autoNomous, self-Learning, OPTImal and compLete Underwater Systems NOPTILUS

FP7-ICT-2009.6: Information and Communication Technologies

Webinar

Situation Understanding (WP6)

E. Orfanoudakis, N. Kofinas, and M. G. Lagoudakis
Telecommunication Systems Institute (TSI), Greece

June 10, 2015
http://tinyurl.com/NOPTILUS

NOPTILUS

SEVENTH FRAMEWORK PROGRAMME

Situation Understanding

NOPTILUS

- **Definition**
  - cognitive ability of inferring high-level descriptions and representations of the current state of the environment
- **Noptilus SU Goal**
  - *recognize interesting events in streams of observations*
  - *observations*: (abstracted) sensor data
  - *events*: patterns in observation streams
- **Quiz: what kind of event is this one?**
  - get eggs, break eggs, discard eggshells, stir eggs, pan on fire, eggs in pan, stir, flip, serve
- **Yes! This is the making of an omelet!**

Noptilus SU Technology

- **Models**
  - Probabilistic Context-Free Grammars (PCFGs)
- **Task 1: Event Recognition**
  - real-time, hierarchical parsing for on-line recognition
- **Task 2: Grammar Learning**
  - off-line learning of PCFGs from past AUV mission logs

PCFG

Probabilistic Context-Free Grammars

Probabilistic CFGs

Context-Free Grammars (CFG)

- formal models for specifying syntax (sequences)
- components:
  - terminal symbols, non-terminal symbols, start symbol
  - production rules

CFG example

- terminals {a,b}, non-terminals {S, A, B}, start symbol S
- rules = {$S \rightarrow A$, $S \rightarrow B$, $A \rightarrow aAb$, $B \rightarrow bBa$, $A \rightarrow ab$, $B \rightarrow ba$}
- encodes/produces all sequences $a^n b^n$ or $b^n a^n$ for $n>0$
- producing bbbaaa: $S \rightarrow B \rightarrow bBa \rightarrow bbBaa \rightarrow bbbaaa$

Probabilistic CFGs (PCFG)

- CFG with a probability value to each production rule

A Simple PCFG

Grammar	Prob	
S → NP VP	0.8	
S → Aux NP VP	0.1	+ 1.0
S → VP	0.1	
NP → Pronoun	0.2	
NP → Proper-Noun	0.2	+ 1.0
NP → Det Nominal	0.6	
Nominal → Noun	0.3	
Nominal → Nominal Noun	0.2	+ 1.0
Nominal → Nominal PP	0.5	
VP → Verb	0.2	
VP → Verb NP	0.5	+ 1.0
VP → VP PP	0.3	
PP → Prep NP	1.0	

Lexicon

Det → the | a | that | this
0.6 0.2 0.1 0.1 Σ=1.0

Noun → book | flight | meal | money
0.1 0.5 0.2 0.2 Σ=1.0

Verb → book | include | prefer
0.5 0.2 0.3 Σ=1.0

Pronoun → I | he | she | me
0.5 0.1 0.1 0.3 Σ=1.0

Proper-Noun → Houston | NWA
0.8 0.2 Σ=1.0

Aux → does
1.0 Σ=1.0

Prep → from | to | on | near | through
0.25 0.25 0.1 0.2 0.2 Σ=1.0

PCFG Parsing

- **Derivation**
  - sequential application of rules to the start symbol
  - *probability*: product of the probabilities of the rules used
- **Sequence**
  - sequence of terminal symbols derived from start symbol
  - *probability*: sum of the probabilities of all its derivations
- **Sequence parsing**
  - given a sequence, find a derivation, if one exists
  - useful for uncovering the structure of the sequence
- **Sequence likelihood**
  - compute the probability of a derivable sequence
  - useful for classifying and ordering sequences

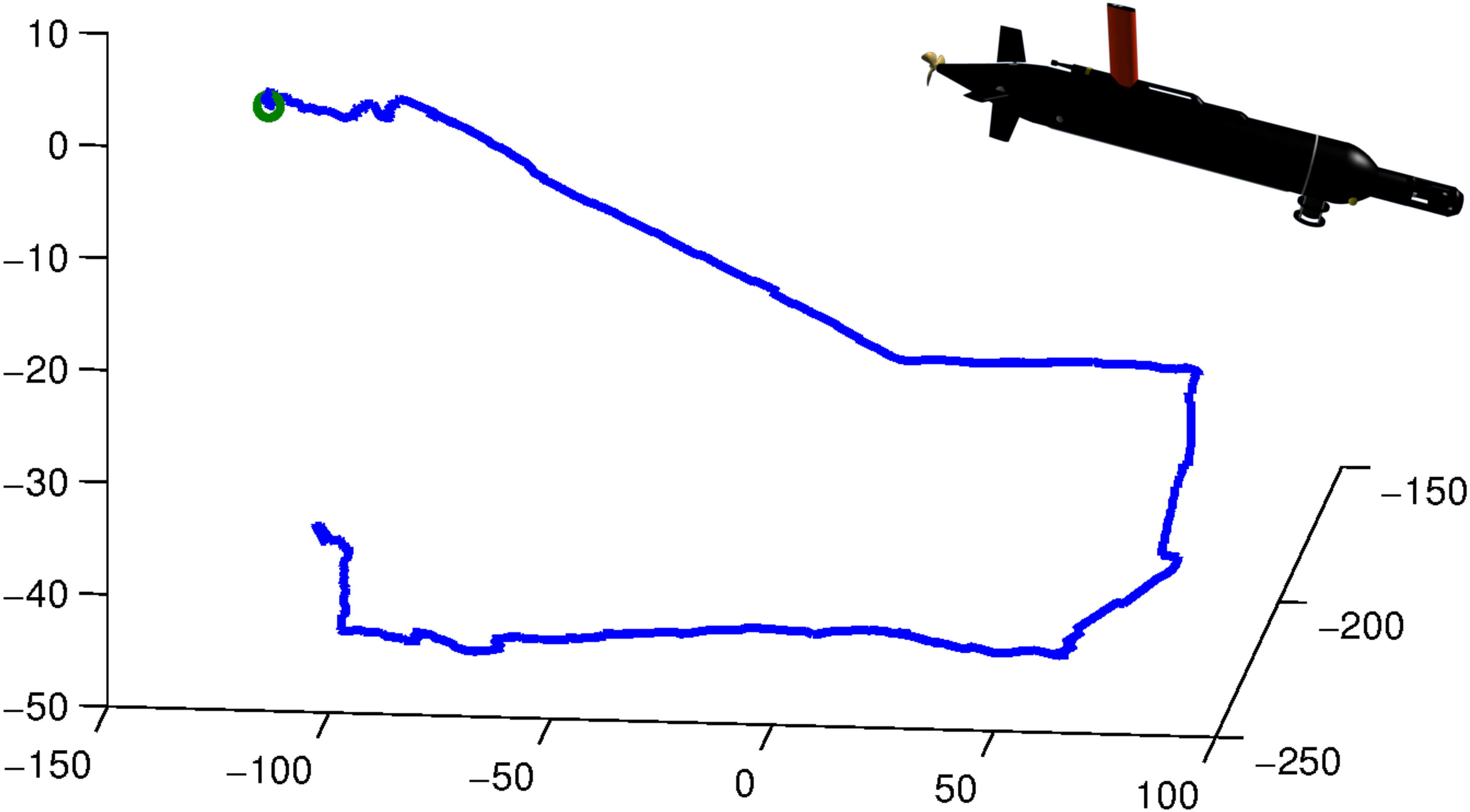
PCFG Benefits

- **Representation**
  - compact and hierarchical representation of sequences
  - human-readable, self-explanatory production rules
- **Algorithms**
  - a variety of parsers for various types of sequences
  - algorithms for learning the probabilities of the rules
  - algorithms for grammatical inference
- **Applications**
  - natural language processing
  - visual human activity recognition
  - …

PCFG Event Recognition

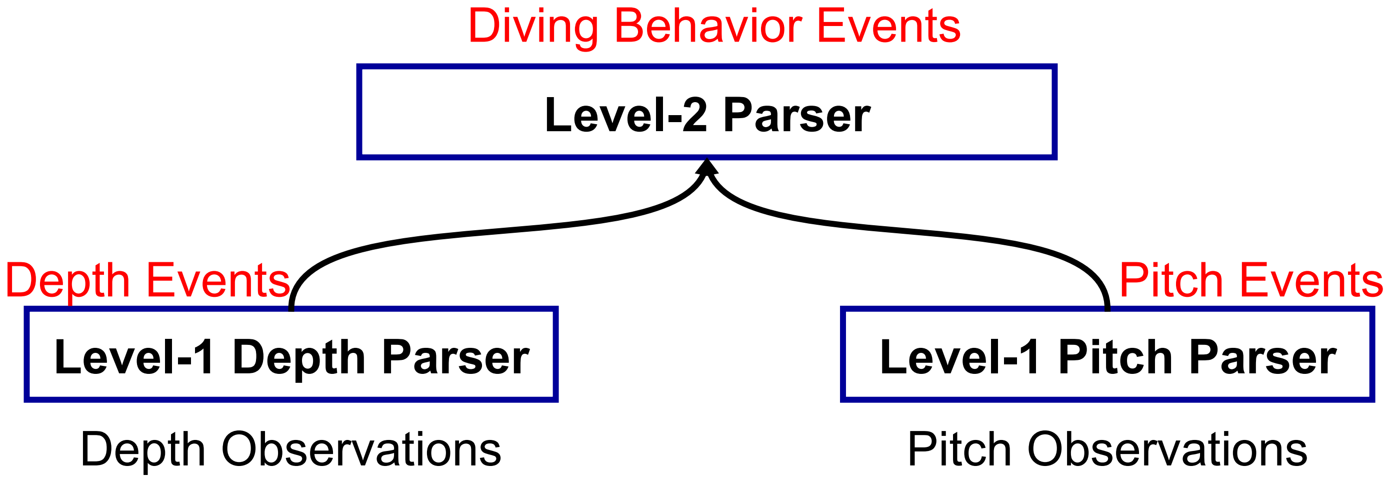
on-line, real-time, on-board parsing

AUV Mission Log

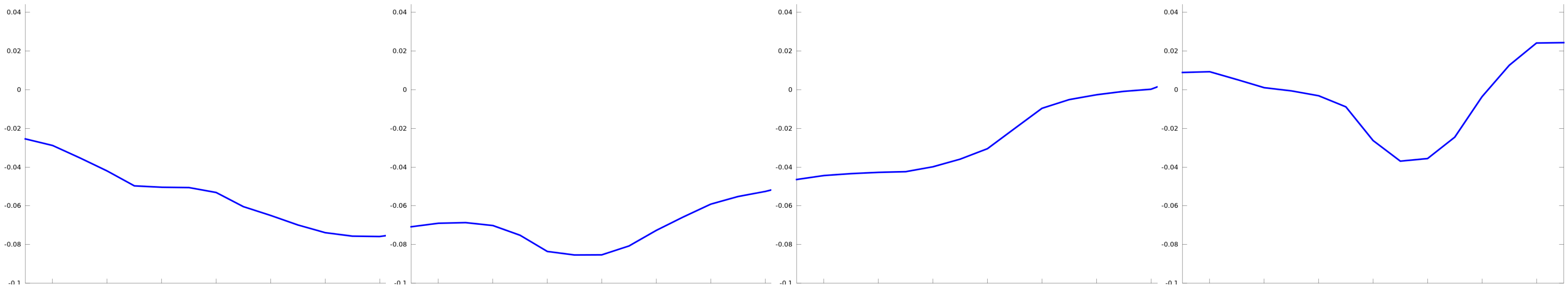


PCFGs for Noptilus

- **Goal**
  - simple event recognition regarding diving behavior
  - focus on joint patterns in depth and pitch
- **Hierarchy**
  - level 1: independent grammars for depth and pitch events
  - level 2: grammar for the combination of Level-1 events



Depth Observation Generation

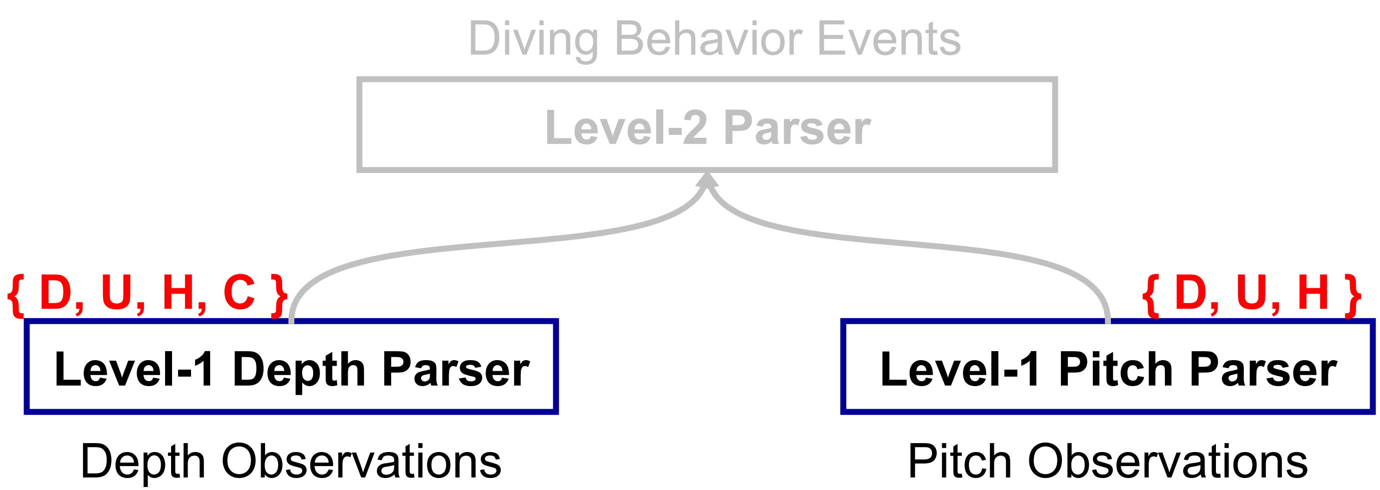


+	0	–	0
0.037474	0.024209	-0.019689	0.028816
0.032219	0.017864	-0.017810	0.035222
0.028115	0.014133	-0.016547	0.041991
0.025039	0.013886	-0.022303	0.049693
0.022243	0.015512	-0.027118	0.050414
0.022903	0.016860	-0.032096	0.050569
0.026767	0.014031	-0.033769	0.053115
0.035741	0.009555	-0.030667	-0.006621
0.037393	0.003580	-0.019416	0.007136
0.034305	-0.000692	-0.000125	0.021382
0.028398	-0.002861	0.008617	0.029188
0.023691	-0.007767	0.017316	0.040740
0.024289	-0.012192	0.025421	0.043735
0.026708	-0.017634	0.028816	0.042784
0.027268	-0.019469	0.035222	0.042784

quantization: average rate of change (**+**, **0**, **-**), averaging window s=15

Level-1 Event Recognition

- **Level-1 events**
  - depth : **D**own, **U**p, **H**over, **C**hange
  - pitch : **D**own, **U**p, **H**over
- **Level-1 parsing**
  - input: observations over a rolling window
  - output: most probable depth/pitch event occurred



Grammar for Depth Events

E → C	**[0.10]**
E → U	**[0.30]**
E → D	**[0.30]**
E → H	**[0.30]**
D → D D	[0.50]
D → d	[0.50]
U → U U	[0.50]
U → u	[0.50]
C → U H D	[0.50]
C → D H U	[0.50]
H → H H	[0.34]
H → U H D	[0.16]
H → D H U	[0.16]
H → h	[0.34]

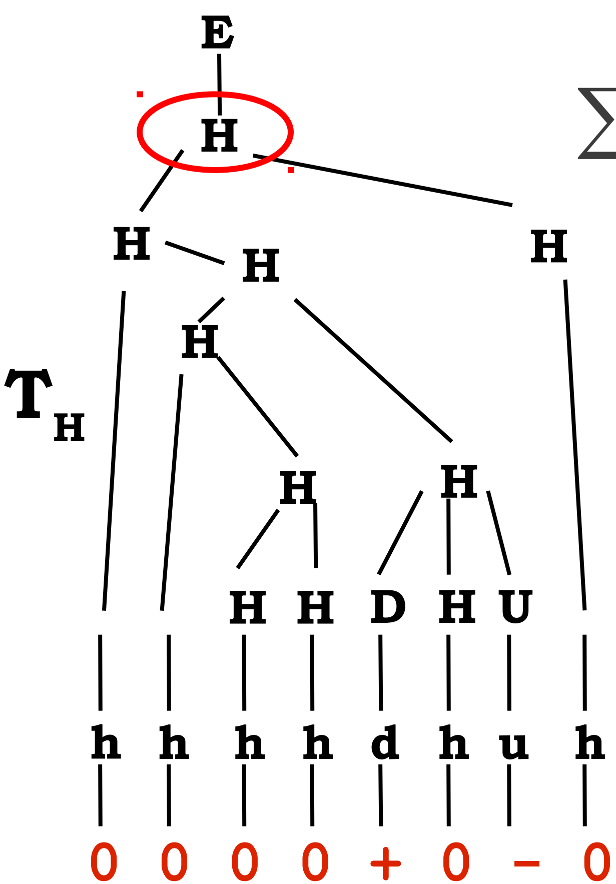
d → **'+'**	[0.85]
d → **'0'**	[0.15]
u → **'–'**	[0.85]
u → **'0'**	[0.15]
h -> **'0'**	[1.00]

Recognize event E by choosing the derivation that maximizes production probability

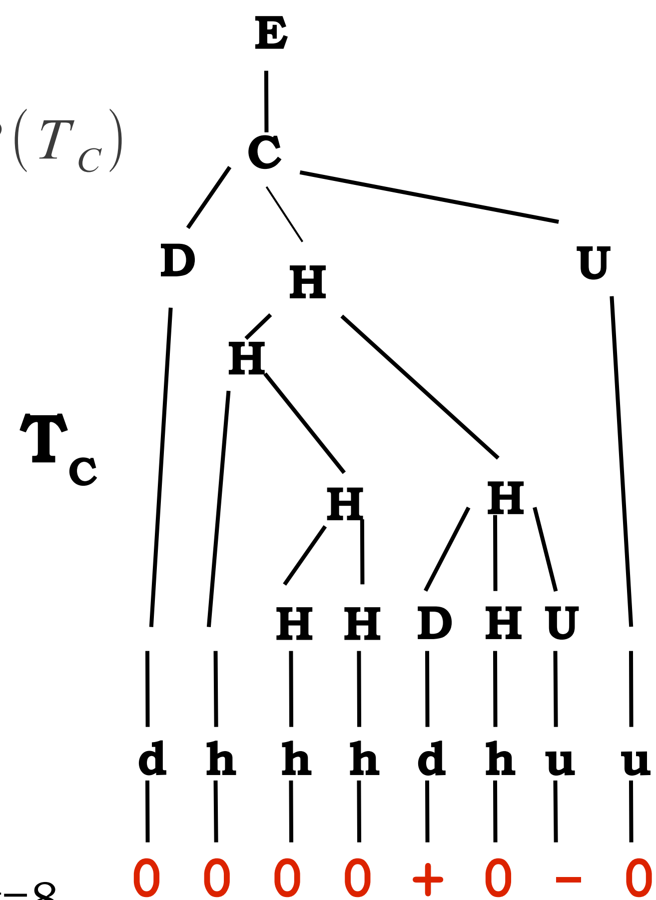
Level-1 Parsing: Depth

0 0 0 0 0 0 + 0 – 0 + + + + + + + + + + + + + + + +

H H H C C C H C D D D D D D D D D D D

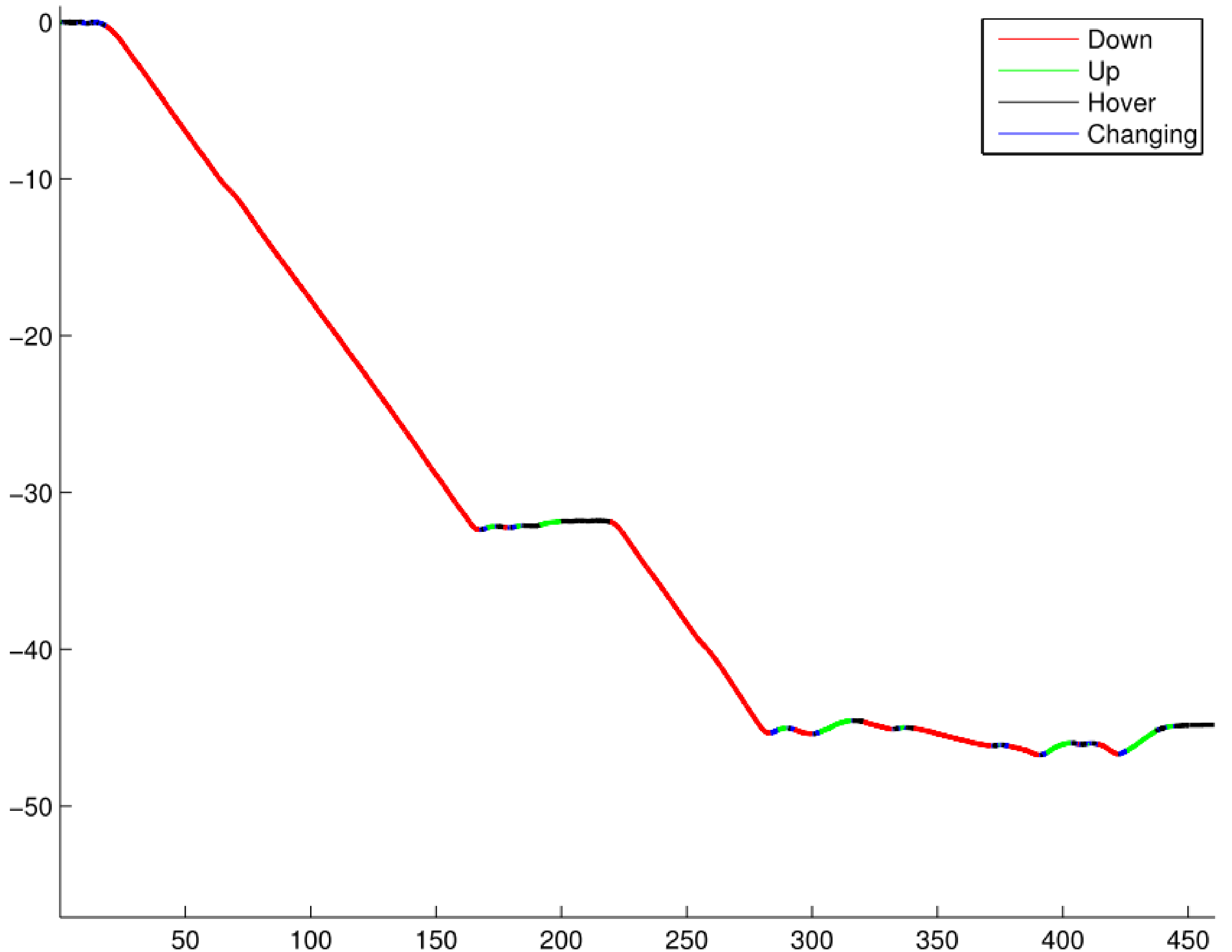


$$\sum_{T_H} P(T_H) > \sum_{T_C} P(T_C)$$

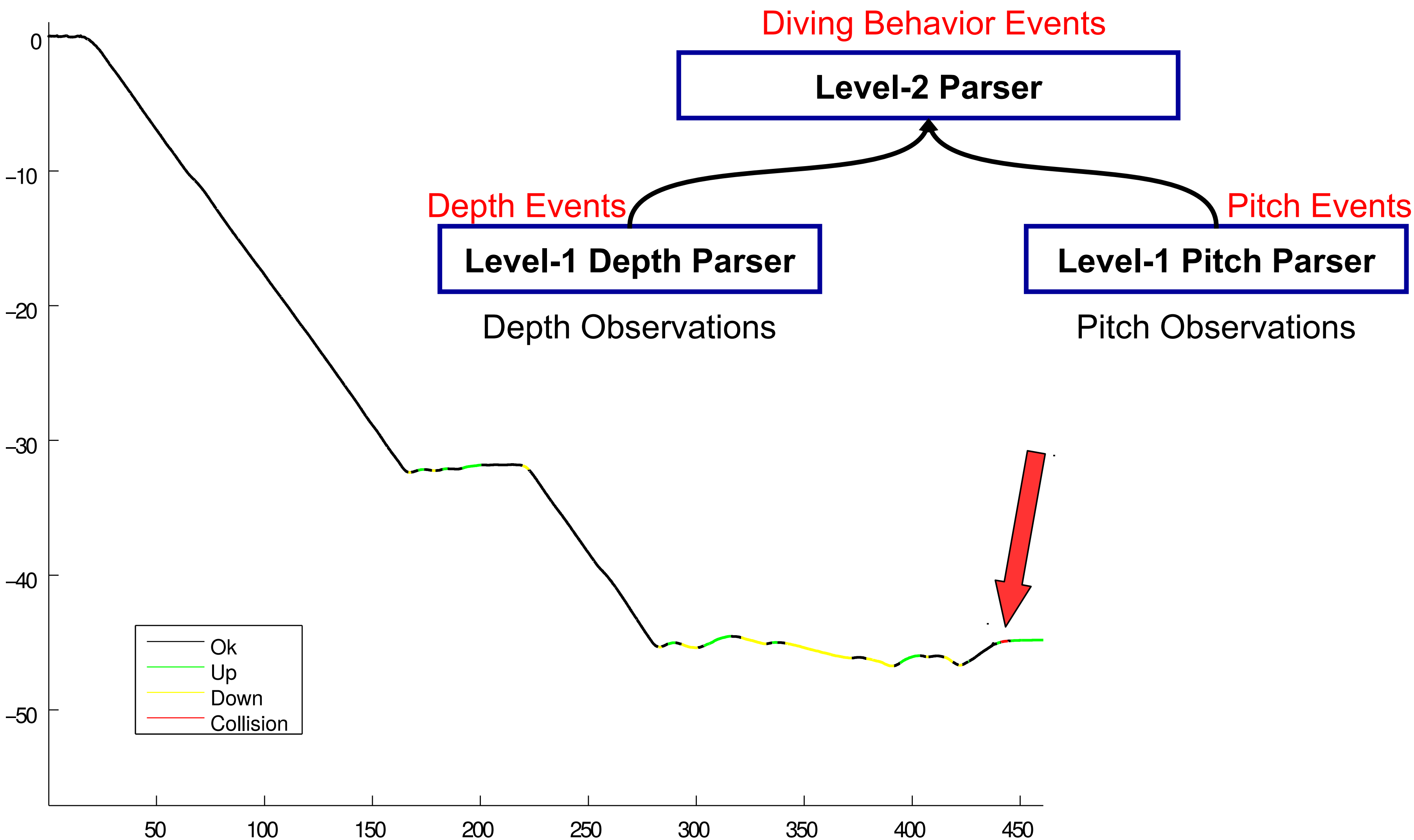


parsing window w=8

Level-1 Annotated Depth



Diving Behavior Event Recognition



PCFG Learning from Logs

off-line, off-board grammatical inference

Normal/Abnormal Events

- **Considerations**
  - what is an interesting event? what do we care about?
  - likely interesting events are unusual and unexpected
  - in most missions almost nothing abnormal occurs
  - *idea*: instead of looking for the abnormal and rare, …
  - … why not look after the normal and frequent?
  - easier to define normal as opposed to abnormal
- **Normal Operation**
  - typical patterns in motion and measurements (PCFG!)
- **Abnormal Operation**
  - any pattern that does not occur in normal operation

Grammar Learning

NOPTILUS

Structured Prediction
- make a prediction about a structured object

Grammatical Inference
- infer a PCFG (symbols, rules, probabilities) from words

Training Data
- corpus of positive (normal) examples only
- must generalize (but not too much) and must not overfit

Challenges
- the space of grammars cannot be generated systematically
- the space of grammars is not a vector space
- the neighborhood of a grammar is hard to define
- cannot tell if the best possible grammar has been reached

Our Bayesian Approach

- **Learning objective**
  - given a corpus O, find G* that maximizes the posterior

$$G^* = \arg\max_G P(G|O)$$

- **Bayes Rule**

$$G^* = \arg\max_G \frac{P(G)P(O|G)}{P(O)} = \arg\max_G P(G)P(O|G)$$

- **Prior of G**

$$P(G) = \frac{1}{2^{|G|}}$$

- **Likelihood of O over G**

$$P(O|G) = \prod_{w \in O} P(w|G)$$

PCFG Initialization (Example)

$$O = \left\{ \begin{array}{cccccccc} a & b & c & c & c & c & & \\ a & a & a & b & c & & & \\ a & a & b & b & b & c & & \\ a & a & b & b & c & c & & \\ a & a & a & b & c & c & c & c \\ a & a & a & a & a & a & b & c \\ a & a & b & b & b & c & & \end{array} \right\}$$

$$G_{init} = (V, \Sigma, R, P, S)$$

$$RP = \left\{ \begin{array}{cccccccccccc} S & \rightarrow & N_1 & N_2 & N_3 & N_3 & N_3 & N_3 & & & (1) \\ & | & N_1 & N_1 & N_1 & N_2 & N_3 & & & & (1) \\ & | & N_1 & N_1 & N_2 & N_2 & N_2 & N_3 & & & (2) \\ & | & N_1 & N_1 & N_2 & N_2 & N_3 & N_3 & & & (1) \\ & | & N_1 & N_1 & N_1 & N_2 & N_3 & N_3 & N_3 & N_3 & (1) \\ & | & N_1 & N_1 & N_1 & N_1 & N_1 & N_1 & N_2 & N_3 & (1) \\ N_1 & \rightarrow & a & & & & & & & & (19) \\ N_2 & \rightarrow & b & & & & & & & & (12) \\ N_3 & \rightarrow & c & & & & & & & & (14) \end{array} \right\}$$

Chunk and Merge Operations

NOPTILUS

➢Chunk

- creates a new non-terminal to replace a sub-sequence

$$\begin{array}{lllllllll} N_1 & \rightarrow & a & a & a & b & c & & (15) \\ N_2 & \rightarrow & a & b & c & a & a & & (22) \\ N_3 & \rightarrow & d & d & a & b & c & a & (9) \end{array} \Longrightarrow \begin{array}{lllllll} N_1 & \rightarrow & a & a & N_4 & & (15) \\ N_2 & \rightarrow & N_4 & a & a & & (22) \\ N_3 & \rightarrow & d & d & N_4 & a & (9) \\ N_4 & \rightarrow & a & b & c & & (46) \end{array}$$

➢Merge

- combines two existing non-terminals into one

$$\begin{array}{llllllll} N_1 & \rightarrow & a & N_2 & a & N_3 & c & (15) \\ N_2 & \rightarrow & a & N_1 & a & N_3 & d & (6) \\ N_3 & \rightarrow & d & N_2 & d & c & d & (32) \end{array} \Longrightarrow \begin{array}{llllllll} N_1 & \rightarrow & a & N_2 & a & N_2 & c & (15) \\ N_2 & \rightarrow & a & N_1 & a & N_2 & d & (6) \\ & \mid & d & N_2 & d & c & d & (32) \end{array}$$

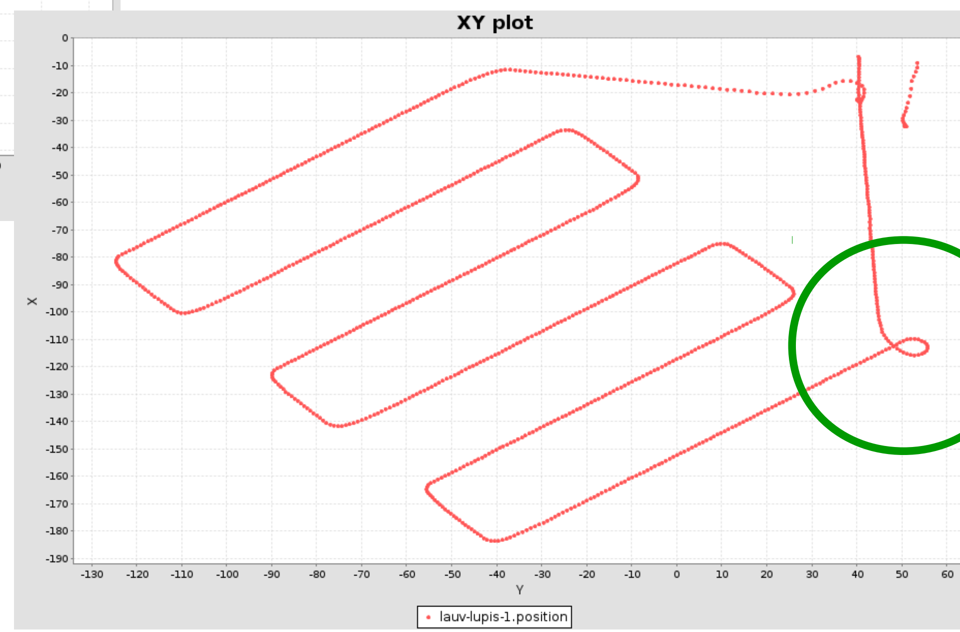
Grammar Learning Approach

- **Local Search**
  - Beam Search strategy to avoid local minima
- **Posterior Gain**
  - incremental computation over previous grammar
- **Recalculation of Probabilities**
  - merge operation invalidates the counts of the rules
  - fix: Inside-Outside algorithm over the corpus
- **Incremental Learning**
  - learning over a batch of the most frequent words
  - elimination of parsed words from training corpus
  - repeat learning and elimination until corpus is empty

AUV Normal and Abnormal Behavior

Normal
[Training]

Abnormal
[Testing]



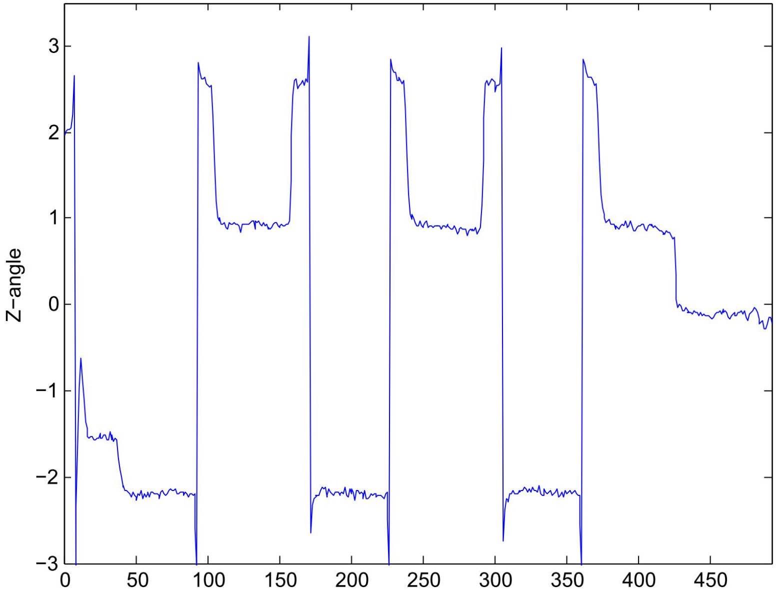
Learned Grammar on AUV Yaw

```
Start Rules:
N24
All Rules:
    N12 -> j (1)
    N13 -> k (1)
...
    N23 -> d (1)

    N24 -> N29 N29 (0.367347)
        -> N26 N26 N26 (0.27551)
        -> N27 N27 N27 (0.122449)
        -> N25 N25 (0.0612245)
...
    N25 -> N33 N14 (0.842105)
        -> N13 N19 N20 (0.157895)
    N26 -> N21 N21 (0.53795)
...
```

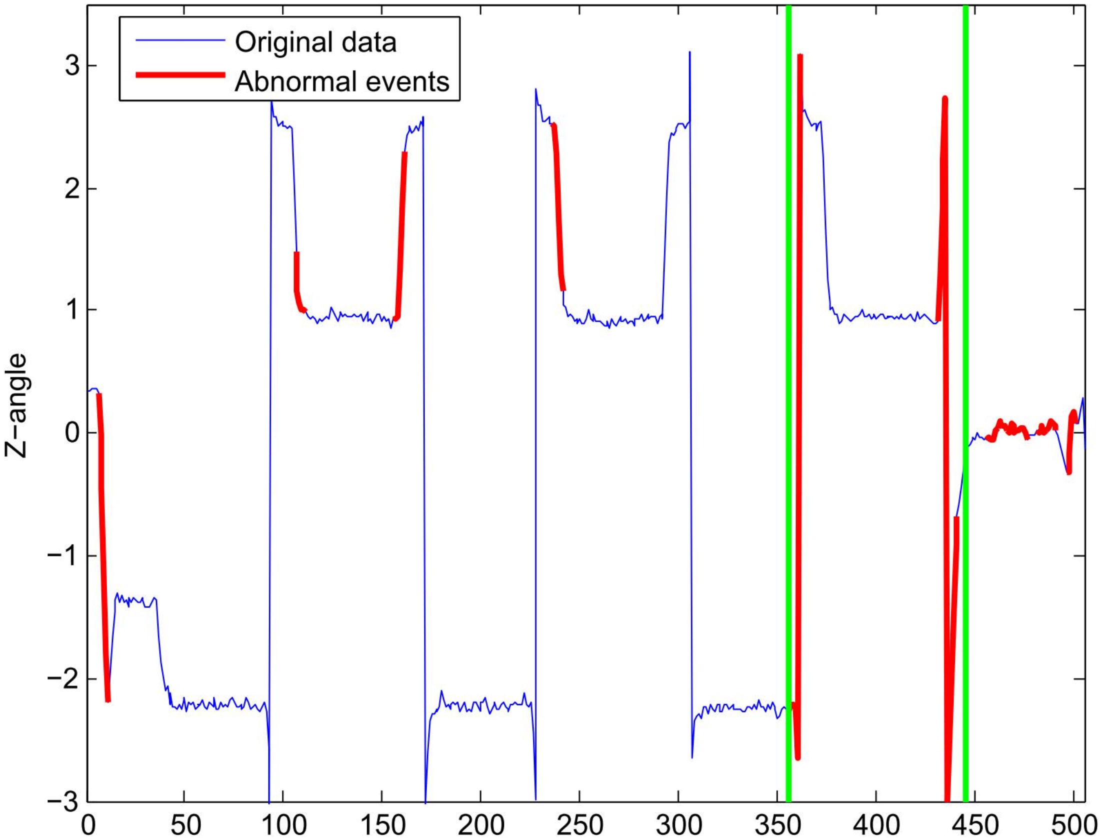
Recognition with Learned Grammar

- data stream: yaw (*z*-angle) of the AUV
- training corpus: 129 examples



Normal
[Training]

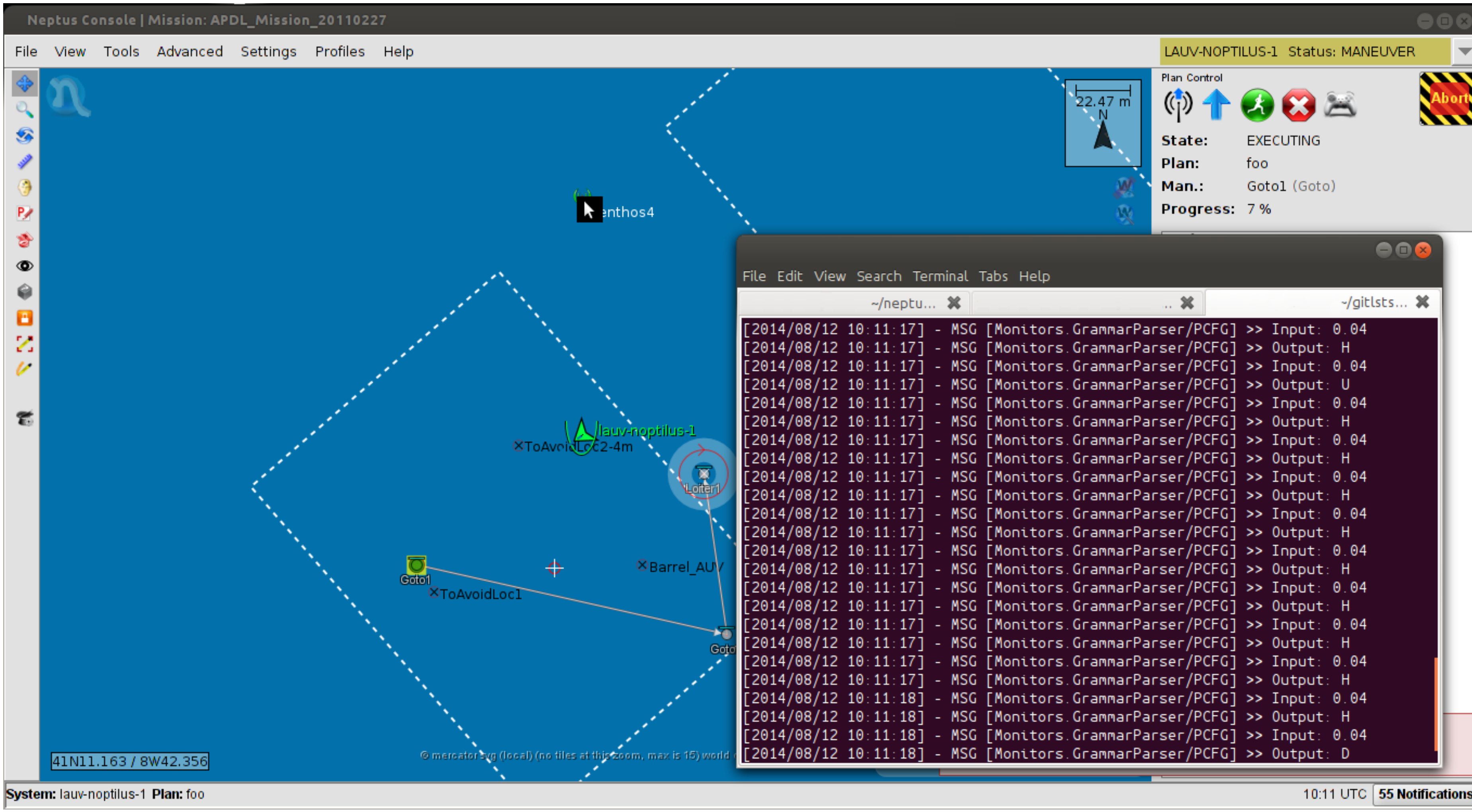
Abnormal [Testing]



Integration

Putting everything together

AUV+Dune+Neptus+Parser



AUV Mission Integration

- **Off-line (before)**
  - identify type of event
  - identify related data
  - collect normal data
  - learn grammar(s)
- **On-line (during)**
  - execute parser onboard
  - recognize events
  - signal detection(s)
- **Off-line (after)**
  - parse past mission logs
  - event detection

- **Event**
  - loss of orientation
  - gyro (z-angle, yaw)
  - normal mission data
  - rules and probabilities
- **Mission**
  - use learned grammar
  - detect abnormality
  - reset state, notify
- **Investigation**
  - detect past occurrences
  - extract event statistics

Thank you!

NOPTILUS