Test and Evaluation/Science and Technology Program

C4I & Software Intensive Systems Test (C4T) Test Technology Area

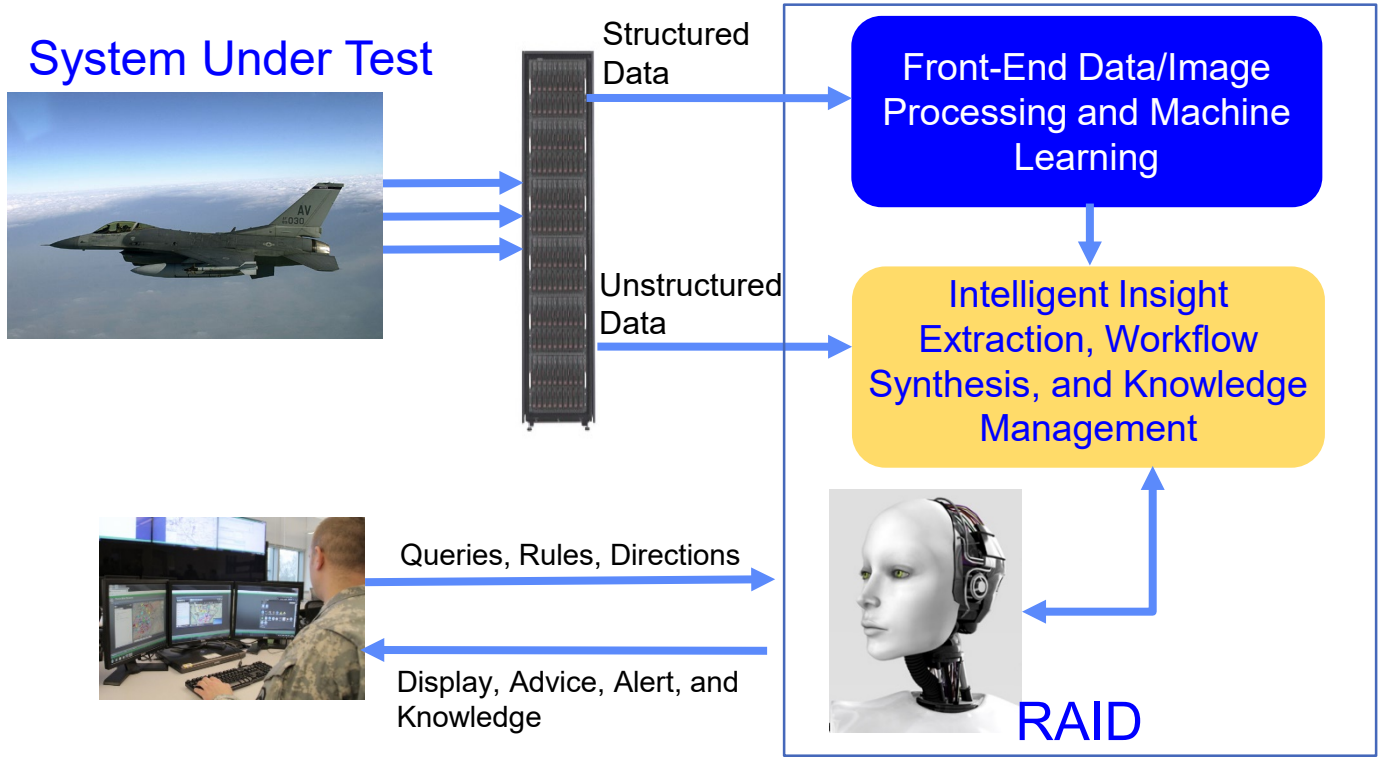
An Automated Artificial Intelligence Assistant for TSPI Post Mission Processing

Farrokh Vatan[a], Brian Kahovec[a], Edward Chow[a], Gary Glazner[b], and Gil Torres[c]

[a]Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA
[b]Edwards Air Force Base, CA, USA
[c]Naval Air Warfare Center, Point Mugu, CA, USA

Real-time Automated Insight Engine for Data to Decision (RAID) Project Description

- RAID will develop an Intelligent, automated assistant for data to decision for
  - Learn T&E know-how, experiences, and relationships from testers and analysts
  - Assist human in processing large amount of test data in complex situations
  - Use data to empirically validate and improve learned knowledge with human assistance
  - Use human-like reasoning to identify insights from structured and unstructured data
  - Enable distributed testers to use shared knowledge to identify critical test information

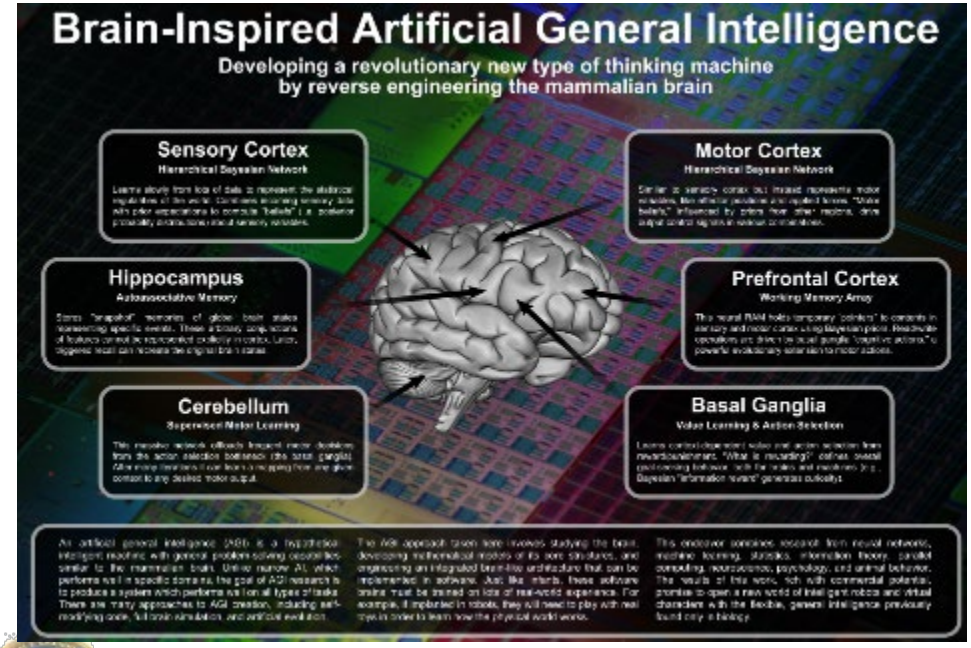


RAID use learned knowledge to assist testers to turn data into decision

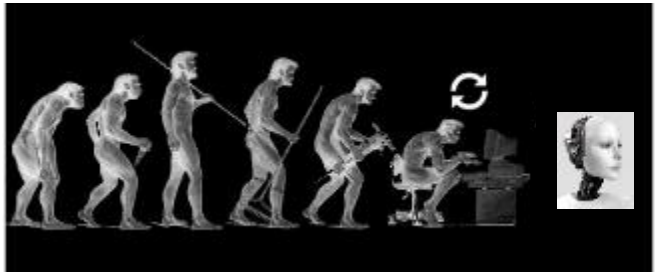
Distribution Statement A

AUDREY (Assistant for Understanding Data through Reasoning, Extraction, & sYnthesis)

- **AUDREY use bio-inspired Neural Symbolic Processing**
  - Mixed neural and symbolic processing at different levels of cognitive reasoning
- **AUDREY leverage human intelligence to achieve better machine intelligence**
- **AUDREY capabilities:**
  - Reasoning and learning new knowledge at the same time
  - Deal with missing or contradictory data
  - Automatically synthesize workflows to answer questions
  - Learn from human and a community of Audrey nodes



The Evolution of AI

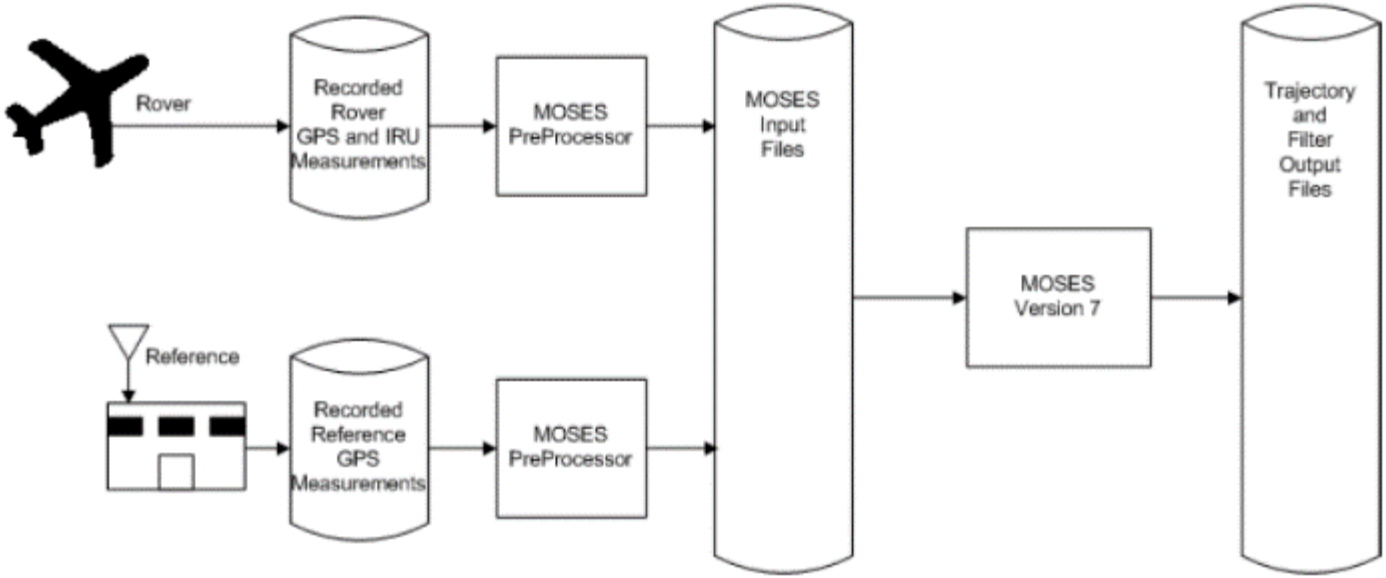


Achieves unprecedented levels of reasoning for previously unsolvable problems

Edwards AFB TSPI Processing Use Case

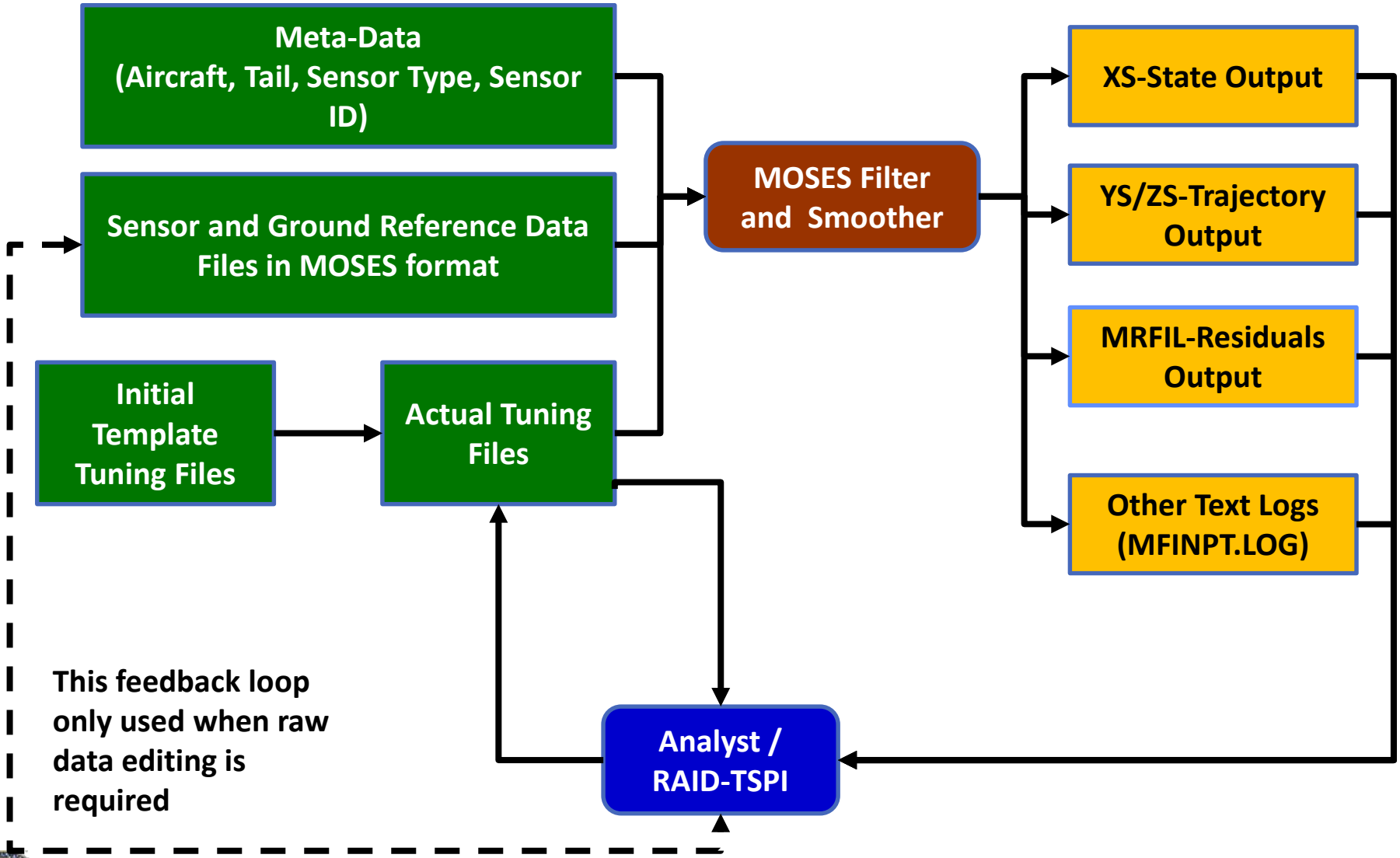
Scenario

- A typical scenario includes a vehicle under test with embedded or attached GPS/IRU measurement units
- Measurement information is recorded onboard or transmitted to a recorder on the ground
- A fixed reference site is also included for collection of GPS measurements to be used for differential corrections and estimation of other GPS measurement errors
- The vehicle under test is typically referred to as the "Rover", while the fixed ground station is referred to as the "Reference Receiver" (RR).



Edwards AFB TSPI Processing Use Case

Analyst/Audrey in the MOSES loop



Edwards AFB TSPI Processing Use Case

TSPI Problem

Tuning Event

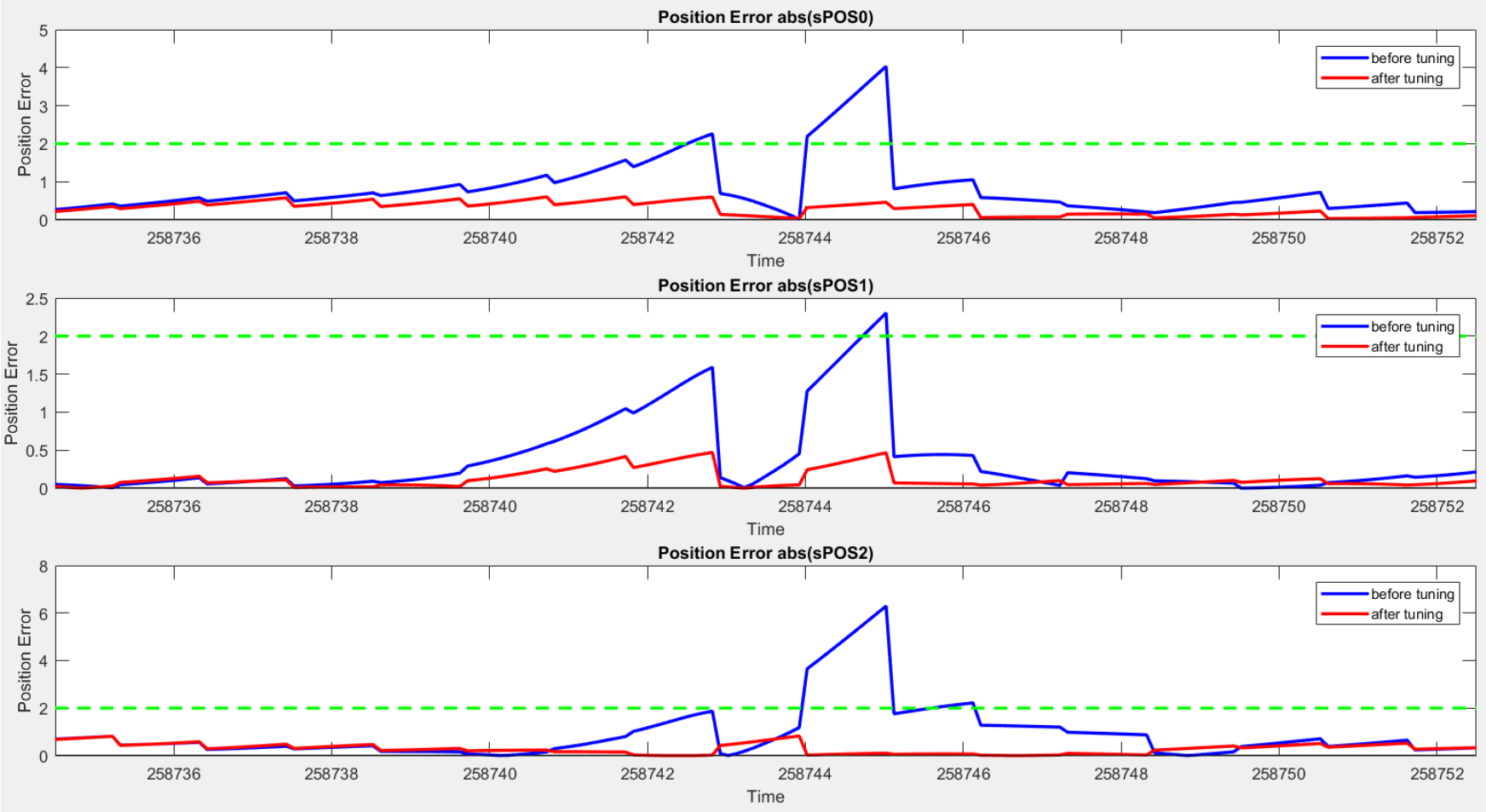
- Large position error
- Large satellite rejection rate
- Satellite residual outside sigma range (standard deviation)
- Spike (plateau) in satellite residual

- **TSPI Problem:**
  - ❑ Change the parameters of the MOSES to remove tuning events

Edwards AFB TSPI Processing Use Case

TSPI Problem

Tuning Event

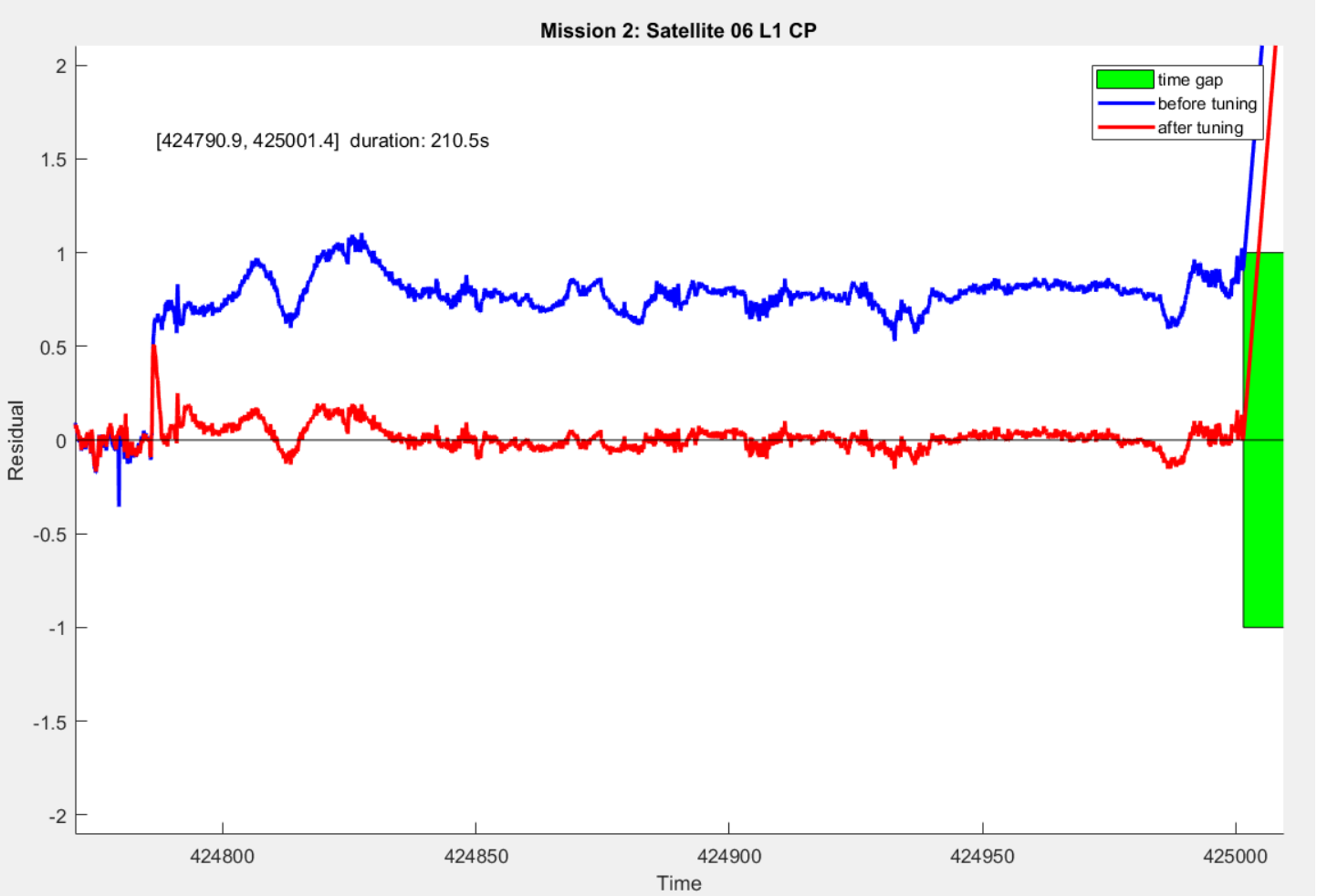


Position Error: before and after tuning

Edwards AFB TSPI Processing Use Case

TSPI Problem

Tuning Event



Satellite Residual Spike (Plateau): before and after tuning

Edwards AFB TSPI Processing Use Case

TSPI Problem

```
******************************   M O S F I L   ******************************

                    Dedicated to Wm. F. Tagliaferro, AFFTC

             MOSES FILTER PROGRAM            V7R4E.9k5-20110919
             AIR FORCE FLIGHT TEST CENTER            EDWARDS AFB
             RANGE DIVISION                       TSPI DEPARTMENT

PROGRAM EXECUTED ON 2017-08-24 AT 08:50:18

--------------------------------------------------------------------------------
CASE:  *    RUN:  1

FILTER TIME SEN SID MES        Z-MES       Z-FIL    Z-RES    Z-SIG   NRES   TOL

249111.1239 GPS   6 PR1 -25155574.5 -25155582.8      8.3      2.2    3.8   3.0
249111.2239 GPS   6 PR1 -25155566.2 -25155577.8     11.7      2.2    5.4   3.0
249111.3239 GPS   6 PR1 -25155556.8 -25155572.9     16.0      2.2    7.3   3.0
249111.4239 GPS   6 PR1 -25155552.1 -25155567.9     15.7      2.2    7.2   3.0
249111.5239 GPS   6 PR1 -25155540.4 -25155562.9     22.4      2.2   10.3   3.0
249111.6239 GPS   6 PR1 -25155527.4 -25155557.9     30.5      2.2   14.0   3.0
249111.7239 GPS   6 PR1 -25155512.6 -25155552.9     40.3      2.2   18.4   3.0
249111.8239 GPS   6 PR1 -25155496.1 -25155547.9     51.7      2.2   23.6   3.0
249119.2239 GPS  13 PR1 -25159078.8 -25159102.3     23.5      7.3    3.2   3.0
249119.3239 GPS  13 PR1 -25159094.4 -25159122.6     28.2      7.3    3.9   3.0
249119.4239 GPS  13 PR1 -25159113.1 -25159142.9     29.8      7.3    4.1   3.0
249119.5239 GPS  13 PR1 -25159127.4 -25159163.2     35.8      7.3    4.9   3.0
249119.6239 GPS  13 PR1 -25159140.8 -25159183.4     42.6      7.3    5.9   3.0
249119.7239 GPS  13 PR1 -25159153.6 -25159203.5     50.0      7.3    6.9   3.0
249119.7239 GPS  19 PR1 -25153045.0 -25153036.8     -8.2      2.2   -3.7   3.0
249119.8239 GPS  13 PR1 -25159165.4 -25159223.9     58.4      7.3    8.0   3.0
249119.8239 GPS  19 PR1 -25153040.1 -25153029.0    -11.1      2.2   -5.0   3.0
249119.9239 GPS  19 PR1 -25153035.5 -25153021.3    -14.2      2.2   -6.3   3.0
249120.0239 GPS  19 PR1 -25153031.7 -25153013.7    -18.0      2.2   -8.0   3.0
249120.1239 GPS  19 PR1 -25153028.3 -25153006.4    -22.0      2.3   -9.8   3.0
```

Satellite Number

Before Tuning

```
******************************   M O S F I L   ******************************

                    Dedicated to Wm. F. Tagliaferro, AFFTC

             MOSES FILTER PROGRAM            V7R4E.9k5-20110919
             AIR FORCE FLIGHT TEST CENTER            EDWARDS AFB
             RANGE DIVISION                       TSPI DEPARTMENT

PROGRAM EXECUTED ON 2017-08-24 AT 12:56:44

--------------------------------------------------------------------------------
CASE:  *    RUN:  1

FILTER TIME SEN SID MES        Z-MES       Z-FIL    Z-RES    Z-SIG   NRES   TOL
```

After Tuning

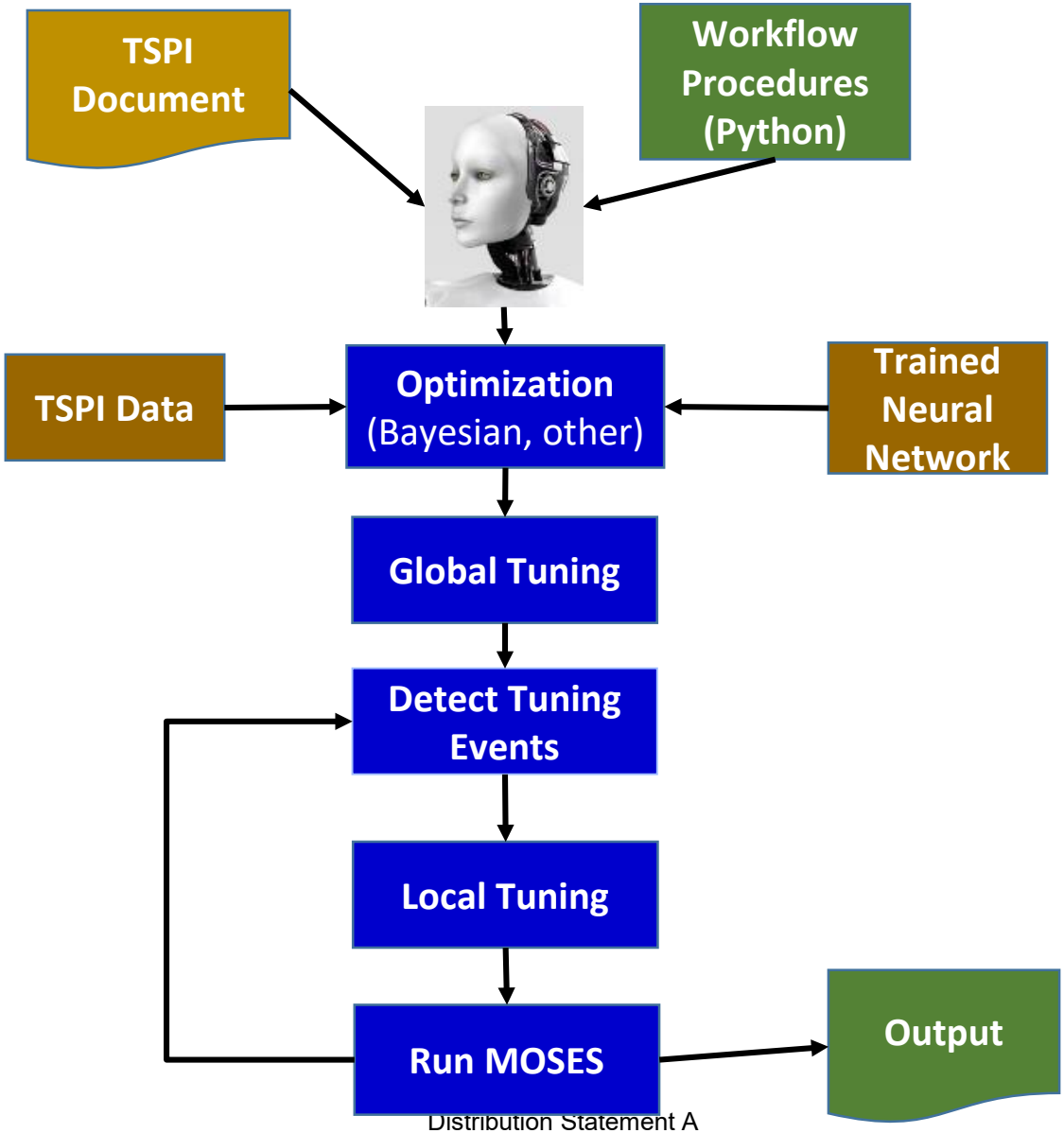
Satellite Rejection: before and after tuning

Audrey Workflow for TSPI

The Tuning Process

- The goal of this effort is to achieve, as described by the Edwards AFB team:
  - **Determine what tuning options are relevant to the characteristics of the mission at hand and the error conditions of the mission**
  - **Based on these options, we are going to create "tuning entries" to handle the anomalies**
- Two types of tuning entries:
  - **Global Tuning: ".INP" files; affects the whole mission**
  - **Local Tuning: "TUNE_SVS" file; affects a window of time**

Workflow


TSPI Document
Workflow Procedures (Python)
TSPI Data
Optimization (Bayesian, other)
Trained Neural Network
Global Tuning
Detect Tuning Events
Local Tuning
Run MOSES
Output

Teaching AUDREY TSPI Processing

Teaching AUDREY

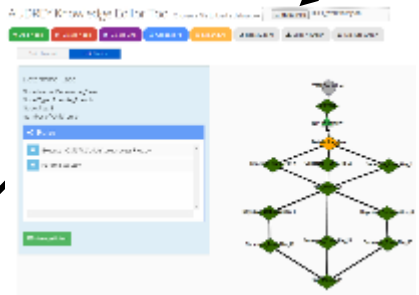
TSPI Document

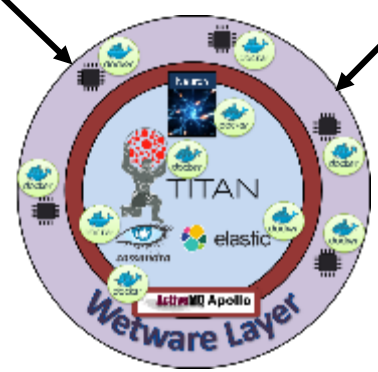
Workflow Procedures (Python)

TSPI Document

Workflow Procedures (Python)

HUNTER
Natural Language Understanding





Knowledge Extraction & Ingestion

- Knowledge extraction is the creation of knowledge from structured and unstructured sources
- Extracted knowledge needs to be in a machine-readable and machine-interpretable format
- The knowledge needs to be stored in a long-term format that facilitates inference and reasoning

- Our goal is to ingest the TSPI document and extract relevant knowledge to facilitate Audrey's reasoning
- Given the extracted knowledge, Audrey will be able to reason and generate ideal workflows

TSPI Post Mission Processing Flow

There are three different processes for tuning the satellite parameters:

1. For In Air data: the data contains four signals for each satellite L1PR, L1CP, L2PR, and L2CP and the altitude is more than 100ft;
2. For Ground data: the data contains four signals for each satellite L1PR, L1CP, L2PR, and L2CP and the altitude is less than 100ft;
3. For ARDS POD: the data contains only one signal for each satellite L1PR and the altitude is more than 100ft.

For tuning In Air data, in the file tune_svs.inp, we have PRN which multiplies the value of sgprgmn in the gpsmodel file to create the resulting sigma value for pseudorange for a specific satellite, and we also have PR_REJ which sets up how the filter rejects pseudorange measurements. PR_REJ will tell the filter to start rejecting satellite measurement values if they are "X" times higher than the sigma value for that data point. So for our case PR_REJ is set to the default of 6.0, which means if a measurement is 6 * 14.14 = 84.85 meters or larger, than the filter will automatically start to reject those measurements. This also holds true with CP_REJ which is for carrier phase and is a rejection multiplier of the CPN in the tune_svs.inp file.

In general, the tuning starts with assigning the value 6 to PRN or CPN and process the satellite residuals. If the resulting position errors are not below the acceptable threshold, then the PRN or CPN value is increased to 7. The process will continue till the PRN or CPN values reach 10, which ends the tuning process.

The ARDS POD data come from a container that is attached to a weapons mount under the wing of the airplane. As such it tends to be noisier than the GLITE data because of wing flexure and vibrations (whereas GLITES are mounted internally in the aircraft). For ARDS data we tend to focus primarily on the position errors and find spikes/plateaus that need to be reduced or tuned. There still may be individual satellite measurements that exceed the sigma values at a given time point, but if there isn't significant filter rejection occurring (long duration plateau), or a spike/plateau in the position errors then we'll ignore it. We consider spikes/plateaus above 3 meters in the position errors to be worth tuning, anything under this value tends to take us too long to try and correct and doesn't significantly affect the final trajectory.

We determined that we could get similar results by simply tuning all satellites around a position error spike and slightly reducing the rejection tolerance, this results in the filter rejecting slightly more measurements during the spike, but it also has a high success rate of reducing or removing position error spikes. As such, we usually tune all satellites in a region around the spike (using sv=00 to tune all satellites) and try varying the de-weight values and tolerance values. Normally setting PRN between 0.5–1.0, and changing PR_REJ between 0.8–3.0 will be enough to remove a position error spike (if it is correctable). This method isn't the optimal solution, but it is by far the fastest for a human to accomplish, and we are normally under a time crunch due to work load.

Knowledge Extraction & Ingestion
Hunter Output

TSPI Post Mission Processing Flow

There are three different processes for tuning the satellite parameters:

1. For In Air data: the data contains four signals for each satellite L1PR, L1CP, L2PR, and L2CP and the altitude is more than 100ft;
2. For Ground data: the data contains four signals for each satellite L1PR, L1CP, L2PR, and L2CP and the altitude is less than 100ft;
3. For ARDS POD: the data contains only one signal for each satellite L1PR and the altitude is more than 100ft.

For tuning In Air data, in the file tune_sys.inp, we have PRN which multiplies the value of sgprgmn in the gpsmodel file to create the resulting sigma value for pseudorange for a specific satellite, and we also have PR_REJ which sets up how the filter rejects pseudorange measurements. PR_REJ will tell the filter to start rejecting satellite measurement values if they are "X" times higher than the sigma value for that data point. So for our case PR_REJ is set to the default of 6.0, which means if a measurement is 6 * 14.14 = 84.85 meters or larger, than the filter will automatically start to reject those measurements. This also holds true with CP_REJ which is for carrier phase and is a rejection multiplier of the CPN in the tune_sys.inp file.

In general, the tuning starts with assigning the value 6 to PRN or CPN and process the satellite residuals. If the resulting position errors are not below the acceptable threshold, then the PRN or CPN value is increased to 7. The process will continue till the PRN or CPN values reach 10, which ends the tuning process.

The ARDS POD data come from a container that is attached to a weapons mount under the wing of the airplane. As such it tends to be noisier than the GLITE data because of wing flexure and vibrations (whereas GLITES are mounted internally in the aircraft). For ARDS data we tend to focus primarily on the position errors and find spikes/plateaus that need to be reduced or tuned. There still may be individual satellite measurements that exceed the sigma values at a given time point, but if there isn't significant filter rejection occurring (long duration plateau), or a spike/plateau in the position errors then we'll ignore it. We consider spikes/plateaus above 3 meters in the position errors to be worth tuning, anything under this value tends to take us too long to try and correct and doesn't significantly affect the final trajectory.

We determined that we could get similar results by simply tuning all satellites around a position error spike and slightly reducing the rejection tolerance, this results in the filter rejecting slightly more measurements during the spike, but it also has a high success rate of reducing or removing position error spikes. As such, we usually tune all satellites in a region around the spike (using sv=00 to tune all satellites) and try varying the de-weight values and tolerance values. Normally setting PRN between 0.5–1.0, and changing PR_REJ between 0.8–3.0 will be enough to remove a position error spike (if it is correctable). This method isn't the optimal solution, but it is by far the fastest for a human to accomplish, and we are normally under a time crunch due to work load.

```
Output.txt
<DOMAIN_KNOWLEDGE>
 <DATA_TYPE_DEF STATEMENT_TYPE='Define_Data_Type' DATA_TYPE='Air_Data'
  INQUALITY='GREATER_THAN' VALUE='1200' UNITS='Inches'>
   <SAT_SIGNALS>
     <SAT_SIGNAL NAME='L1pr'/>
     <SAT_SIGNAL NAME='L1cp'/>
     <SAT_SIGNAL NAME='L2pr'/>
     <SAT_SIGNAL NAME='L2cp'/>
   </SAT_SIGNALS>
 </DATA_TYPE_DEF>
 <DATA_TYPE_DEF STATEMENT_TYPE='Define_Data_Type' DATA_TYPE='Ground_Data'
  INQUALITY='LESS_THAN' VALUE='1200' UNITS='Inches'>
   <SAT_SIGNALS>
     <SAT_SIGNAL NAME='L1pr'/>
     <SAT_SIGNAL NAME='L1cp'/>
     <SAT_SIGNAL NAME='L2pr'/>
     <SAT_SIGNAL NAME='L2cp'/>
   </SAT_SIGNALS>
 </DATA_TYPE_DEF>
 <DATA_TYPE_DEF STATEMENT_TYPE='Define_Data_Type' DATA_TYPE='ARDS_Pod_Da
  INQUALITY='GREATER_THAN' VALUE='120
   <SAT_SIGNALS>
     <SAT_SIGNAL NAME='L1pr'/>
   </SAT_SIGNALS>
 </DATA_TYPE_DEF>
 <DEEP_KNOWLEDGE KNOWLEDGE_NET='1'/>
 <DEEP_KNOWLEDGE KNOWLEDGE_NET='2'/>
 <DEEP_KNOWLEDGE KNOWLEDGE_NET='3'/>
 <DEEP_KNOWLEDGE KNOWLEDGE_NET='4'/>
</DOMAIN_KNOWLEDGE>
```

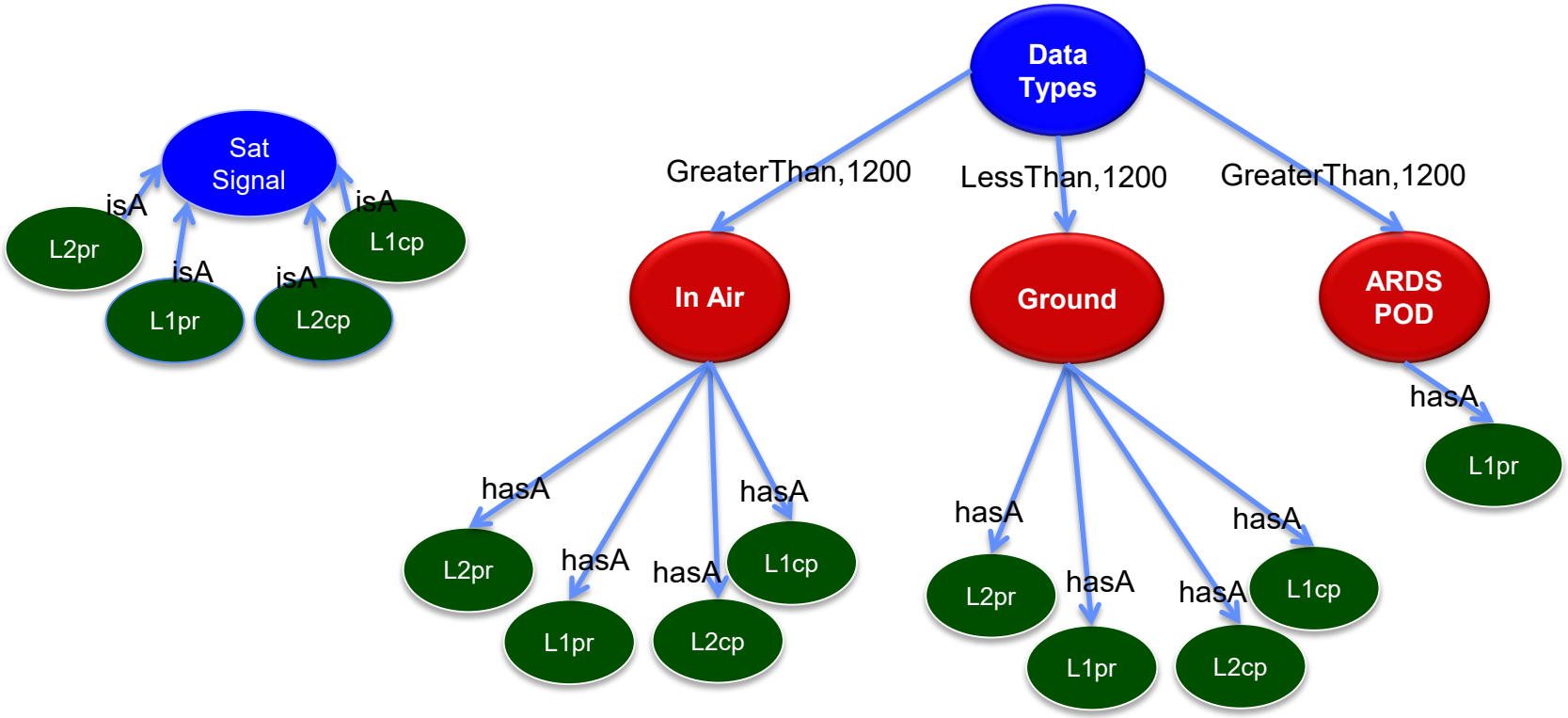
In Air data contains four listed signals

Ground data contains four listed signals

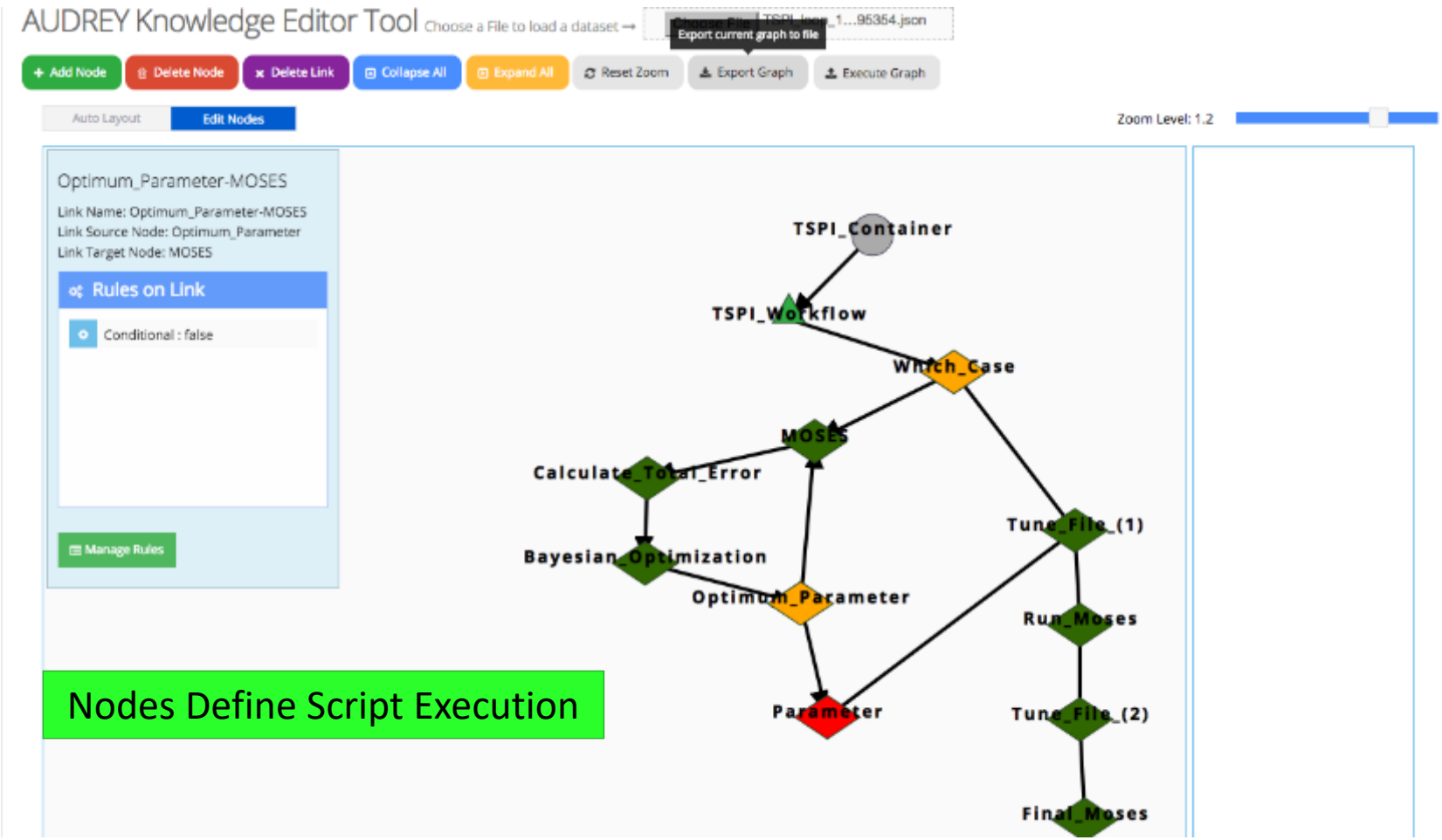
ARDS POD data contains one listed signal

Knowledge Extraction & Ingestion Node-Link Relationships for Inference

- Extracted XML from Hunter is parsed and converted into a Node-Link relationship for storage in our graph-database, Cortex
- Cortex provides the means to store relationships and world knowledge in a format that facilitates reasoning and inference



Procedure Knowledge (Python)



Nodes Define Script Execution

AUDREY TSPI Processing

Global Tuning
Bayesian Optimization

- We will utilize a Bayesian optimization method to find the optimal value this global parameter
  - **As an example, we consider two parameters:**
    - **SGCF0: Sigma of initial clock drift**
    - **SGCF: Sigma of clock drift**
- The first parameter only affects the initial part of the data (first few minutes), while the second one affects the whole data.

Global Tuning
Bayesian Optimization

- Objective function (target):

$$x^* = \arg\min_x f(x)$$

where the function $f(x)$ is the total position error of after assigning the value x to SGCF

- Given observations $(x_i, y_i = f(x_i))$ for $i = 1, \cdots, t$, build a probabilistic model for the objective function $f(x)$

- Minimize an easy to compute acquisition/utility function $u(x)$ based on the posterior distribution for sampling the next point:

$$x_{t+1} = \arg\min_x u(x)$$

- For acquisition function we use the Lower Confidence Bound defined as

$$u(x) = m(x) + \kappa \cdot \sigma(x)$$

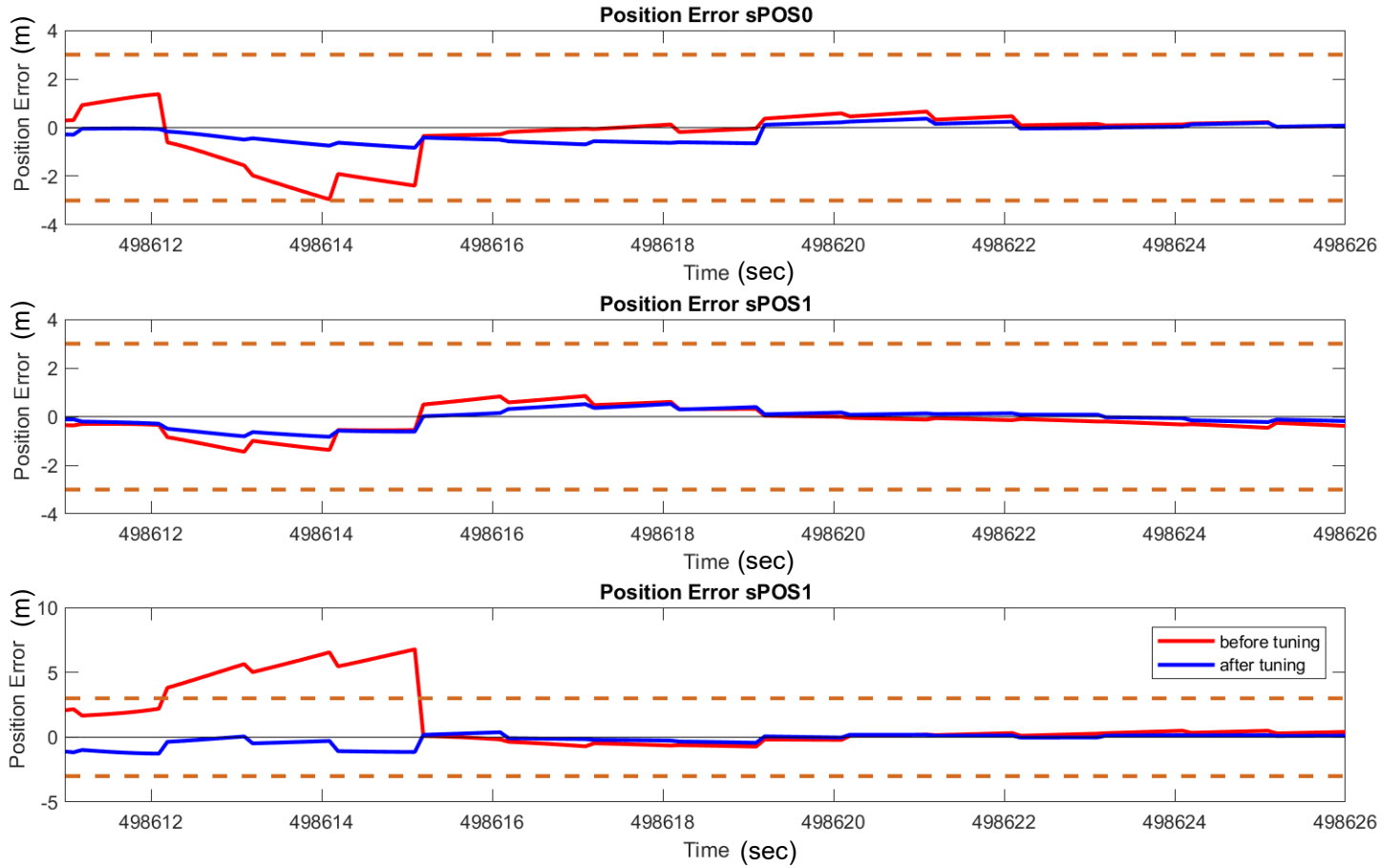
where $m(x)$ is the posterior mean, $\sigma(x)$, $\sigma(x)$ is the posterior variance, and κ is a controlling parameter

```
&GPSCLK


SGCRGMN = 10.0,
SGPRGMN = 1.5,
SGCPHMN = 0.015,
SGDOPMN = 0.025,
SGDRGMN = 0.03,
SGCP0 = 1.E9,
SGCF0 = 30.0,
SGCF = 0.5,
SGCAS(1) = 0.0005,0.0005,0.0005,
SGIPB0 = 0.0,
RGCP = 0.3,
RGCPXG = 0.3,
RGIPB = 1.E-3,
TGCF = 1800.0,
TGCAS(1) = 1800.0,1800.0,1800.0,
TGAPN = 1800.0,

/

&GPSLEV


DPABA(1) = 0.0,0.0,0.0,
SGLEV(1) = 0.00,0.00,0.00,
```

File gpsmodel.inp

Global Tuning: Bayesian Optimization

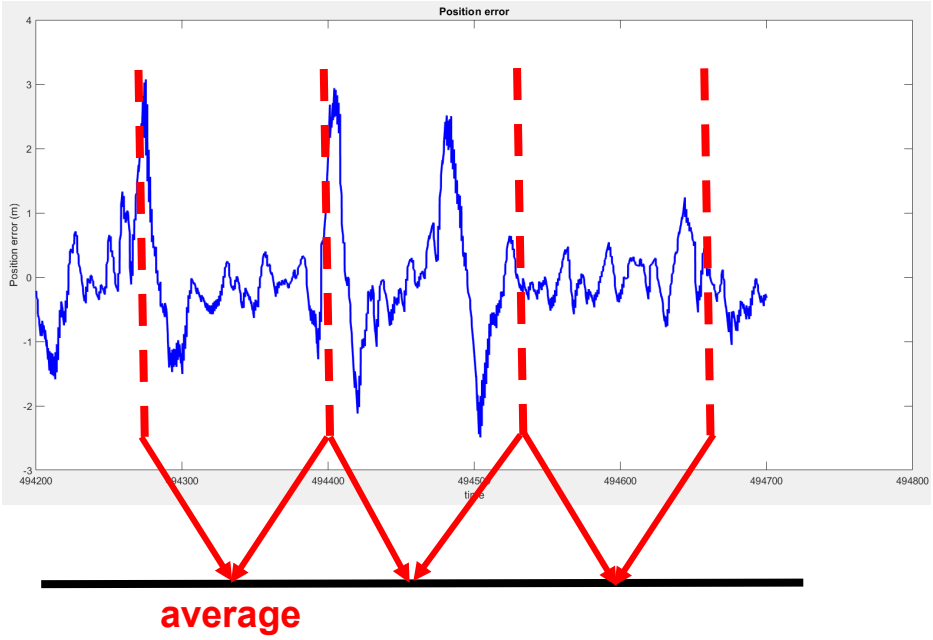
Reduction of Position Error



Reduction of position error after global tuning (SGCF0, SGFC)

Neural Networks
Implementation

- We implemented a Python recurrent neural network (RNN) based on *Tensorflow* package
- To make the size of the problem manageable, we reduced the length of each mission from several thousand to 100
  - We replaced the the average of the absolute value of the position error over a smaller range



Neural Networks Implementation

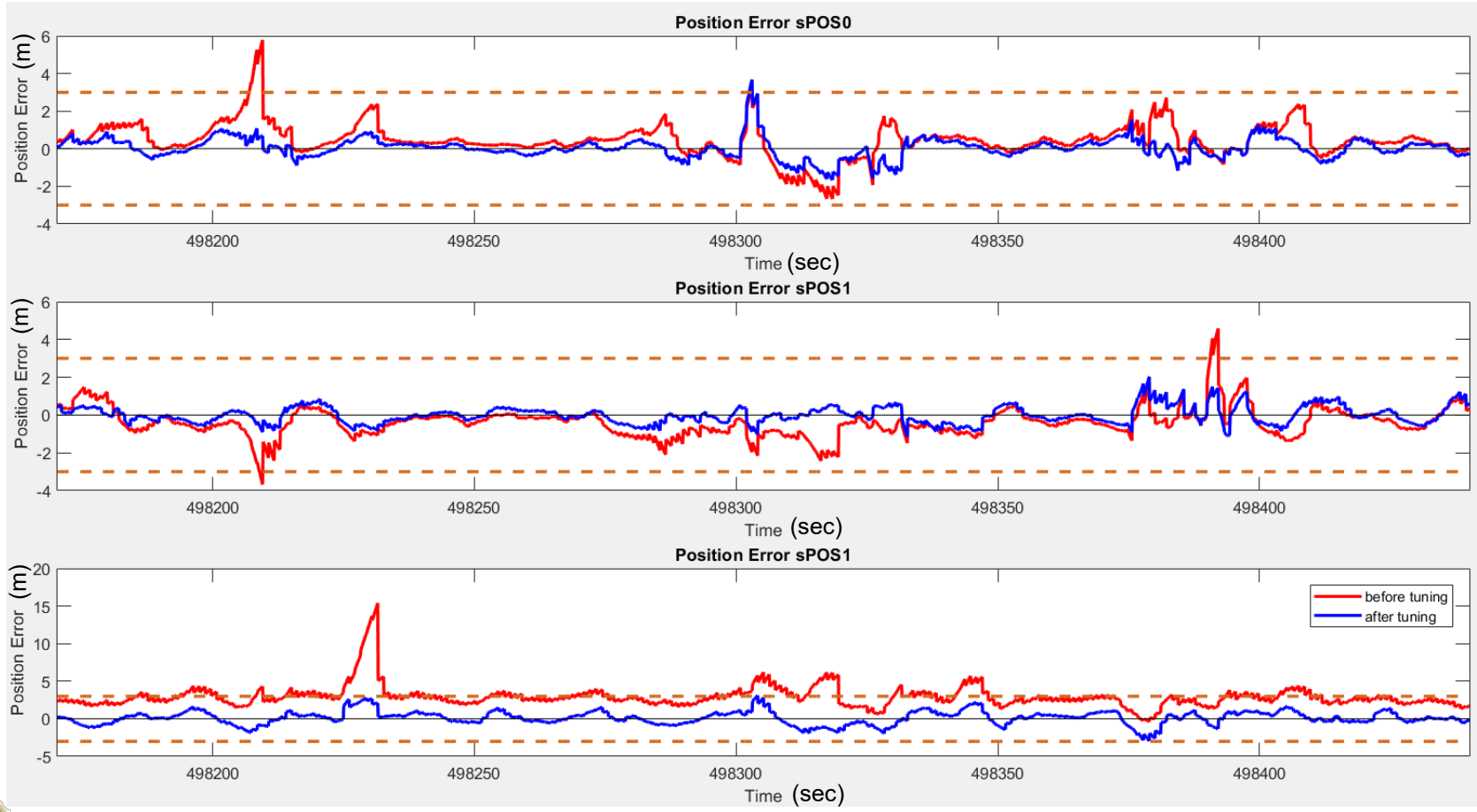
- We trained the RNN on 480 missions and tested it on the remaining 20 missions
- The whole architecture works properly
- As any neural network optimization, larger set of training data will improve the accuracy
- Later the Edwards team could use this implementation in their large set of missions, which we do not have access, to train a more accurate net

Training Neural Networks

- We applied two different methods for training the neural networks
  - **The neural network trained on analyst inputs**
  - **The neural network trained on outputs from Bayesian Optimization**
- Then, for evaluation, we compared the results with Bayesian optimization method and the analyst inputs

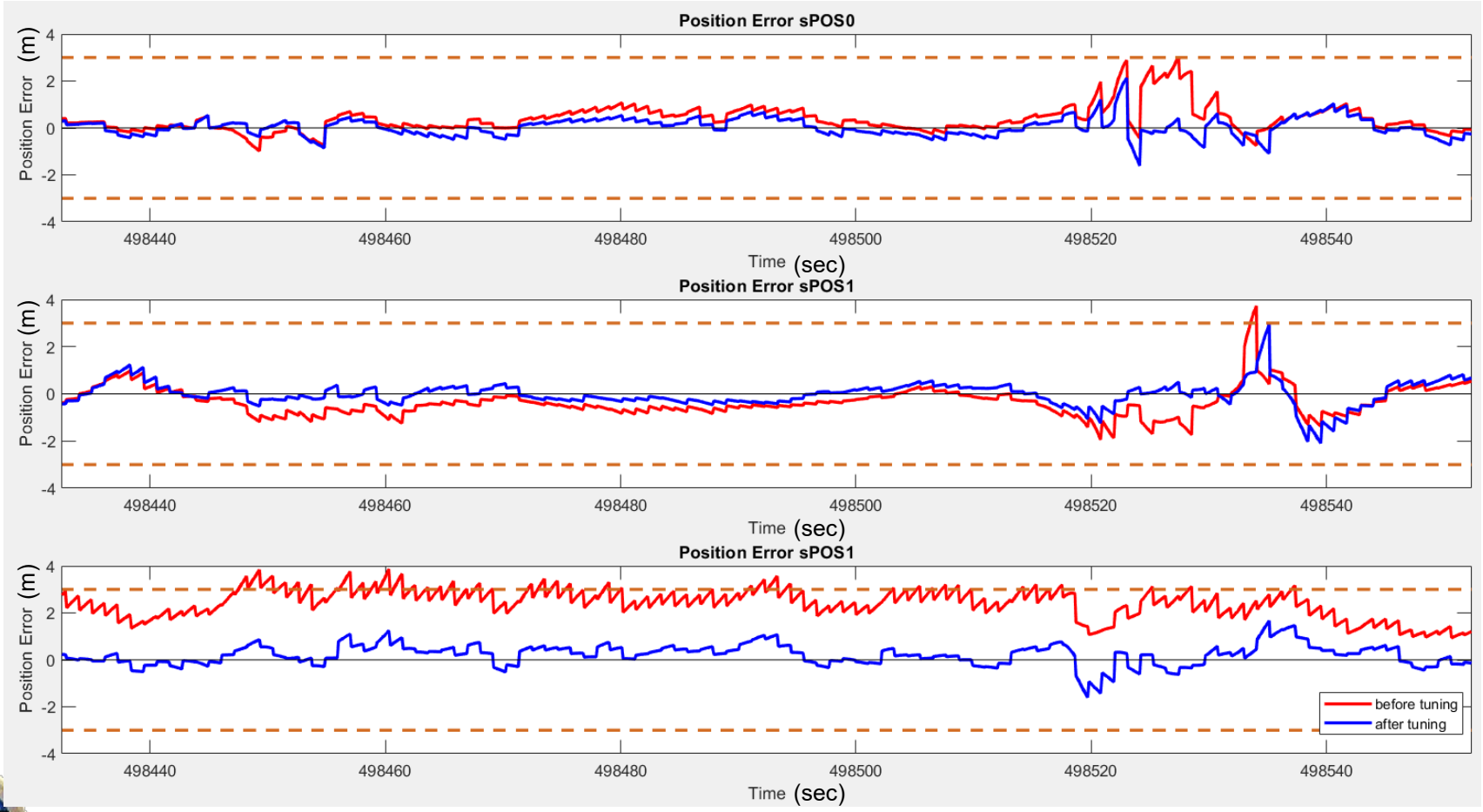
Training Neural Networks

- This figure shows the effect of neural network global tuning parameters on removing large positions errors



Training Neural Networks

- This figure shows the effect of neural network global tuning parameters on removing large positions errors

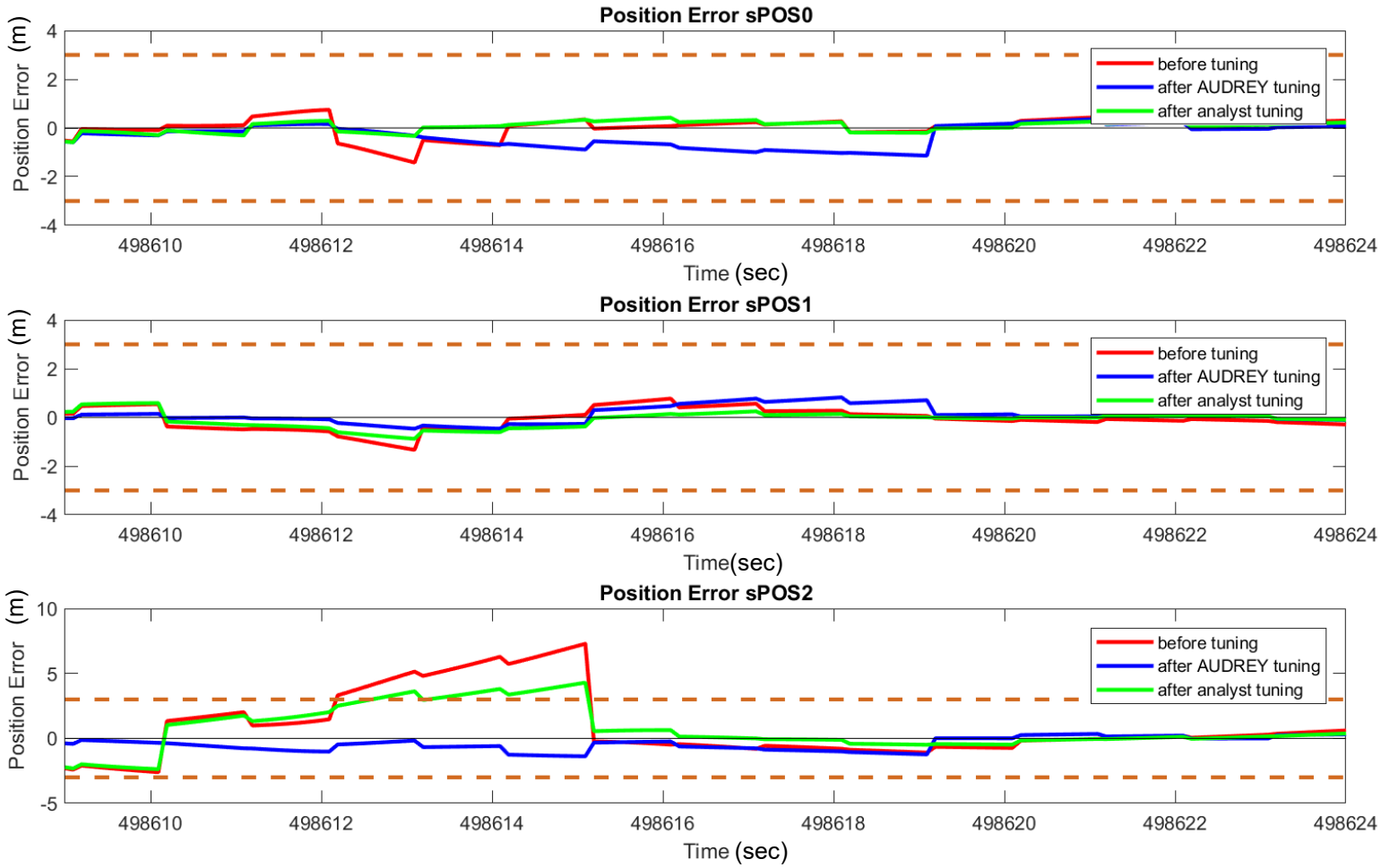


Combining Global and Local Tuning

- We finished developing the components of the code for optimization of the global and local tuning parameters
  - In the current version only few global parameters are considered
  - In later version of the code, more parameters would be optimize
- We are working on new Bayesian optimization method for finding global tuning, called “Derivative-Free,” which early results are very promising

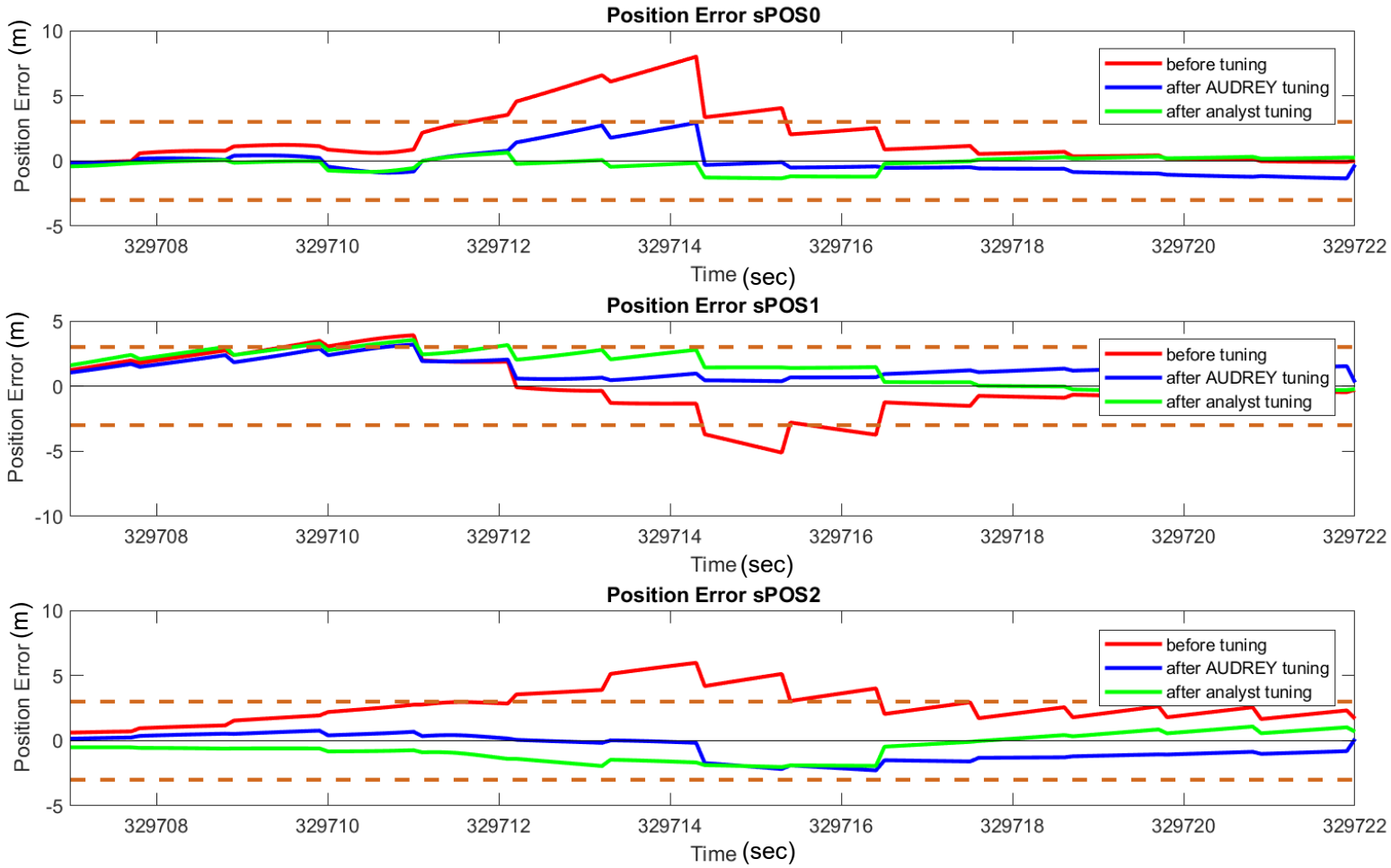
Some Results

Combining Global and Local Tuning



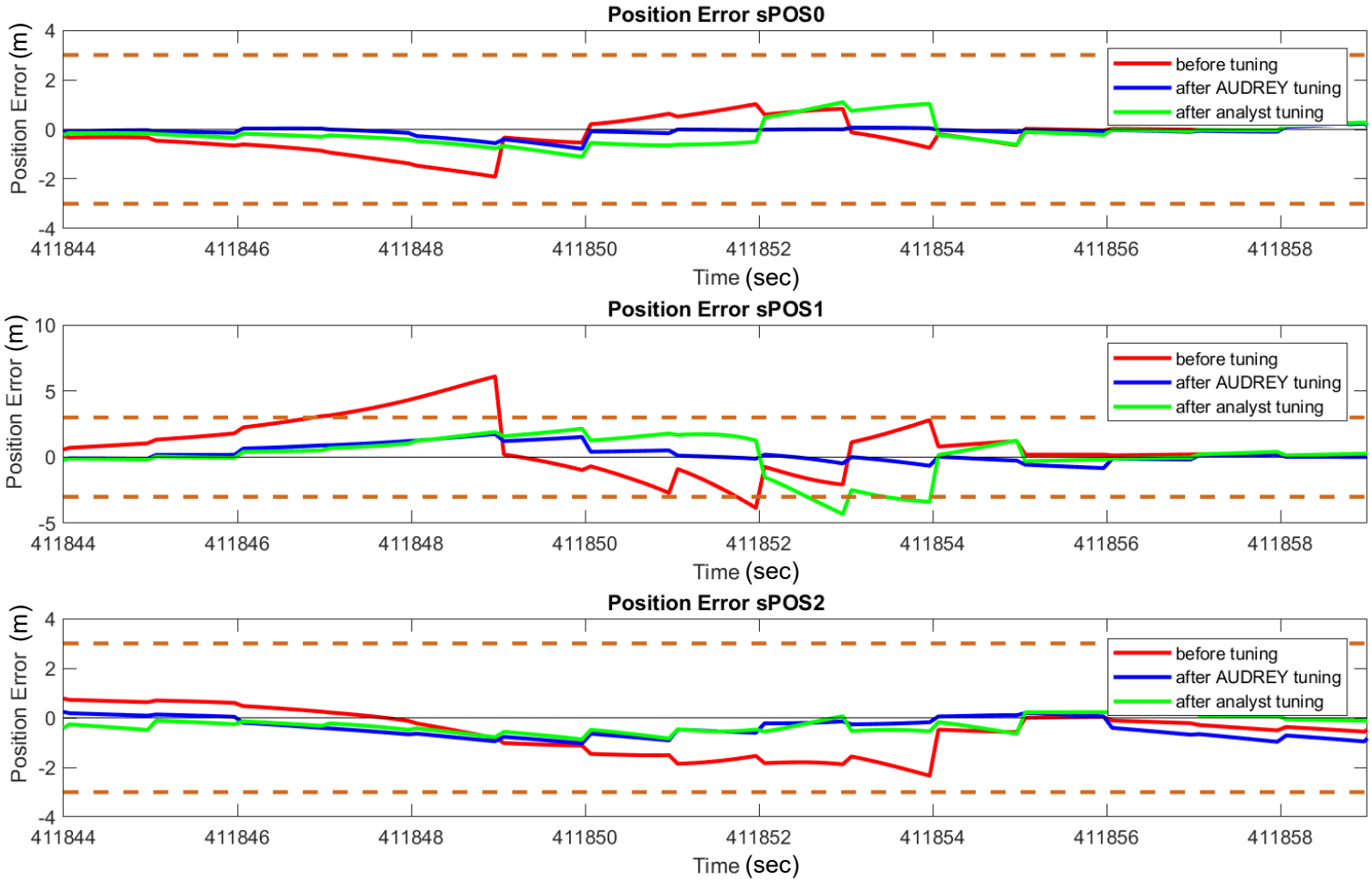
Some Results

Combining Global and Local Tuning



Some Results

Combining Global and Local Tuning



Some Results

- We experimented with our implementation of Bayesian optimization for global tuning, using 20 missions
- We used the 18 global parameters for labeling the missions
- The results are in the following table

	Total Error	Actual Decrease	Percentage Decrease	MOSES Runs
Default	2042754.976	0	0%	1
Analyst	2002357.203	40397.77285	2%	?
Bayes Opt	1787984.35	254770.6258	12%	5
RNN				1

Evaluating Audrey's TSPI

- We tested our Audrey TSPI code with 9 different data sets
- Our code successfully processed all data sets. For each data sets Audrey TSPI took about 10 minutes to produce the final tune file, while for some data sets analysts spend about two working days to do the same

- **Part of Edwards' TSPI data analyst evaluation of Audrey's performance:**
  - ❑ **… I can say that Audrey is accurately detecting tuning events. All of the spikes that an analyst would deem worthy of tuning were being tuned**
  - ❑ **…Overall I would say that having Audrey attempt data analysis looks very promising**
  - ❑ **… I believe at this point your group has shown proof of concept that Audrey can accurately detect tuning events, now all that needs to be shown is that it can smartly determine the optimal tuning values**

Performance	
Audrey	Analyst
~ 10 minutes	~1-16 hours

Wrap Up

- ★ We presented a RAID-Audrey use case for TSPI post mission processing.
- ★ We demonstrated a process to capture knowledge and automate the TSPI post processing process.
- ★ The RAID-Audrey project is designed to address T&E Big Data and Knowledge Management Problems