CAST - PDR

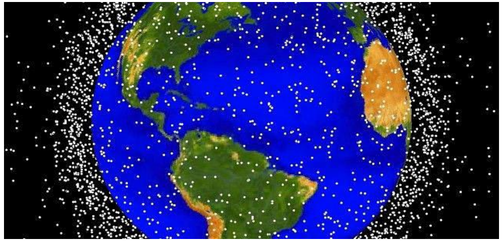
Ann and H.J. Smead
Aerospace Engineering Sciences
UNIVERSITY OF COLORADO BOULDER

Team members: Trace Valade, Adam Holdridge, Angel Hoffman, Cameron Turman, Conner Martin, Griffin Van Anne, Hugo Stetz, Isaac Goldner, Jason Balke, Reade Warner, Roland Bailey, Sam Hartman
Advisor: Prof. John Mah
Customer: John Reed and United Launch Alliance

Project Description

Project Motivation and Value

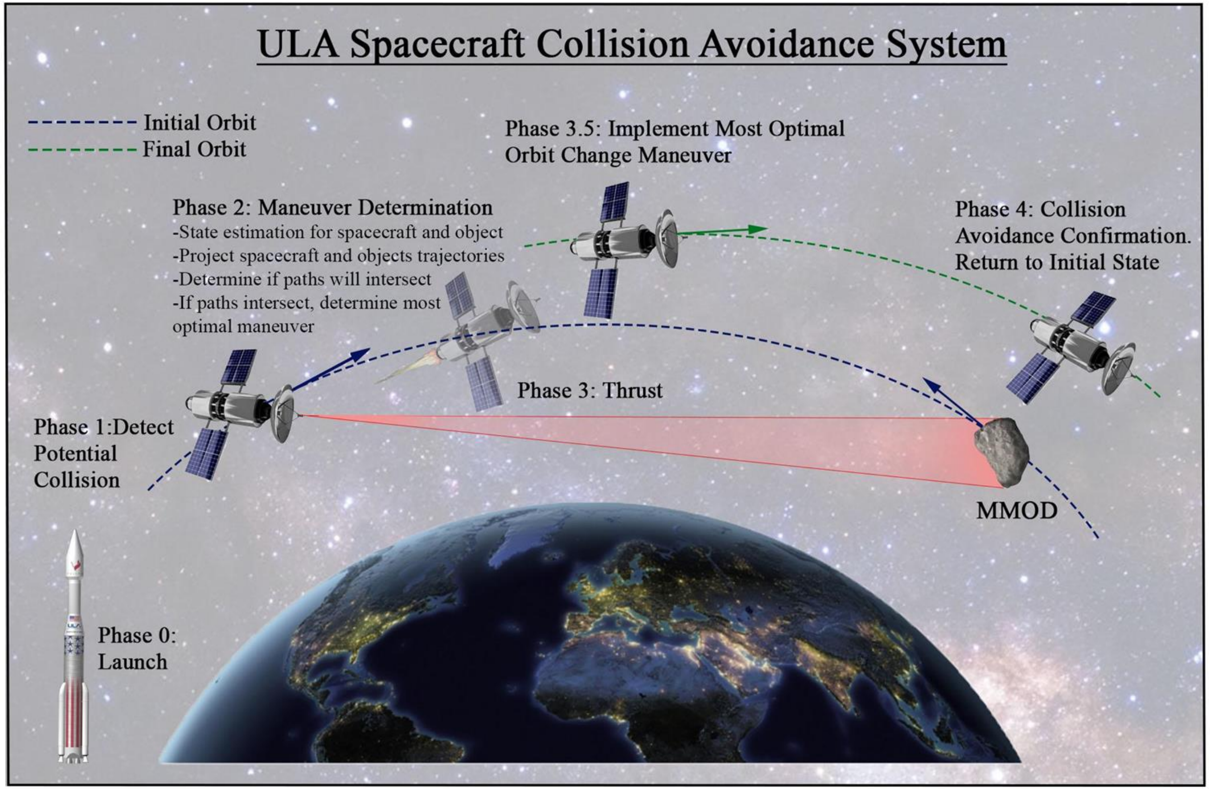
- Space is cluttered. At orbital velocities, any colliding object may pose a mission ending threat to spacecraft.
- Spacecraft must be capable of detecting and avoiding a possible collision
- Application of detect and react technology to real spacecraft could:
    - Decrease the probability of collision by having more accurate state estimation
    - Reduce the frequency of avoidance maneuvers (and therefore orbital corrections), saving fuel
    - Be supplemented with data from ground-tracked objects
    - Allow for semi or fully autonomous collision avoidance



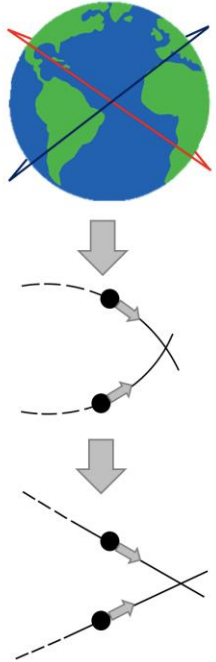
Project Objectives

- Create a 3D simulation capable of **representing a collision scenario** in space.
- Create a physical/visual 2D demonstration that implements a **detect and react** algorithm for a collision scenario with debris, satellite, or launch vehicle.
- Use developed 3D software model and 2D testbed demonstration to **provide detailed specifications** that would represent what the requirements of a spacecraft (thrust and sensing) would need to execute successful avoidance.

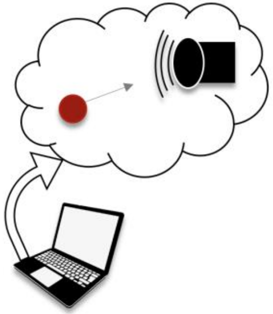
CONOPS: Overview



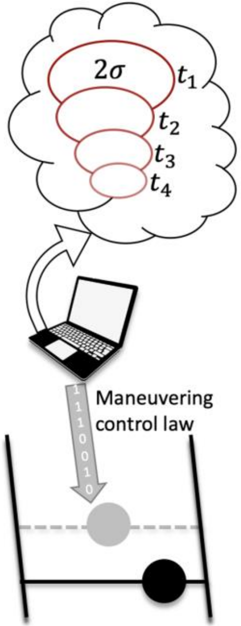
CONOPS: Project Year



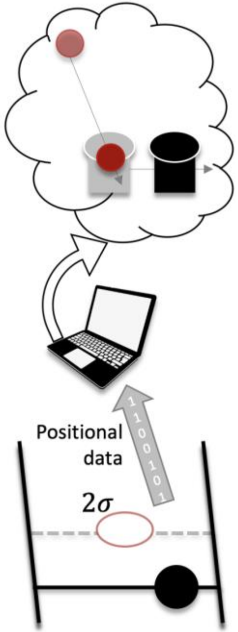
Establish 2D orbital collision scenario parameters



Software-defined object launch and sensing through sensor model

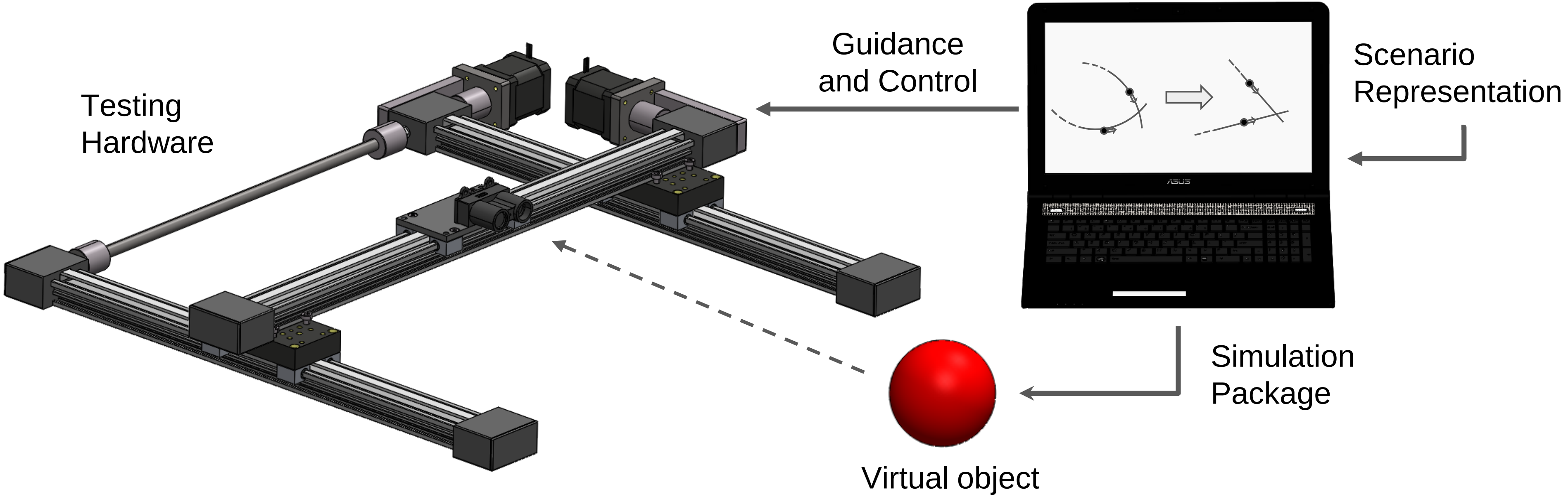


Physical testbed maneuvering to avoid covariance ellipse



Verification of covariance ellipse avoidance

Baseline Design: Overview



Testing Hardware

Guidance and Control

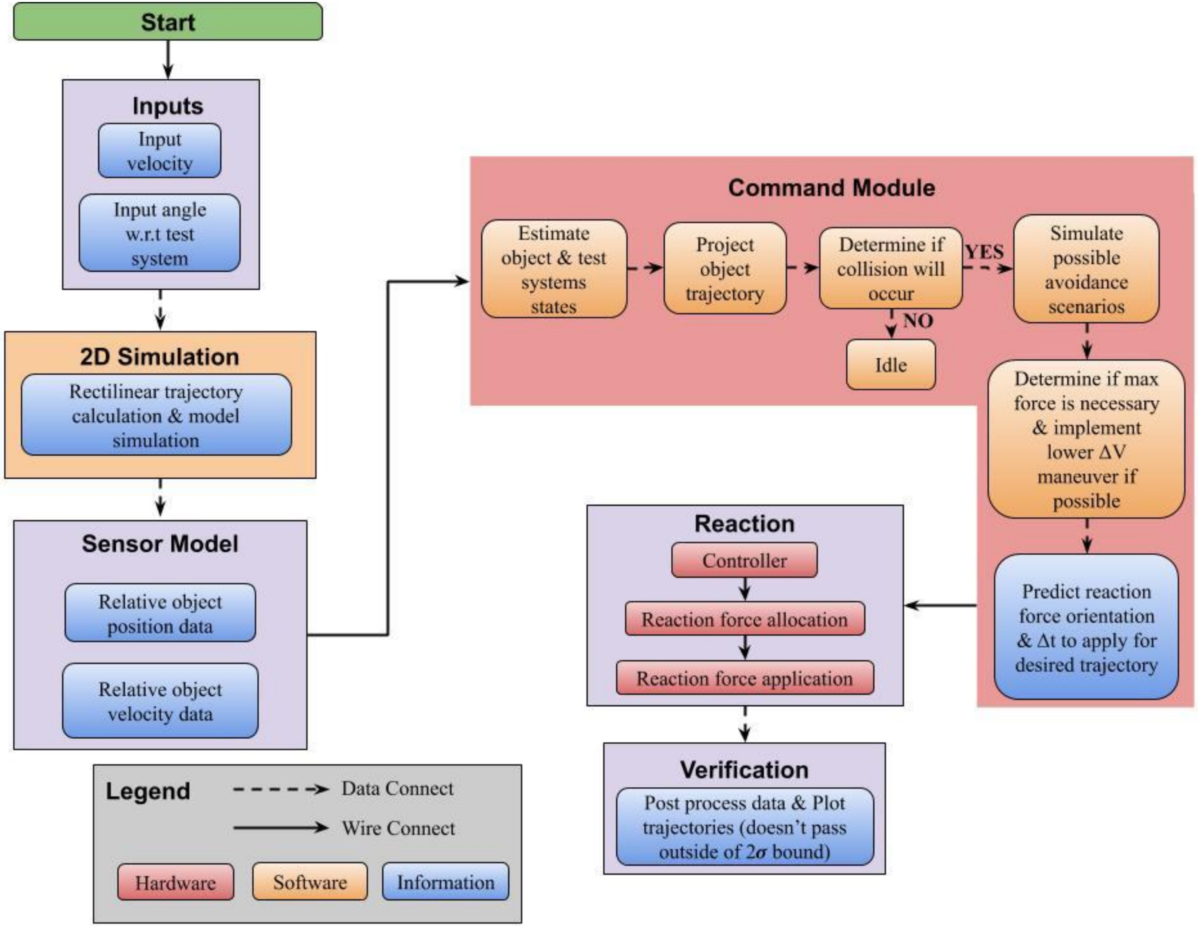
Scenario Representation

Simulation Package

Virtual object

On ground testing environment

FBD



Start

Inputs
Input velocity
Input angle w.r.t test system

2D Simulation
Rectilinear trajectory calculation & model simulation

Sensor Model
Relative object position data
Relative object velocity data

Command Module
Estimate object & test systems states
Project object trajectory
Determine if collision will occur
NO
Idle
YES
Simulate possible avoidance scenarios
Determine if max force is necessary & implement lower ΔV maneuver if possible
Predict reaction force orientation & Δt to apply for desired trajectory

Reaction
Controller
Reaction force allocation
Reaction force application

Verification
Post process data & Plot trajectories (doesn't pass outside of 2σ bound)

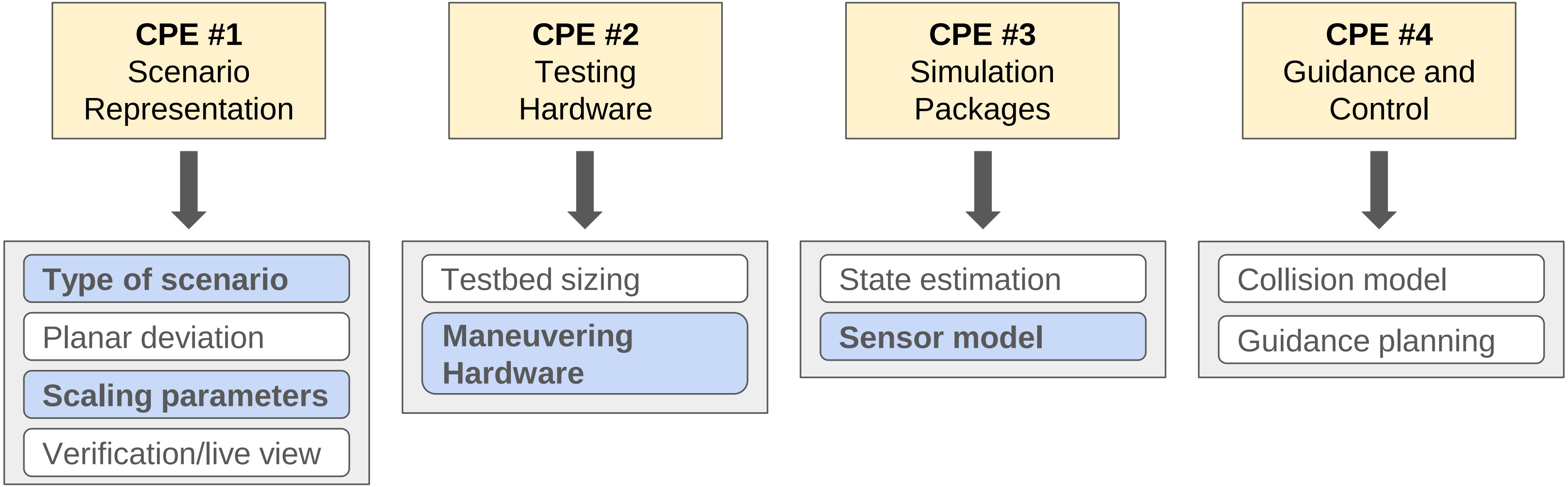
Legend
- - - -> Data Connect
——> Wire Connect
Hardware
Software
Information

Functional Requirements

FR1	A 3D orbital collision scenario shall be reduced to a 2D simulation and implemented in a 2D test bed while scaling collision parameters.
FR2	The test system shall be capable of detecting incoming objects in a representative collision scenario on the test bed.
FR3	The test system shall be capable of avoiding a collision in a representative collision scenario.

Baseline Design

Baseline Design: Critical Project Elements



CPE #1
Scenario Representation

- **Type of scenario**
- Planar deviation
- **Scaling parameters**
- Verification/live view

CPE #2
Testing Hardware

- Testbed sizing
- **Maneuvering Hardware**

CPE #3
Simulation Packages

- State estimation
- **Sensor model**

CPE #4
Guidance and Control

- Collision model
- Guidance planning

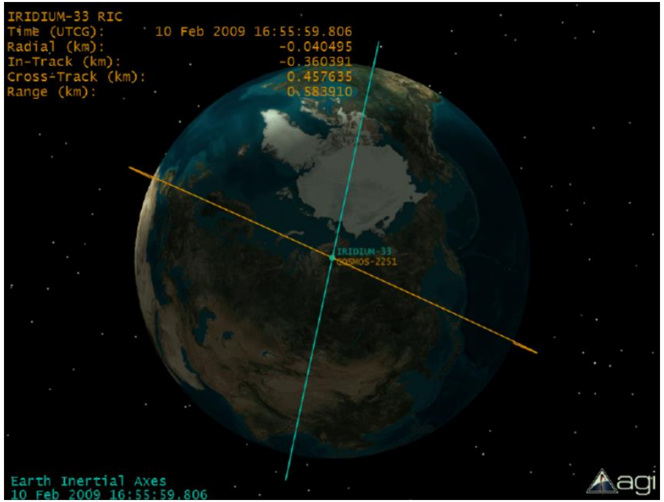
Baseline Design: Type of scenario

FR1 A 3-dimensional orbital collision scenario shall be <u>reduced to a 2-dimensional simulation</u> with the associated collision parameters replicated in a test bed where the body is restricted to 2-dimensions

<u>Purpose</u>: A specific orbital collision scenario is chosen to model on the 2-dimensional test bed

<u>Selection</u>: Active satellite avoiding debris, cross-track collision in a circular orbit at an altitude of 1200 km

<u>Reasoning</u>: Cross-track high probability conjunctions with debris occur frequently in orbit (via CelesTrak high probability conjunction list)



IRIDIUM-33 RIC
Time (UTCG): 10 Feb 2009 16:55:59.806
Radial (km): -0.040495
In-Track (km): -0.360391
Cross-Track (km): 0.457635
Range (km): 0.583910

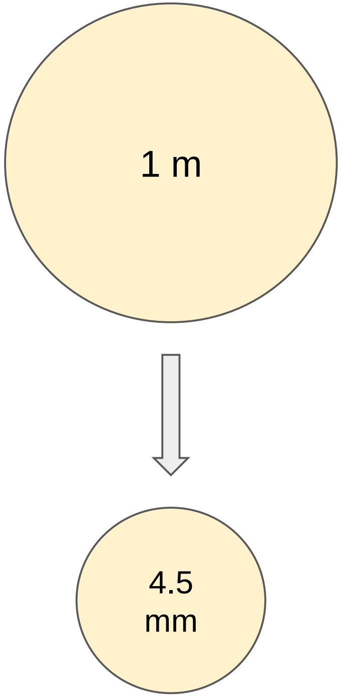
Credit: CelesTrak

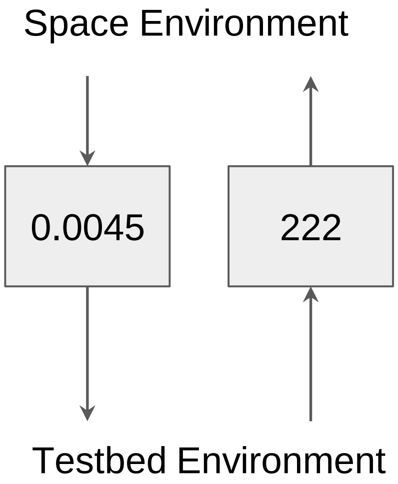
Baseline Design: 2-D Similarity Parameters

Parameter		**Dependencies**	**Purpose**
Effective View Angle		1) Target size 2) Relative distance	Governs sensor ability
Acceleration		1) Force 2) Mass	Maneuvering ability
Time	Sensor time	1) Sensing distance 2) Relative velocity	Maximum response time
	Maneuvering time	1) Maneuvering distance 2) Maneuvering time	Minimum response time

Baseline Design: Scaling Factor

- Scaling scenario by factor of 222:1
- Easy to visualize as 1 meter goes to size of 1 “BB” ball projectile
- Keeps objects physically representable in the testbed environment
    - Added value from extended project perspective





Baseline Design: Maneuvering Hardware

FR3 The test system shall be capable of avoiding a collision in a representative collision scenario without the test system technology performance levels exceeding those of current, full-scale spacecraft hardware

Purpose: Avoid a modeled and incoming collision object using a system that accurately represents the motion of the spacecraft system in a scaled environment.

Selection: Linear Rail System

Reasoning: High positional accuracy (only limited by motor resolution), high level of control of motion, and speed capacity. Relatively adequate potential size. Encoder avoidance verification.

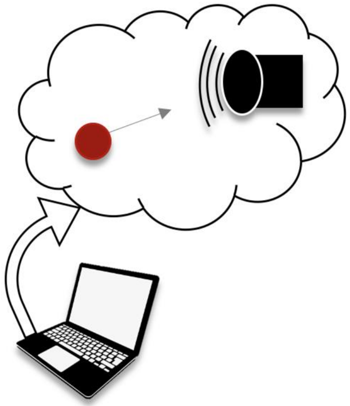
Baseline Design: Simulation Packages

FR2 The test system shall be capable of <u>detecting incoming objects</u> (active or inactive) in a representative collision scenario on the test bed

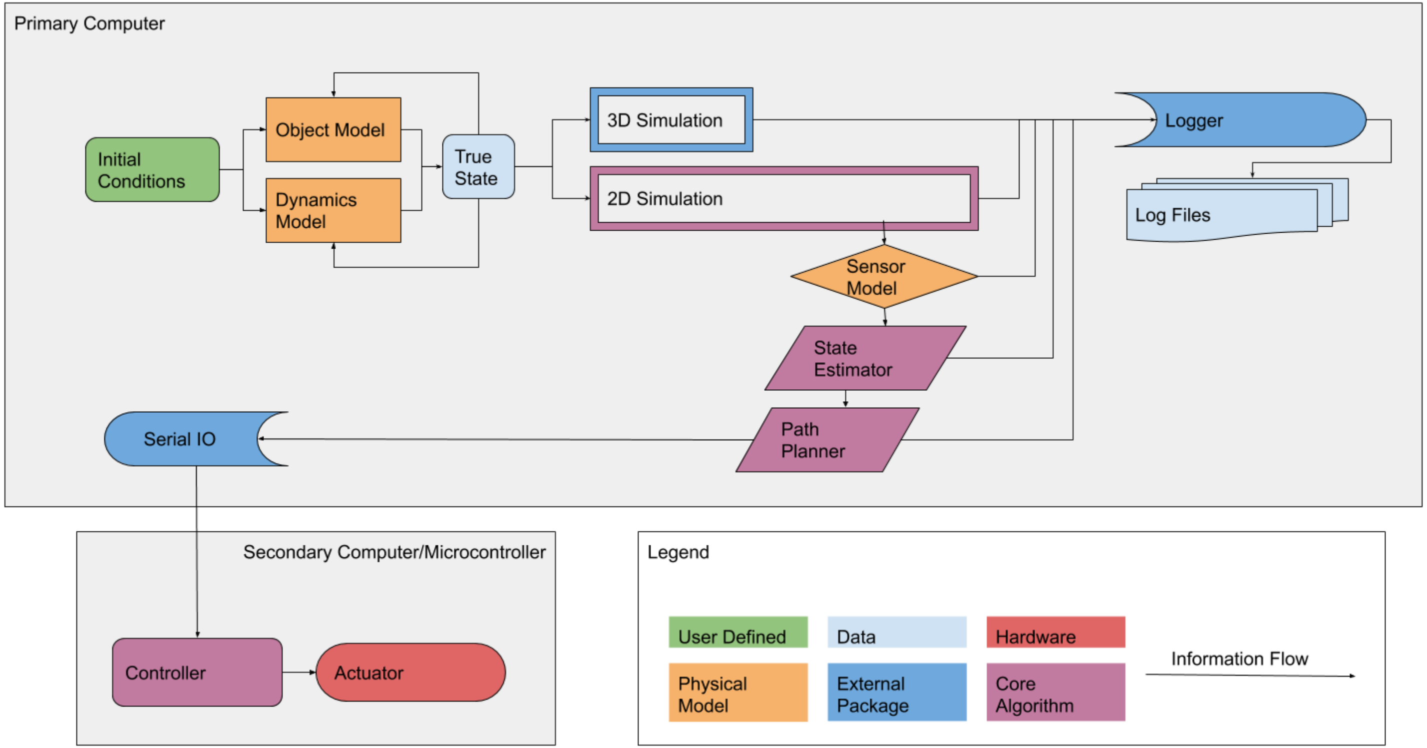
<u>Purpose</u>: Model an incoming object at a relative velocity defined by orbital cross-track scenario

<u>Selection</u>: Software-defined collision

<u>Reasoning</u>: Selecting a capable sensor is outside the scope of this project. Instead, a “rubber sensor” can be used to guide the dialog on necessary sensor parameters.

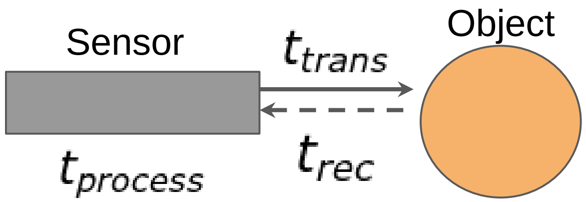


Baseline Design: High Level Architecture



Baseline Design: Sensor Model

- Take sensor parameters as input and model the sensor output to feed into state estimation algorithm
- Utilize governing equations for a typical LiDAR sensor
- Simulated measurement noise will have to be verified with baseline, physical COTS sensor
  - This baseline sensor will NOT drive sensor parameter selection!

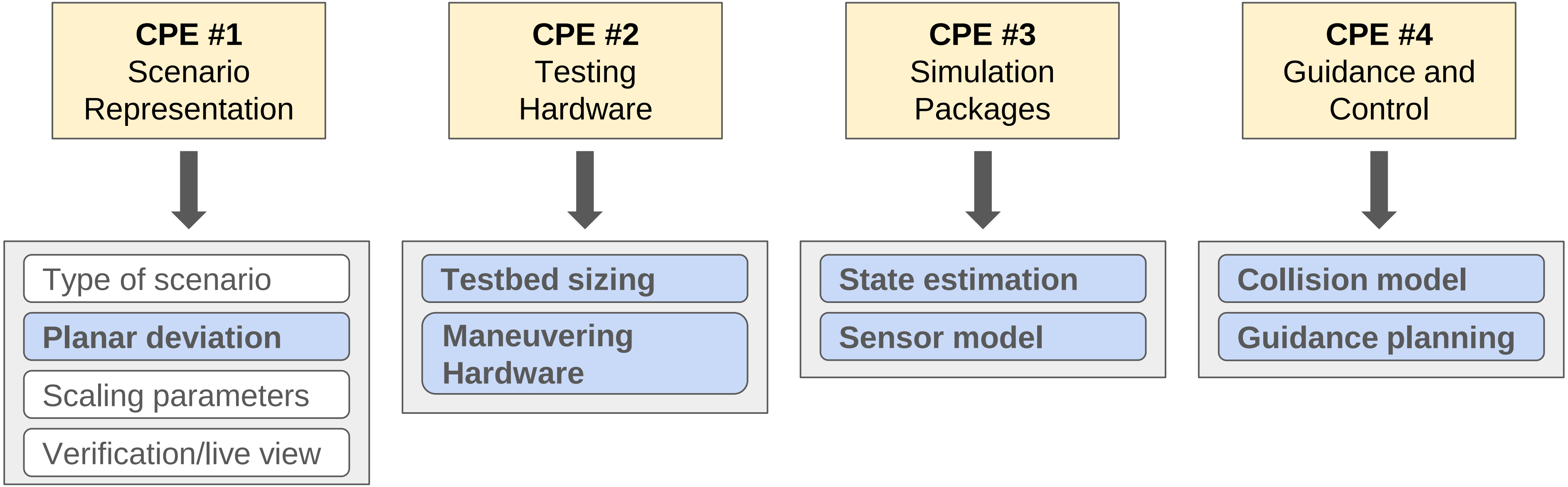


$$\Delta t = t_{process} + t_{rec} + t_{trans}$$

$$\text{Range} = c\frac{\Delta t}{2}$$

Feasibility Analysis

Feasibility: Critical Project Elements



CPE #1
Scenario Representation

- Type of scenario
- **Planar deviation**
- Scaling parameters
- Verification/live view

CPE #2
Testing Hardware

- **Testbed sizing**
- **Maneuvering Hardware**

CPE #3
Simulation Packages

- **State estimation**
- **Sensor model**

CPE #4
Guidance and Control

- **Collision model**
- **Guidance planning**

CPE: Scenario Representation Feasibility Analysis

Planar Deviation Analysis

FR 1, DR 1.1, DR 1.2, DR 1.3	2D scenario, is representative of the 3D environment for 100 seconds

Purpose:

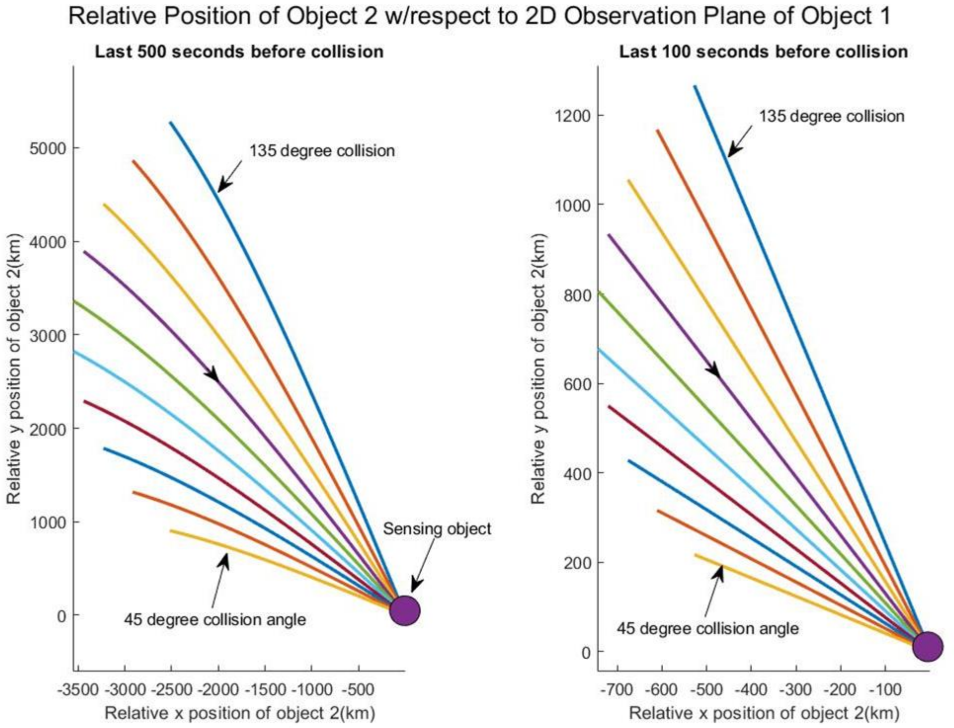
- Can the full-scale collision be accurately scaled to a 2D rectilinear test-bed?

Assumptions:

- Both objects orbiting at same altitude
- Both objects in circular orbits
- Simplified 2-body problem without perturbations

Results:

- Under 100s → R^2 of 0.999, max deviation of 4°
- Small planar deviation
- Very linear relative motion
- Accurate 2D representation of 3D environment for 100 seconds is feasible



CPE: Testing Hardware Feasibility Analysis

Testbed Sizing

Purpose/question to answer:

- Determine if the test bed is capable producing maximum maneuvering outputs on a 1x1m test bed for the minimum 10s requirement.

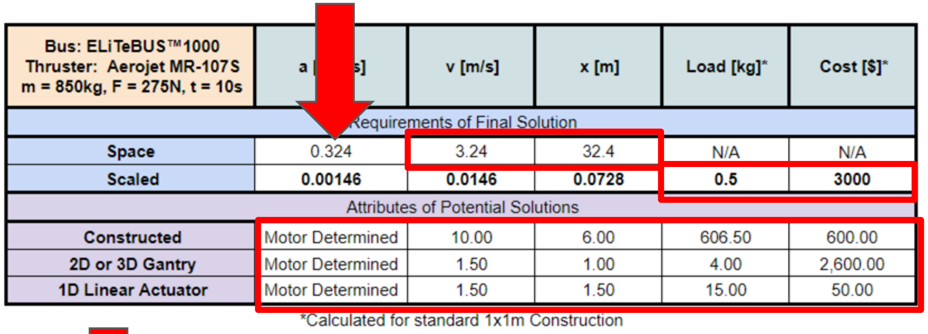
Assumptions:

- Thrusting in-line, full time
- 1-D, constant acceleration motion
- Relative path only
- Negligible external force

Results:

- 10s Reaction Feasible:
  - Velocity, distance, load, and cost
  - 37.1 s max case limitation (1x1m Testbed)
- 100s Reaction Feasible BUT Infeasible for Max Thrust Case
  - 0.0444 m/s^2 in space
  - 0.0002 m/s^2 scaled

DR 1.1, 1.3	Preserve a timescale between bounds of 10-100 seconds



Bus: ELiTeBUS™1000 Thruster: Aerojet MR-107S m = 850kg, F = 275N, t = 10s	a [[illegible]s]	v [m/s]	x [m]	Load [kg]*	Cost [$]*
Requirements of Final Solution					
Space	0.324	3.24	32.4	N/A	N/A
Scaled	**0.00146**	**0.0146**	**0.0728**	**0.5**	**3000**
Attributes of Potential Solutions					
Constructed	Motor Determined	10.00	6.00	606.50	600.00
2D or 3D Gantry	Motor Determined	1.50	1.00	4.00	2,600.00
1D Linear Actuator	Motor Determined	1.50	1.50	15.00	50.00

*Calculated for standard 1x1m Construction

$$F = ma$$

$$v_f = v_0 + at$$

$$x_f = x_0 + v_0 t + \frac{1}{2}at^2$$

Maneuvering Hardware

FR 3	Capable of avoiding a collision without exceeding performance of current, full-scale, spacecraft hardware

Purpose/question to answer:

- Determine if there exists a suitable motor solution that can meet the acceleration, speed, and resolution requirements of testbed.
- Determine if a fitting solution is implementable give the team’s skill set and time.

Assumptions:

- Assumed the mass driven by motors
- Assumed belt/pulley drive
- 1 motor drives each axis
- Negligible friction

	Speed	**Resolution**	**Torque**
Requirement	0.054 m/s	0.063 mm	0.01 oz in
Stepper - 5mm spindle	0.1 m/s	0.04 mm	Torque hard to control
Servo - 10 mm spindle	0.21 m/s	0.05 mm	6 oz in

Results:

- Servo Motor
  - Closed loop control for Speed, Position, and Acceleration
- Stepper Motor

CPE: Simulation Packages Feasibility Analysis

Languages and Packages

Languages considered:

- [✓ Matlab]
- [✓ C/C++]
- Fortran
- R
- Julia
- Java
- [✓ Python]
- PHP

NOTE: Downselected to circled options based on group experience/desire to learn, available packages, ease of interface, accessibility, etc.

Three Dimensional Orbital Simulation

- Basilisk: C++/Python, Developed at CU
- [✓ GMAT:] Standalone, NASA Heritage, Matlab & Python Interfaces
- Tudat: C++, Unofficial Matlab Interface, JSON Interface, Estimation Library
- STK: Standalone, Popular in industry, not available for delivery to customer

Sensor Modeling

- [✓ Matlab:] Parametric Sensor Models (Sensing Fusion and Tracking Toolbox)
- Tudat: Simplified RF Model - Ground Based Tracking

Object Velocity Uncertainty

DR 1.4	The test bed shall be capable of repeatedly creating a relative velocity with less than 2.5% deviation from the scaled, test bed relative velocity

Purpose/question to answer:

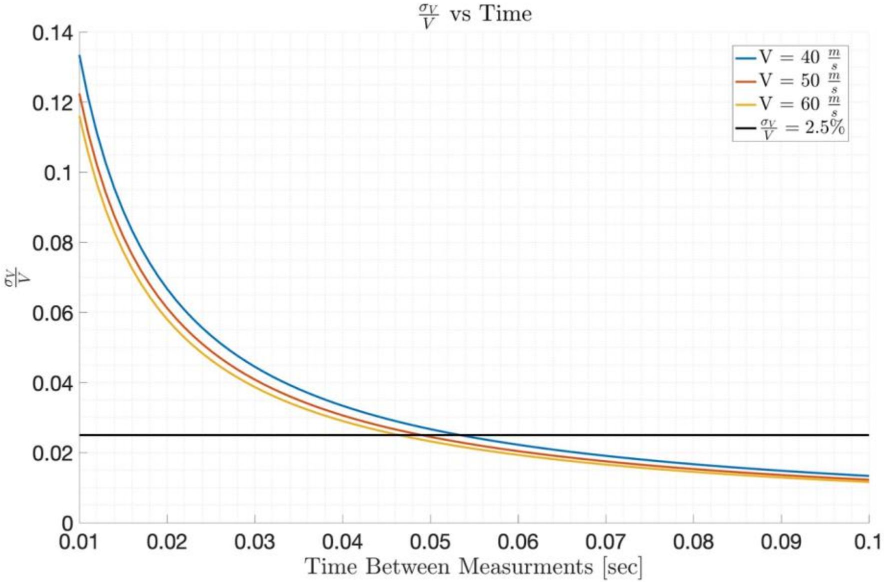
- Determine time between measurements to keep percent deviation under 2.5%

Assumptions:

- Independent random variables
- Constant velocity
- Scaled velocity: 40 m/s to 60 m/s
    - 8.88 - 13.3 km/s in represented scenario

Results:

- Required time << time of collision
- From the table values, these times are feasible for off the shelf sensors.



	V = 40 m/s	V = 50 m/s	V = 60 m/s
2.5% Time	0.0534 s	0.0490 s	0.0464 s

State Estimation

SR 1.6.2, 1.6.3	Software package shall generate and utilize an estimate of the incoming object state to implement avoidance

Purpose/question to answer:

- What state estimation method is best suited for our needs?

Assumptions:

- Minimum 10 second collision
- Sampling rate between 10 Hz - 1 kHz

Results:

- Baseline: LLS used to give feasibility for PDR
- Best implementation: use batch to initialize and KF to propagate dynamics.
- Feasible

Kalman Filter (KF)	**Batch Method (BM)**	**Linear Least Squares (LLS)**
Able to estimate evolving state Does not require data storage	Converges faster than KF More accurate than KF	Easy to implement Can update estimate of static parameter
Will break if bad initial guess	More computationally expensive than KF Requires batch of data	Least accurate method

Positional Uncertainty

DR 3.1	The testing hardware shall be capable of generating sufficient force to avoid the 2σ covariance ellipse

Purpose/question to answer:

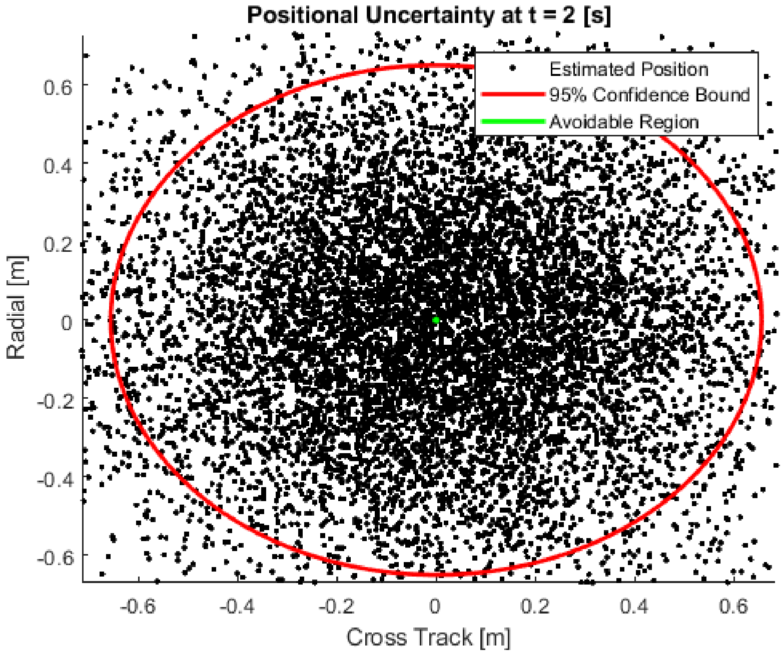
- Determine what sensor parameters are necessary to provide sufficient reaction time to avoid the 95% confidence bound.

Assumptions:

- Error follows normal distribution
- Rectilinear motion
- Time of measurement is known to the millisecond
- Least squares estimation
- Sensor capabilities match that of commercially available sensors

Results:

- The capabilities of commercial sensors make avoiding the incoming object not feasible



Positional Uncertainty

DR 3.1	The testing hardware shall be capable of generating sufficient force to avoid the 2σ covariance ellipse

Purpose/question to answer:

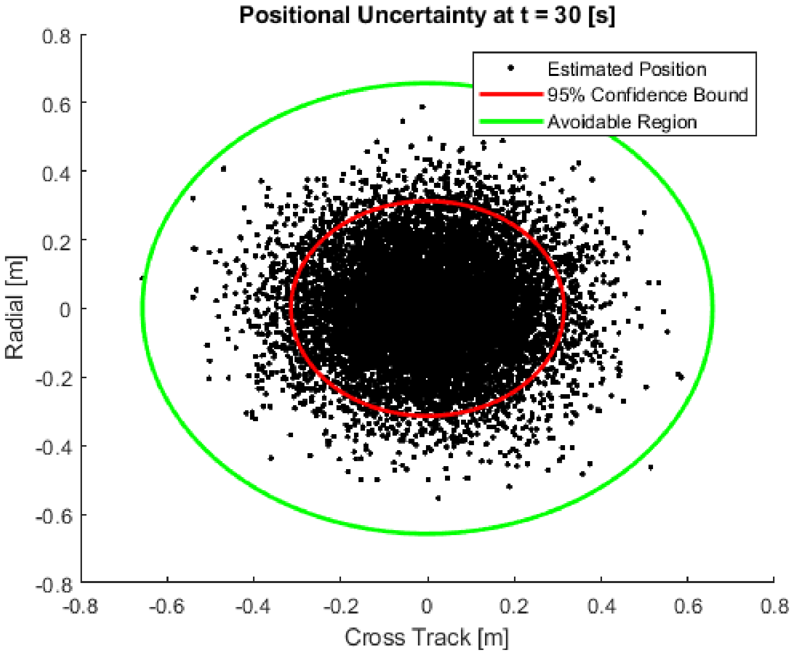
- Determine what sensor parameters are necessary to provide sufficient reaction time to avoid the 95% confidence bound.

Assumptions:

- Error follows normal distribution
- Rectilinear motion
- Time of measurement is known to the millisecond
- Least squares estimation
- Sensor **range** extended to full 100s timescale

Results:

- By extending the sensing capabilities of our model, avoiding the 95% confidence bound for the object at time of collision is feasible



CPE: Guidance and Control Feasibility Analysis

Collision Model

DR 1.4, 1.5	Test bed will create a collision scenario with a probability of 95% and within 2.5% of representative relative velocity

Purpose/question to answer:

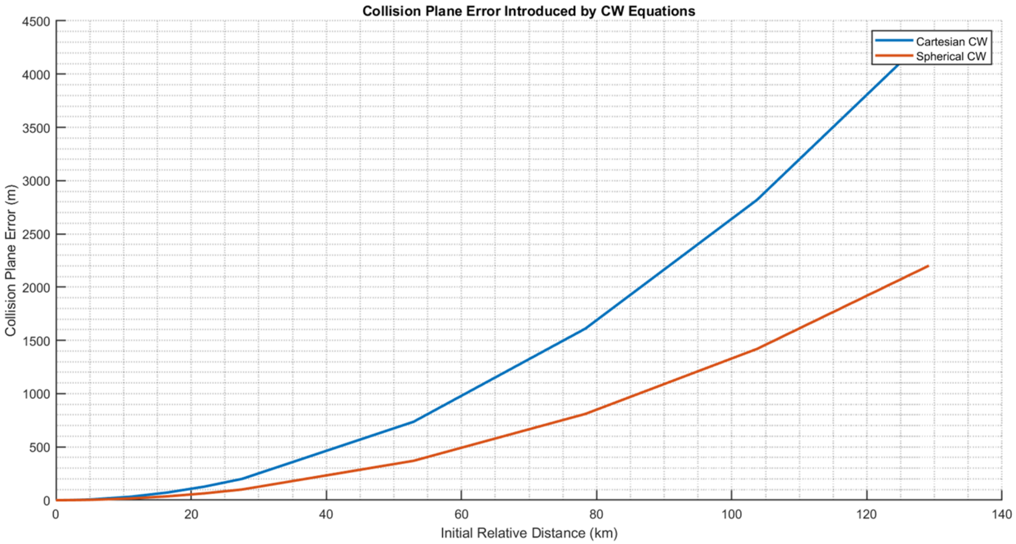
- An appropriate model for the relative motion of the two objects is essential for determining the probability of collision

Assumptions:

- The model assumes the motion of the objects can be modeled by the two body problem and that the objects are within the planar region

Results:

- CW equations generate far too much error in either the rectilinear or curvilinear formulation.
- However directly integrating the non-linear orbital dynamics is feasible.



Guidance Planning

DR 1.6, SR 3.2.1	The test bed will compute the maneuver required to avoid a collision

Purpose/question to answer:

- Determine the appropriate action the spacecraft must take in order to avoid a collision

Assumptions:

- Perturbations are negligible.
- Errors in conjunction time are negligible.

Results:

- 2003 paper by Patera and Peterson utilize the gradient of the conjunction probability density to determine maneuver direction, then iteratively determine magnitude, this result can be improved, but the baseline algorithm is feasible.

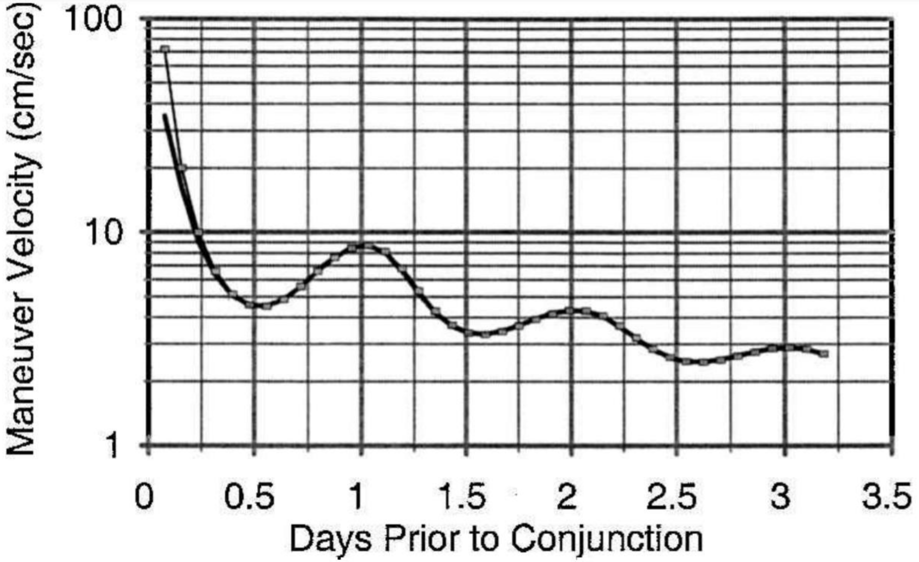
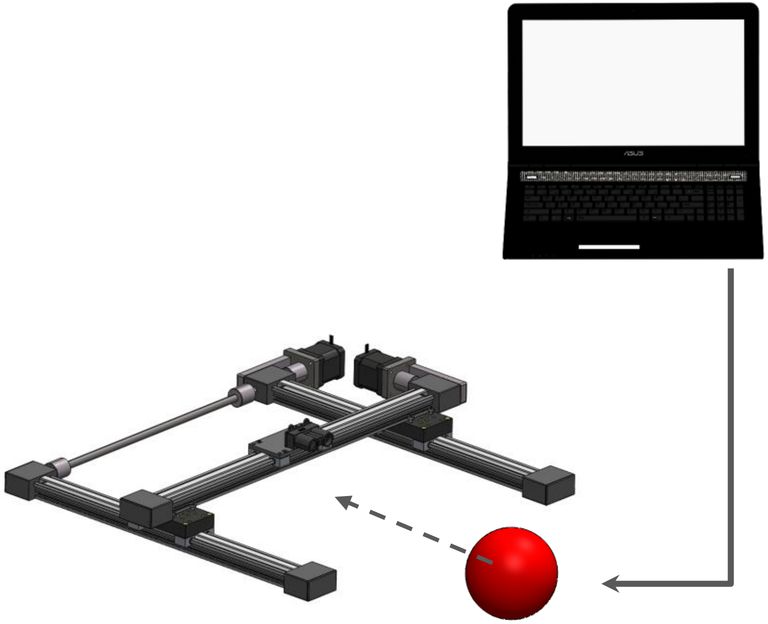


Image: Patera, Russell & Peterson, Glenn. (2003). Space Vehicle Maneuver Method to Lower Collision Risk to an Acceptable Level. Journal of Guidance Control and Dynamics - J GUID CONTROL DYNAM. 26. 233-237. 10.2514/2.5063.

Status Summary and Strategy

Baseline Design Summary

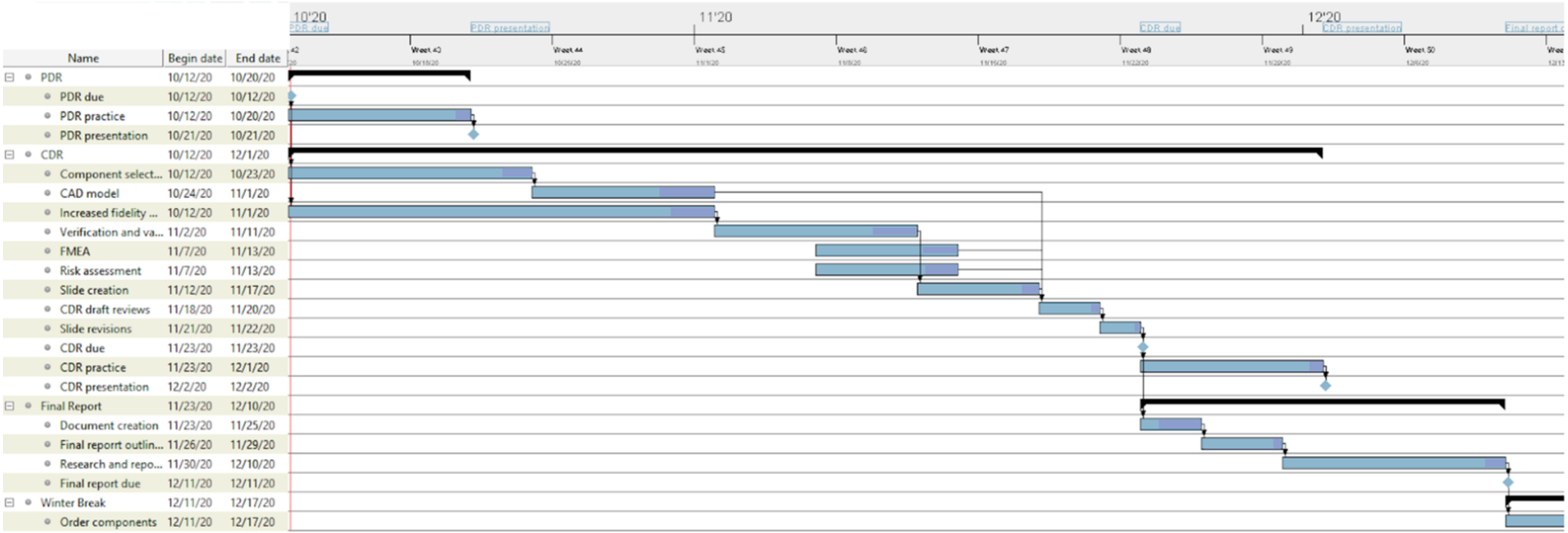
- 1200 km altitude cross-track collision modeled in 2-dimensions
- Implementation of detect-and-react algorithm
  - Sensor model detects software-defined object collision
  - Physical, baseline sensor used to verify simulated measurement noise
- Maneuvering test-bed guides recommendation for required sensor
- Verification of successful avoidance via viewing of trajectories (live plot)



Next Steps

- Testbed motor controller selection
- Improved state estimation implementation
- Ballistics modeling
- Electronics communications protocols
- Computational time feasibility
- Verification and validation plans
- Risk assessment

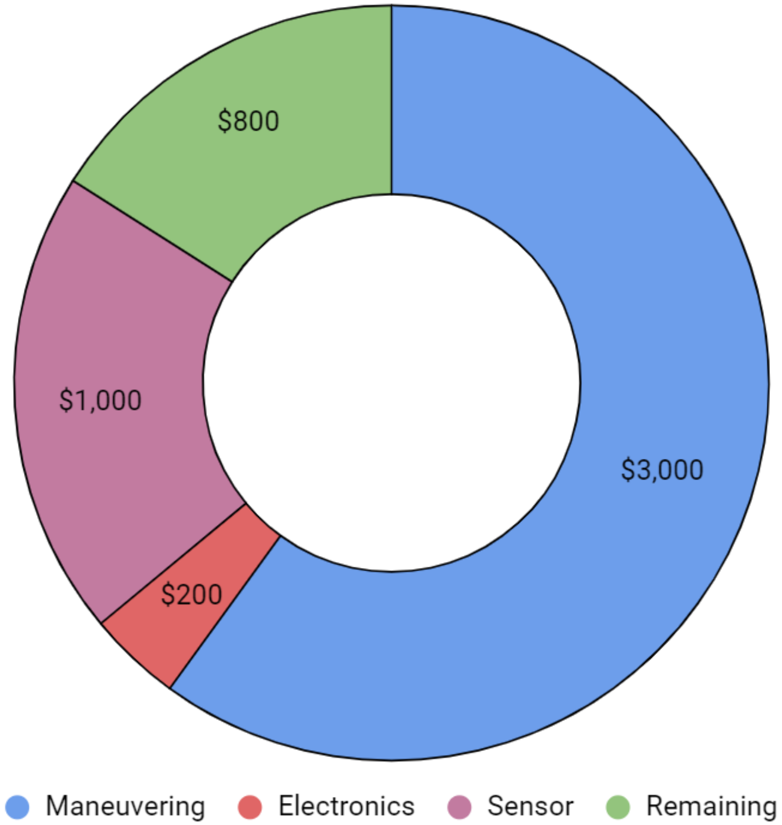
Gantt Chart



Name	Begin date	End date
⊟ PDR	10/12/20	10/20/20
PDR due	10/12/20	10/12/20
PDR practice	10/12/20	10/20/20
PDR presentation	10/21/20	10/21/20
⊟ CDR	10/12/20	12/1/20
Component select...	10/12/20	10/23/20
CAD model	10/24/20	11/1/20
Increased fidelity ...	10/12/20	11/1/20
Verification and va...	11/2/20	11/11/20
FMEA	11/7/20	11/13/20
Risk assessment	11/7/20	11/13/20
Slide creation	11/12/20	11/17/20
CDR draft reviews	11/18/20	11/20/20
Slide revisions	11/21/20	11/22/20
CDR due	11/23/20	11/23/20
CDR practice	11/23/20	12/1/20
CDR presentation	12/2/20	12/2/20
⊟ Final Report	11/23/20	12/10/20
Document creation	11/23/20	11/25/20
Final reporrt outlin...	11/26/20	11/29/20
Research and repo...	11/30/20	12/10/20
Final report due	12/11/20	12/11/20
⊟ Winter Break	12/11/20	12/17/20
Order components	12/11/20	12/17/20

Budget

	Budget ($)	Expected ($)	Margin (%)
Maneuvering ● Linear Rails ● Motors ● Pre-Packaged	3000	2000	66%
Electronics ● Microcontroller ● Wiring	200	75	37%
Sensor ● LiDAR	1000	500	50%
Total	4200	2575	61%
Remaining	800	2425	48%



Questions?

Backup Slides

Requirements

FR1 A 3-dimensional orbital collision scenario shall be reduced to a 2-dimensional simulation with the associated collision parameters replicated in a test bed where the body is restricted to 2-dimensions

- **DR1.1** The test bed shall support representative collision scenarios that preserve a time scale (ratio of distance to relative velocity) up to 100-150 seconds (LEO) and/or 1600 seconds (GEO).
- **DR1.2** The test bed shall support representative collision scenarios that preserve an optical view angle (ratio of cross section diameter to distance) of 0.226-0.29 (LEO) and/or 0.05 (GEO) arcseconds at the furthest point
- **DR1.3** The lower bound of the time scale shall be 10 seconds.
- **DR1.4** The test bed shall be capable of repeatedly creating a relative velocity with less than 2.5% deviation from the scaled, test bed relative velocity.
- **DR1.5** When modeling an orbital scenario with a collision trajectory the test system shall create a collision with 95% (2 σ) success.
- **DR1.6** The test system shall implement a detect and react procedure.
  - **DR1.6.1** The test system software shall receive sensor data about an incoming object, including relative velocity and position.
- **DR1.7** The test system shall be fully functional after repeated detect and react procedures, where full functionality is defined as the ability to sense position and velocity data for an incoming object, integrate this data into the avoidance algorithm software, and perform an avoidance maneuver.
- **DR1.8** The total cost of the test bed system shall be less than $5000.

Requirements

FR2 The test system shall be capable of detecting incoming objects (active or inactive) in a representative collision scenario.

- **DR2.1** The test system shall be capable of making at least two position and relative velocity measurements in sufficient time to avoid collision.
- **DR2.2** The test system shall sense the time scale (defined in DR1.1) with an uncertainty less than 0.2 seconds.
- **DR2.3** In order for the test system to sense an incoming object, a reorientation maneuver shall not be required.

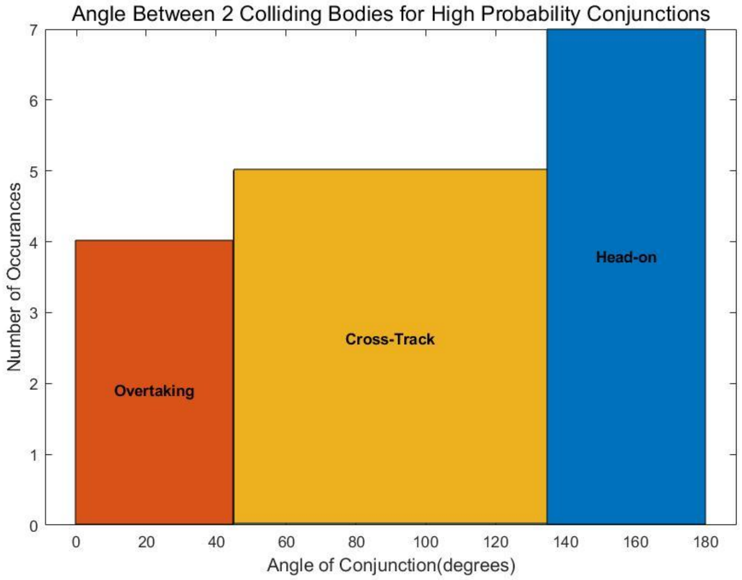
FR3 The test system shall be capable of avoiding a collision in a representative collision scenario without the test system technology performance levels exceeding those of current, full-scale spacecraft hardware.

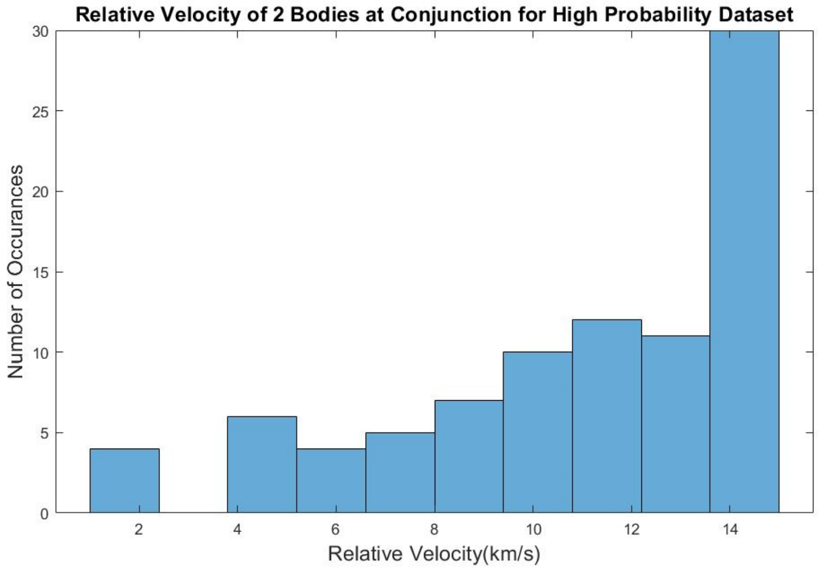
- **DR3.1** The test system shall generate sufficient force to avoid a collision with the covariance ellipse of the sensor package.
- **DR3.2** The first course of action in an avoidance maneuver shall not be to apply the largest capable force in the direction perpendicular to the relative velocity

High Probability Conjunctions

CelesTrack - Satellite Orbital Conjunction Reports Assessing Threatening Encounters in Space(SOCRATES)

- Lists most likely conjunctions to occur for one week span
- Head-on conjunction most common, followed by cross-track then overtaking


Angle Between 2 Colliding Bodies for High Probability Conjunctions


Relative Velocity of 2 Bodies at Conjunction for High Probability Dataset

State Estimation

Method	Pros	Cons
Kalman Filter	● Does not require data storage ● Able to estimate an evolving state ● More accurate than linear least squares	● Will break if a bad initial guess is used ● More difficult to implement than linear least squares
Batch Method	● Converges faster than the Kalman filter ● More accurate than the Kalman filter	● More Computationally Expensive than the Kalman filter (requires storage of data and inverting growing matrices) ● Cannot begin until a batch of data arrives
Linear Least Squares	● Can update estimate of a static parameter ● Easy to implement ● Less complex than the Kalman filter	● Least accurate

Scenario scaling

Assumptions:

- Both objects are in circular orbits
- Both objects have same altitude
- Object paths propagated from same location, collision assessment happens at 1 orbital period

Orbit Generation:

1. Objects given initial position and velocity vectors
2. Runge-Kutta solution via ode45(MATLAB)

2-body circular orbit Governing Eq:

$$|v| = \sqrt{\frac{GM}{R}} \quad \vec{a} = -\frac{v^2}{|r|}\hat{r}$$

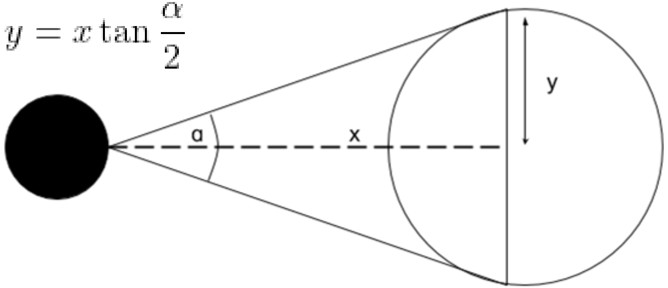
$$T = 2\pi\sqrt{\frac{|r|^3}{\mu}}$$

Analysis:

- Object 2 motion relative to object 1
- Object 2 relative position projected onto viewing plane of object 1

Sensor Model: Geometry

$$y = x \tan \frac{\alpha}{2}$$



Clohessy–Wiltshire equations

$$\ddot{x} = 3n^2x + 2n\dot{y}$$

$$\ddot{y} = -2n\dot{x}$$

$$\ddot{z} = -n^2z$$

$$n = \sqrt{\frac{\mu}{a^3}}$$

Sensor Package: Available Options

Some viable options:

- RPLidar A1M8 360 degree Laser scanner
    - Built in serial port and USB interface. Open source SDK and tools. Can interface with Arduino (C/C++) and firmware libraries can be found online.

- LiDAR Lite v3HP
    - Can be interfaced with an Arduino (C/C++)

- Lightware LW20-C
    - Serial and I2C interfaces. Software packages available for integration with PC

- LeddarTech VU8 Channel LiDAR Module
    - USB, CAN, serial interfaces. Interfaces using Leddar Enabler SDK which provides a programming interface

Sensor Package:

Sensor Uncertainty					
Sensor Package	**Range Uncertainty**	**FOV Uncertainty**	**Beam Divergence Uncertainty**	**Sampling Rate**	**Time Uncertainty**
RPLiDAR A1M8	0.5 mm		1 deg	Linear - 5.5 Hz	1 ms
LiDAR Lite V3HP	5 cm (Range<2m) 2.5 cm (Range>=2m)		0.3 deg	1 kHz	1 ms
LW20 SF20	10 cm		0.3 deg	48 Hz	1 ms
Vu8	5 cm	1 deg (HFOV) 0.6 deg (VFOV)		100 Hz	1 ms

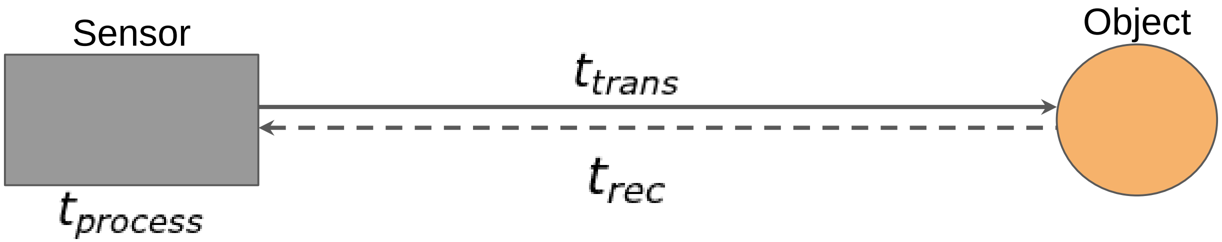
Sensor Package: Model

Error Equation on TOF Distance Measurement:

$t_{process} = t_p + \delta t_p$

$t_{trans} = t_t + \delta t_t$

$t_{rec} = t_r + \delta t_r$



$$\Delta t = t_{process} + t_{rec} + t_{trans}$$

Assumptions:

- Independent Random Variables

$$Range = SOL\frac{\Delta t}{2}$$

Since time received & time transmitted but range is only one of these

$$\delta\Delta t = \sqrt{(\frac{\partial\Delta t}{\partial t_{process}}\delta t_p)^2 + (\frac{\partial\Delta t}{\partial t_{trans}}\delta t_t)^2 + (\frac{\partial\Delta t}{\partial t_{rec}}\delta t_r)^2}$$

$$\delta Range = \frac{\partial Range}{\partial\Delta t}\delta\Delta t$$

$$\delta Range = \frac{1}{2}SOL\sqrt{(\delta t_p)^2 + (\delta t_t)^2 + (\delta t_r)^2}$$

Phase Shift Velocity Uncertainty

Error Equation on Phase Shifting Measurements:

Assumptions:

- Angle between the horizontal and the object θ=0 (object is in line with the sensor
- $V_{MovingTarget} << c$ (c=speed of light)
- The velocity is only the radial component of the velocity
- Variables are independent and random

$$V_{MovingTarget} = \frac{(f_{rec} - f_{trans})c}{2}$$

$$f_{rec} = f_r + \delta f_r$$

$$f_{trans} = f_t + \delta f_t$$

$$\delta V_{MovingTarget} = \sqrt{(\frac{\partial V_{MovingTarget}}{\partial f_{rec}} \delta f_r)^2 + (\frac{\partial V_{MovingTarget}}{\partial f_{trans}} \delta f_t)^2}$$

$$\delta V_{MovingTarget} = \frac{c}{2}\sqrt{(\delta f_r)^2 + (\delta f_t)^2}$$

Mean Bias Error

Error Equation for Mean Error:

Assumptions:

- Object is launched at a velocity that remains constant

$$ME = \frac{(Vel_{meas1} - Vel_{actual}) + (Vel_{meas2} - Vel_{actual}) + (Vel_{measN} - Vel_{actual})}{N}$$

Where
N=Number of measurements
ME=Mean error of the velocity (aka mean deviation MD, mean bias, or bias)

Velocity Uncertainty

Error Equation from the calculation of velocity using multiple measurements:

Assumptions:

- All range measurements and time measurements are independent random variables

$$\delta V_{MovingTarget} = \frac{x_1 - x_0}{t}$$

$$x_1 = x_{1,val} + \sigma_{x_1}$$

$$x_0 = x_{0,val} + \sigma_{x_0}$$

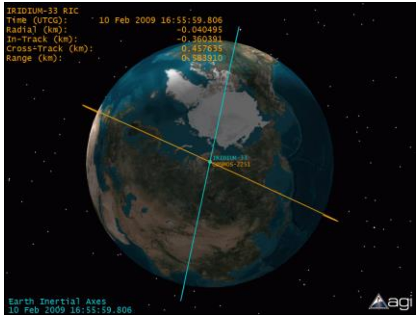
$$t = t_{val} + \sigma_t$$

$$\sigma_v = \sqrt{\left(\frac{\partial V_{MovingObject}}{\partial x_1}\sigma_{x_1}\right)^2 + \left(\frac{\partial V_{MovingObject}}{\partial x_0}\sigma_{x_0}\right)^2 + \left(\frac{\partial V_{MovingObject}}{\partial t}\sigma_t\right)^2}$$

$$\sigma_v = \sqrt{\left(\frac{1}{t}\sigma_{x_1}\right)^2 + \left(\frac{-1}{t}\sigma_{x_0}\right)^2 + \left(\frac{-(x_1 - x_0)}{t^2}\sigma_t\right)^2}$$

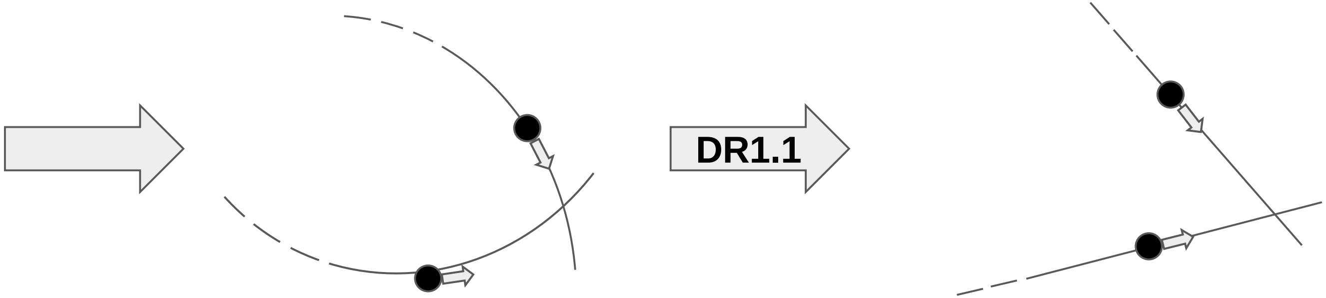
$$\frac{\sigma_v}{v} = \frac{1}{v}\sqrt{\frac{2\sigma_x^2}{t^2} + \frac{v^2\sigma_t^2}{t^2}}$$

Specific Objectives



```
IRIDIUM-33 RIC
Time (UTCG):      10 Feb 2009 16:55:59.806
Radial (km):                     -0.040495
In-Track (km):                   -0.360391
Cross-Track (km):                 0.457635
Range (km):                       0.583910
```

Credit: CelesTrak

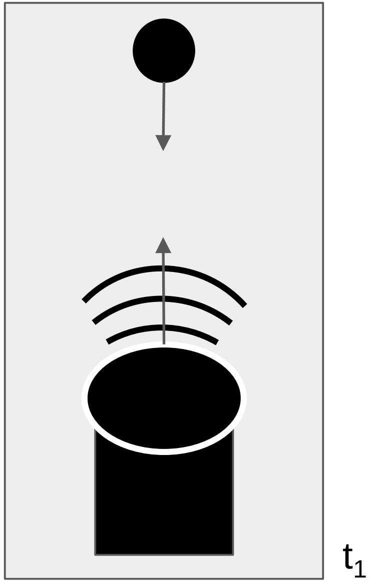


DR1.1 The test bed shall support representative collision scenarios that preserve a time scale (ratio of distance to velocity) up to 100-150 seconds (LEO) and/or 1600 seconds (GEO)

<u>Upper</u> time scale bound needed to ensure rectilinear motion is preserved

Specific Objectives

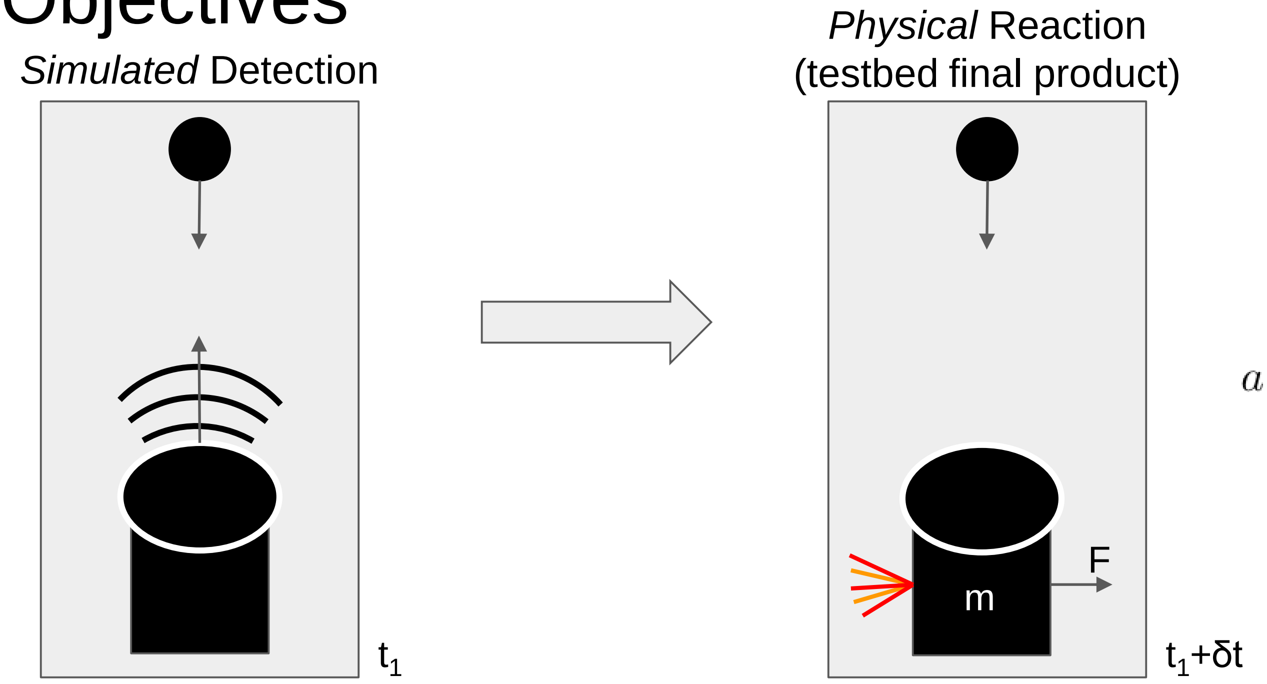
Simulated Detection



- Uncertainty parameters for existing sensors are used to develop sensor model
  - CAST is not developing a sensor package
  - Physical sensor will be used as a subsystem test for sensor model
- Objective is to develop representative testbed
- Successful testbed depends on ability to scale up

Detection is assumed to be possible

Specific Objectives



$$a = \frac{F}{m}$$

DR1.3 The lower bound of the time scale shall be 10 seconds

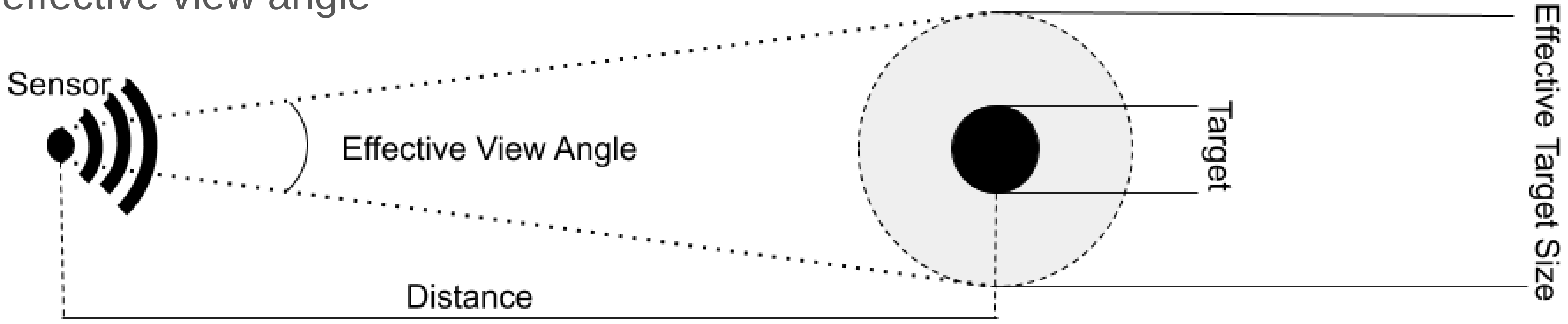
<u>Lower</u> time scale needed to preserve current spacecraft propulsive abilities

2-D Similarity Parameters - Sensor

Sensor behavior is governed by two major parameters

- Effective target size
  - Depends on the sensor being used
- Distance from target

Together these parameters create the dimensionless similarity parameter: effective view angle



2-D Similarity Parameters - Maneuvering

The ability for an object to maneuver is governed by two parameters

- Available actuation force
- Mass

Together via the familiar equation of Newton’s second law these parameters create the dimensional similarity parameter of acceleration

2-D Similarity Parameters - Time

The third parameter, required to fix the test environment, is time.

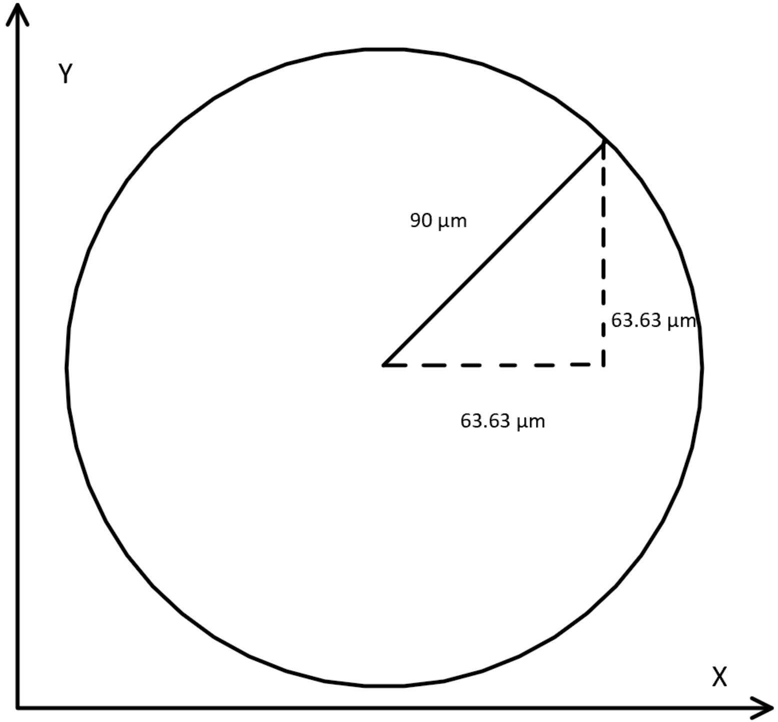
Time can be approached via two methods

- Sensing distance and mean relative velocity
- Maneuvering distance and acceleration
  - Maneuvering distance is governed by the error ellipsoid of the sensor

The sensor time provides the maximum response time, while the maneuvering time provides the minimum response time.

Motor Feasibility:

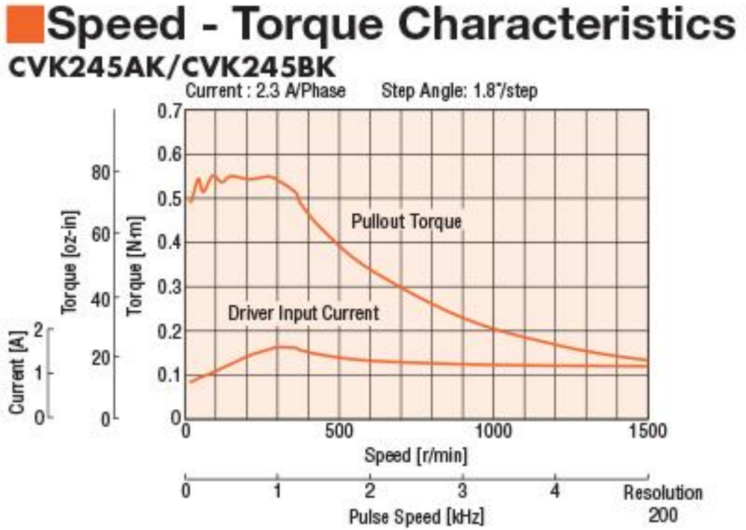
- General Design
  - 1 motor in each axis
  - Motors include encoders
- Feasibility Parameters
  - Speed
    - 0.1 m/s
  - Resolution
    - 90 μm - 2% error
    - 63.63 μm in each x,y
  - Acceleration
    - 0.00122 m/s^2
    - Bottom Motor T = 0.0022 oz in
    - Top Motor T = 0.0095 oz in



Positional Resolution Component Circle

Motor Feasibility: Stepper

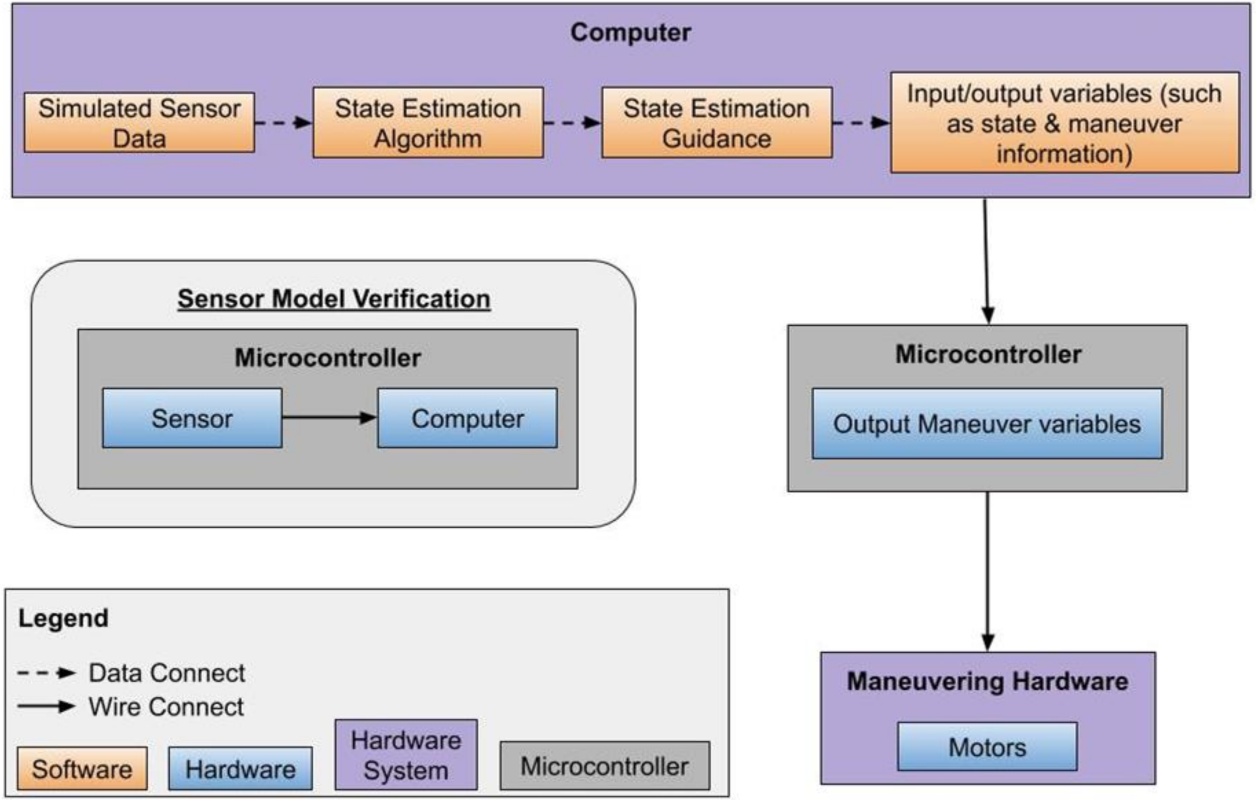
- Basic Operation
  - Rotor teeth and Stator teeth
  - Best driven by commercial drivers
- Feasibility
  - Resolution ☑
    - 5mm spindle - 40 μm/rev at ¼ steps
  - Speed ☑
    - 100 mm/s - 2.55 kHz step rate
  - Acceleration ⊖
    - Torque @ ¼ steps is 38.2% max Torque
    - Torque decreases with speed



● The pulse input circuit responds up to 1 MHz with a pulse duty of 50%.

$$T_{INC} = T_{HFS} \cdot \sin\left(\frac{90}{SDR}\right)$$

Electronics Block Diagram



Computer

Simulated Sensor Data - - → State Estimation Algorithm - - → State Estimation Guidance - - → Input/output variables (such as state & maneuver information)

Sensor Model Verification

Microcontroller

Sensor → Computer

Microcontroller

Output Maneuver variables

Maneuvering Hardware

Motors

Legend

- - → Data Connect

→ Wire Connect

Software

Hardware

Hardware System

Microcontroller

Electronics

Sensor Communications:

- Sensors were found at multiple price ranges and specs
    - Capable of interfacing with a microcontroller such as an arduino, Raspberry Pi, or simply just a USB connection with the necessary packages included with the sensor.
    - There were 360 degree LiDAR sensors as well as stationary LiDAR sensors found with these capabilities
- The languages used for the chosen sensors were C/C++ and python.
    - Most sensors were capable of interacting with a microcontroller that uses C/C++
    - The group is familiar with C/C++ from the curriculum at CU which makes this a feasible option
- From the 8 sensors that were looked into, the lowest voltage needed to operate the sensor was 5V and the highest voltage needed was 12V

Electronics

Controllers:

- Given the easy integration of LiDAR sensors, finding a controller(s) to handle sensor testing is not an issue.
  - Arduino is easily-integratable with a number of LiDAR sensors, and provides many microcontroller options.
  - Plenty of alternatives to Arduino exist that are fully compatible with Arduino programs.

- Plenty of controllers exist that can provide motor needs as well.
  - And for those that can't, compatibility tools are available, like the CAN Arduino interface and motor H-bridge circuits.

Electronics

Results:

- Microcontrollers and wires are low cost
- Sensors use common communication protocols
- The team has experience using the commonly-applied languages, controllers, and PC interactions
- All connections are simply wired

Motor Feasibility Calculations

- Mass breakdown
  - Top drive mass = ~1.2 kg
    - Top Slide = 0.09 kg
    - Sensor = 0.5 kg
    - Mounting = 0.5 kg
    - Timing belt = 0.025 kg
  - Bottom drive mass = ~5 kg
    - Top drive mass = 1.2 kg
    - 2 slides = 0.18 kg
    - 2 mounts = 1 kg
    - Top motor = 0.5 kg
    - 2 belts = 0.05 kg
    - Top rail = 1 kg
    - Misc = 1 kg

- Forces and Torques
  - Needed a in any x,y = 0.00146 m/s^2
  - Drive pulley radius ~ 10mm
  - Top drive mass
    - F = 1.752 mN
    - τ = 17.52 E -6 N*m = 0.0025 oz*in
  - Bottom drive mass
    - F = ~7 mN
    - τ = 67 E -6 N*m = 0.0095 oz*in