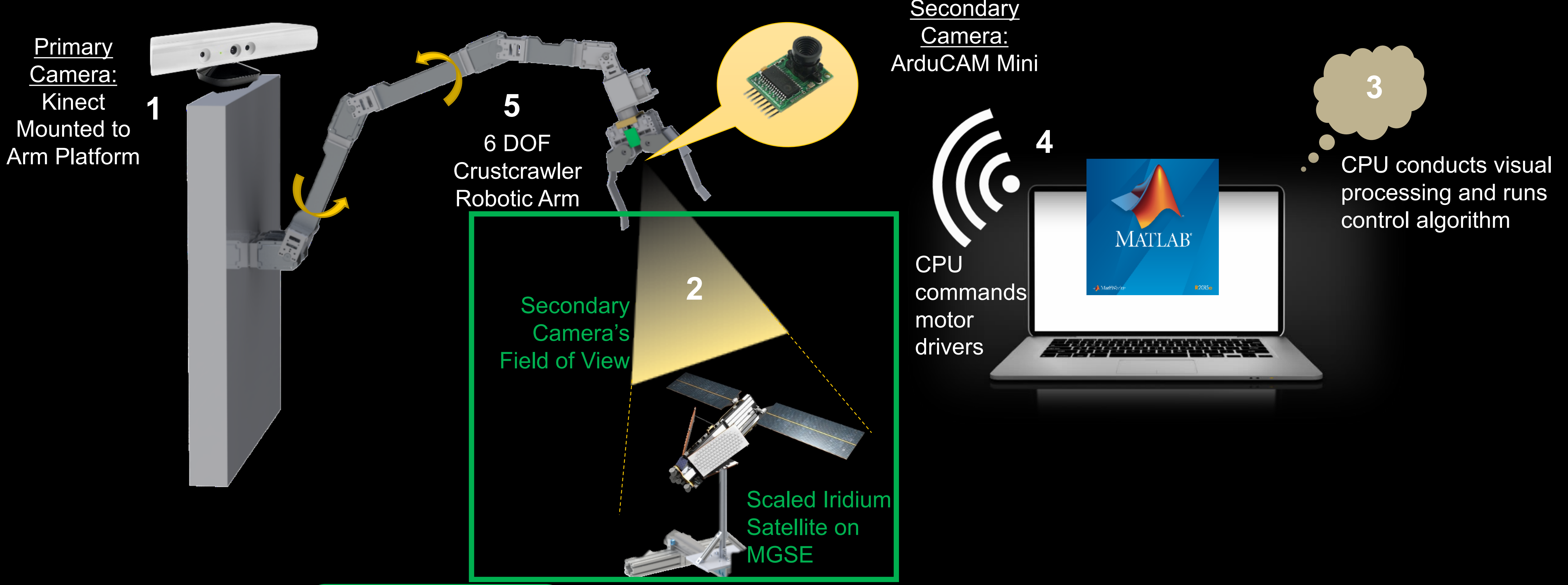
2. Algorithm identifies feature on satellite in FOV

3. Algorithm calculates how to grapple

4. CPU commands arm move as necessary to grapple feature

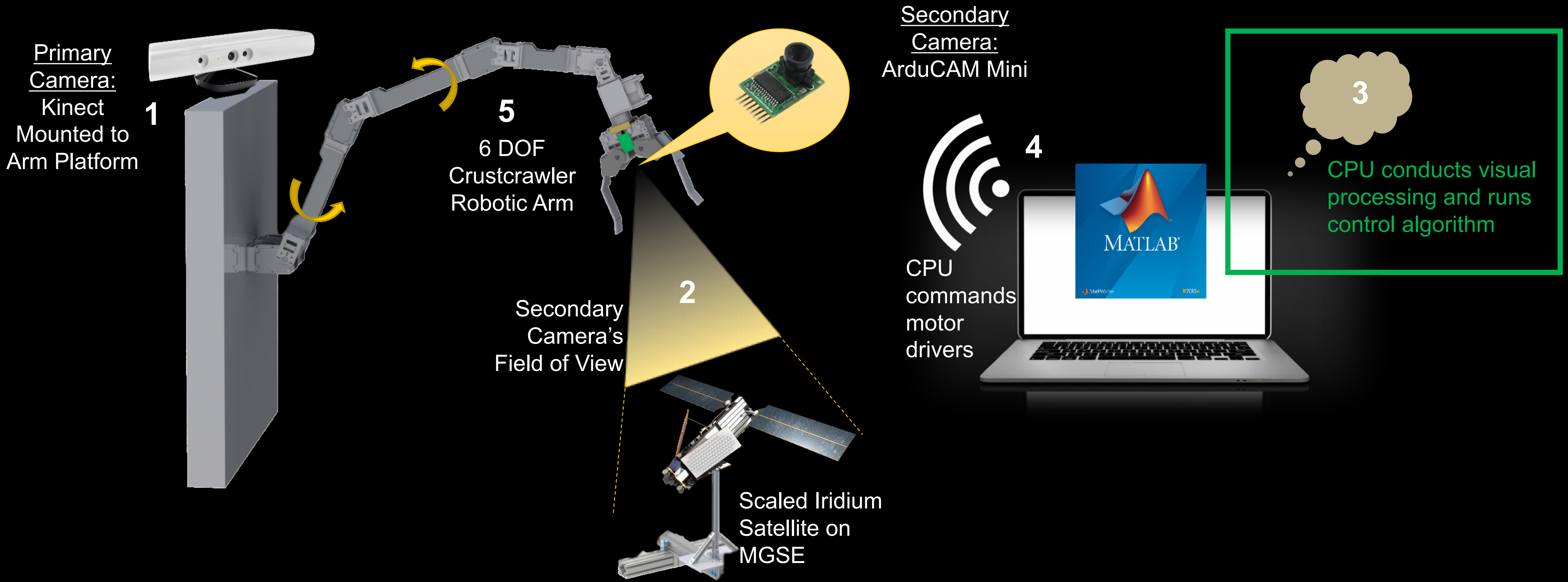
5. Robotic arm receives commands, rotates as necessary to grapple feature

Baseline Design Overview



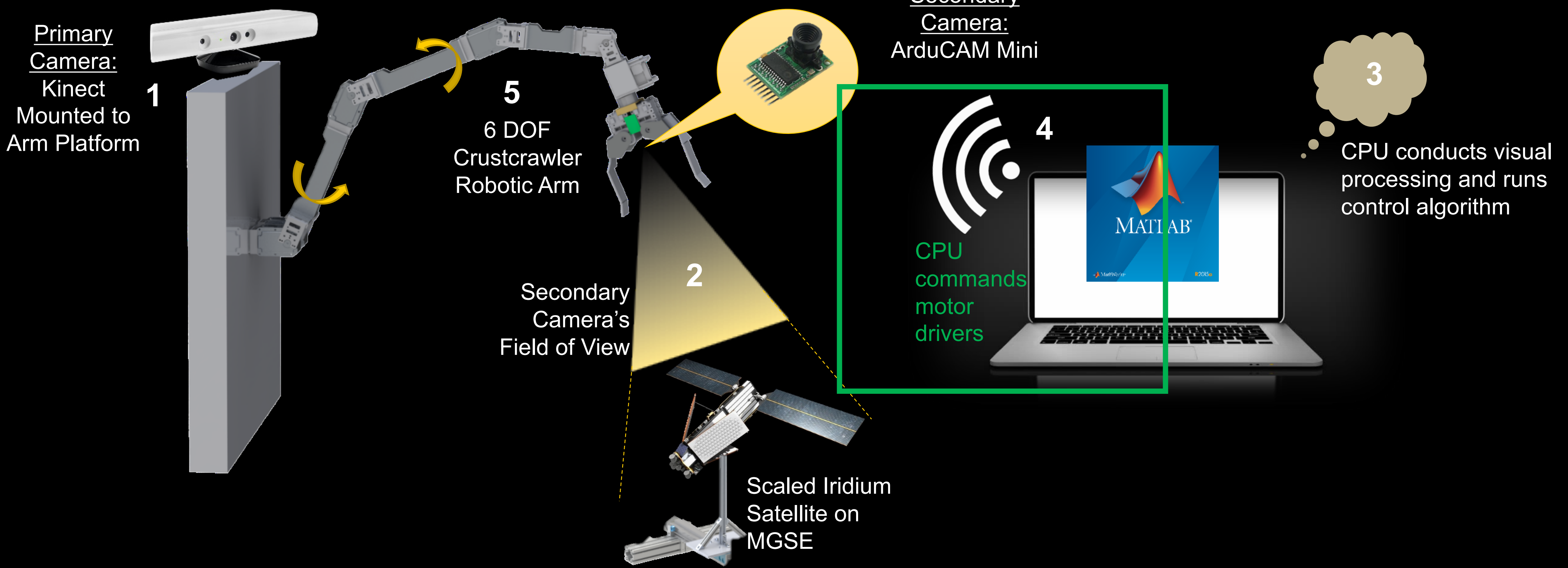
1. Visual system searches for grappling feature on satellite	**2.** Algorithm identifies feature on satellite in FOV	**3.** Algorithm calculates how to grapple	**4.** CPU commands arm move as necessary to grapple feature	**5.** Robotic arm receives commands, rotates as necessary to grapple feature

Baseline Design Overview



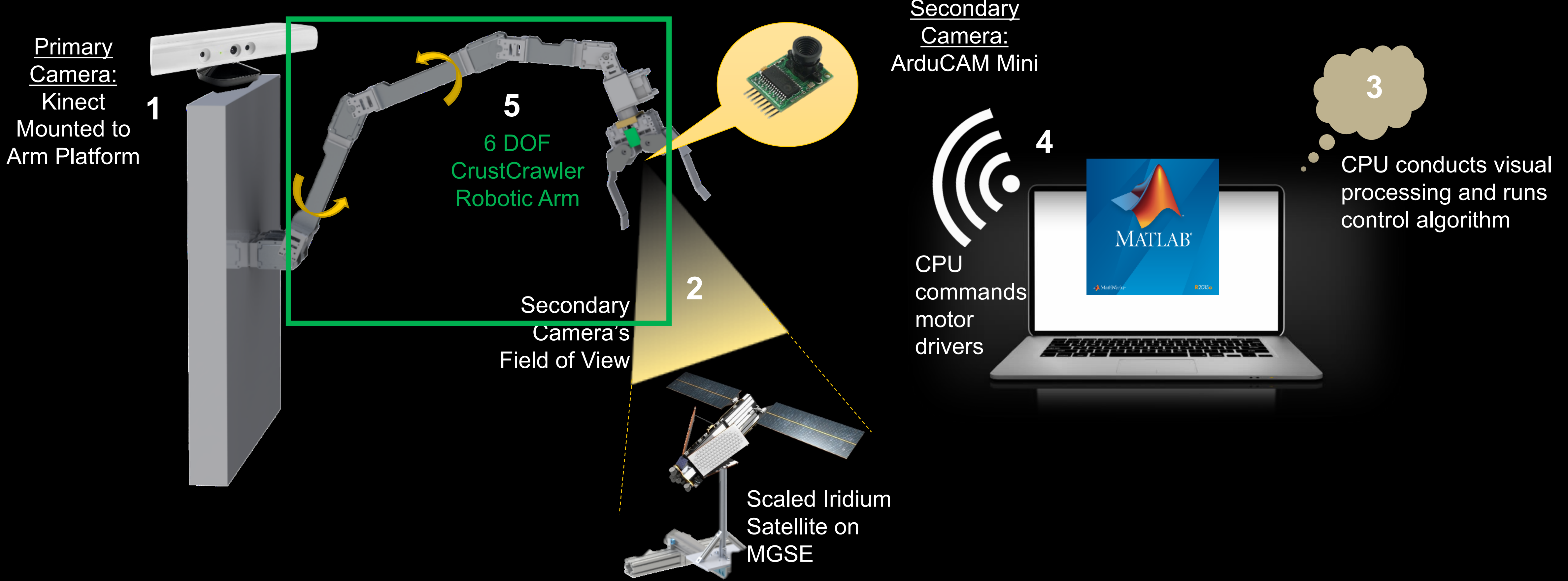
1. Visual system searches for grappling feature on satellite	**2.** Algorithm identifies feature on satellite in FOV	**3.** Algorithm calculates how to grapple	**4.** CPU commands arm move as necessary to grapple feature	**5.** Robotic arm receives commands, rotates as necessary to grapple feature

Baseline Design Overview



1. Visual system searches for grappling feature on satellite	**2.** Algorithm identifies feature on satellite in FOV	**3.** Algorithm calculates how to grapple	**4.** CPU commands arm move as necessary to grapple feature	**5.** Robotic arm receives commands, rotates as necessary to grapple feature

Baseline Design Overview



1. Visual system searches for grappling feature on satellite	**2.** Algorithm identifies feature on satellite in FOV	**3.** Algorithm calculates how to grapple	**4.** CPU commands arm move as necessary to grapple feature	**5.** Robotic arm receives commands, rotates as necessary to grapple feature

Evidence of Feasibility

Project Definition | Baseline Design | **CPE Overview** | CPE 1 Feasibility | CPE 2 Feasibility | CPE 3 Feasibility | Status & Future Work

Critical Project Elements Overview

Three Critical Project Elements

- CPE 1: Feature Recognition
  - Addresses Objectives 1 and 2
- CPE 2: Mechanical Robotic Arm
  - Addresses Objectives 4
- CPE 3: Control Systems
  - Addresses Objective 3 and 4

1. Take visual data confirming the target object is within FOV.	**2. Identify pre-defined grappling feature.**
3. Determine prediction path to feature location.	**4. Autonomously capture the feature via robotic arm**

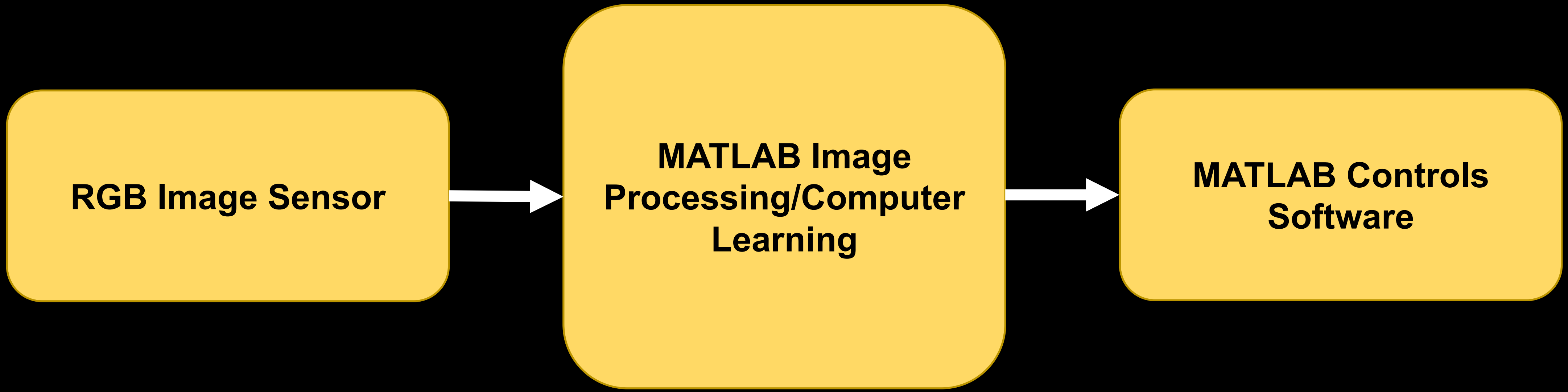
Project Definition | Baseline Design | **CPE Overview** | CPE 1 Feasibility | CPE 2 Feasibility | CPE 3 Feasibility | Status & Future Work

CPE 1: Feature Recognition

Cassidy Hawthorne (Visual Processing Lead)

CPE 1: Feature Recognition

Feature Recognition Process Flow



CPE 1: Feature Recognition

Computer Vision Systems Toolbox

- Object detection and recognition
- Tracking
- Camera calibration and 3D vision
- Display and graphics
- Analysis
- Code generation

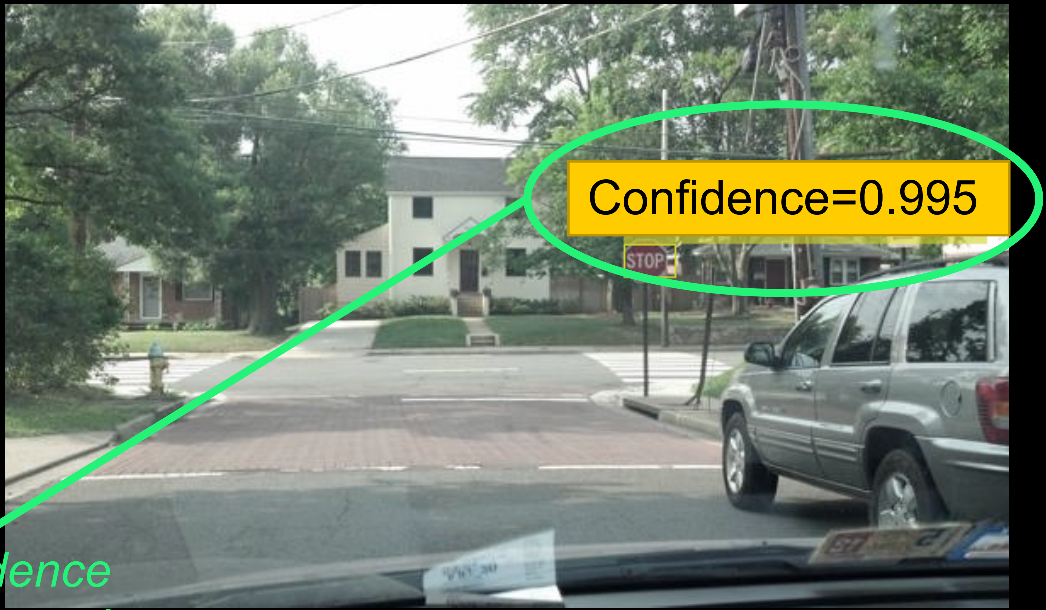
Image Processing Toolbox

- Deblurring and enhancement
- Image registration
- Transformations
- Image segmentation
- Measuring image features
- Working with large images

CPE 1: Feature Recognition

Object Detection Methods

Method 1: Deep Learning



Confidence percentage in identifying object

Fig. 5: Deep Learning Example via MATLAB

Method 2: Feature Matching

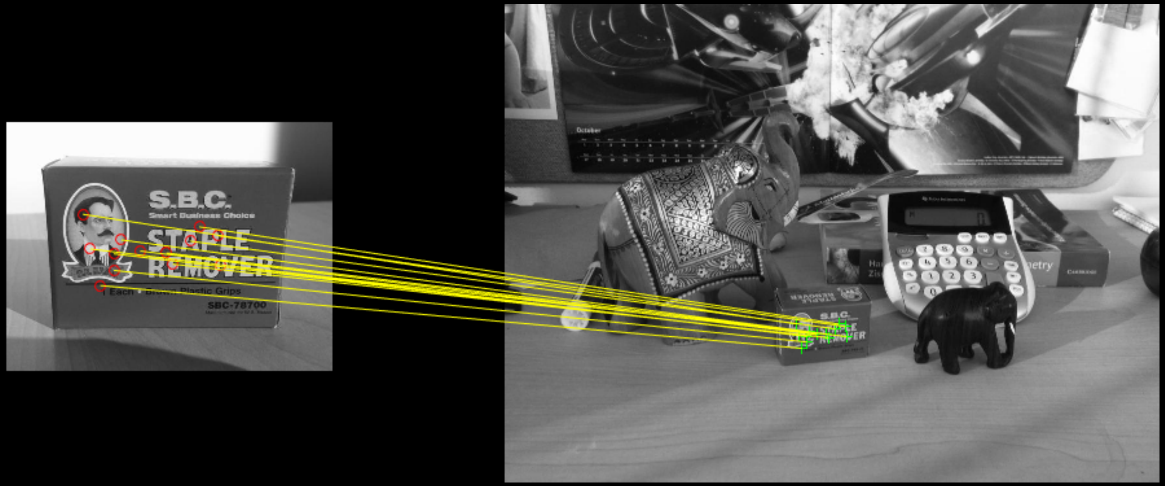


Fig. 6: Feature Matching Example via MATLAB

Project Definition | Baseline Design | CPE Overview | **CPE 1 Feasibility** | CPE 2 Feasibility | CPE 3 Feasibility | Status & Future Work

CPE 1: Feature Recognition

Feature Matching: Object Identification
Testing with Coke Box

- Computer Vision System Toolbox
- MATLAB find strongest features in each image
  - 100 strongest points in left image
  - 300 strongest points in right image
- Test performed 10-8-17

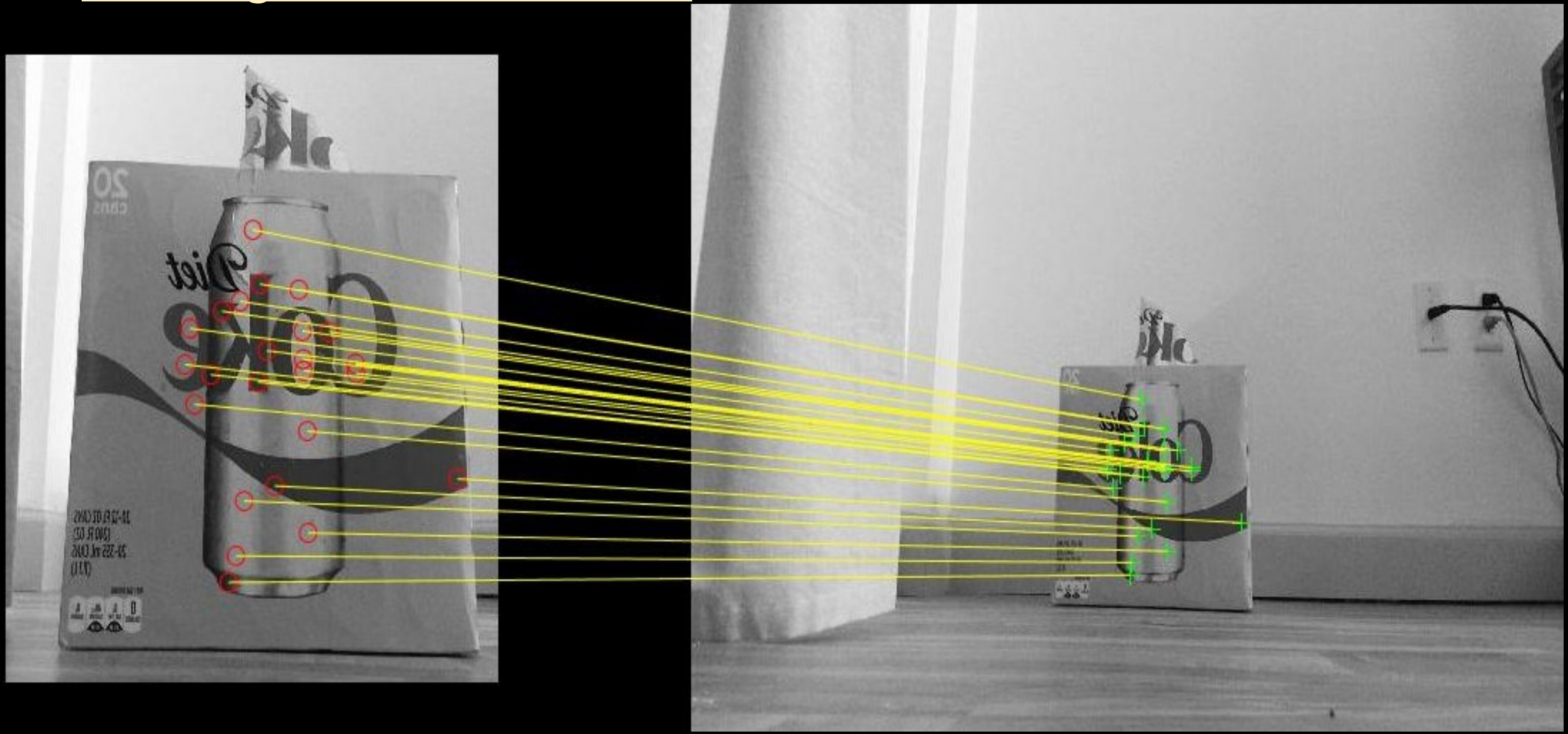


Fig. 7: Coke Box Test: 100 Strongest Points

Fig. 8: Coke Box Test: 300 Strongest Points

CPE 1: Feature Recognition

Feature Matching: Object Identification Testing with Funnel

- Computer Vision System Toolbox
- Minimum of 3 matches
- MATLAB find strongest features in each image
  - 100 strongest points in left image
  - 300 strongest points in right image
- Test performed 10-8-17

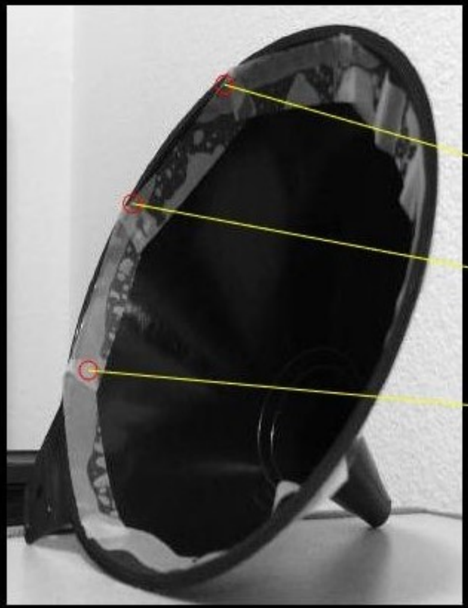


Fig. 9: Funnel Test Object Image

Fig. 10: Funnel Test Full Image with Object

CPE 1: Feature Recognition

Risks & Concerns

- Very little visual processing experience on KESSLER team
- Mitigation:
  - Extensive documentation through MATLAB
  - On-campus Subject Matter Experts

Future Work

- Develop preliminary feature database
- Test feature database with test images
- Machine learning with object identification

CPE 2: Mechanical Robotic Arm

Abdiel Agramonte-Moreno (Mechanical Design Lead)

Project Definition → Baseline Design → CPE Overview → CPE 1 Feasibility → **CPE 2 Feasibility** → CPE 3 Feasibility → Status & Future Work

CPE 2: Mechanical Robotic Arm

CASCADE 5DOF Robotic Arm

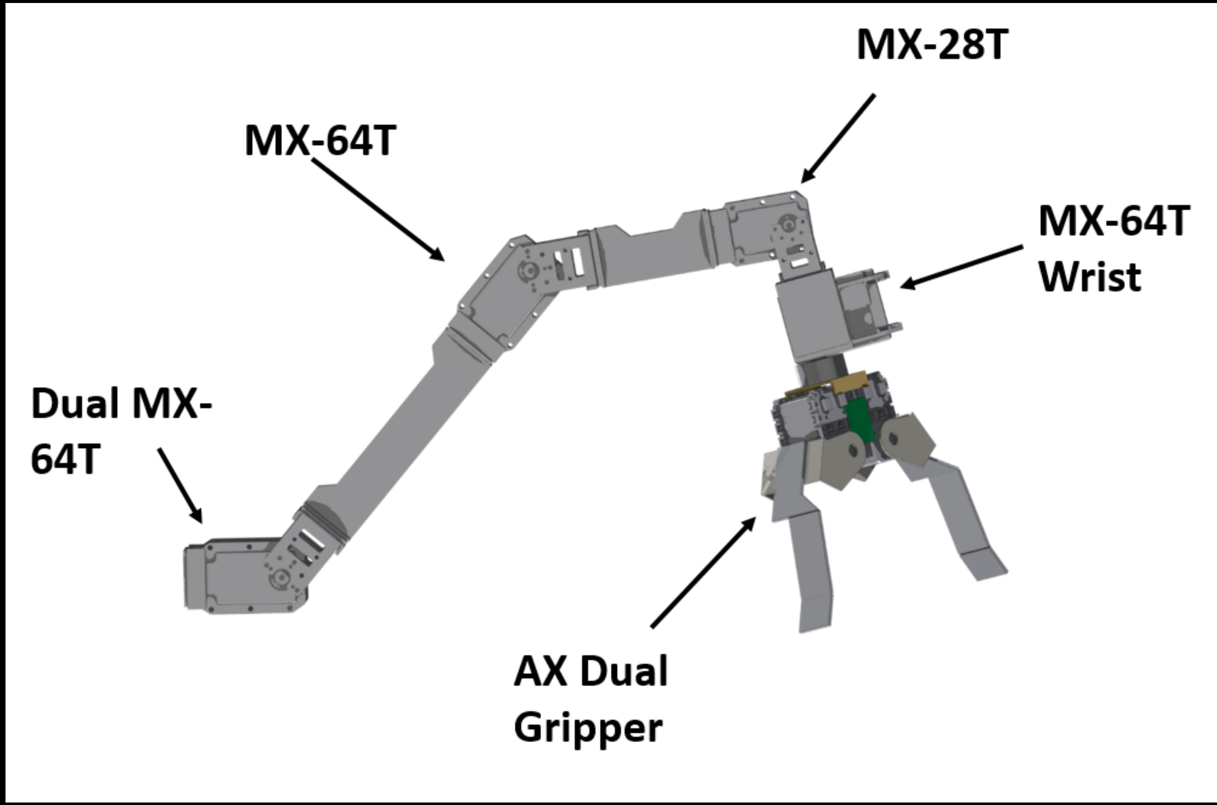


Fig. 11: CASCADE 5 DOF Robotic Arm with Actuators Labeled

- Length: 22 inches
- Mechanically interfaced to rotating platform to allow 360 deg. motion
- Can capture objects in linear path
- COTS Modular Robotic Arm

CPE 2: Mechanical Robotic Arm

KESSLER 6DOF Robotic Arm

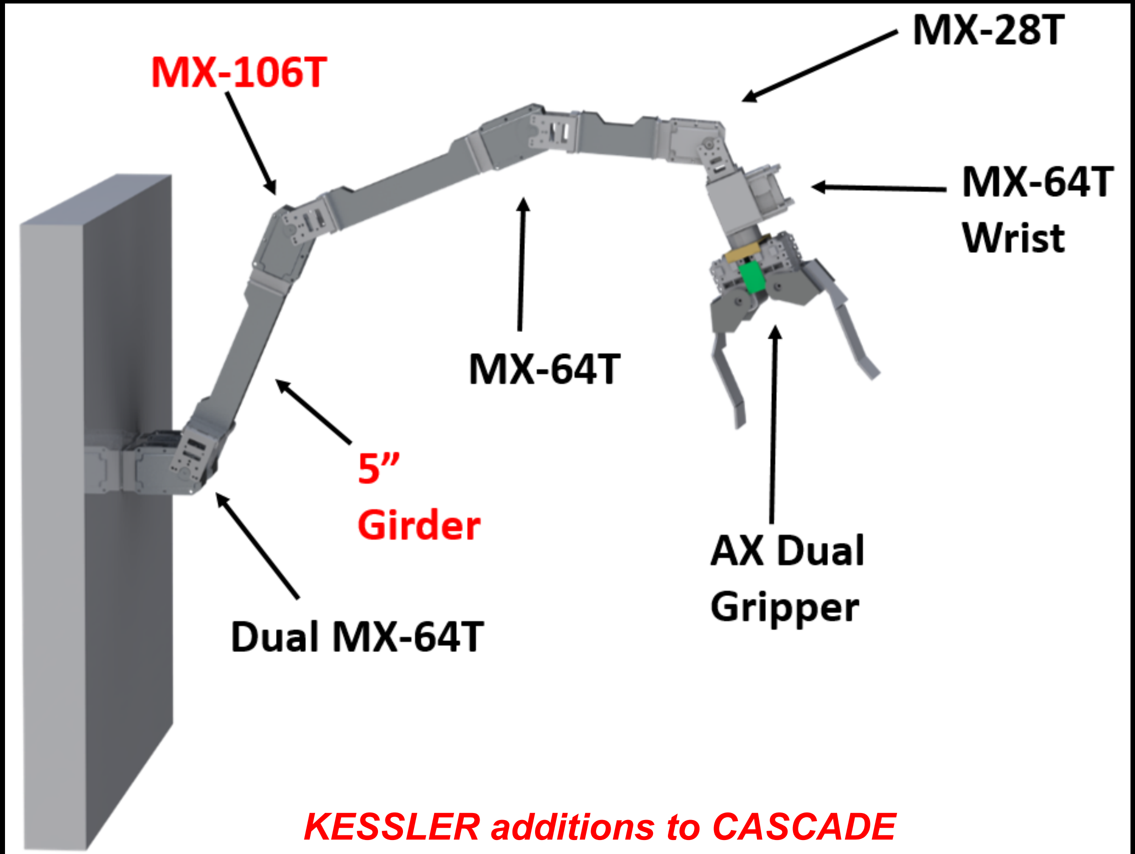
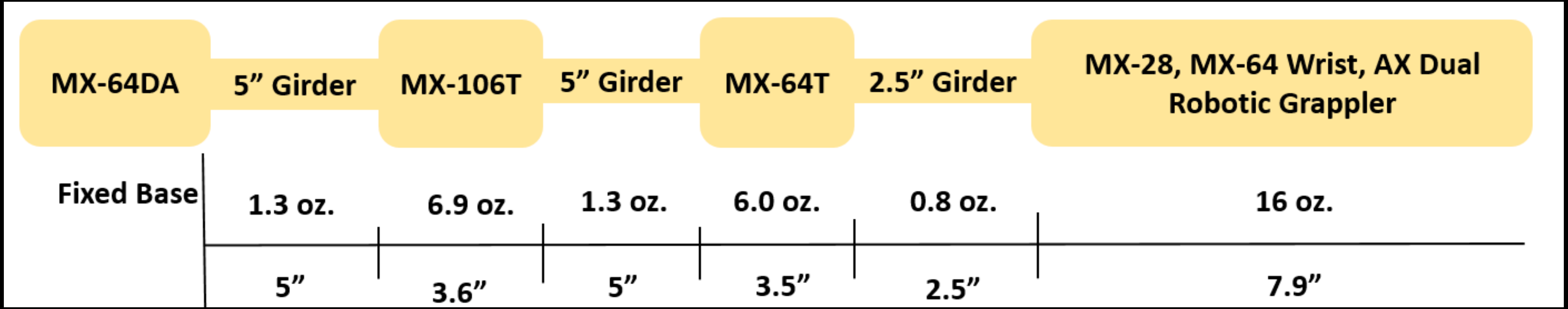


Fig. 12: KESSLER 6 DOF Robotic Arm with Actuators & Girder Labeled

- Length: 31 inches
  - 9 inches more than CASCADE
- New additions: MX-106T, 5 inch Girder
- Mechanically interfaced to rotating platform to allow 360 deg. motion
- Can capture objects from side faces (perpendicular to baseplate plane)

CPE 2: Mechanical Robotic Arm

Actuator Loading Feasibility Analysis



- Total Length: 27.5” (Measured from fixed base; neglect MX-64DA Length)
- Total Supported Weight: 32.3 oz.

Spacecraft FOS: 1.4

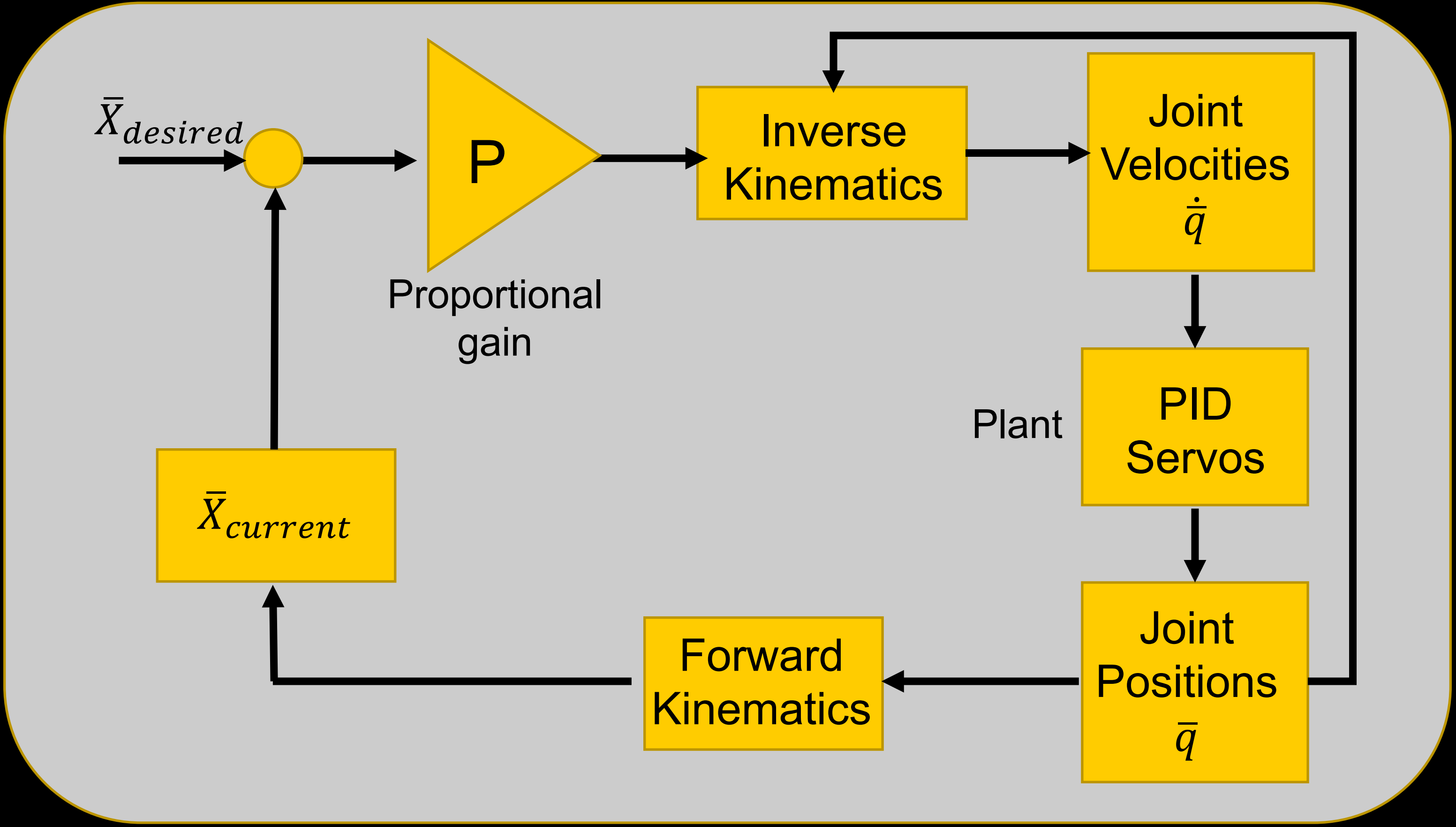
Specification	MX-64DA	MX-106T	MX-64T (heritage)
Max Stall Torque (oz.in.)	2,200	1,400	1,030
Torque Experienced (oz.in)	888	455	173
Factor of Safety	2.33	3.11	5.95

CPE 3: Control Systems

Abigail Johnson (Software I&T Lead)
Thanh Cong Bui (Hardware I&T Lead)

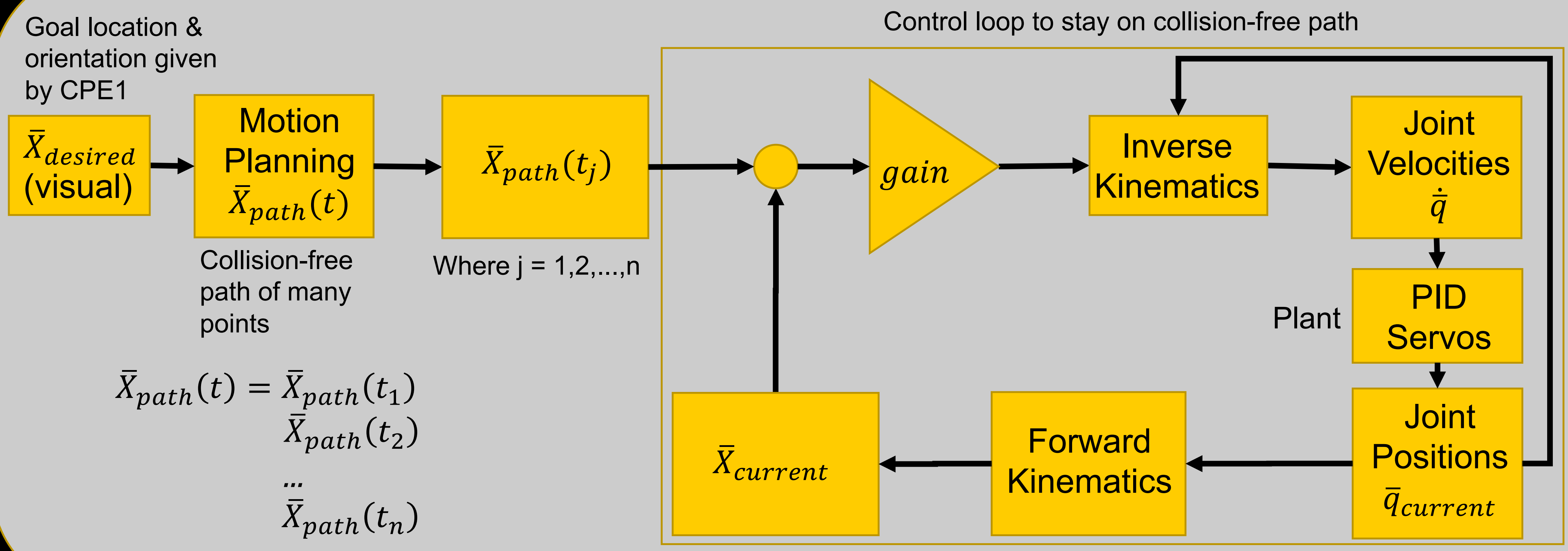
CPE 3: Control Systems

CASCADE Controls



CPE 3: Control Systems

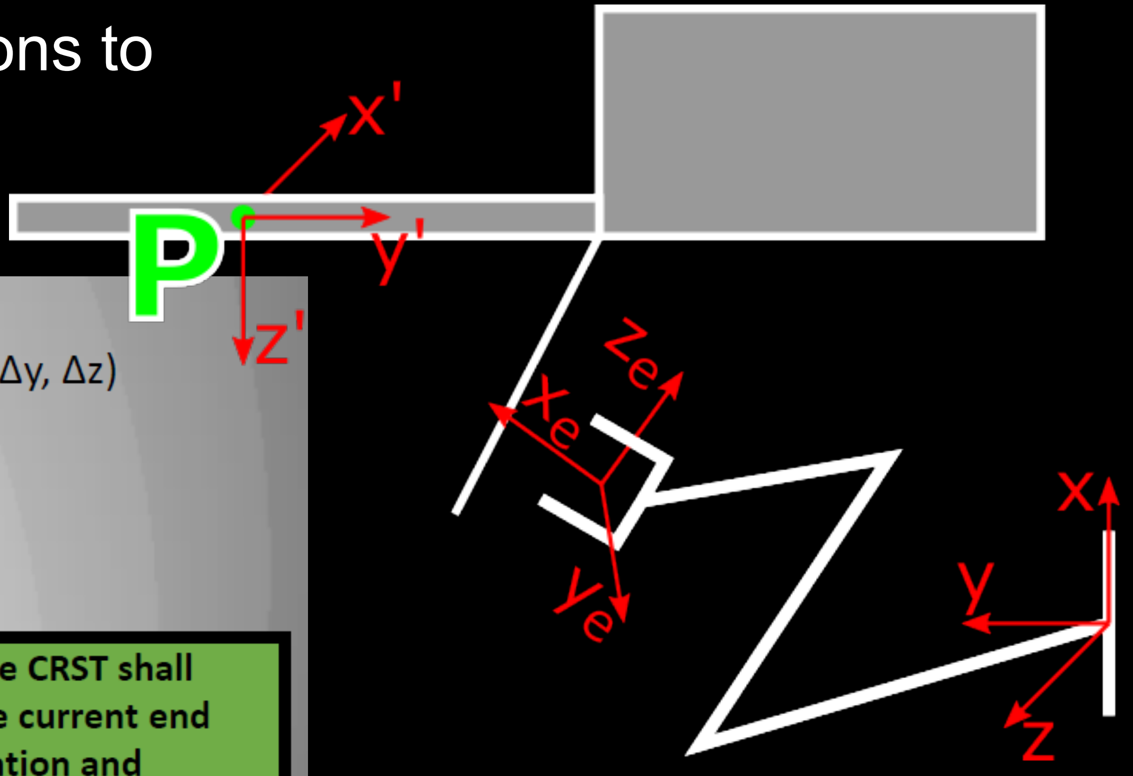
KESSLER Controls



$$\bar{X}_{path}(t) = \begin{matrix} \bar{X}_{path}(t_1) \\ \bar{X}_{path}(t_2) \\ \ldots \\ \bar{X}_{path}(t_n) \end{matrix}$$

CPE 3: Control Systems

Forward kinematics convert joint positions to Cartesian position and orientation



The transformation matrix from one joint to the next is the augmentation of rotation matrix R and translation vector (Δx, Δy, Δz)

$$T = \begin{pmatrix} R_{11} & R_{12} & R_{13} & \Delta x \\ R_{21} & R_{22} & R_{23} & \Delta y \\ R_{31} & R_{32} & R_{33} & \Delta z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The transformation from inertial coordinates to end effector coordinates is then:

$$T_{end}^{inertial} = T_0^{inertial} T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_{end}^5$$

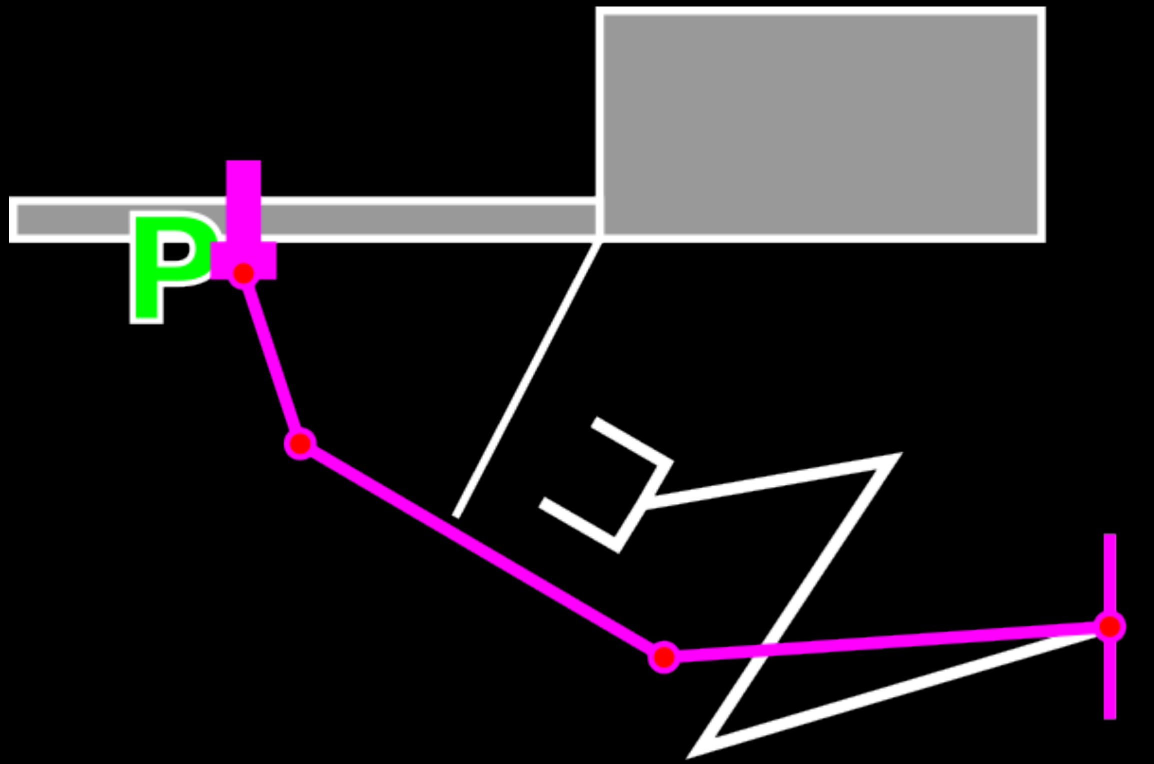
This can be used to calculate x, y, z coordinates and roll, pitch, yaw of end effector

DR 1.3.1: The CRST shall calculate the current end effector location and orientation during the demonstration.

CPE 3: Control Systems

- Inverse kinematics: Position and orientation to required joint positions
- Provides a goal for the motion planning to work towards

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{bmatrix} = \left[F_{6\times6}\right]^{-1} \begin{bmatrix} x \\ y \\ z \\ \chi \\ \psi \\ \zeta \end{bmatrix}$$



Project Definition | Baseline Design | CPE Overview | CPE 1 Feasibility | CPE 2 Feasibility | **CPE 3 Feasibility** | Status & Future Work

CPE 3: Control Systems

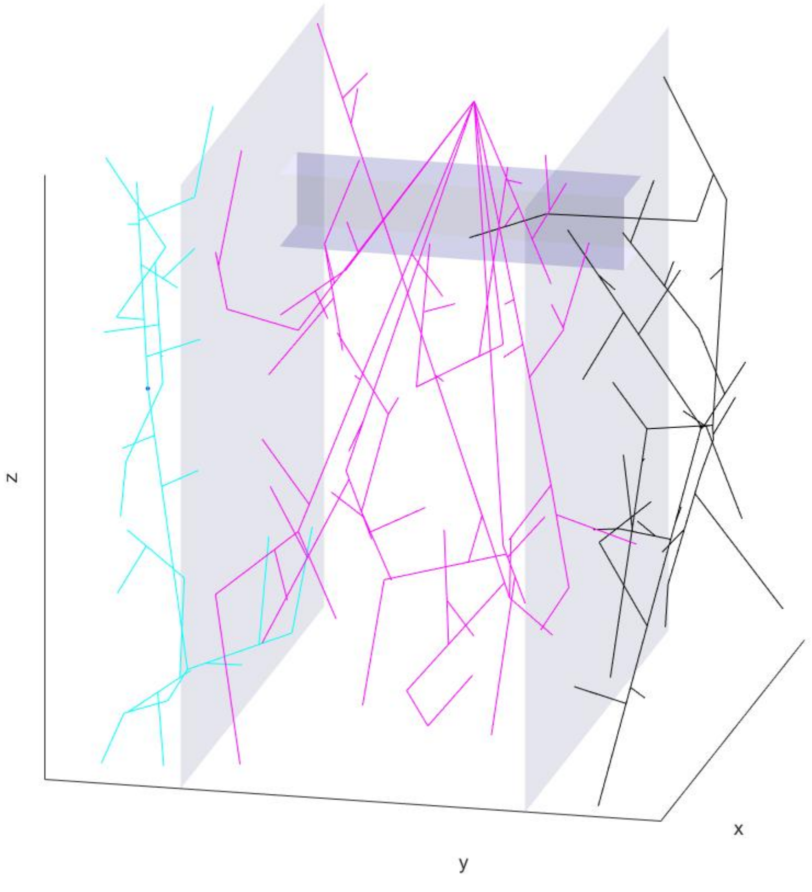


Fig. 13 RRT with 3 Active Trees (MATLAB), wall collision avoidance

Rapidly Exploring Random Tree (RRT) – Collision Free Path Determination

- Use to find collision-free path
- Pick random points and tries to connect to current tree
- Optimize final path

CPE 3: Control Systems

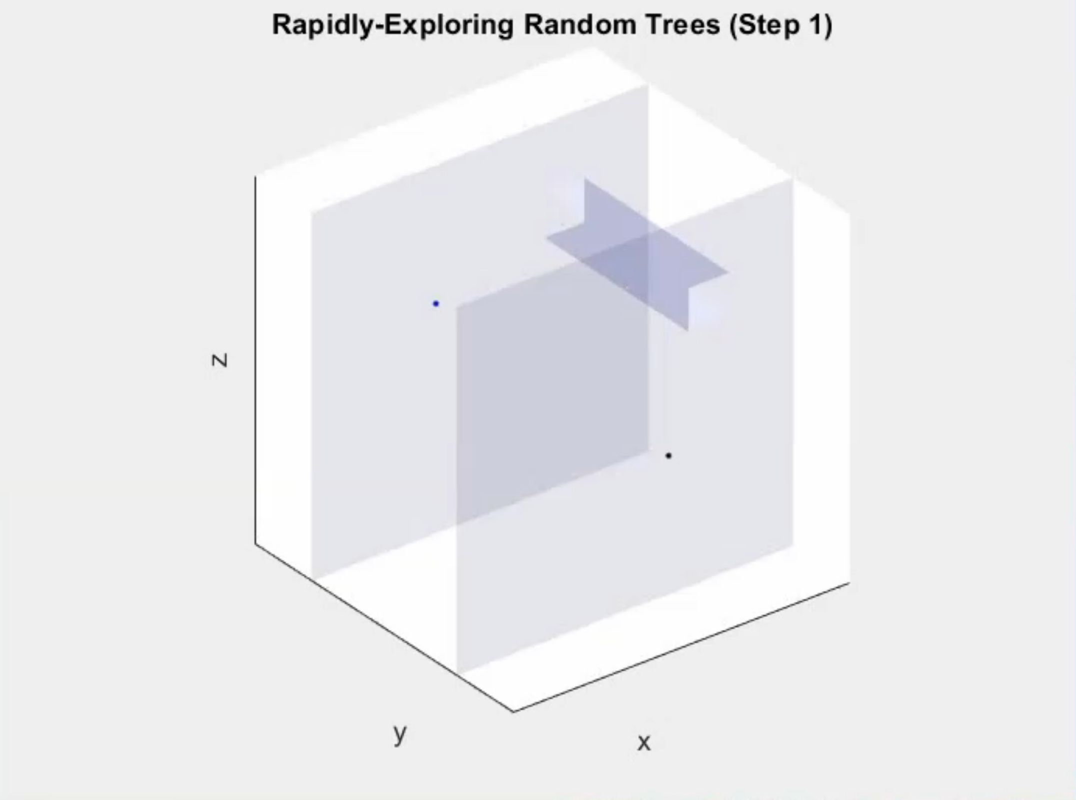


Fig. 14 RRT Animation (KESSLER MATLAB Simulation)

Rapidly Exploring Random Tree (RRT) – Collision Free Path Determination

- Use to find collision-free path
- Pick random points and tries to connect to current tree
- Optimize final path

Project Definition | Baseline Design | CPE Overview | CPE 1 Feasibility | CPE 2 Feasibility | **CPE 3 Feasibility** | Status & Future Work

CPE 3: Control Systems

- Motion Planning Algorithm Considerations
  - Joint sets achievable & collision-free
  - Motion(area swept out) is collision-free
  - Find area swept out by motion to check collision

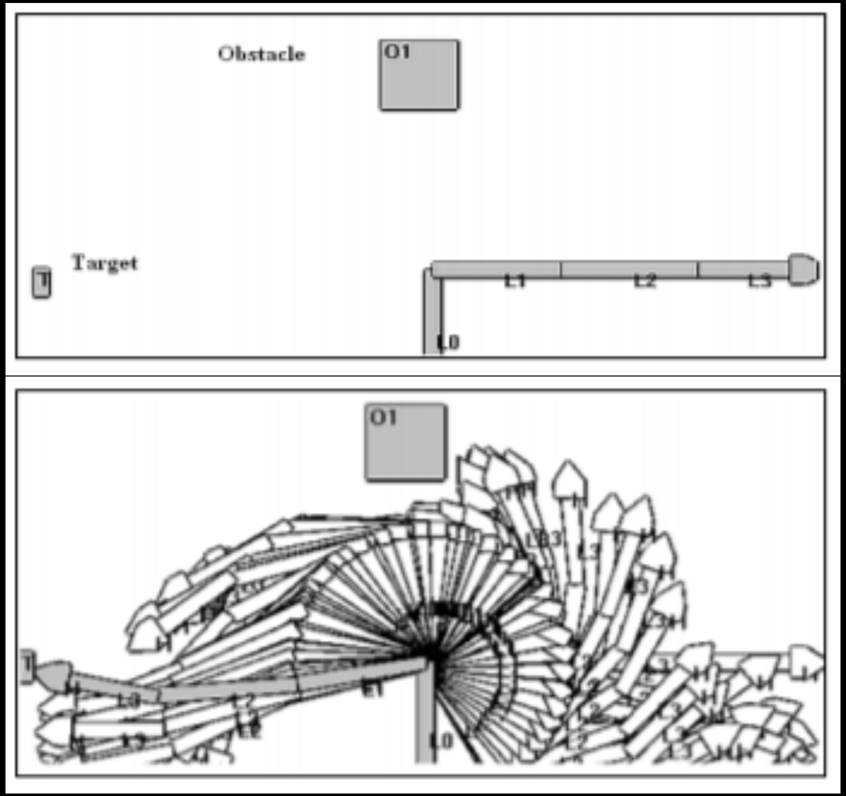


Fig. 15 Example of area swept out during motion

Project Definition | Baseline Design | CPE Overview | CPE 1 Feasibility | CPE 2 Feasibility | **CPE 3 Feasibility** | Status & Future Work

CPE 3: Control System

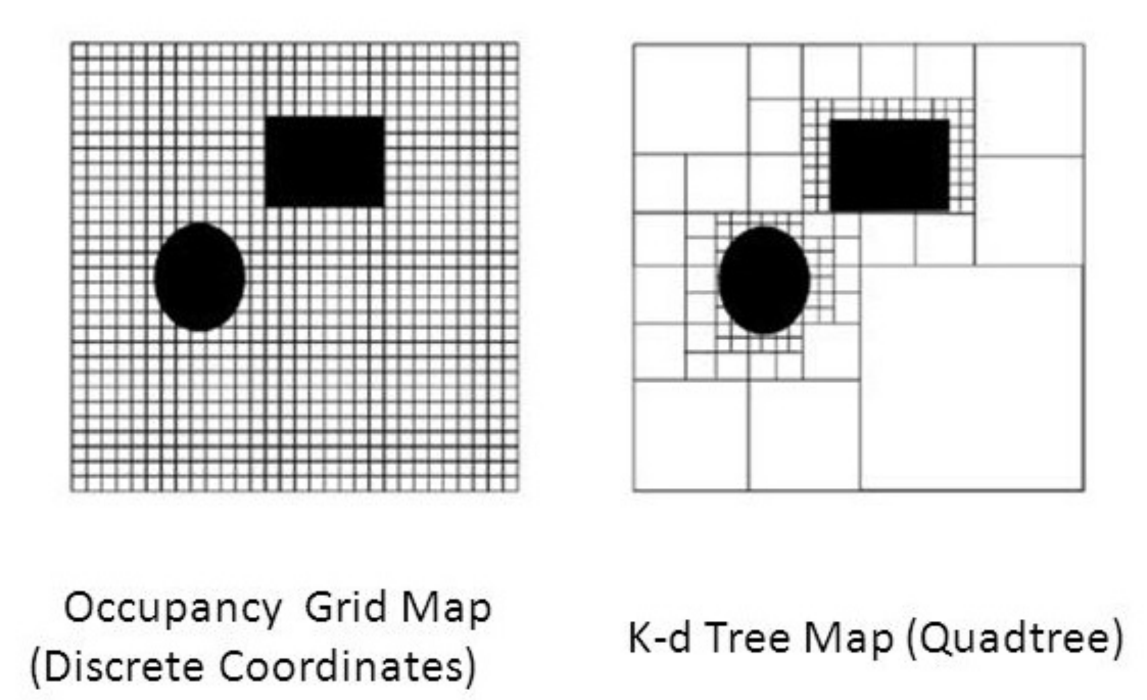


Fig. 16 K-D 2 Dimensional Tree [4]

- K-Dimensional Tree (KDT)
  - Generalization of a Binary Search Tree
  - Maps location to nearest neighbor
  - Used with pathfinding algorithms like A*

Status and Future Work

Glenda Alvarenga (Project Manager)

Feasibility Study Recap

- CPE 1: Feature Recognition
  - Addresses Objectives 1 and 2
- CPE 2: Mechanical Robotic Arm
  - Addresses Objectives 4
- CPE 3: Control System
  - Addresses Objective 3 and 4

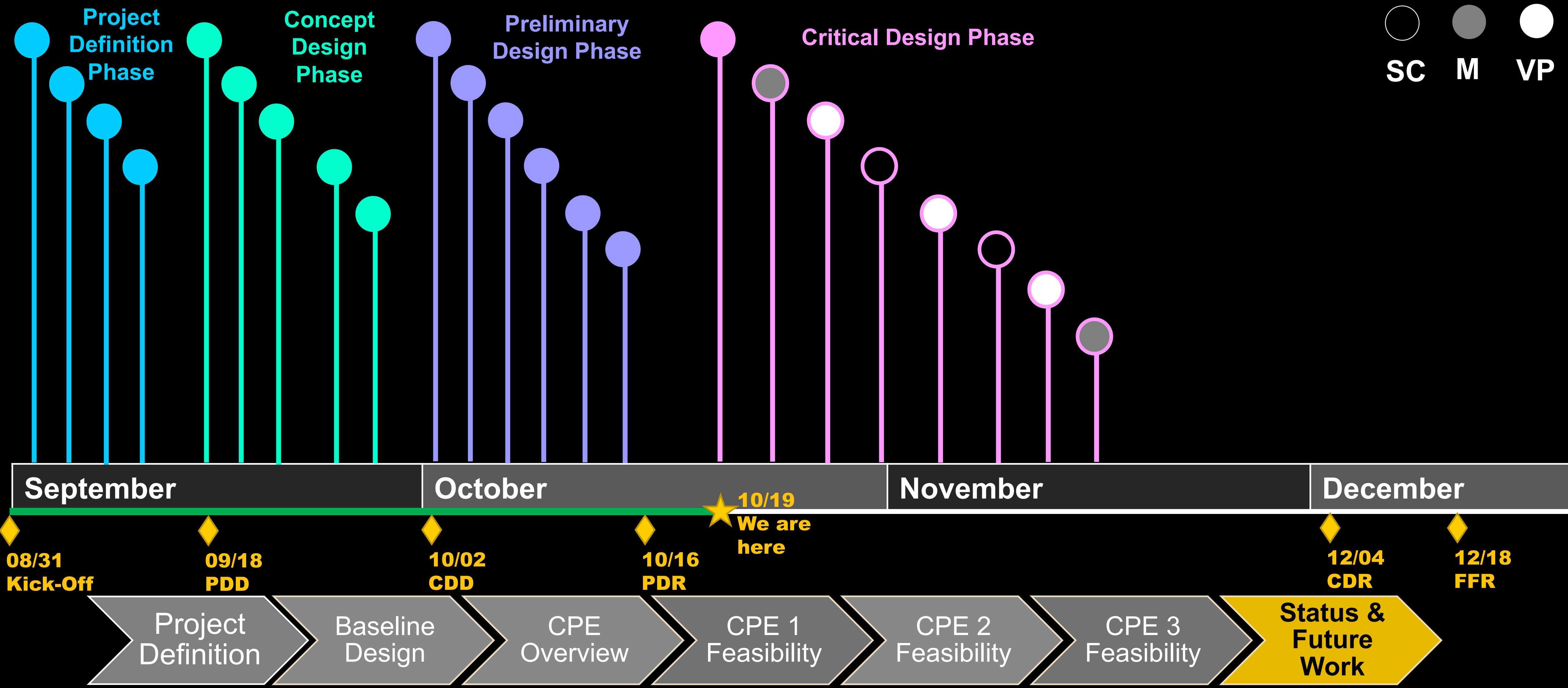
- CPE 1:
  - Feature recognition and machine learning are achievable via MATLAB
  - Initial testing conducted
- CPE 2:
  - Heritage hardware and load analysis confirms 6 DOF design
- CPE 3:
  - 6 DOF architecture has two viable approaches that have been used in robotics

Project Timeline

- KESSLER Unique Operations
  - Have access to hardware provided by customer
    - Kinect Imaging Hardware acquired
    - 5 DOF CrustCrawler Robotic Arm – may require repairs
    - May accelerate project development
  - Have access to a portion of heritage code
    - May not be fully applicable to KESSLER
    - Provides starting point

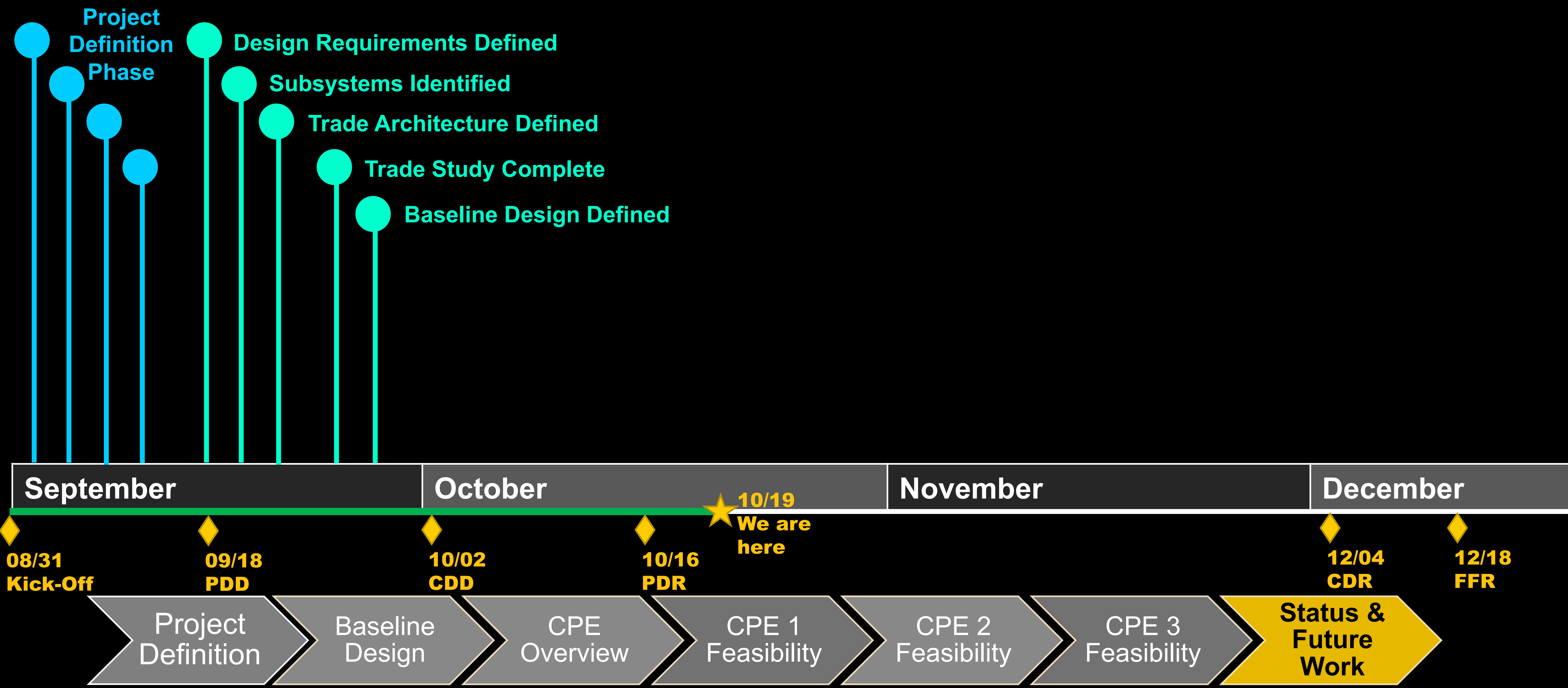
Project Definition → Baseline Design → CPE Overview → CPE 1 Feasibility → CPE 2 Feasibility → CPE 3 Feasibility → **Status & Future Work**

Fall Schedule

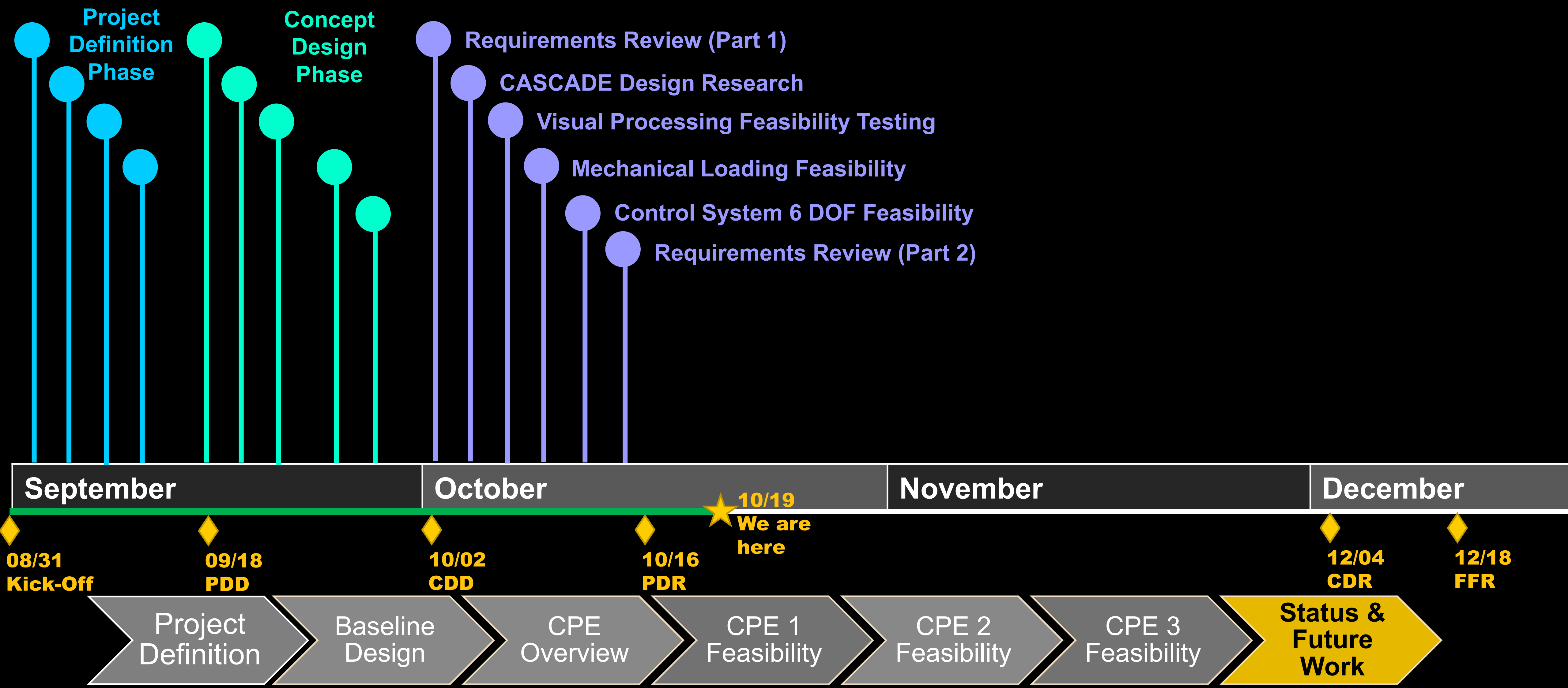


Fall Schedule

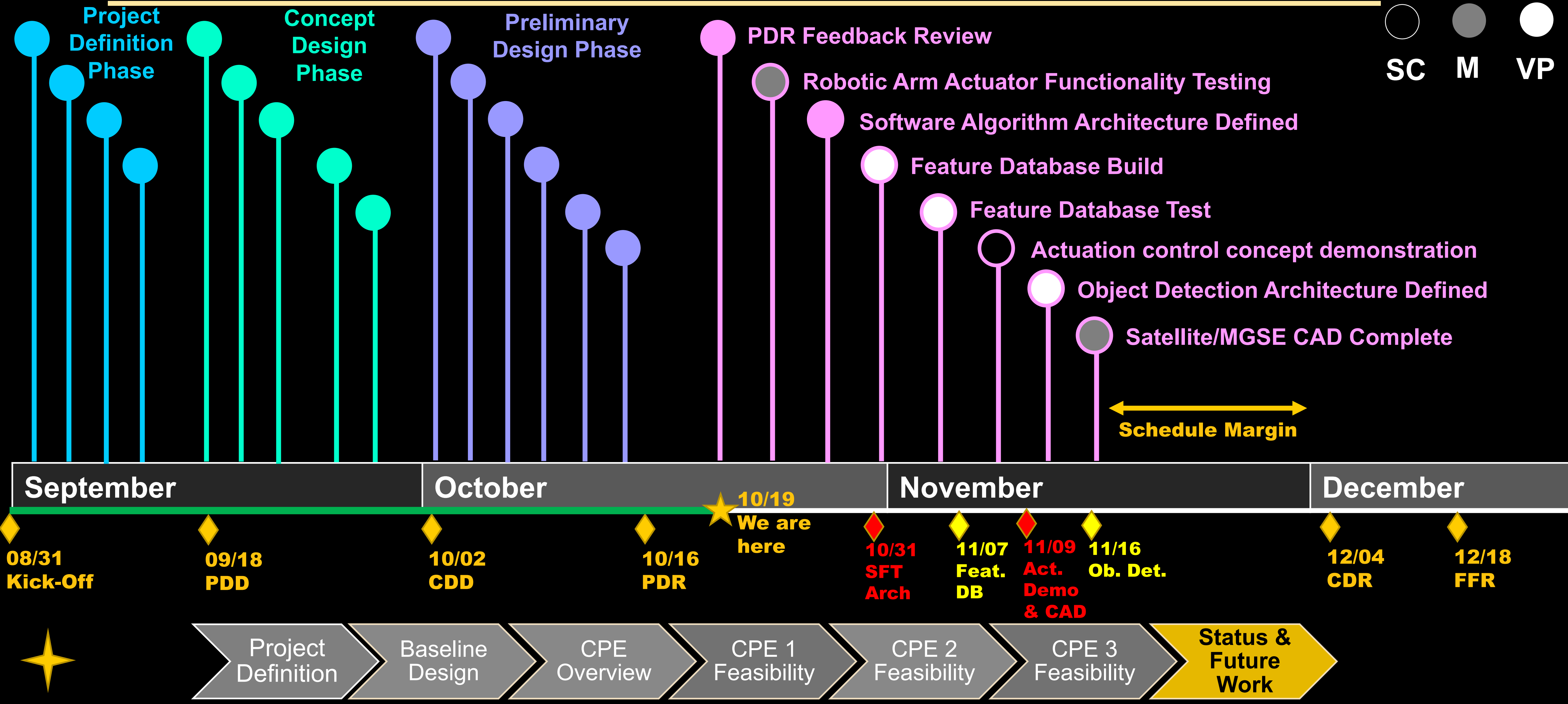
Fall Schedule



Fall Schedule



Fall Schedule



Project Operations Fall 2017

Efforts for Internal Design Reviews Leading Up to CDR

- Visual Processing
  - Develop ‘feature’ database and begin testing of feature recognition
  - Incorporate machine learning attributes to algorithm
- Mechanical
  - Develop simulations for 6 DOF motion
  - Develop end-to-end mechanical design (including MGSE, S/C, etc.)
- Software Control
  - Test robotic arm commands (electrical support required)
  - Develop path determination algorithm/simulation
- Ground & Test Support
  - Build feature prototypes to support visual processing

Spring 2018: Manufacturing, Test, & Integration

Project Definition → Baseline Design → CPE Overview → CPE 1 Feasibility → CPE 2 Feasibility → CPE 3 Feasibility → **Status & Future Work**

Financial Feasibility

Subsystem	Cost
Visual Processing	$116.00
Mechanical	$581.00
Software Control	$0.00
Ground & Test Support	$309.00
Misc	$0.00
% Margin	25%
Total Projected Cost	$1,257.50

- Starting Budget: $5,000.00
- Remaining Budget: $3,742.50
- Heritage hardware saves ~$800.00
- Worst Case estimates
- Percent Margin:
  - Will be refined post PDR

Organization

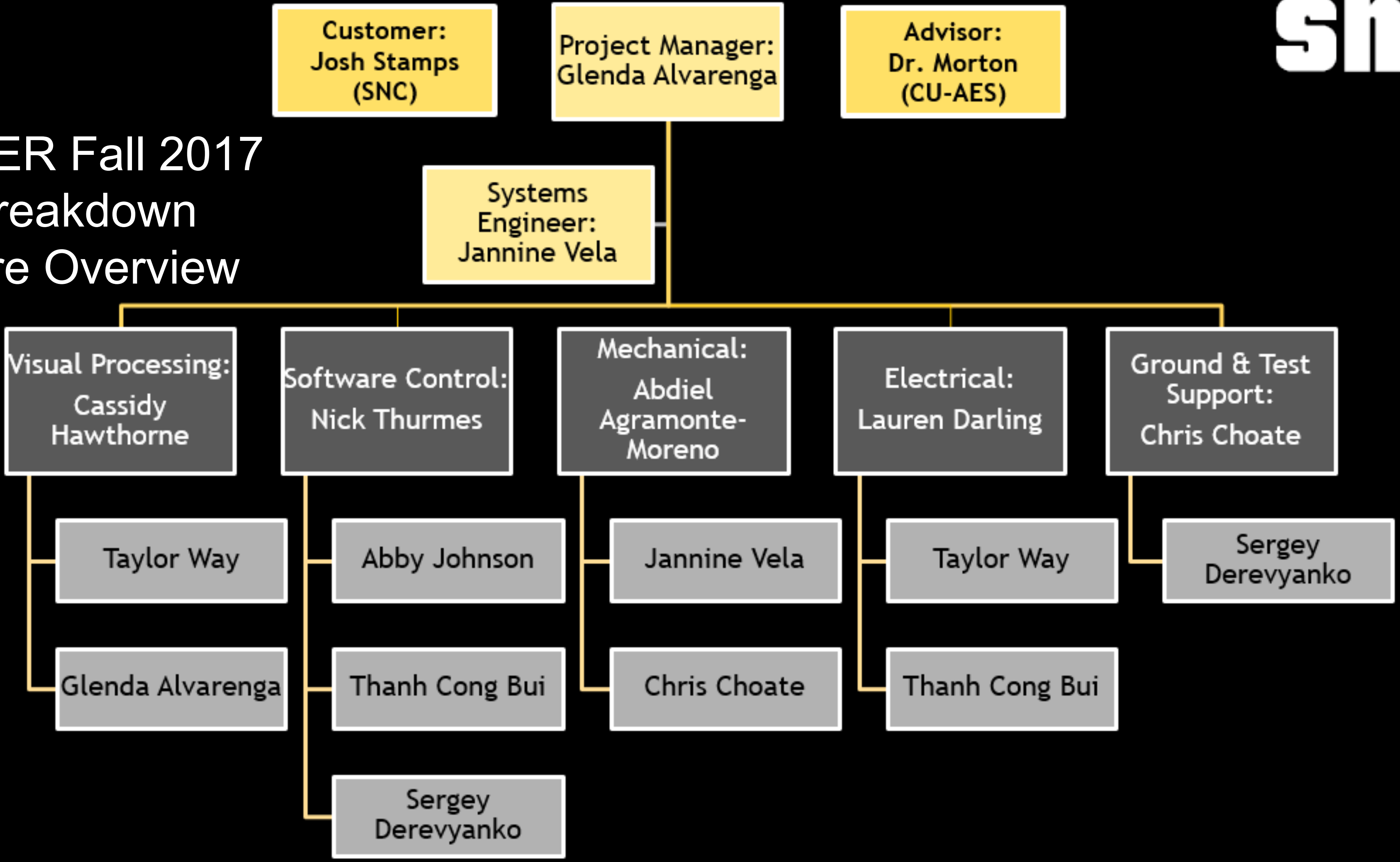
Course Defined

- Project Manager
  - Glenda Alvarenga
- Systems Engineer
  - Jannine Vela
- Financial Lead
  - Taylor Way
- Test & Safety Lead
  - Sergey Derevyanko
- Manufacturing Lead
  - Christopher Choate

KESSLER Defined

- Mechanical Design Lead
  - Abdiel Agramonte-Moreno
- Electrical Design Lead
  - Lauren Darling
- Image Processing Lead
  - Cassidy Hawthorne
- Software Control Lead
  - Nicholas Thurmes
- Software I&T Lead
  - Abigail Johnson
- Hardware I&T Lead
  - Thanh Cong Bui

KESSLER Fall 2017 Work Breakdown Structure Overview



Thank You!

Q/A Session

References

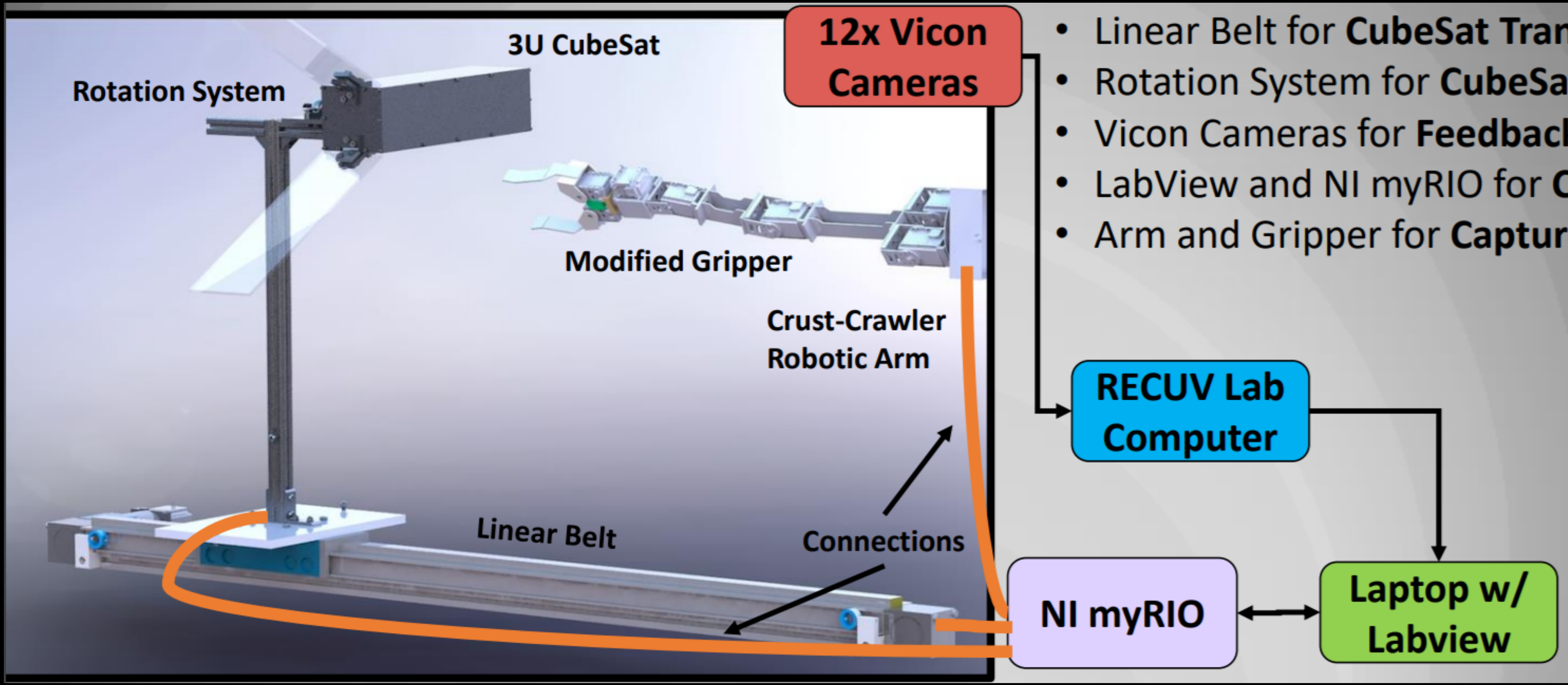
1. https://www.nasa.gov/mission_pages/station/news/orbital_debris.html
2. http://spaceflight101.com/falcon-9-orbcomm-flight2/orbcomm-g2-satellites-finish-in-orbit-checkouts-adjust-orbital-positions/
3. http://seradata.com/SSI/wp-content/uploads/2013/10/iridiumoriginal.jpg
4. http://geo.tuwien.ac.at/opals/html/ref_odm.html
5. http://www.crustcrawler.com/
6. http://pointclouds.org/documentation/tutorials/kdtree_search.php
7. http://msl.cs.uiuc.edu/rrt/
8. https://www.mathworks.com/products/computer-vision.html
9. https://developer.microsoft.com/en-us/windows/kinect/develop
10.

Back-Up Charts

Project Description – Back-Up

Project Definition

CASCADE Overview



- Linear Belt for **CubeSat Translation**
- Rotation System for **CubeSat Spin**
- Vicon Cameras for **Feedback**
- LabView and NI myRIO for **CDHS**
- Arm and Gripper for **Capture**

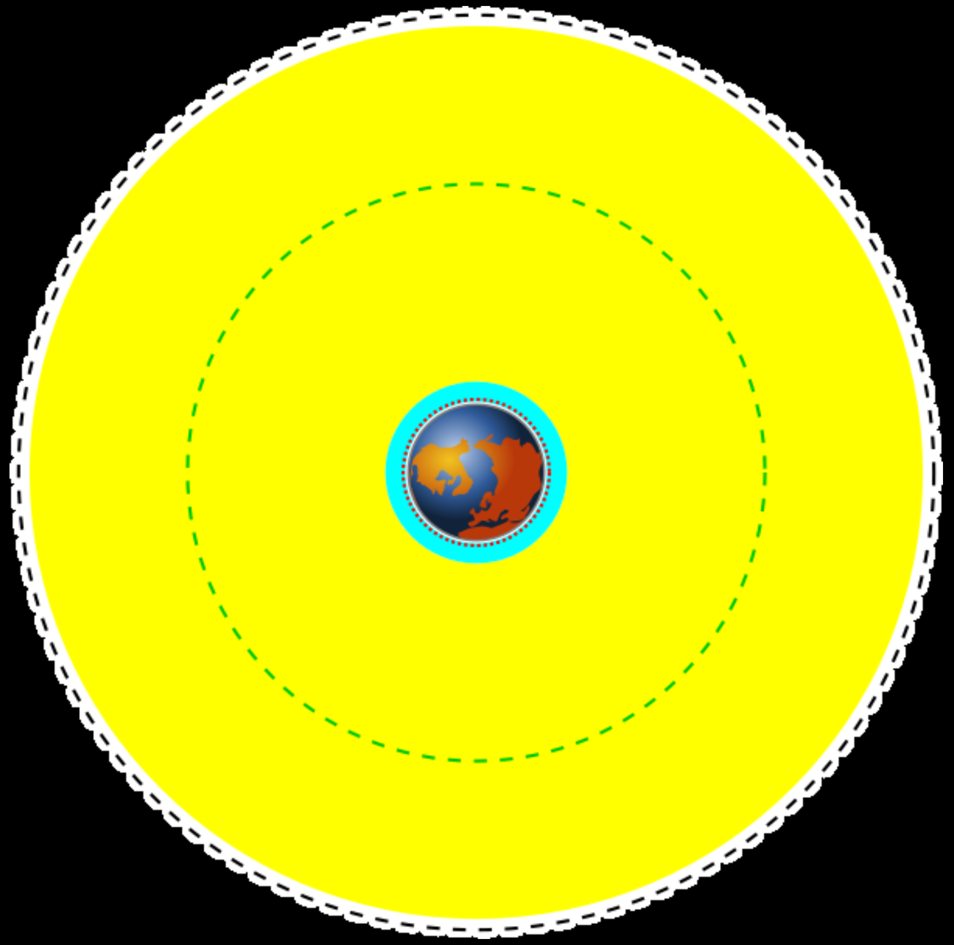
Project Definition

Project Assumptions

- Target object is in-front & within reach of the robotic arm; this entails that this scenario is valid if the target object and the chase vehicle are in the space orbit and in proximity to each other.
- Target object is stationary wrt the chase vehicle; this entails that this scenario is valid if the target object is 3-axis stabilized (or the chase vehicle has matched rotation at one axis if 2-axis stabilized)
- Chase vehicle operations (target and capture) occurs during Sun-soak in LEO

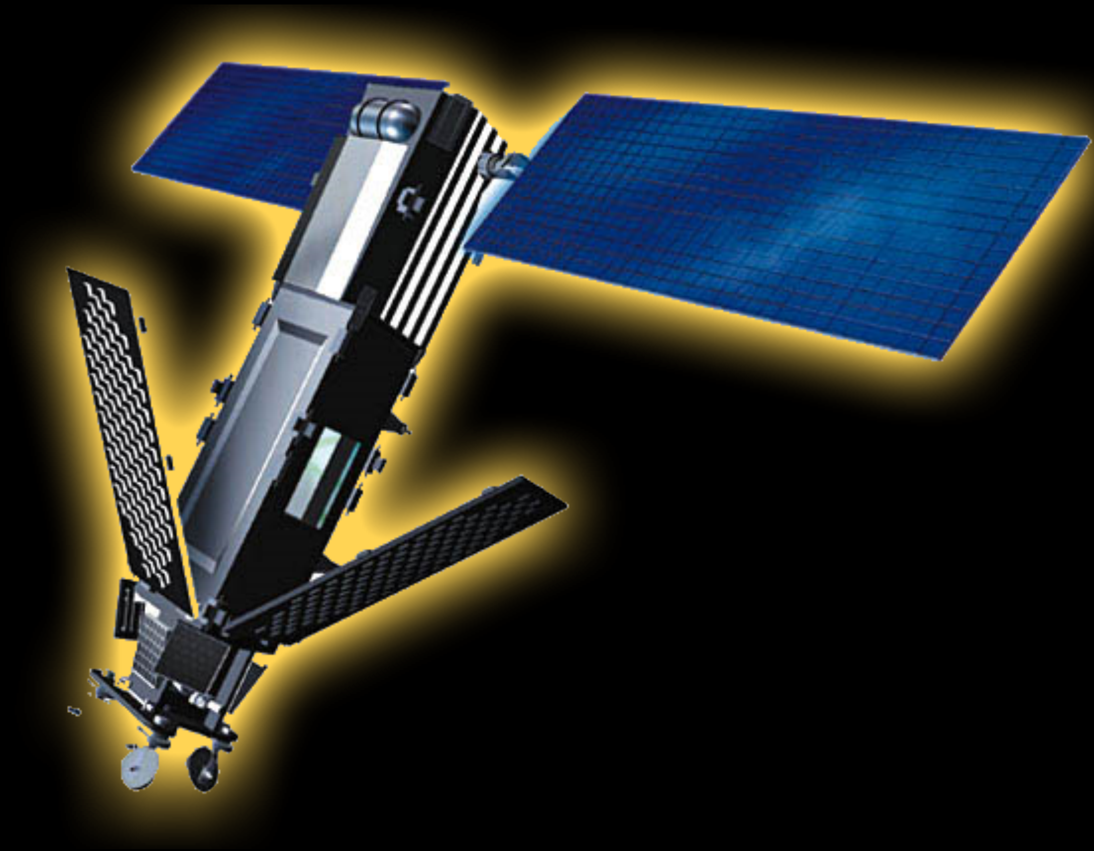
Baseline Design

- There are 1459 active satellites in orbit around the Earth
  - 804 satellites in Low Earth Orbit (150-2000 km)
  - 96 satellites in Medium Earth Orbit (2000-35785 km)
  - 518 satellites in Geosynchronous Earth Orbit (>35785km)
  - 41 satellites in Eccentric Orbits
- Of the 804 satellites in Low Earth Orbit, the most common series are:
  - The **Iridium series** with 67 Satellites
  - The ORBCOMM FM series with 40 satellites
  - The Yaogan series with 36 satellites
  - The Rodnik series with 21 satellites



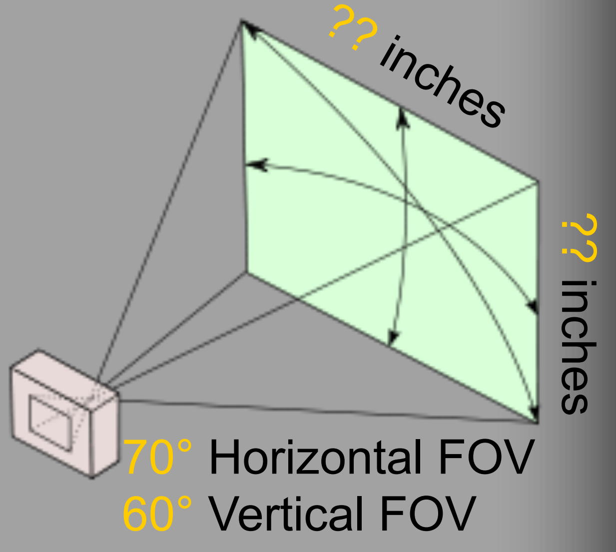
Baseline Design

- Iridium is the single most common type of satellite in Low Earth Orbit
- 8.33% of Satellites in Low Earth Orbit are of the Iridium Series
- Contains all 3 grapple features from grapple feature trade study
  - Solar Panel Joints
  - Bus Support Structure
  - Antenna
- Easiest satellite to find information about.

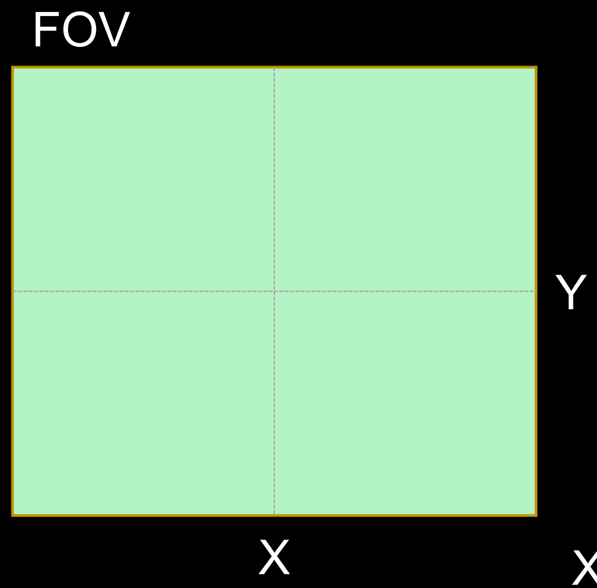


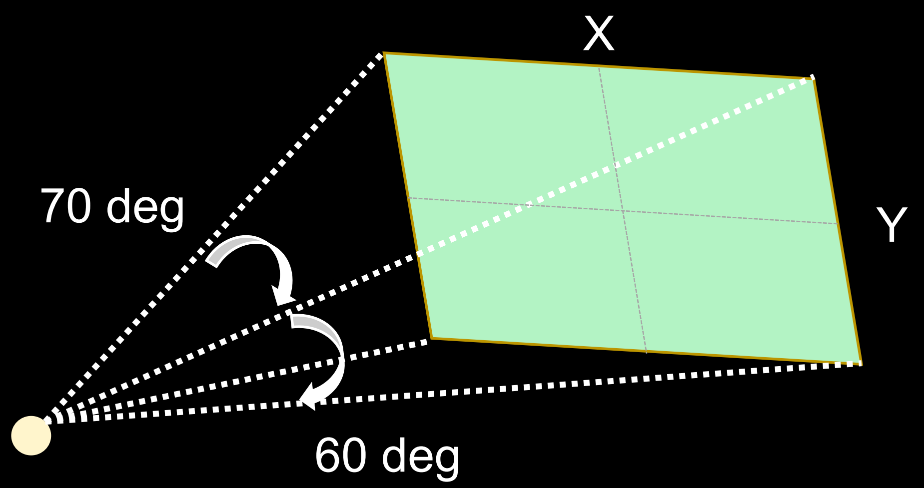
Baseline Design

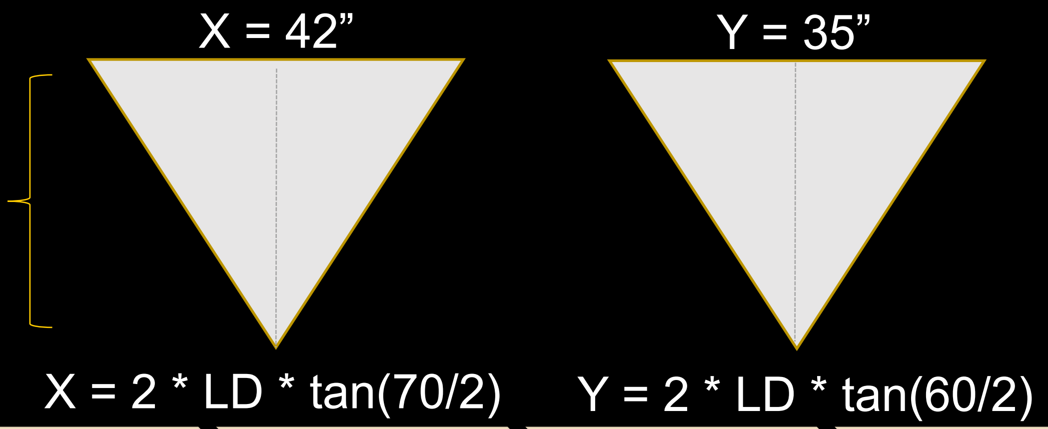
- One of Iridium Antenna's is 6'1" tall by 2'10" wide
- Using visual approximation, Iridium's Bus is about 12' 2" tall, from the top to the base
- Mockup is 30% scale, so it will be 43.8" tall
- Kinect has field-of-view (FOV) of 70° by 60°
- At maximum arm range of 31", the Kinect can see a 42" by 35" area
- Kinect will be able to see the entire bus of the Iridium model for distances greater than ??"



Levels of Success Metric Determination







Max Dim on Satellite = 1.86*2 m = 146.45”

Scale = 42/147 = .29 ~ 0.3 = 30%

Levels of Success

	Visual Processing		Control	Control & Robotic Arm
	Object Identification	**Processing**	**Path Prediction**	**Command Execution**
Level 1	Identify at least two surfaces on the satellite with varying depths in 3D space.	Identify the distance between the closest point of the satellite and the base of the robotic arm (± 4mm).	Define travel path of robotic arm for end-effector to arrive at closest point on the satellite.	Demonstrate end-effector can move to closest point while facing the parallel plane.
Level 2	Identify grappling feature recognition on target satellite.	Determine grappling feature location and orientation to within ±4mm & ± 5 deg.	Define travel path of robotic arm for end-effector to obtain as well as end-effector orientation required to arrive and grapple feature.	Grapple feature in parallel plane within ± 90 deg end-effector roll angle.
Level 3	Identify collision features on target satellite.	Define keep-out zone to within ± 4mm of collision feature surface, and select grappling feature of less collision risk.	Define constrained travel path of robotic arm for end-effector to obtain to arrive at grapple feature as well as end-effector orientation required.	Grapple feature in perpendicular plane (demonstrate additional ROM)

Project Definition

Functional Requirements

Req. ID	Requirement	Verification Method
F1	The **visual processing algorithm** shall identify the surface of a satellite in the primary camera's (RGB) field of view (FOV) and within the robotic arm's reach.	Measurement
F2	Control **algorithm** shall **define a path** to the location of a grappling feature.	Inspection
F3	Robotic arm shall **autonomously navigate and secure** at least one preselected grappling feature on the satellite with the motion of the robotic claw heading vector remaining normal to the robotic arm's mounting platform.	Measurement
F4	The KESSLER system shall have a **total mission time** no greater than **53 minutes**, based off the average LEO orbital period.	Timing Analysis
F5	KESSLER shall **attempt** a total of **3 end to end** process **operations** within the total mission time with an individual process operation time of 17 +/- 2 minutes	Demonstration/Test

Design Requirement

Req. ID	Requirement
D1.1	The visual system shall identify the location (x,y,z) and orientation (Euler angles) of an object in 3D space
D1.1.1	The system shall determine a body coordinate frame an origin of the target object
D1.1.2	The system shall identify feature edges to within TBD inches.
D1.2	The visual system shall be capable of communicating with the control system.

Design Requirement

Req. ID	Requirement
D2.1	The end-effector orientation and locations with respect to the grappling feature shall be determined in 3D space to within +-13mm and +- 5deg.
D2.1.1	Algorithm shall transform and image data (TBR) to body coordinate frame.

Design Requirement

Req. ID	Requirement
D3.1	The robotic arm shall be capable of receiving commands from the control system.
D3.1.1	The robotic arm shall be capable of initiating operations based off commands relayed from the CPU.
D3.1.2	The robotic arm shall terminate operation upon command from the CPU.
D3.2	The grappling feature shall be representative of common features found on the Iridium Constellation Satellite form factor.
D3.2.1	The Iridium Constellation Satellite shall be scaled by 0.30.
D3.3	Robotic arm shall move in path outlined by positioning algorithm
D3.4	The end effector shall be able to capture objects of (F2.2.a) size
D3.4.1	End effector shall have a fully deployed range of 9 inches
D3.4.2	End effector shall secure object without compromising structure of grappled object.

Design Requirement

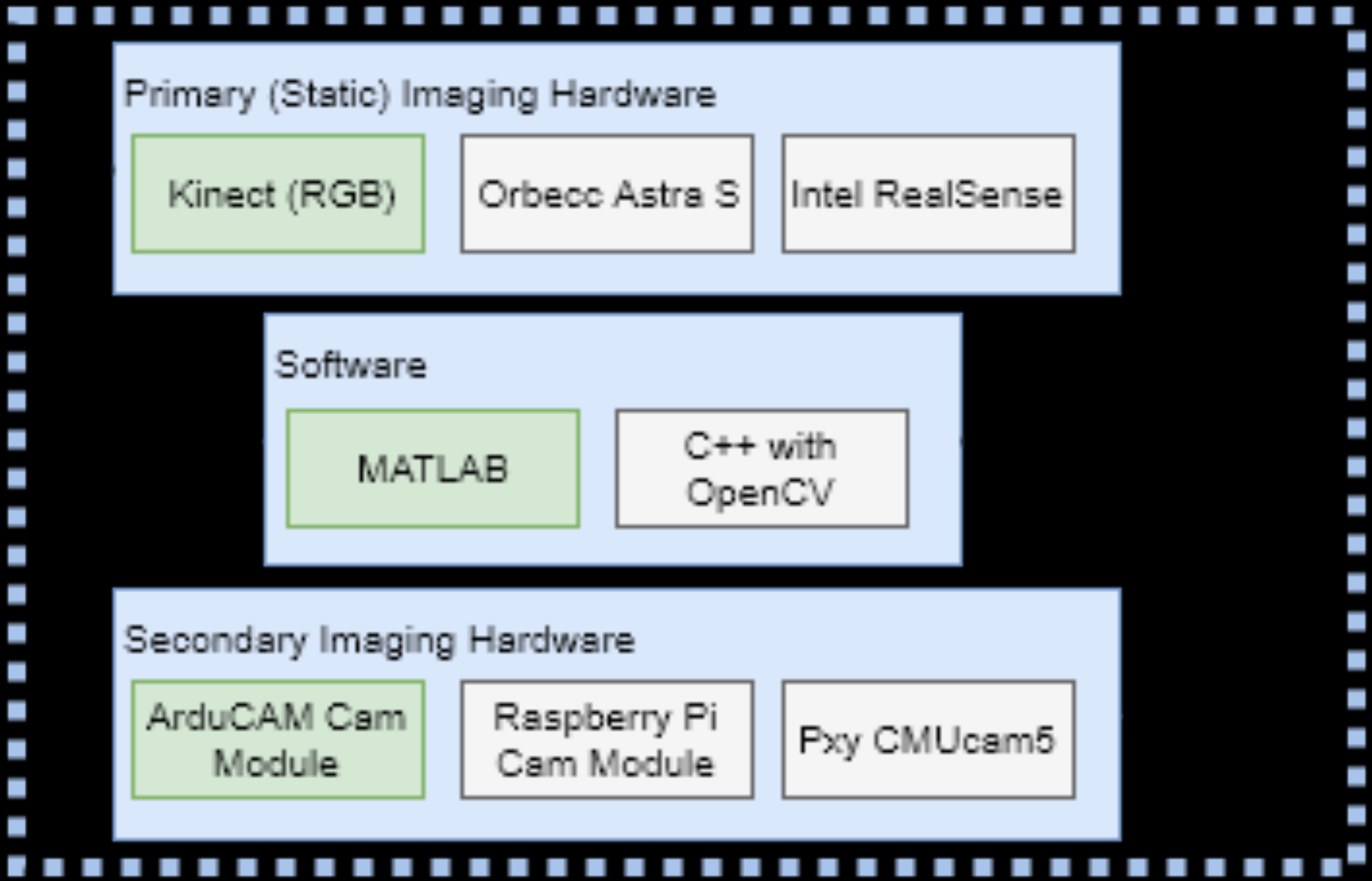
Req. ID	Requirement
D4.1	Image identification, grappling maneuver, and capture will take no more than 17 +/- 2 minutes to be executed.

Design Requirement

Req. ID	Requirement
D5.1	KESSLER shall **complete** no less than **2 end to end** process **operations** within the total mission time.

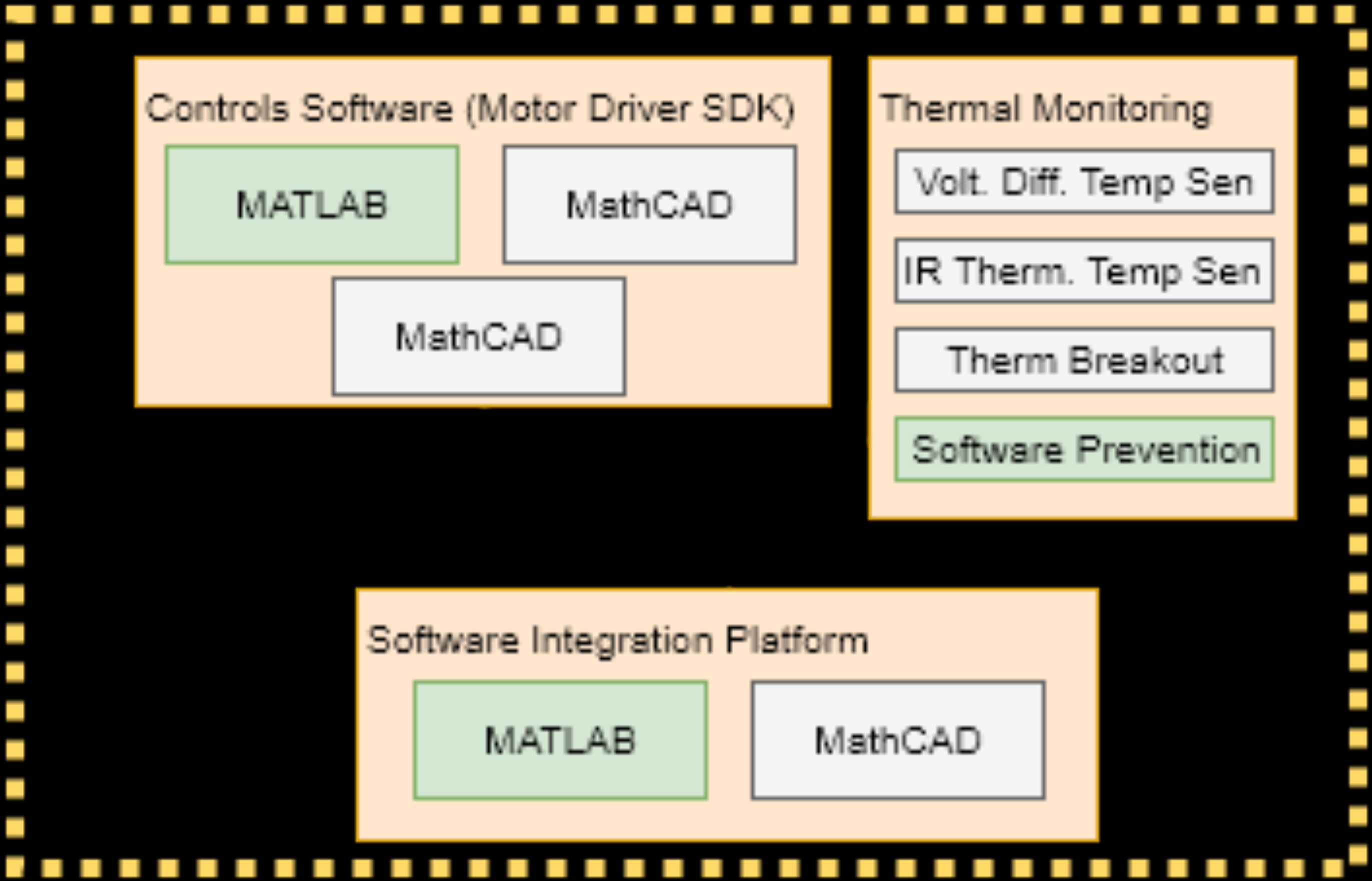
Baseline Design

Visual Processing



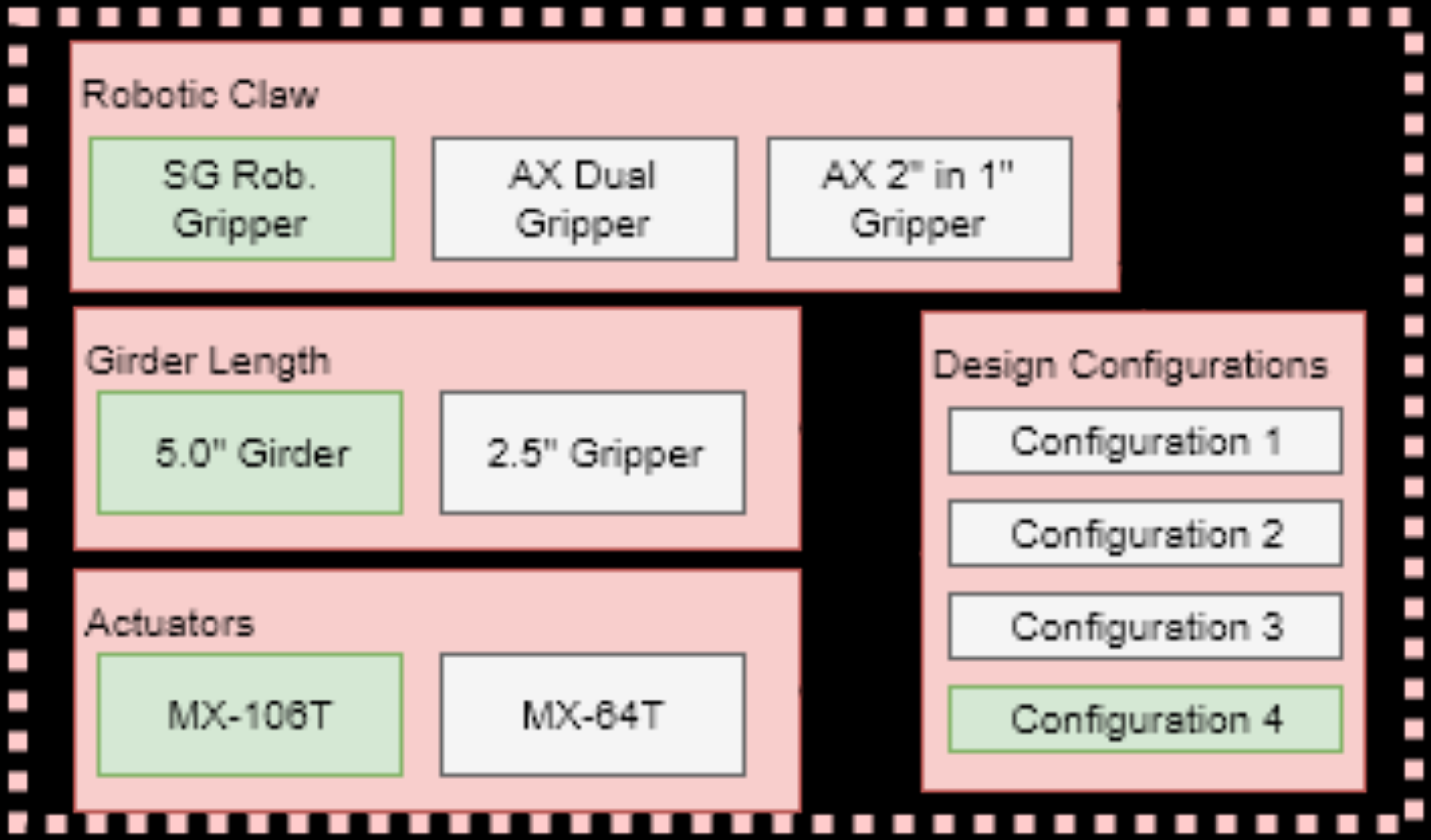
Baseline Design

Software Control



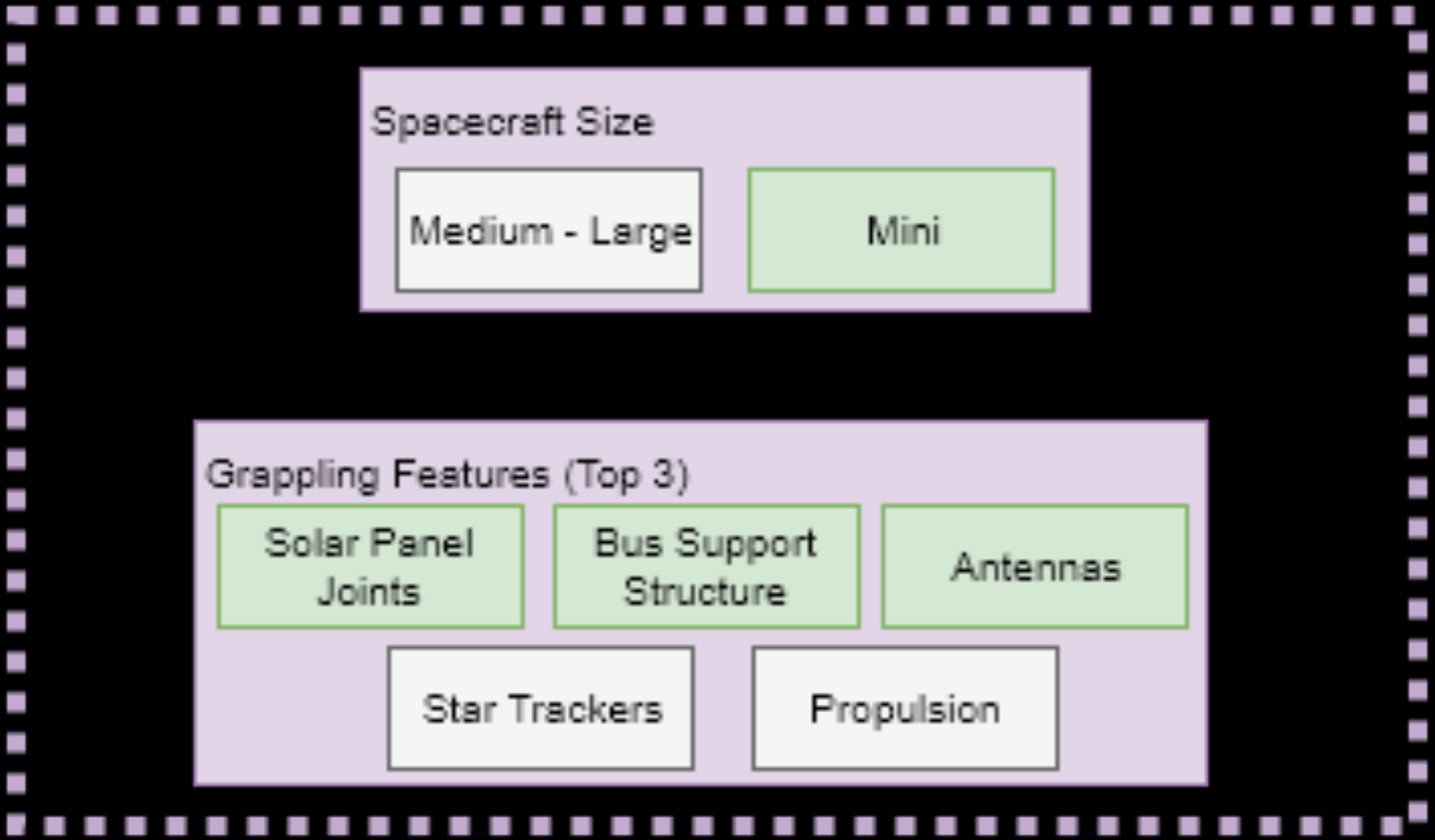
Baseline Design

Robotic Arm



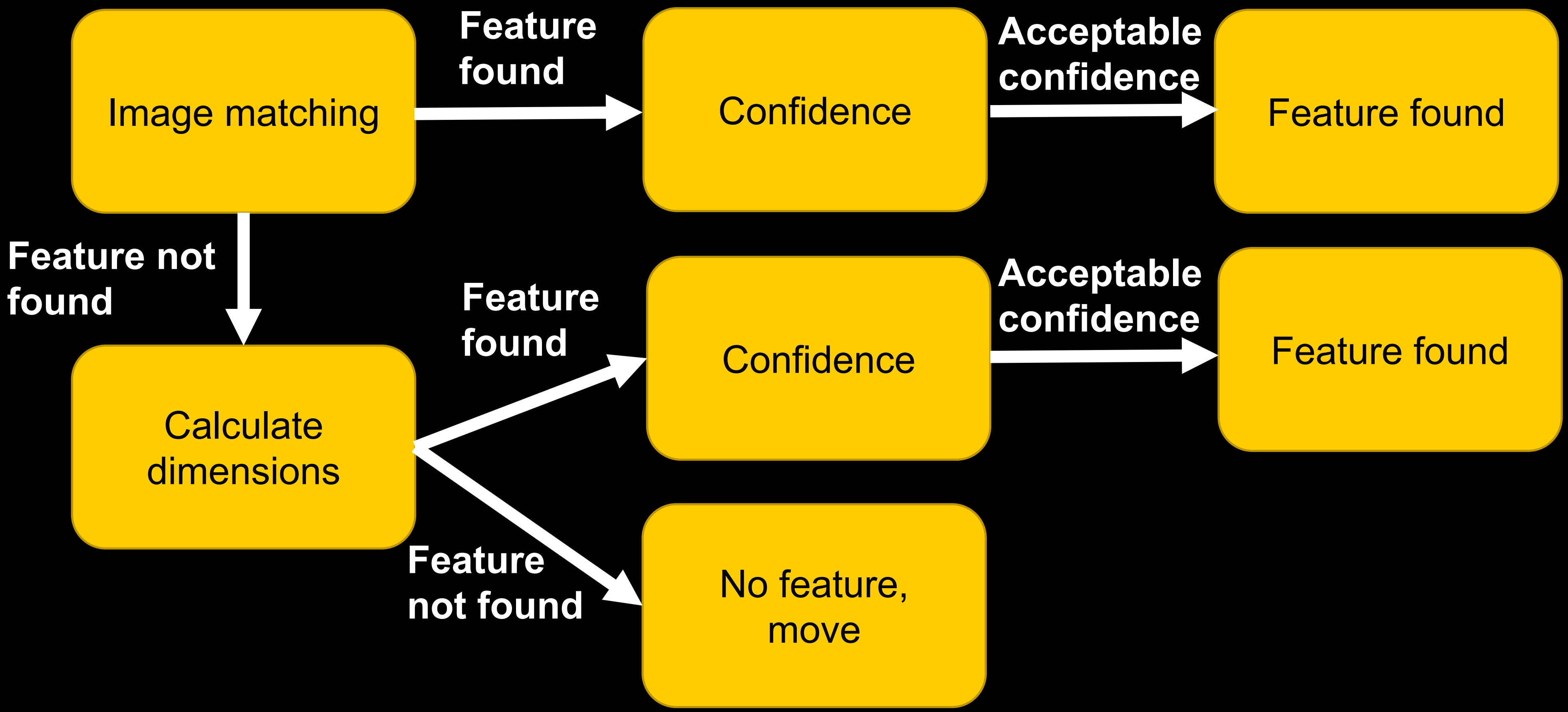
Baseline Design

Ground & Test Support



CPE 1: Back-Up

CPE 1: Feature Recognition



CPE 1: Feature Recognition

Imaging Hardware Trade Study

	Weight (%)	Microsoft Kinect	Orbbec Astra S	Intel RealSense SR300
Cost	10	10	9	10
User Documentation	30	10	7	4
Picture Quality	30	7	7	10
Supporting Software	30	10	10	1
Weighted Total	100	9.1	8.1	5.5

CPE 1: Feature Recognition

Secondary Imaging Hardware Trade Study

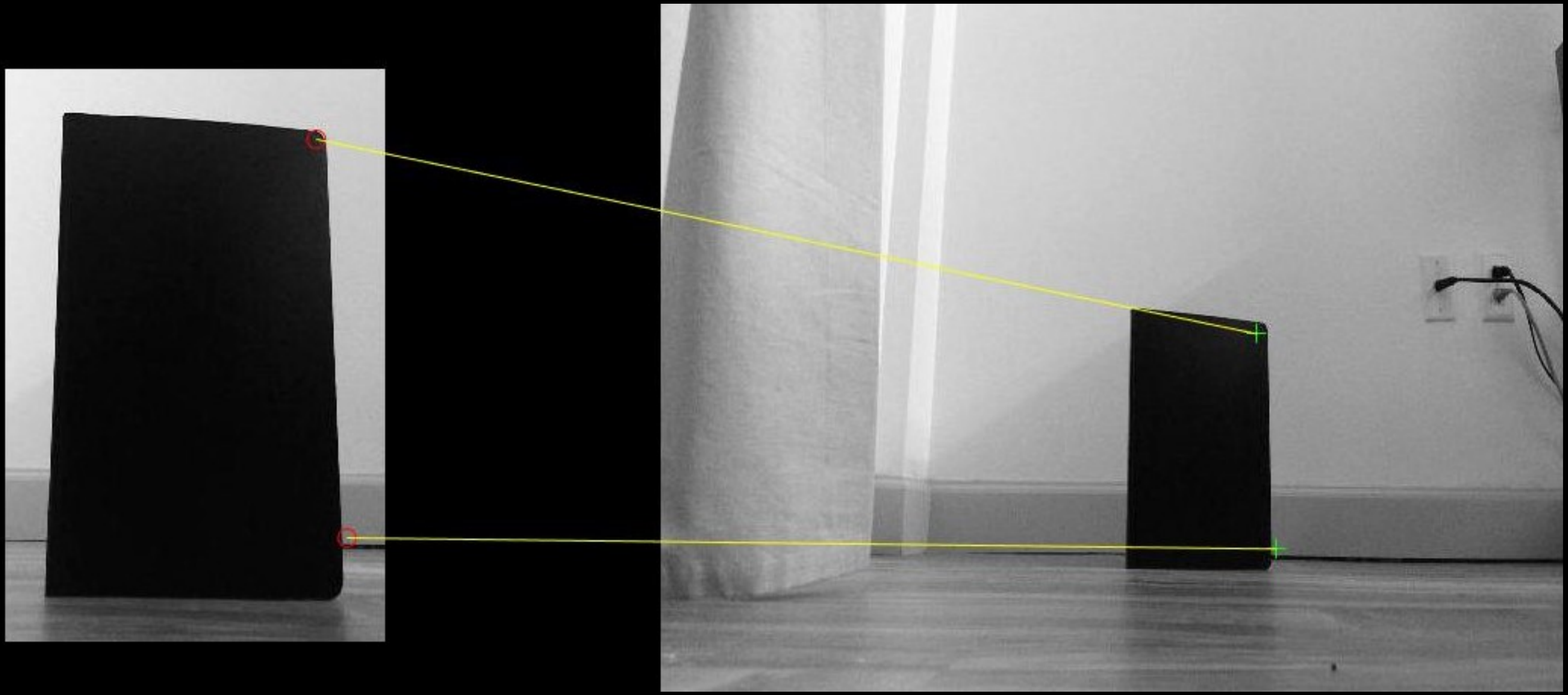
	Weight (%)	ArduCAM 3CMOS OV7670	Raspberry Pi Cam Module V2	CMOS Camera Module	ArduCAM Mini 2MP OV2640	Pixy CMUcam5
Resolution	40	3	5	5	5	5
Microcontroller Compatibility	20	5	5	1	5	5
Dimension	10	2	4	2	5	1
Weight	10	5	5	3	5	3
Cost	5	5	5	3	5	3
Power	15	4	-	2	4	3
Weighted Total	100	3.75	4.15	3.15	4.85	4

CPE 1: Feature Recognition

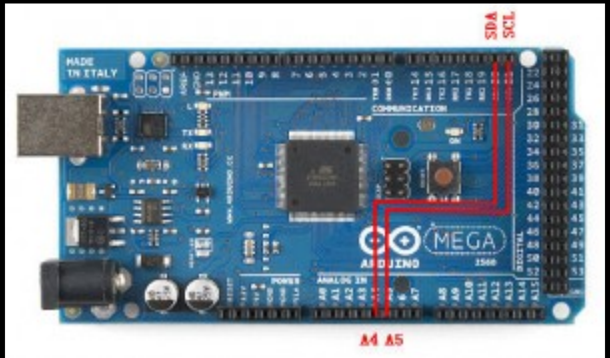
Imaging Software Trade Study

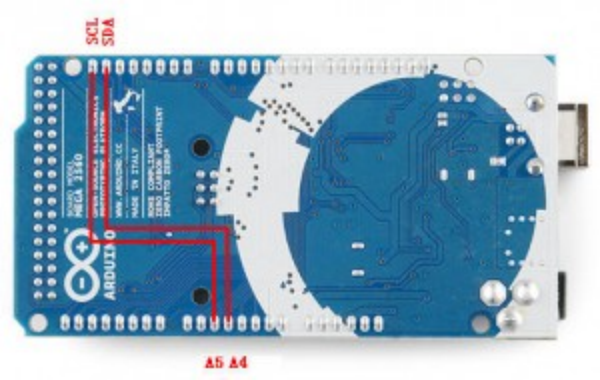
	Weight (%)	MATLAB	C++ with OpenCV
Documentation	20	10	4
Visualization/Debugging Tools	30	10	3
Availability of Library Functions/Toolboxes	30	10	8
Runtime	5	3	9
Difficulty of Use	15	7	3
Weighted Total	100	9.2	5

CPE 1: Feature Recognition



Electrical Secondary
Visual Sensor Interface





Feasibility of usable sensor data comes from acquiring hub schematic for sensor relay to code base.

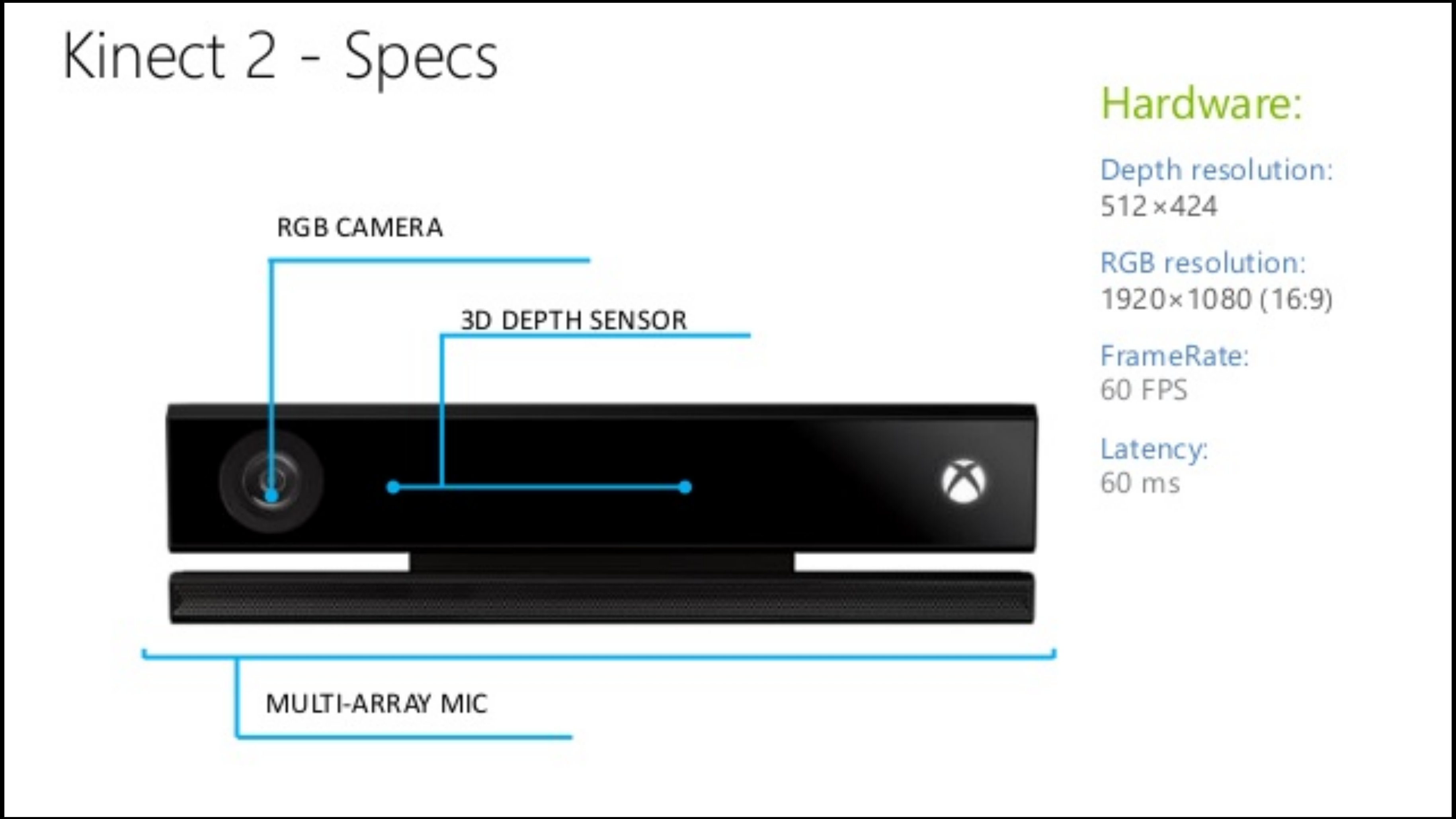
Must read information from sensors, interpret the data, and send through solution algorithm. Feasibly via USB.

Feasibility: ArduCAM shield use hardware I2C interface, which is 20(SDA),21(SCL) on MEGA board. ArduCAM shield use hardware SPI interface for SD/TF read and write, which is 10(SS),11(MOSI),12(MISO),13(SCK) on UNO board. But on MEGA2560 board they are 53(SS),51(MOSI),50(MISO),52(SCK). When ArduCAM shield used on MEGA board, user should use software SPI, changes should be made as follows:

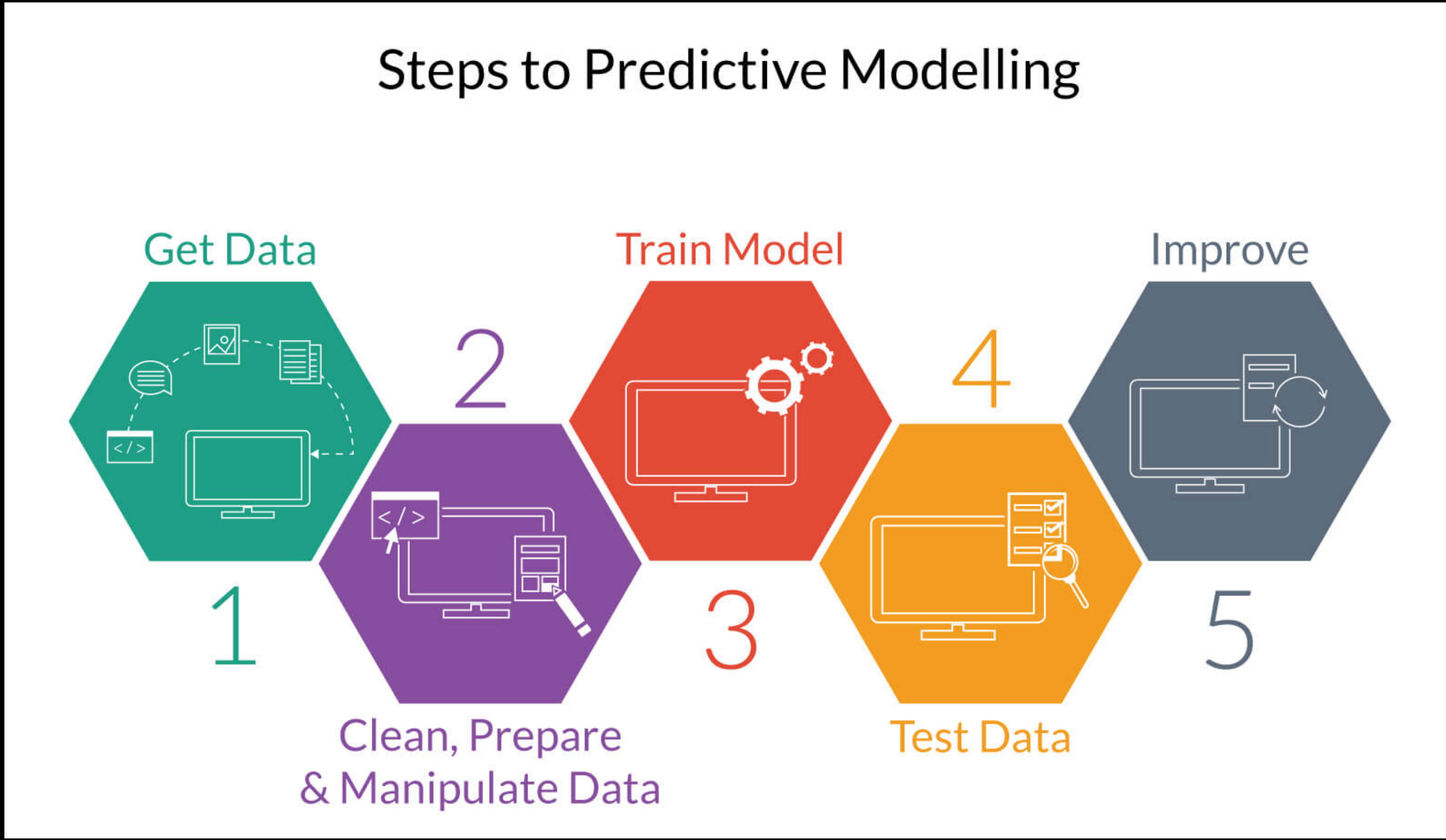
http://www.arducam.com/how-to-connect-arducam-shield-to-mega-2560/#more-509

http://www.arducam.com/camera-modules/2mp-ov2640/

CPE 1: Feature Recognition



CPE 1: Feature Recognition



CPE 2: Back-Up

CPE 2: Mechanical Robotic Arm

5 Degrees of Freedom	6 Degrees of Freedom	7 Degrees of Freedom
Current CASCADE design	Possible KESSLER addition	Possible KESSLER addition
Restricted to grappling along straight path	Restricted to grappling in only one configuration	No restrictions, redundant system, can grapple in multiple configurations
No arm modifications, no cost	Some arm modifications, low cost	Many arm modifications, high cost

- Inverse kinematics require desired position (x, y, z) and orientation (χ , ψ , ζ)
- 6 variables, 6 unknowns: 6 DOF needed
- 7 degrees of freedom creates additional solutions, but cost and weight budget make desired design 6 degrees of freedom

Trade Study -- Robotic Claw

	Weighting	SG Robotic Gripper	AX Dual Robotic Gripper	AX 2” in 1” Robotic Gripper
Cost	10%	1	3	2
User Documentation	20%	3	5	3
Extension Range	30%	3	5	4
Motor Interface	20%	5	3	1
Contact Surface Area	20%	3	4	3
Weighted Total	100%	3.90	4.35	3.00

Trade Study -- Arm Girder Length

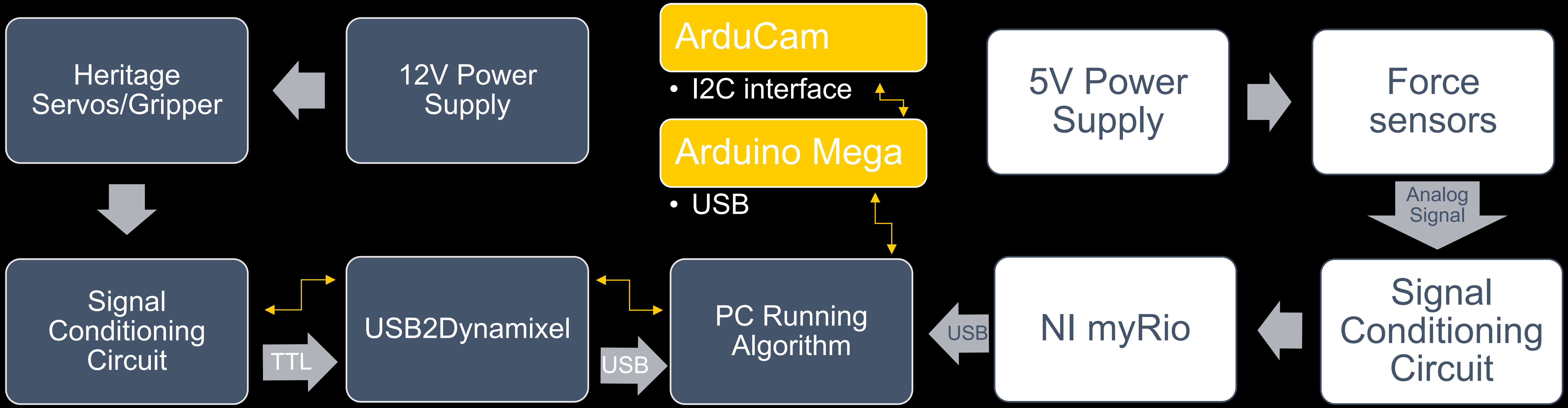
	Weighting	2.5” Girder	5.0” Girder
Cost	30%	5	5
Range	40%	4	4
Weight	20%	5	5
Weighted Total	100%	4.1	4.2

Trade Study -- Actuator

	Weighting	MX-64T Actuator	MX-106T Acuator
Cost	20%	5	4
Performance	25%	4	5
Weight	30%	3	5
Gear Ratio	25%	4	3
Weighted Total	100%	3.9	4.2

Inherited hardware Diagraming

Feasibility analysis for motor reuse determined from signal input/output continuity testing. Evaluation of each motor proved electrically stable with no shorts. Motors were evaluated for proof of feasibility in CASCADE.

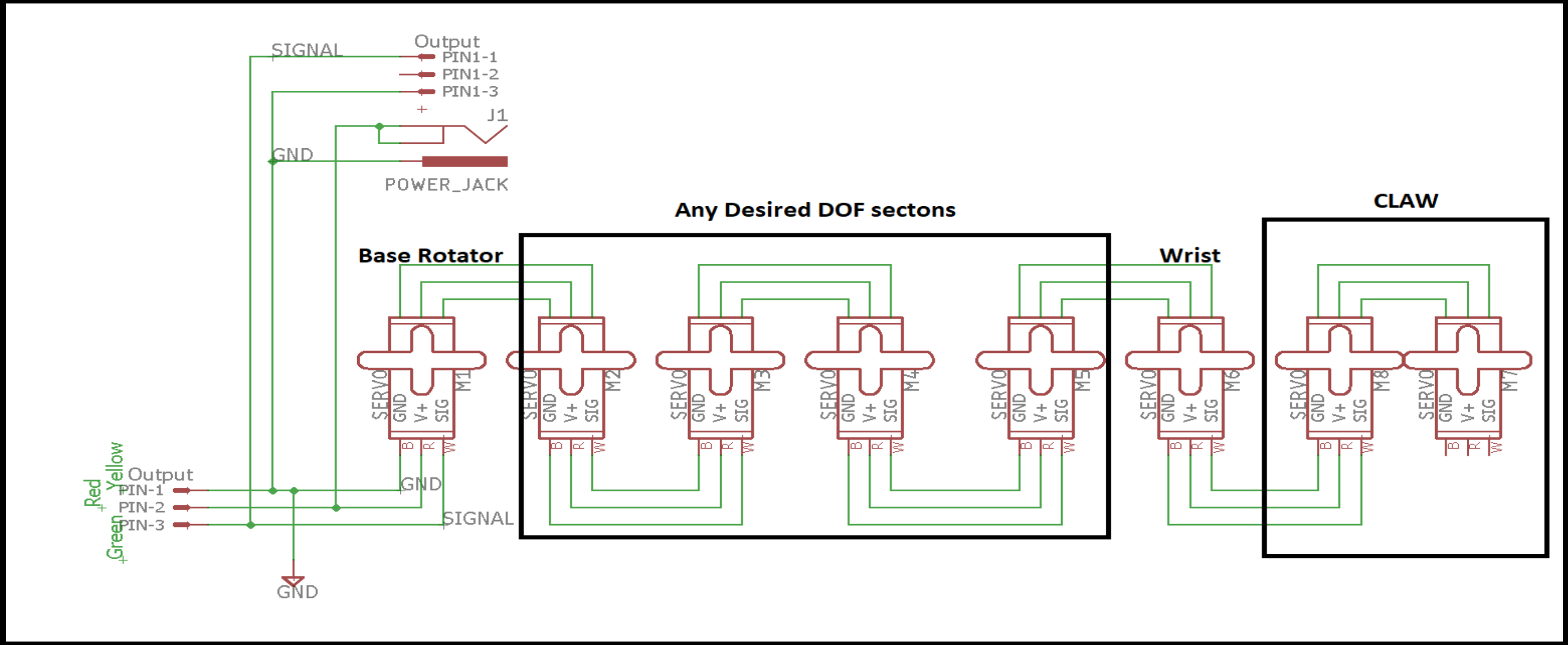


Servo Stall Torque: 226 oz-in

Additional Electrical Motor Interface

Addition of degree of freedom motor can be daisy-chained into the currently existing system.

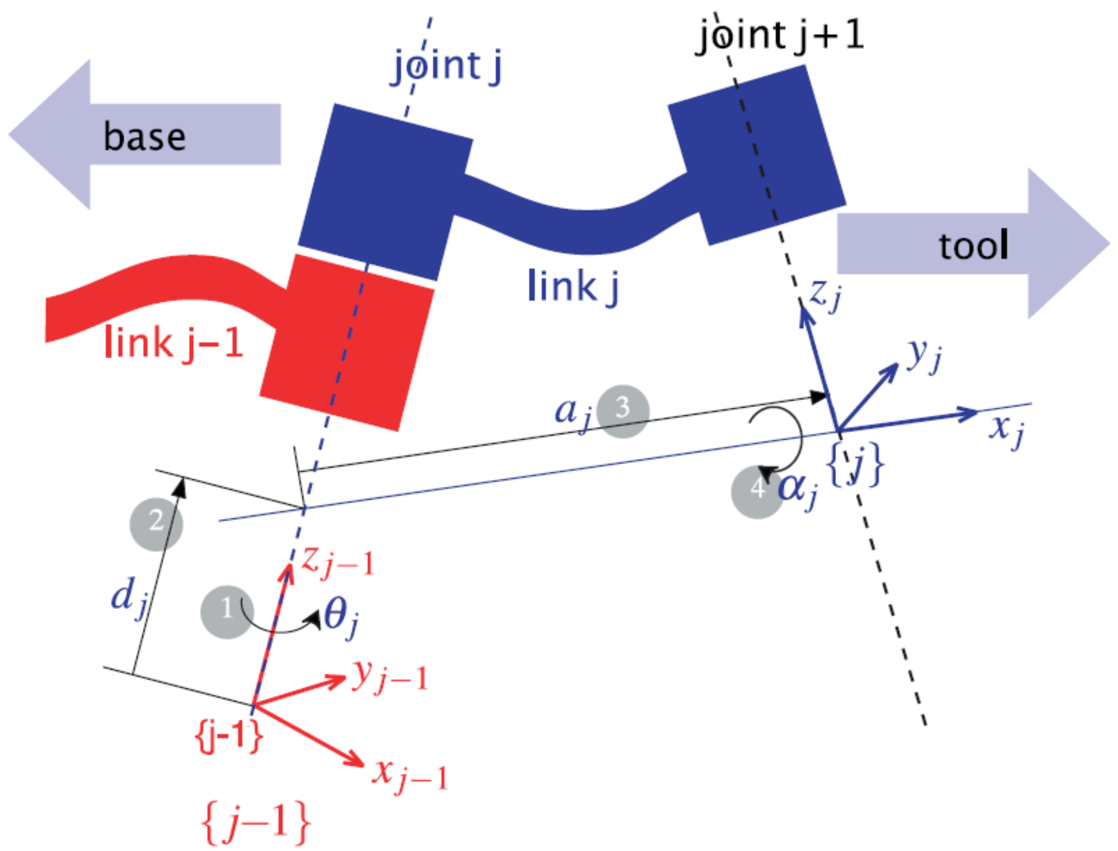
5 DOF Robotic Arm



CPE 3 - Backup

CPE 3: Control Systems

Denavit–Hartenberg parameters



Joint angle	θ_j	The angle between the x_{j-1} and x_j axes about the z_{j-1} axis	Revolute joint variable
Link offset	d_j	The distance from the origin of frame $j-1$ to the x_j axis along the z_{j-1} axis	Prismatic joint variable
Link Length	a_j	The distance between the z_{j-1} and z_j axes along the x_j axis	constant
Link Twist	α_j	The angle from the z_{j-1} axis to the z_j axis about the x_j axis	constant
Joint type	σ_j	$\sigma=0$ for a revolute joint, $\sigma=1$ for a prismatic joint	constant

CPE 3: Control Systems

Servo Command Protocol

START	ALL	T.LEN	SYNC	INSTRUCTION	LENGTH	ID 1	P 1	P 2	ID N	P 1	P 2	C.SUM

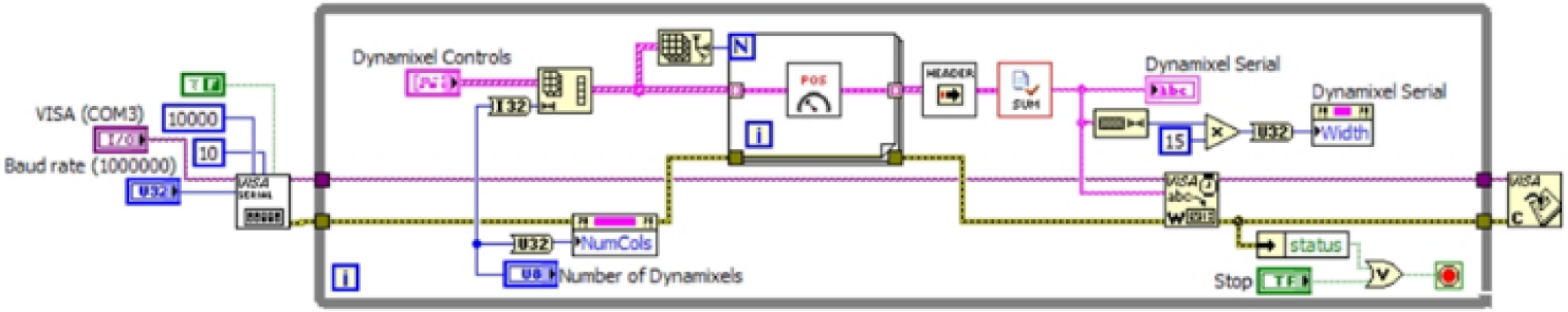
START: (Hex 0XFF 0XFF) The double FF initializes communication between the COM and the Dynamixels.
ALL: Broadcast ID (Hex-0XFE
) to all Dynamixels, disables return of status packets
T.LEN: Total Length, Uses the formula (NxL)+N+4, where N is number of Dynamixels, L is the length of the Control Table Bytes, and 4 is the length of the header bytes to be used by the checksum (Manual uses the formula (L+1)xN+4 this achieves the same solution as the formula used).
SYNC: The Sync Write (Hex 0x83) defines the command being used
INSTRUCTION: Starting Byte Address from Control Table (Goal Position (L) 0X1E).
LENGTH: L is the length of the Control Table bytes used
ID P1 P2 IDN P1 P2: Dynamixel ID and Position Control (Position sub VI)
C.SUM: (CheckSUM sub VI)

http://www.ni.com/white-paper/12557/en/

- Commands for a write instruction.
- SYNC and INSTRUCTION bits can be changed for different servo commands
- Check Sum transmitted at the end of each write command for error checking.

CPE 3: Control Systems

Servo Commands-Labview

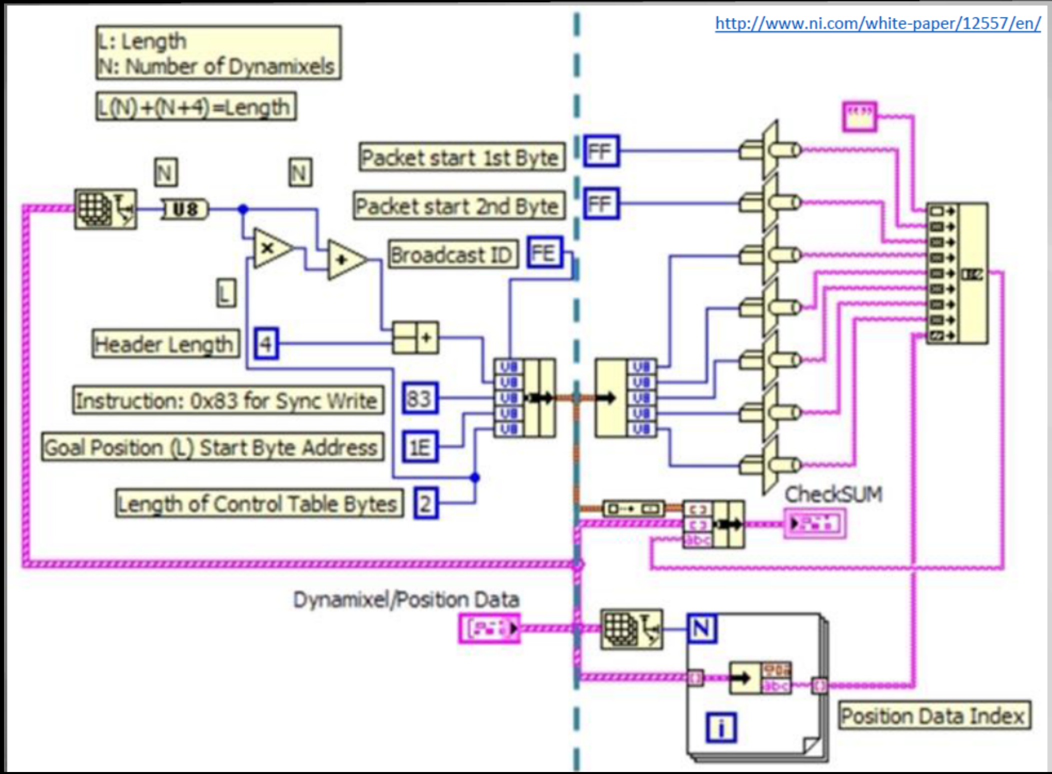


http://www.ni.com/white-paper/12557/en/

- Top Level LabView for writing to servos

CPE 3: Control Systems

Servo Commands-Labview

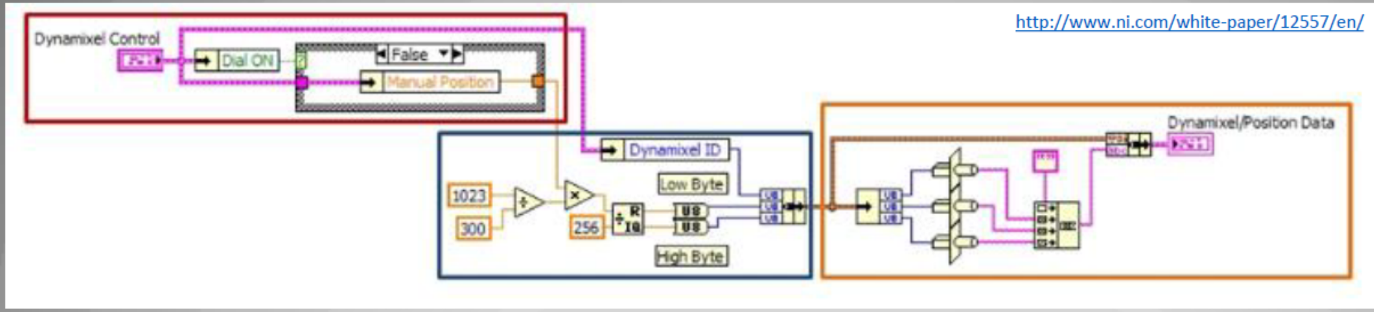


- Header VI
- Left of dotted line: SYNC and INSTRUCTION parameters set
- Right of dotted line: Data is bundled and concatenated into final package to be sent to the servos.

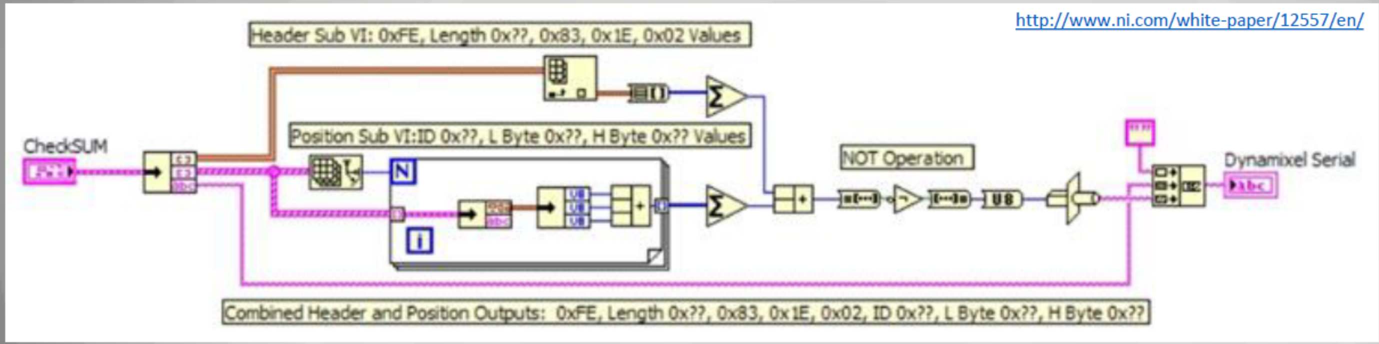
Project Definition | Baseline Design | CPE Overview | CPE 1 Feasibility | CPE 2 Feasibility | **CPE 3 Feasibility** | Status & Future Work

CPE 3: Control Systems

Servo Commands-Labview



- Position Data VI for forming desired position into bytes, ours will be for velocities instead.



- Check SUM VI for framing and bit errors.

Project Definition | Baseline Design | CPE Overview | CPE 1 Feasibility | CPE 2 Feasibility | **CPE 3 Feasibility** | Status & Future Work

CPE 3: Controls

=6DOF with spherical wrist

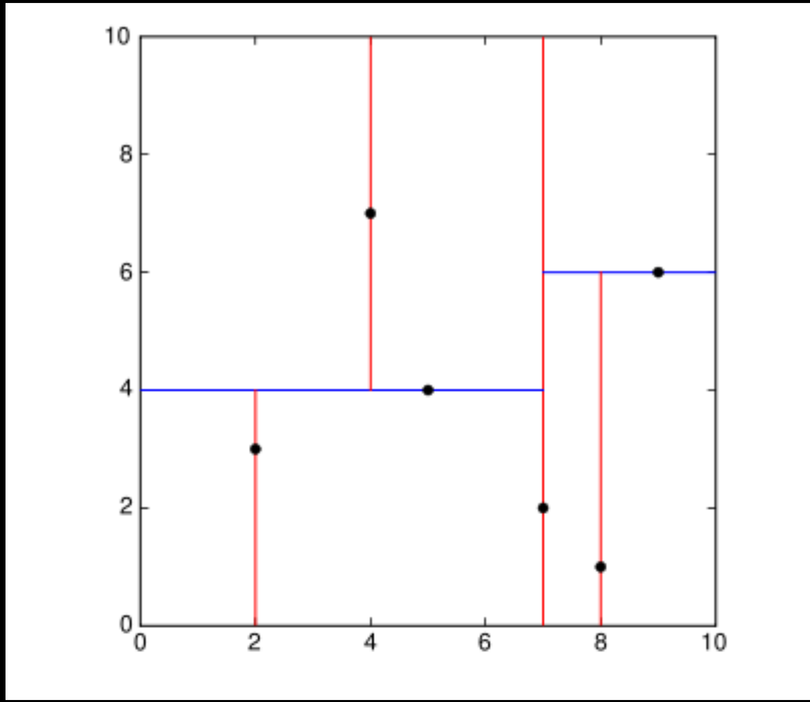
- Closed-form solution
- <= 8 arm poses for a desired Xe
  - Joint limits, link collisions, singularity
  - Not all Xe can be achieved in reachable space

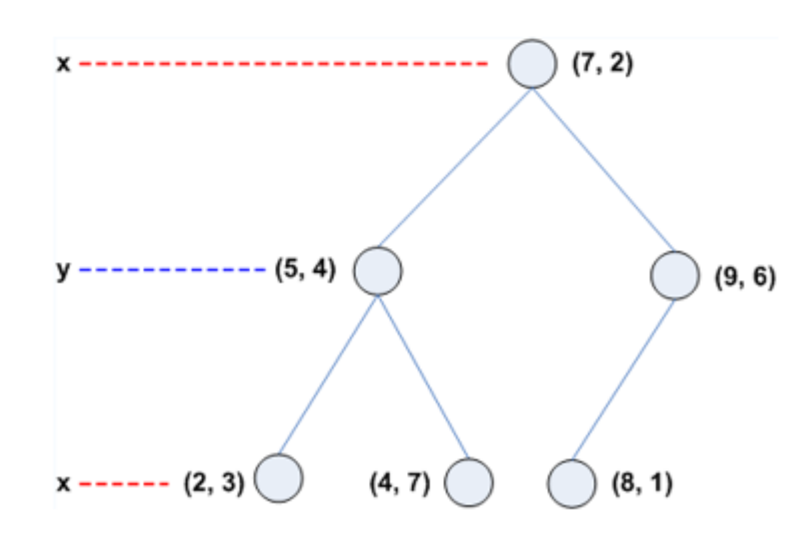
>6DOF with spherical wrist

- Numerical solution
- > 8 arm poses for a desired Xe
  - Can work with singularities
  - Null-space motions used to move joints w/o affecting Xe
    - Good for avoiding collisions

CPE 3: Controls

KDT Functionality





CPE 3: Control Systems

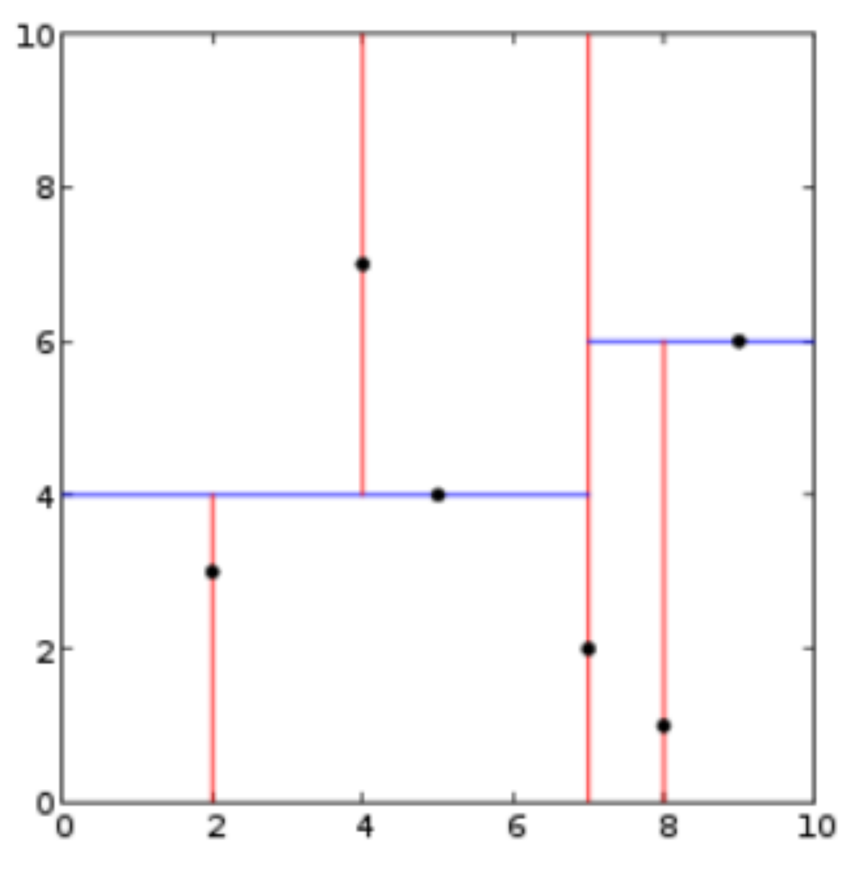
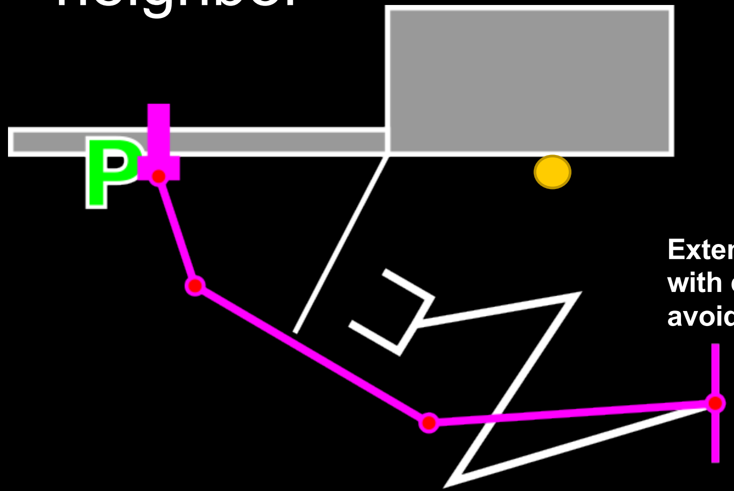


Fig. 15 K-D 2 Dimensional Tree [4]

- K-Dimensional Tree (KDT)
  - Generalization of a Binary Search Tree
  - Maps location to nearest neighbor



Extended arm motion with collision avoidance via KDT

CPE 3: Control Software

- Visual system will output desired position and orientation variables
- Inverse kinematics equations use 6 equations with the 6 degrees of freedom to find each joint angle
- Resulting joint angles passed to control software motion planning algorithm

From Image Processing

Desired position: x, y, z
Desired orientation: χ , ψ , ζ

→

Inverse Kinematics

Desired joint angles

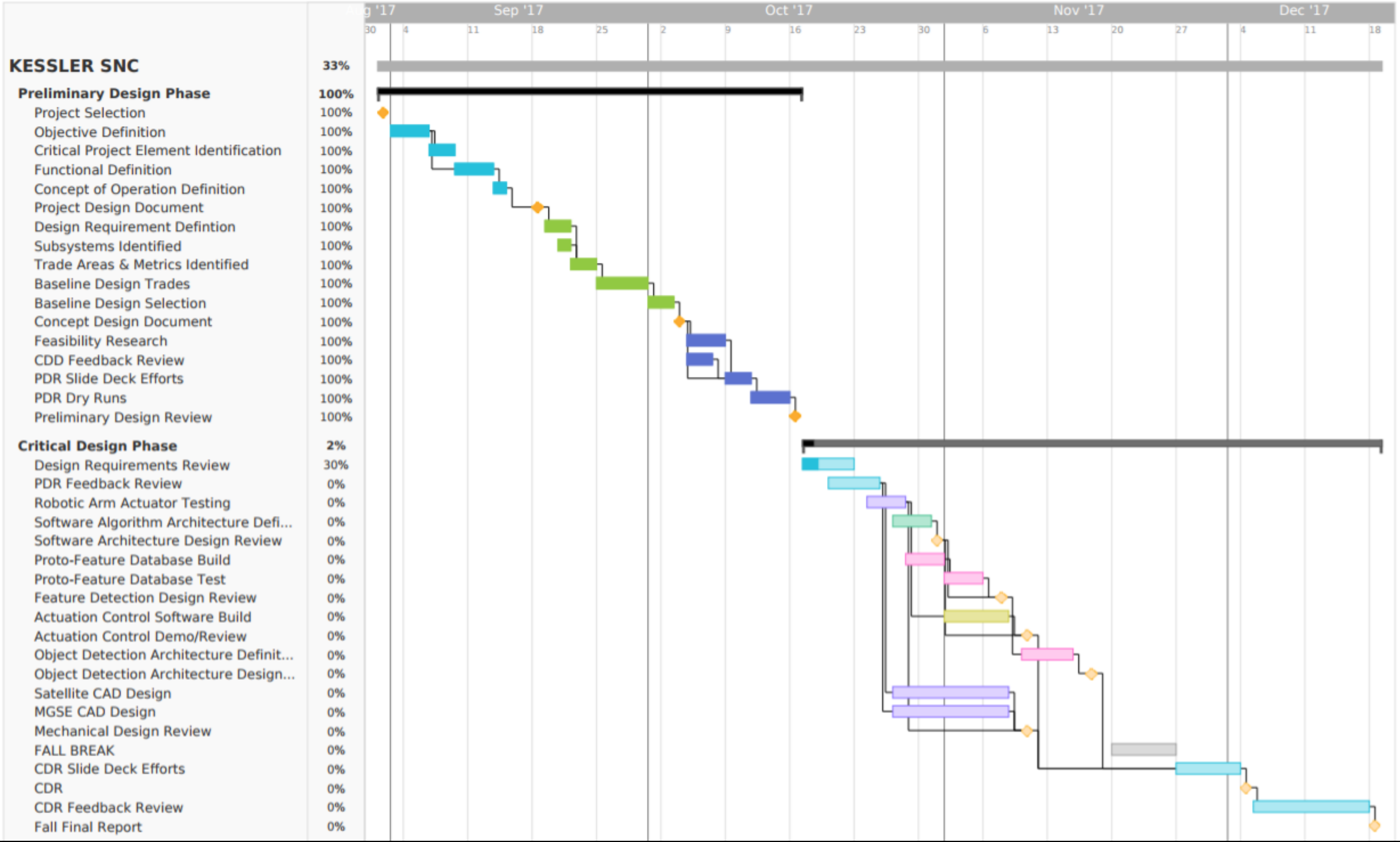
$$\begin{bmatrix} q1 \\ q2 \\ q3 \\ q4 \\ q5 \\ q6 \end{bmatrix} = F \begin{bmatrix} x \\ y \\ z \\ \chi \\ \psi \\ \zeta \end{bmatrix}$$

→

Motion Planning

Uses joint angles to find intermediate claw positions

Status & Future Work Backup


Sep '17
Oct '17
Nov '17
Dec '17
30 4 11 18 25 2 9 16 23 30 6 13 20 27 4 11 18
KESSLER SNC 33%
Preliminary Design Phase 100%
Project Selection 100%
Objective Definition 100%
Critical Project Element Identification 100%
Functional Definition 100%
Concept of Operation Definition 100%
Project Design Document 100%
Design Requirement Defintion 100%
Subsystems Identified 100%
Trade Areas & Metrics Identified 100%
Baseline Design Trades 100%
Baseline Design Selection 100%
Concept Design Document 100%
Feasibility Research 100%
CDD Feedback Review 100%
PDR Slide Deck Efforts 100%
PDR Dry Runs 100%
Preliminary Design Review 100%
Critical Design Phase 2%
Design Requirements Review 30%
PDR Feedback Review 0%
Robotic Arm Actuator Testing 0%
Software Algorithm Architecture Defi... 0%
Software Architecture Design Review 0%
Proto-Feature Database Build 0%
Proto-Feature Database Test 0%
Feature Detection Design Review 0%
Actuation Control Software Build 0%
Actuation Control Demo/Review 0%
Object Detection Architecture Definit... 0%
Object Detection Architecture Design... 0%
Satellite CAD Design 0%
MGSE CAD Design 0%
Mechanical Design Review 0%
FALL BREAK 0%
CDR Slide Deck Efforts 0%
CDR 0%
CDR Feedback Review 0%
Fall Final Report 0%