A General Framework for Trajectory Optimization with Respect to Multiple Measures

SBIR Phase II AF01-156
Technical Presentation
By David Diel
2011 JUN 29

Introduction

- Multi-sensor navigation
- GPS alternatives such as vision
- Competition for better algorithms
- Greater accuracy at a lower cost
- Nonlinear filtering
- Open source development

TOMMAS

Trajectory **O**ptimization **M**anager for **M**ultiple **A**lgorithms and **S**ensors

Mission Statement





"Deliver a software testbed that helps a user to define *optimal* for a specific navigation problem by selecting components from a library of dynamic models, sensing algorithms, and optimization algorithms."



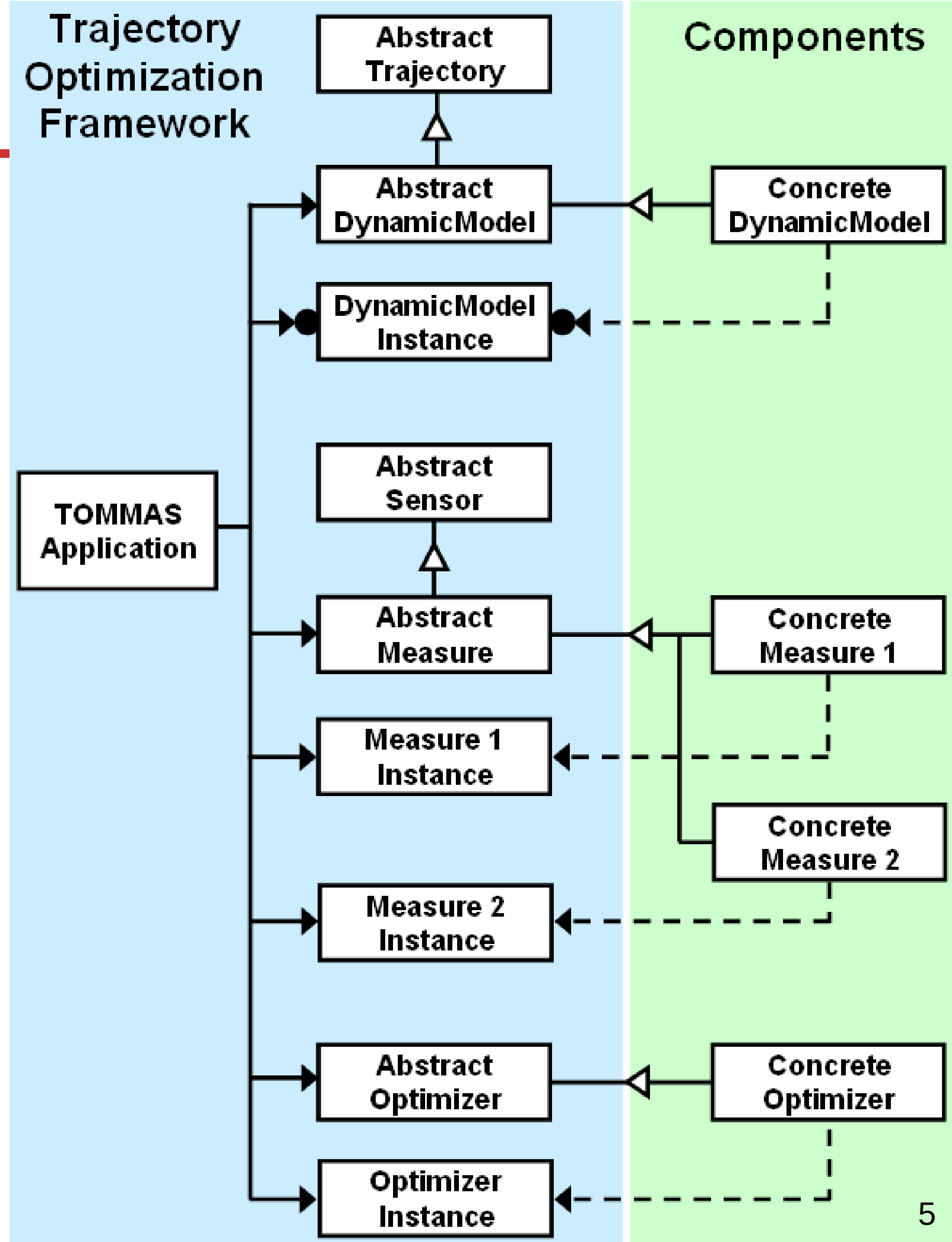

Potential Benefits

- For the Systems Engineer
  - Ease of creating Interface Control Documents (ICDs).
  - Automated testing of algorithms and sensors together.
  - Encouraging innovation by leveling the playing field.
  - Lower cost GOTS devices due to commoditization in long-term.

- For the Warfighter
  - Convenience of device interoperability and potential hot-swapping.
  - Mitigates sensor cutouts by augmenting GPS/INS with cameras, altimeters, LIDAR, RF signals of opportunity, etc.

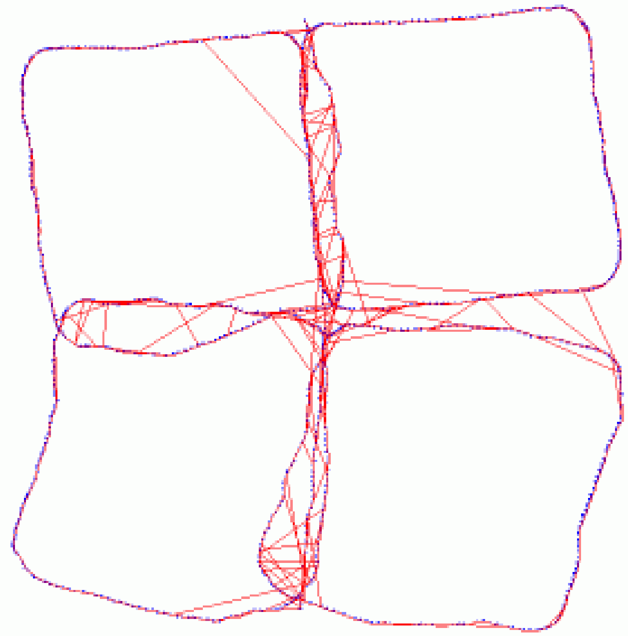
Approach

- Break existing solutions into interchangeable parts.
- Use object-oriented design to create abstract interfaces in C++, Matlab, and Python.
- Implement polymorphism via the Factory Design Pattern.



Navigation by Trajectory Optimization

- Assumptions
  - Newtonian mechanics
  - Global reference frame
  - Rigid body dynamics
  - Time domain $[t_0, t_K]$
  - 6-DoF range in $\mathbb{R}^3 \times \mathbb{S}^3$
  - Finite forces (C^1 continuity)

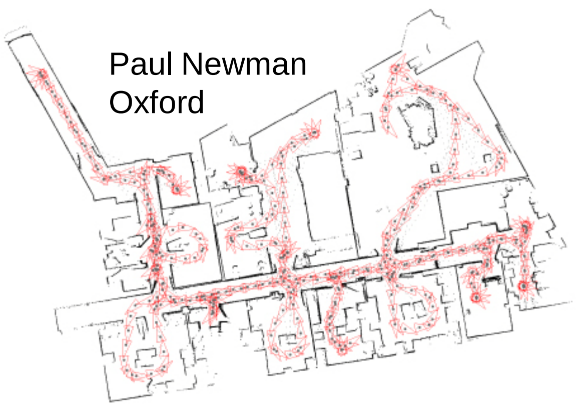


- Given
  - Stochastic dynamics
  - Multiple calibrated sensors

- Find
  - Maximum likelihood trajectory
  - Accuracy of results

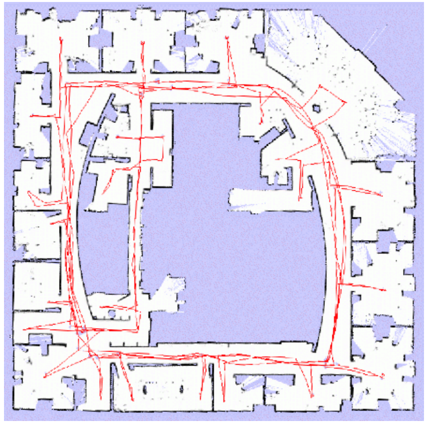
Framing the Navigation Problem

- Efficient solutions to sub-problems exist in the literature (some also map the environment):



Giorgio Grisetti
La Sapienza
University of Rome

Michael Kaess
GA Tech / MIT

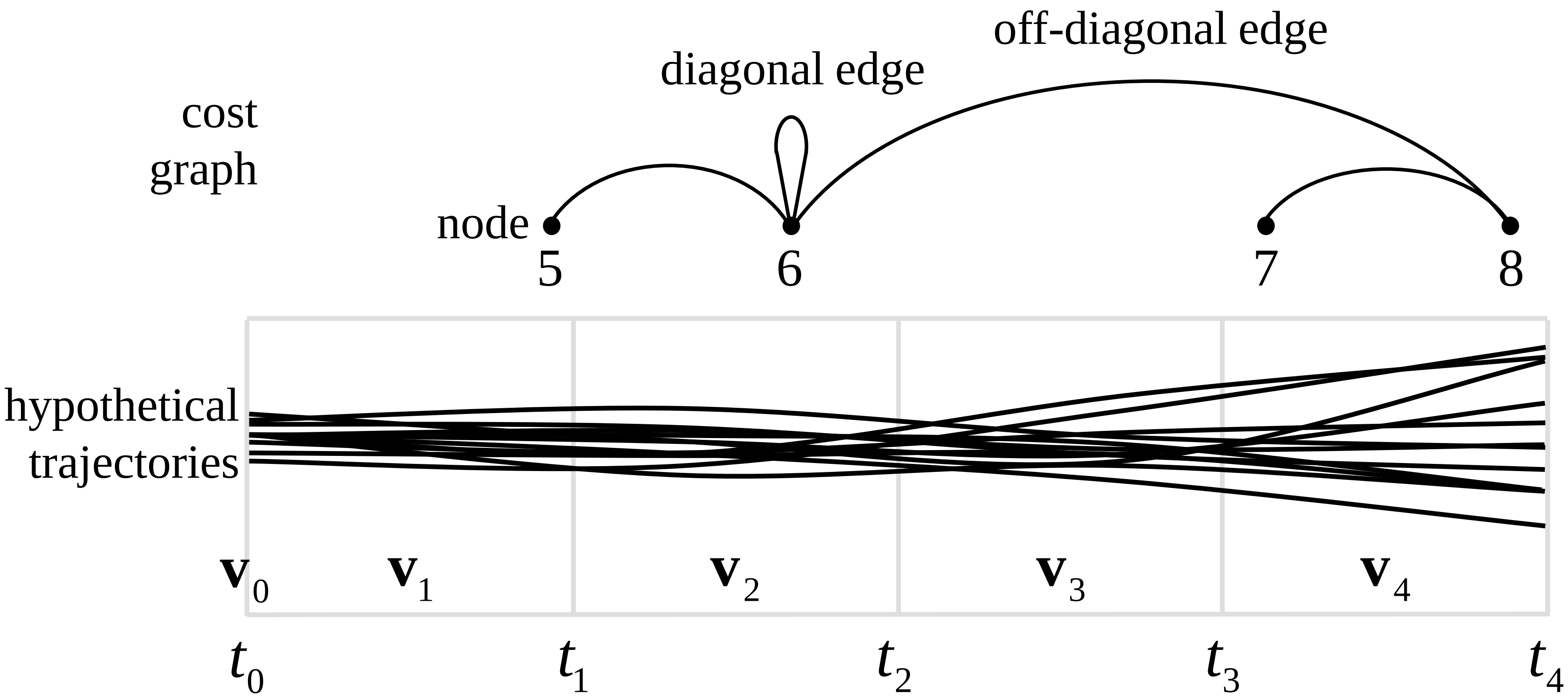


- The generalized problem can be expressed in this form:

$$\mathbf{v}^* = \operatorname*{argmin}_{\mathbf{v}\in\mathbb{V}} \left\{ \sum_{k\in\mathbb{K}} r(\mathbf{v},k) + \sum_{m\in\mathbb{M}} \sum_{(a,b)\in\mathbb{A}} s_m(\mathbf{u},\mathbf{x},(a,b)) \right\}$$

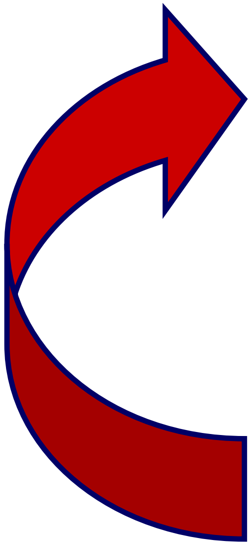
$$\mathbf{x} = \mathbf{F}(\mathbf{v},\mathbf{u})$$

Visualizing the Problem Structure



- Each block of parameters corresponds to a discrete time period.
- Measurements have graph structure (node = time, edge = measure).
- Trajectories are not required to have a graph structure.

Optimization Concept



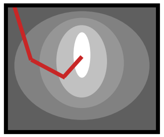
Step 1) Generate initial parameters.

Step 2) Compute prior costs from DynamicModel.

Step 3) Compute costs from each Measure.

Step 4) Generate new parameters.

- Within this framework, there are many possible strategies:



Gradient

Genetic

Simplex

Other

DynamicModel Details

- Stochastic motion model in nonlinear functional form:

$$\mathbf{x} = \mathbf{F}(\mathbf{v}, \mathbf{u})$$

- Can depend on sensor data (i.e. strapdown mechanization):
- Driven by parameters $\mathbf{v}$ that are not known in advance (i.e. noise, disturbance, perturbation).
- Prior statistical information about the parameters:

$$r(\mathbf{v}, k) = -\log\left(\frac{\mathrm{P}_{\mathbf{v}}(\mathbf{v} | k)}{\left\|\mathrm{P}_{\mathbf{v}}(\mathbb{V} | k)\right\|_{\infty}}\right)$$

Measure Details

- Sensing model in nonlinear functional form:

$$s_m\left(\mathbf{u},\mathbf{x},(a,b)\right) = -\log\left(\frac{\mathrm{P}_{\mathbf{u}|\mathbf{v}}\left(\mathbf{u}\,\middle|\,\mathbf{x},(a,b),m\right)}{\left\|\mathrm{P}_{\mathbf{u}|\mathbf{v}}\left(\mathbb{U}\,\middle|\,\mathbf{x},(a,b),m\right)\right\|_{\infty}}\right)$$

- It evaluates how much the data deviates from a statistical model given the trajectory.
- As a mathematical measure, it systematically assigns nonnegative numbers (costs) to each element in its domain. The domain is the space of possible data.
- Multiple points in the trajectory space may map to the same point in the measure space, so a measure may not be sensitive to some trajectory deviations.

Incremental Measure: Visual Tracking

Feature Sampling

Epipolar Projection

Participation

- **Use** the framework for free under a permissive license agreement.
- **Contribute** as a developer and stay on the cutting edge.
- **Own** the intellectual property for components that you produce.
- **Download** the code from our open source repository.

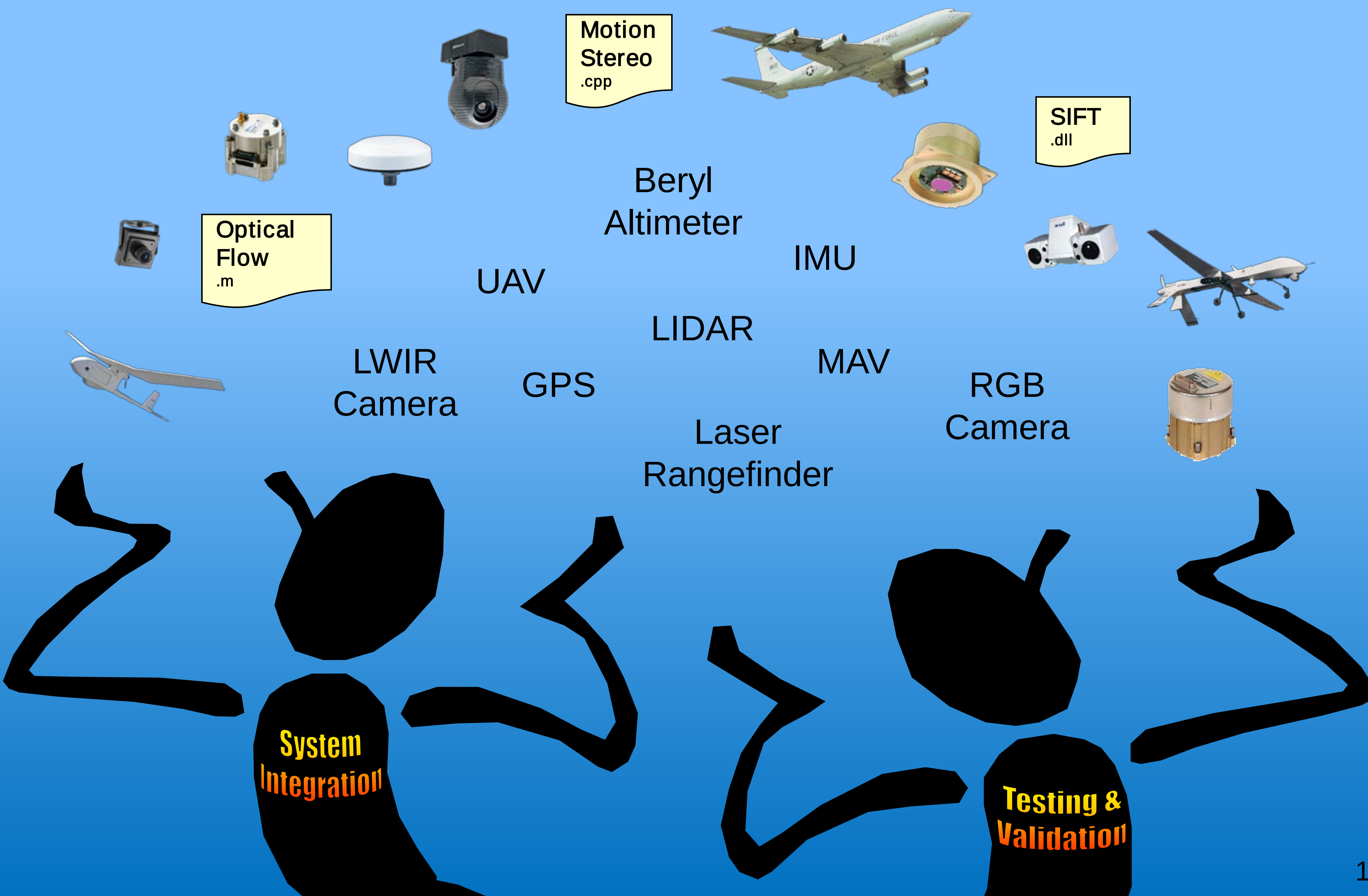
Keyword on Google Code:

functionalnavigation

functionalnavigation

(Additional Slides)

Managing Multiple Algorithms and Sensors



System Integrator Perspective

Optimizer $\rightarrow$
DynamicModel $\rightarrow$
Measure_1 $\rightarrow$
Measure_2 $\rightarrow$
Measure_m $\rightarrow$
$\vdots$ $\rightarrow$
Measure_M $\rightarrow$

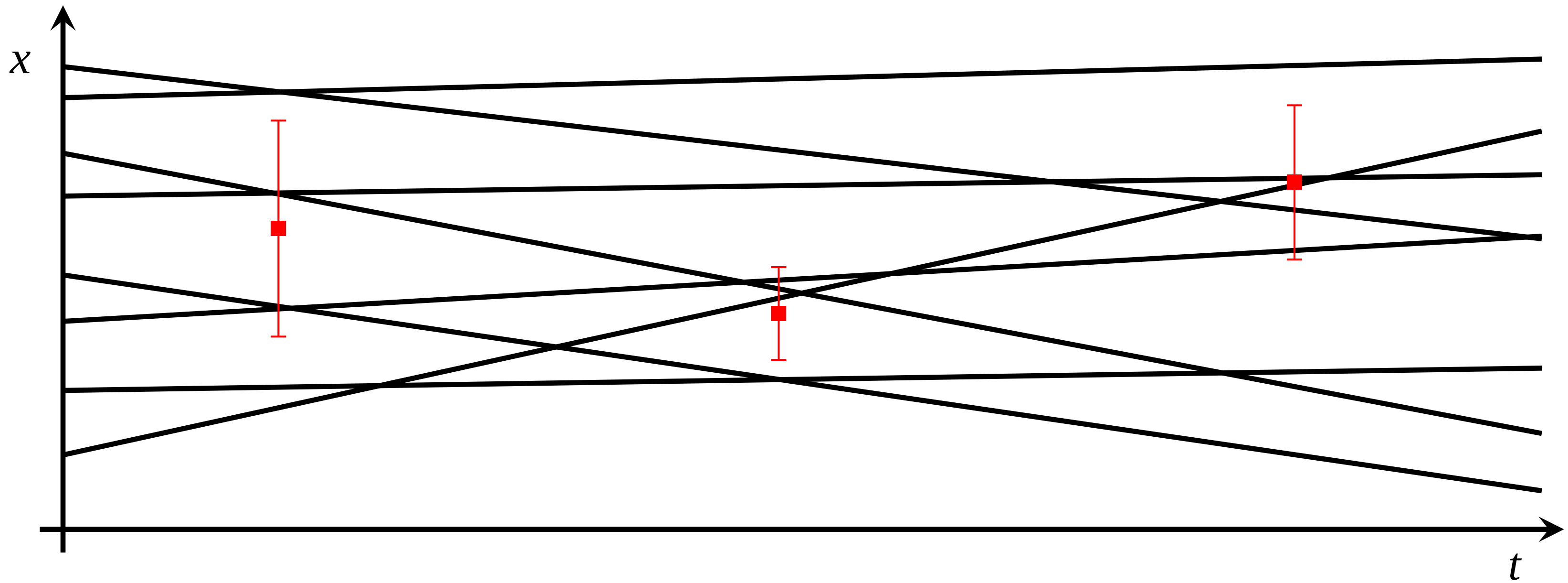
TOMMAS

$\rightarrow (\text{Trajectory},\text{Cost})_1$
$\rightarrow (\text{Trajectory},\text{Cost})_2$
$\rightarrow (\text{Trajectory},\text{Cost})_j$
$\rightarrow \vdots$
$\rightarrow (\text{Trajectory},\text{Cost})_J$

Configuration Set

Solution Set

Trajectory Optimization: Weighted Least Squares



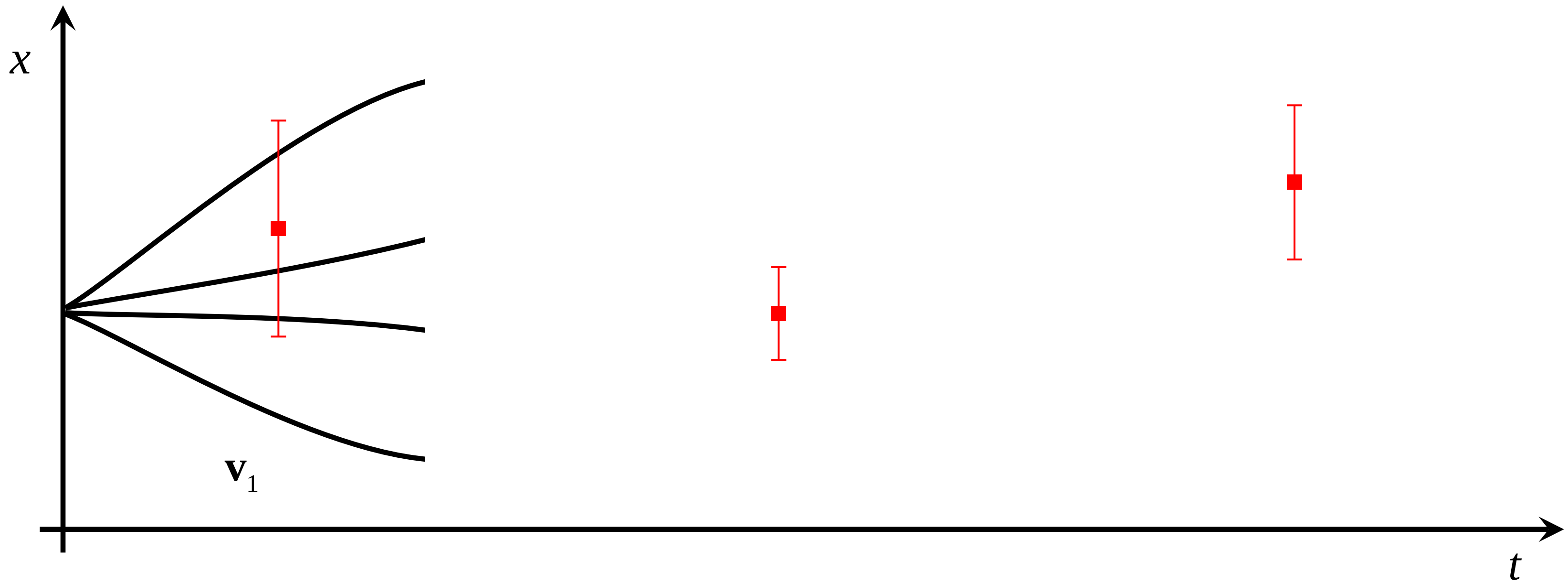
Sensor Data $\quad \mathbf{u} = \left\{ \begin{matrix} t_0,\ t_1,\ t_2, \\ p_0, p_1, p_2, \\ \alpha_0, \alpha_1, \alpha_2 \end{matrix} \right\}$ "Time, Position, and Weight"

Dynamic Model $\quad \mathbf{x} = \mathbf{v}_0 + \mathbf{v}_1 t$ "Constant Velocity"

Functional Measure $\quad s_0\left(\mathbf{u}, \mathbf{x}, (a, a)\right) = \left(\mathbf{x}(t_a) - p_a\right)^2 \alpha_a$ "Weighted Distance Squared"

Optimization Problem $\quad \mathbf{v}^* = \underset{\mathbf{v} \in \mathbb{R}^2}{\operatorname{argmin}} \left\{ \sum_{a=1}^{3} s_0\left(\mathbf{u}, \mathbf{x}, (a, a)\right) \right\}$ "Weighted Least Squares"

Trajectory Optimization: Weighted Least Squares



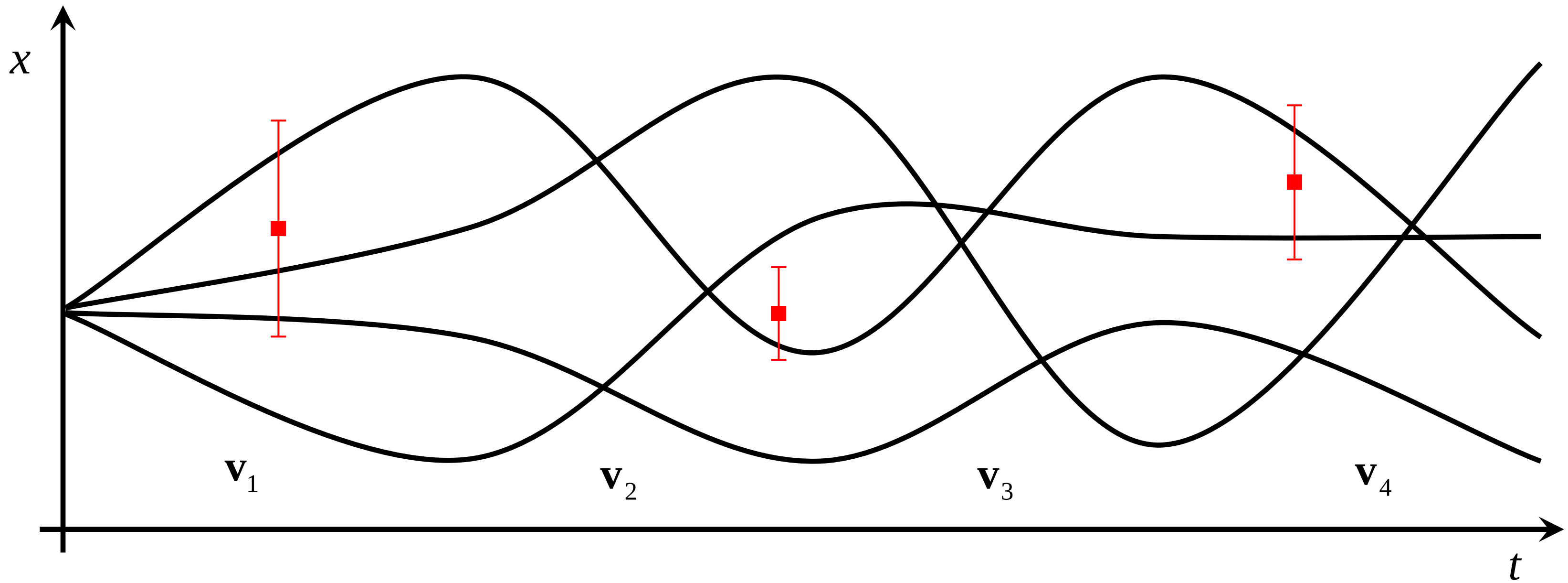
Sensor Data $\quad \mathbf{u} = \left\{ \begin{matrix} t_0,\ t_1,\ t_2, \\ p_0, p_1, p_2, \\ \alpha_0, \alpha_1, \alpha_2 \end{matrix} \right\}$ "Time, Position, and Weight"

Dynamic Model $\quad \mathbf{x} = \mathbf{F}(\mathbf{v}, \mathbf{u})$ "Generalized Rigid Body Dynamics"

Functional Measure $\quad s_0\left(\mathbf{u}, \mathbf{x}, (a, a)\right) = \left(\mathbf{x}(t_a) - p_a\right)^2 \alpha_a$ "Weighted Distance Squared"

Optimization Problem $\quad \mathbf{v}^* = \underset{\mathbf{v} \in \mathbb{V}}{\operatorname{argmin}} \left\{ \sum_{a=1}^{3} s_0\left(\mathbf{u}, \mathbf{x}, (a, a)\right) \right\}$ "Weighted Least Squares"

Trajectory Optimization: Generalized Least Squares



Sensor Data $\mathbf{u} = \left\{ \begin{matrix} t_0, \ t_1, \ t_2, \\ p_0, p_1, p_2, \\ \alpha_0, \alpha_1, \alpha_2 \end{matrix} \right\}$ "Time, Position, and Weight"

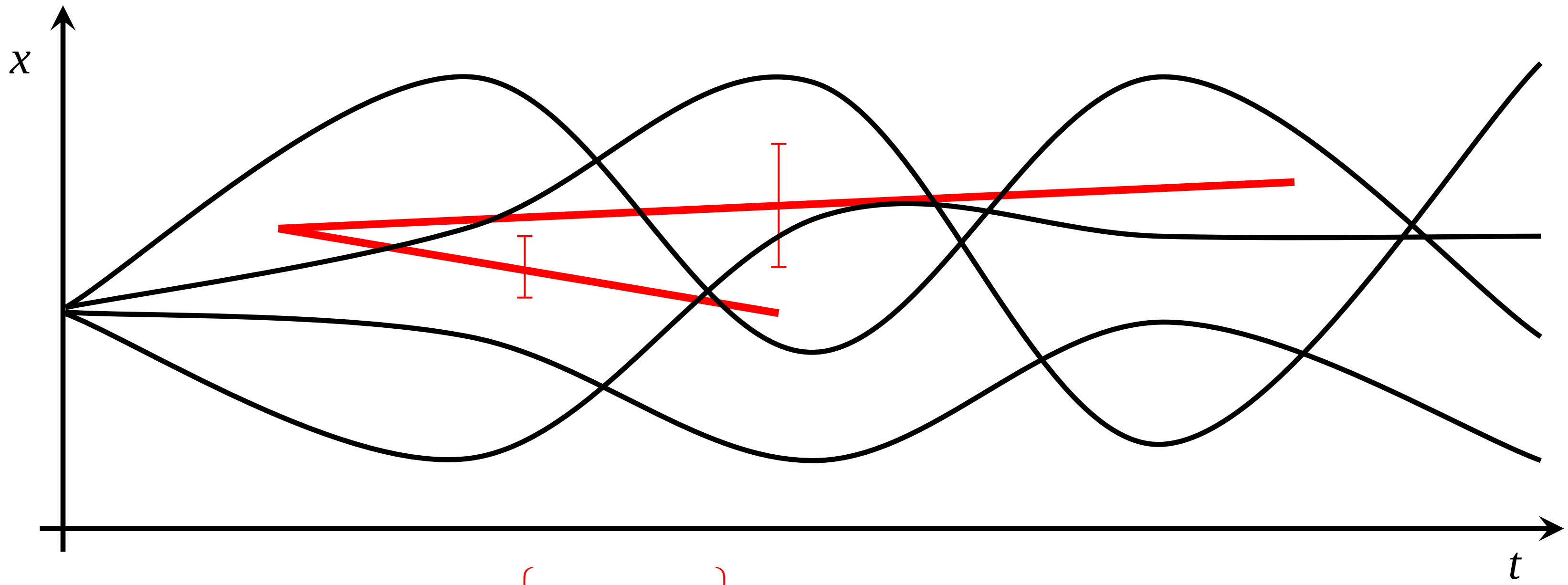
Dynamic Model $\mathbf{x} = \mathbf{F}(\mathbf{v}, \mathbf{u})$ "Generalized Rigid Body Dynamics"

$r(\mathbf{v}, k)$ "Prior Motion Information"

Functional Measure $s_0(\mathbf{u}, \mathbf{x}, (a, a)) = (\mathbf{x}(t_a) - p_a)^2 \alpha_a$ "Weighted Distance Squared"

Optimization Problem $\mathbf{v}^* = \underset{\mathbf{v} \in \mathbb{V}}{\operatorname{argmin}} \left\{ \sum_{k \in \mathbb{K}} r(\mathbf{v}, k) + \sum_{a=1}^{3} s_0(\mathbf{u}, \mathbf{x}, (a, a)) \right\}$ "Weighted Least Squares with Prior"

Trajectory Optimization: Relative Least Squares



Sensor Data $\mathbf{u} = \begin{Bmatrix} t_0,\ t_1,\ t_2, \\ \psi_{(0,1)}, \psi_{(0,2)}, \\ \alpha_{(0,1)}, \alpha_{(0,2)} \end{Bmatrix}$ "Time, Relative Position, and Weight"

Dynamic Model $\mathbf{x} = \mathbf{F}(\mathbf{v}, \mathbf{u})$ "Generalized Rigid Body Dynamics"

$r(\mathbf{v}, k)$ "Prior Motion Information"

Functional Measure $s_0(\mathbf{u}, \mathbf{x}, (a,b)) = \left(\mathbf{x}(t_b) - \mathbf{x}(t_a) - \psi_{(a,b)}\right)^2 \alpha_{(a,b)}$ "Relative Trajectory Measure"

Optimization Problem $\mathbf{v}^* = \underset{\mathbf{v}\in\mathbb{V}}{\operatorname{argmin}} \left\{ \sum_{k\in\mathbb{K}} r(\mathbf{v}, k) + \sum_{(a,b)\in\mathbb{A}} s_0(\mathbf{u}, \mathbf{x}, (a,b)) \right\}$ "Weighted Least Squares with Prior"

Framework Variables and Functions

$\mathbb{K} = \{k : k \in \mathbb{Z}_+, k \le K\}$ discrete time index

$\mathbb{T} = \{t : t \in \mathbb{R}, t_0 \le t \le t_K\}$ continuous time index

$\mathbb{M} = \{m : m \in \mathbb{Z}_+, m < M\}$ measure index

$\mathbb{A} = \left\{ \begin{matrix} (a,b) : a,b \in \mathbb{Z}_+, \\ A_m \le a \le b \le B_m, m \in \mathbb{M} \end{matrix} \right\}$ ordered pair of data nodes

$\mathbb{U} = \{\mathbf{u} : \mathbb{Z}_+ \to \{0,1\}\}$ raw sensor data

$\mathbb{V} = \{\mathbf{v} : \mathbb{K} \to \mathbb{Z}_+^{D_k}, k \in \mathbb{K}\}$ dynamic model parameters

$\mathbb{X} = \{\mathbf{x} : \mathbb{T} \to \mathbb{R}^3 \times \mathbb{S}^3\}$ continuous body trajectory

$\mathbf{F} : \mathbb{V} \times \mathbb{U} \to \mathbb{X}$ functional dynamic model

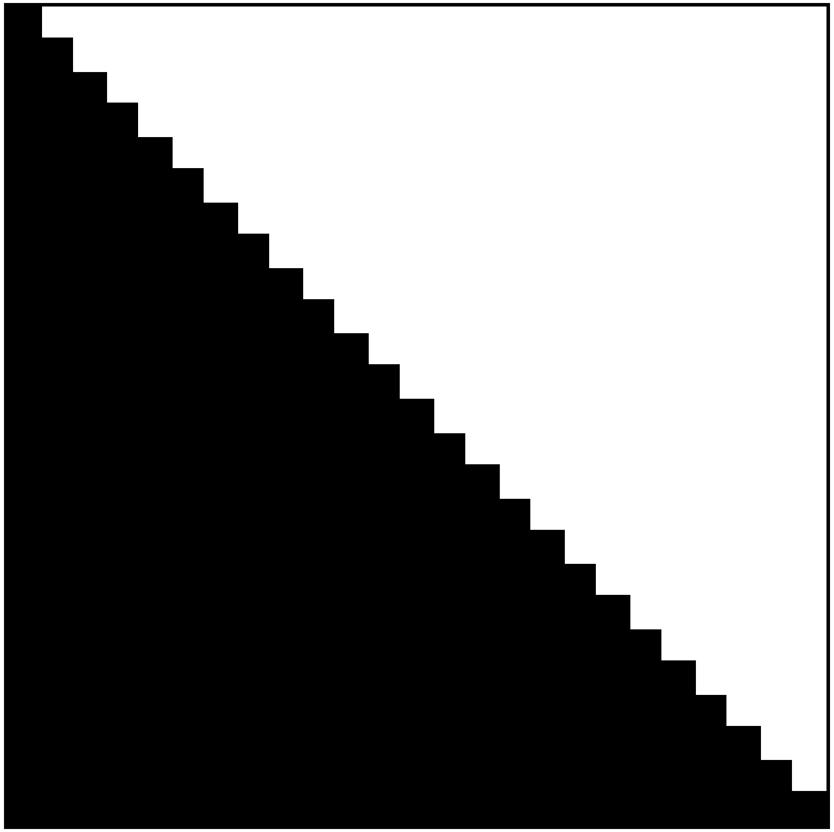
$\mathbf{p} : \mathbb{T} \to \mathbb{R}^3$ body position

$\mathbf{q} : \mathbb{T} \to \mathbb{S}^3$ body orientation

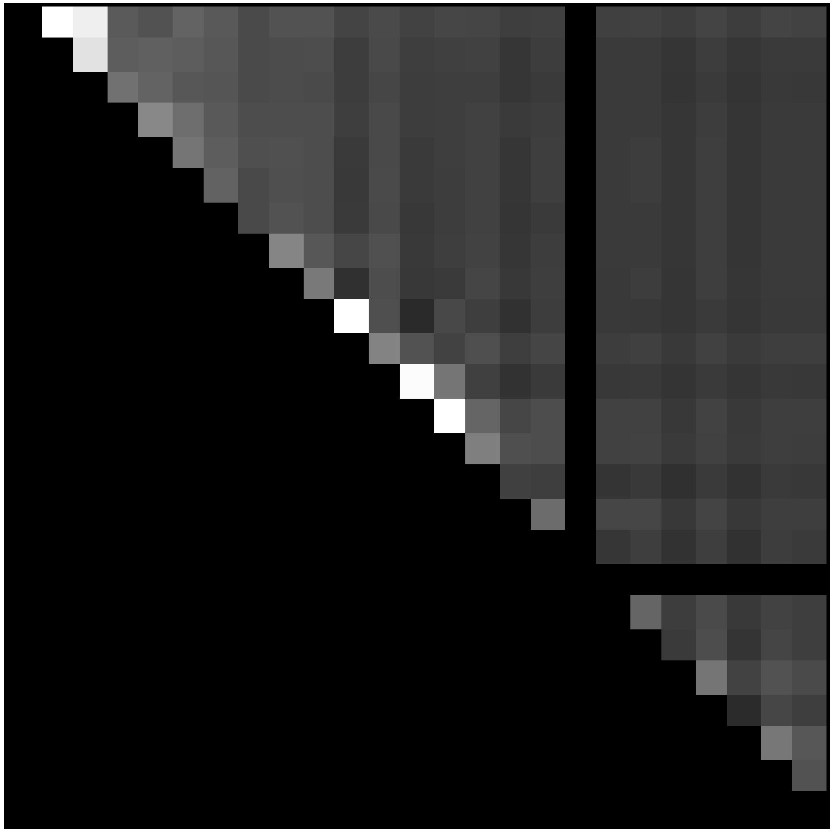
$r : \mathbb{V} \times \mathbb{K} \to \mathbb{R}_+$ prior measure

$s : \mathbb{M} \times \mathbb{U} \times \mathbb{X} \times \mathbb{A} \to \mathbb{R}_+$ conditional measure

Visualizing the Problem Structure



Adjacency Matrix



Cost Matrix

Absolute Measure: “I am near McDonalds”

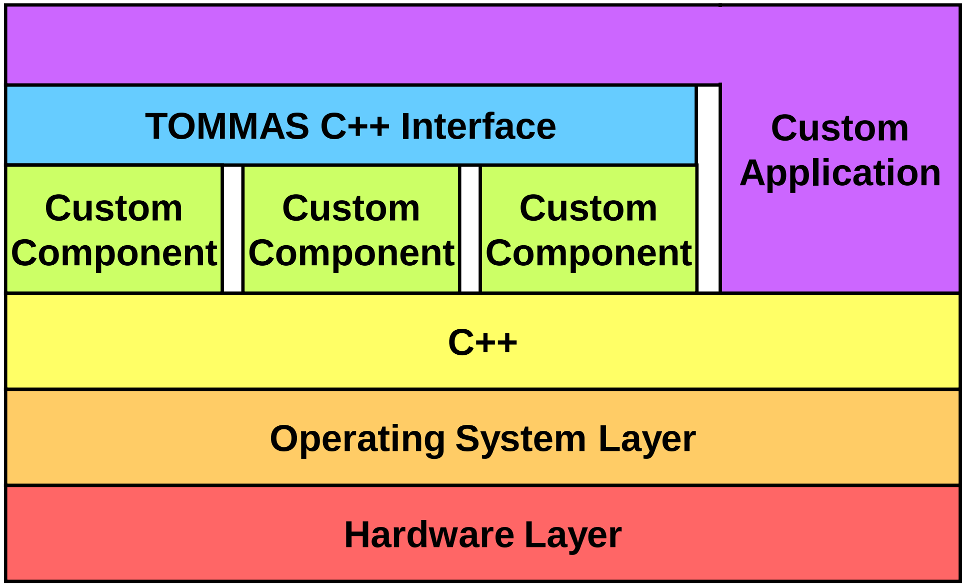
“Now I see a large Citgo sign and Soverign Bank…”

<u>Reference</u>
Stephen Von Worley. The Contiguous United States Visualized by Distance to the Nearest McDonald's. http://www.datapointed.net. Data courtesy of AggData. http://www.aggdata.com. Used with permission.

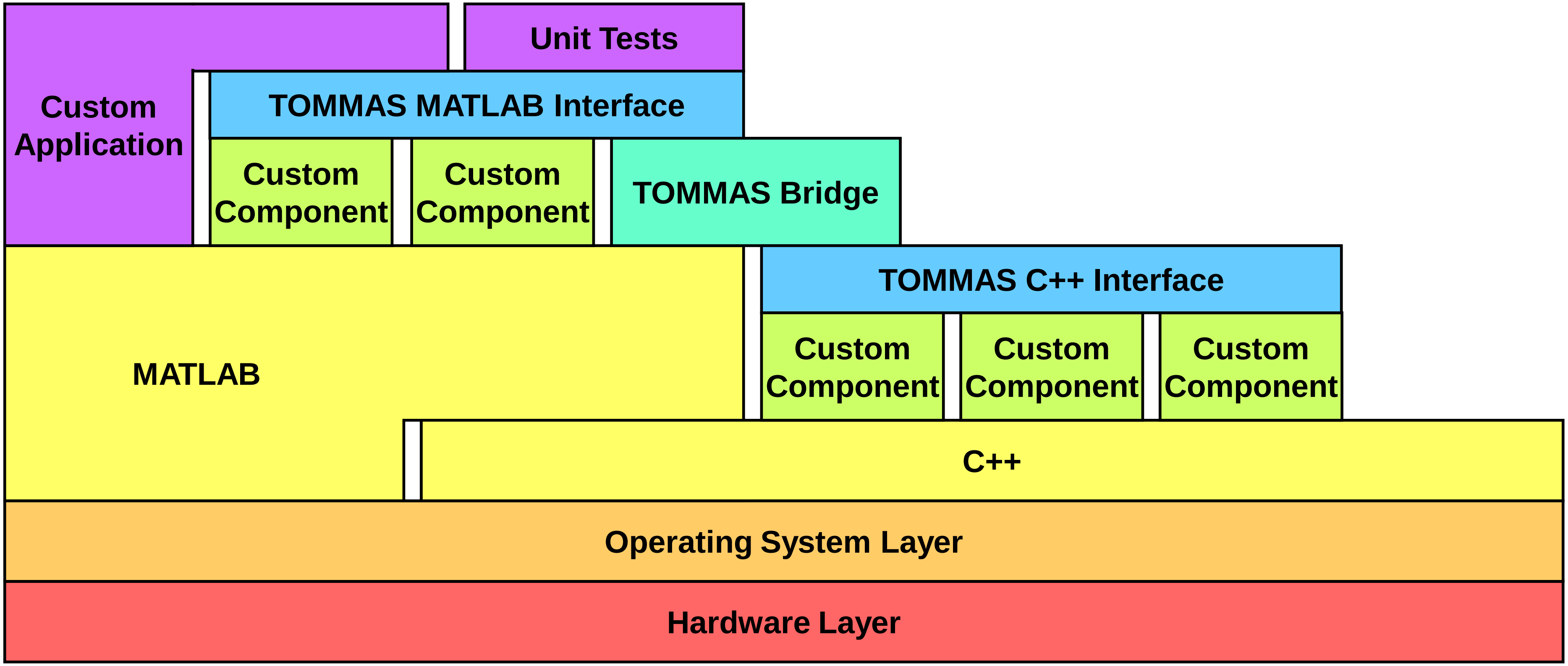
Software Licensing and Distribution

- Goals
  - Enable collaboration with universities and corporations
  - Encourage open and closed source development
- Policy for optimization framework and demo subsystems
  - Hosted at http://code.google.com/p/functionalnavigation/
  - License: BSD
  - Open source? YES
  - Copyleft for derivative works? NO
  - Allow linked code with different licenses? YES
  - Restrict use of owner’s name by others? YES
- Policy for proprietary subsystems developed by SSCI
  - Hosted at https://svn.ssci.com/repos/functionalnavigation
  - License: SBIR Data Rights
    - All rights reserved for 5 years
    - Contact SSCI with commercial inquiries

Layer Diagram for Embedded Systems



Layer Diagram for Developers



How to Implement a TOMMAS Component

- Review the paper in JNC 2011:
  - “A General Framework for Trajectory Optimization with Respect to Multiple Measures”
- Navigate to the online repository
  - http://code.google.com/p/functionalnavigation
- Review the Wiki pages
- Download the TOMMAS framework code
- Refer to the internal code documentation in the C++ header files located in the “trunk/+tom” and “trunk/+antbed” directories
- Use the components in “trunk/components” as templates
  - Algorithm library includes: optical flow, bundle adjustment, EKF, gradient descent, genetic algorithm, simplex, SLAM methods, generic dynamics, and flight dynamics