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

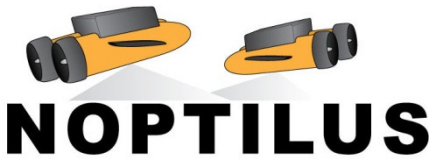
FP7-ICT-2009.6: Information and Communication Technologies

1st Project Review

WP6 (Situation Understanding) Progress Report

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

November 6, 2012
FEUP, Portugal



Outline

- **WP6: Situation Understanding**
- **Probabilistic Context-Free Grammars**
- **PCFG Event Recognition**
  - Proof-of-Concept Example
  - Multiple Experiments
- **WP6 Progress and Plans**

WP6

Situation Understanding

Situation Understanding

- **Definition**
  - cognitive ability of inferring high-level descriptions and representations of the current state of the environment
- **Abilities**
  - analysis of timed data
  - event recognition
- **Observations**
  - stream of discrete symbols
- **Events**
  - patterns in observations

WP6 Tasks

- **Task 6.1: Probabilistic Context-Free Grammars**
  - formal models for specifying sequences
  - appropriate for recognizing patterns in observations
- **Task 6.2: Structured Prediction**
  - step-by-step approach to building structured objects
  - appropriate for automatically learning grammars
- **Task 6.3: Integration**
  - PCFG event recognition using learned grammars
- **WP6 current status**
  - significant progress in Task 6.1
  - first steps in Task 6.2
  - no work yet in Task 6.3

PCFG

Probabilistic Context-Free Grammars

Probabilistic CFGs

- **Context-Free Grammars (CFG)**
  - formal models for specifying syntax
  - components
    - terminal symbols
    - non-terminal symbols
    - production rules
    - start symbol
- **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

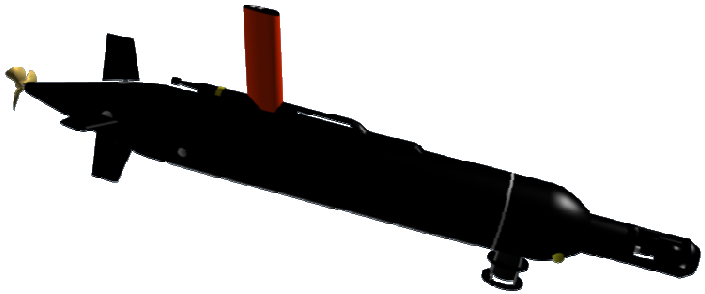
PCFG Benefits

- **Representation**
  - compact and hierarchical representation of sequences
  - self-explanatory production rules
- **Algorithms**
  - a variety of parsers for various types of sequences
  - algorithms for learning the probabilities of the rules
- **Applications**
  - natural language processing
  - visual human activity recognition
- **Noptilus**
  - PCFGs for AUV event recognition
  - abstract state description, reduction of communication
  - causal grammar specification, diagnostic sequence parsing

PCFG Event Recognition

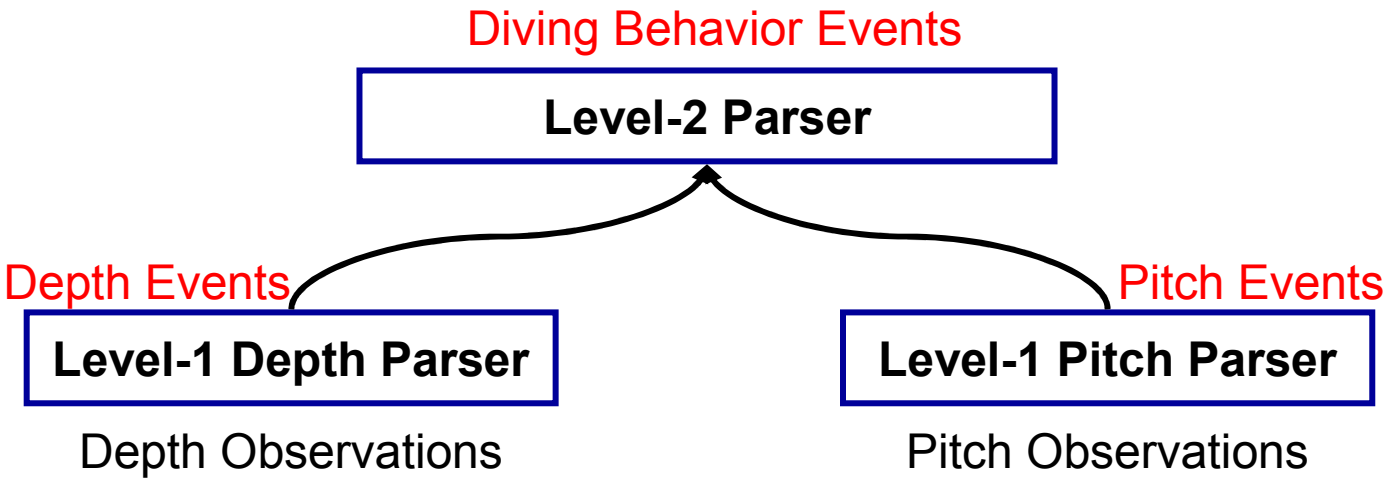
A Proof-of-Concept Example

AUV Mission Log



PCFGs for Noptilus

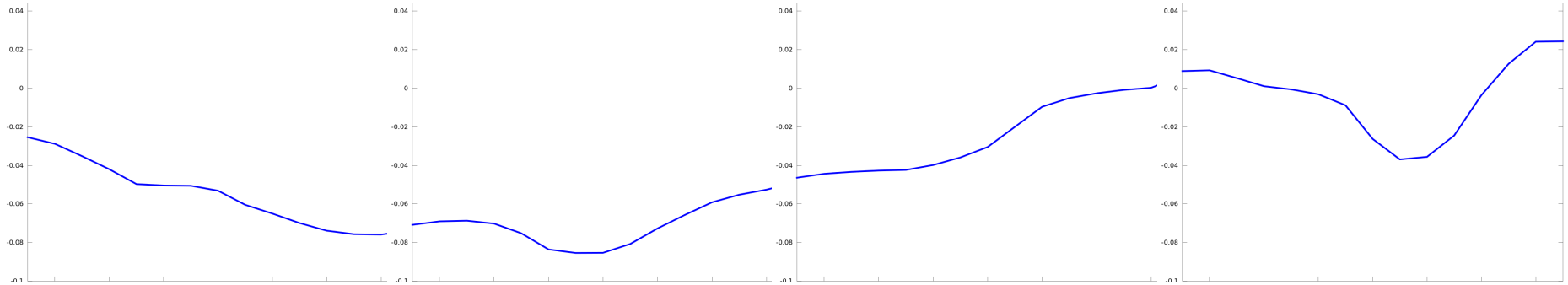
- **Goal**
  - simple event recognition regarding diving behavior
  - focus on joint patterns in depth and pitch
- **Hierarchy**
  - level 1: independent grammars for depth and pitch events
  - level 2: grammar for the combination of Level-1 events



Generating Level-1 Observations

- **Raw data**
  - depth measurements
  - pitch measurements
- **Abstraction**
  - depth: compute the rate of change, average over a window
  - pitch: average over a window
- **Level-1 observations**
  - **–** : negative (value<-ε)
  - **0** : zero (|value|<ε) for some ε>0
  - **+** : positive (value>+ε)

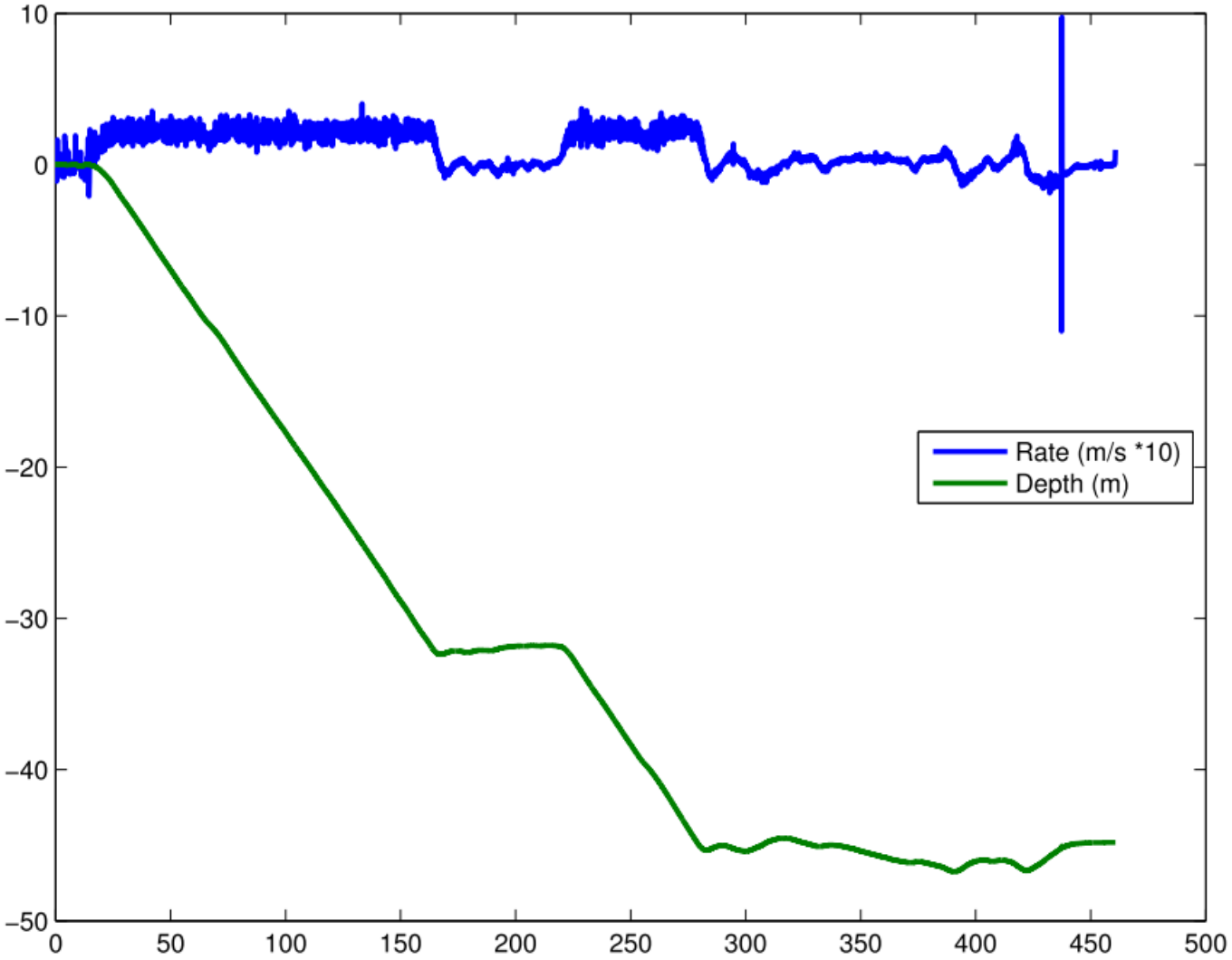
Depth Observation Generation



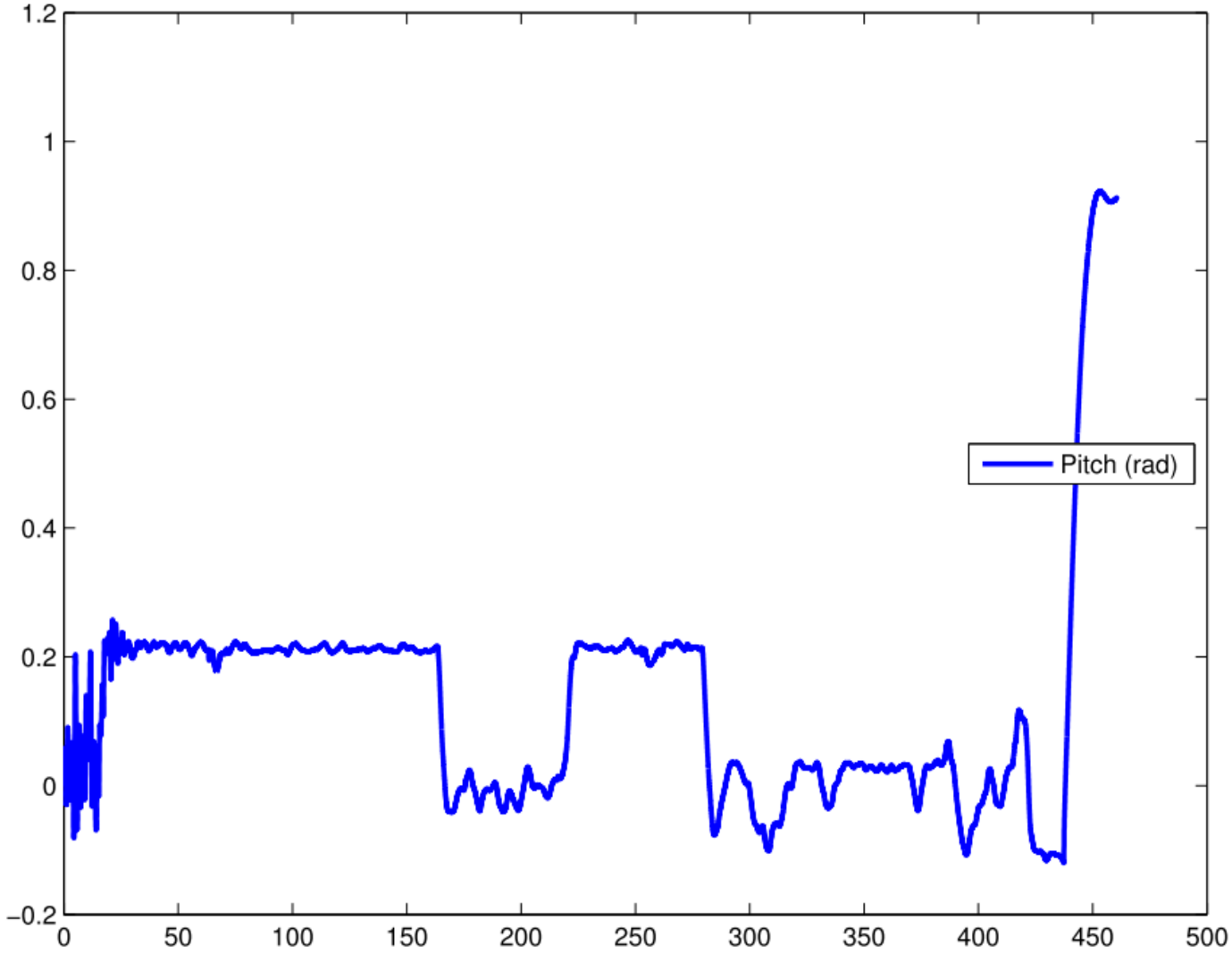
+	0	–	0
0.037474	0.024209	-0.019689	0.028816
0.032219	0.017864	-0.017810	0.035222
0.028115	0.014133	-0.016547	0.041991
0.025039	0.013886	-0.022303	0.049693
0.022243	0.015512	-0.027118	0.050414
0.022903	0.016860	-0.032096	0.050569
0.026767	0.014031	-0.033769	0.053115
0.035741	0.009555	-0.030667	-0.006621
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

averaging window s=15

Depth Sensor Data

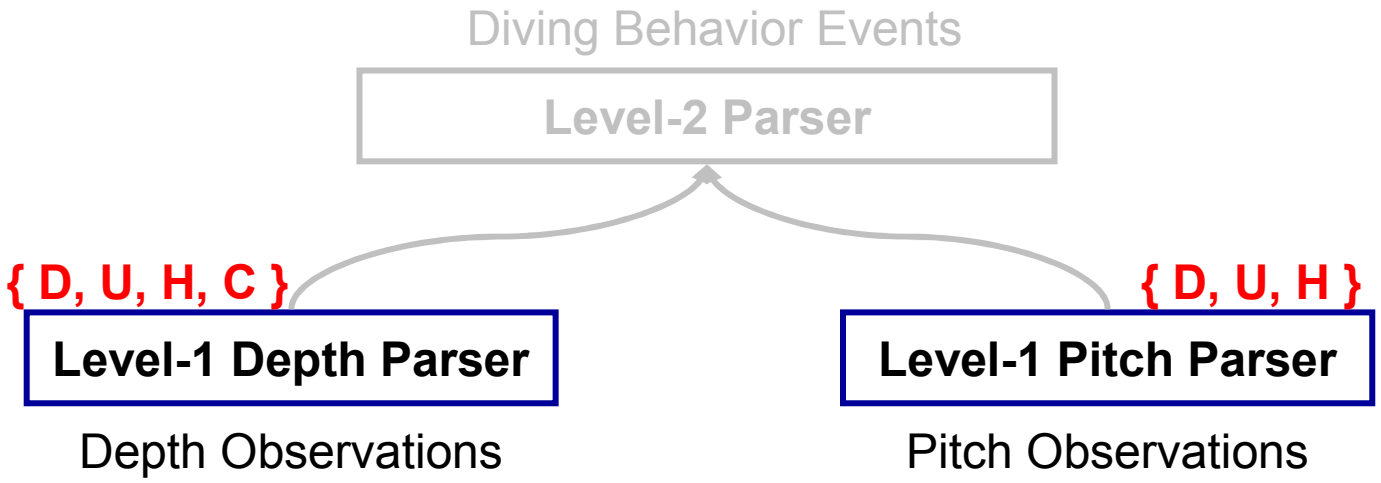


Pitch Sensor Data



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



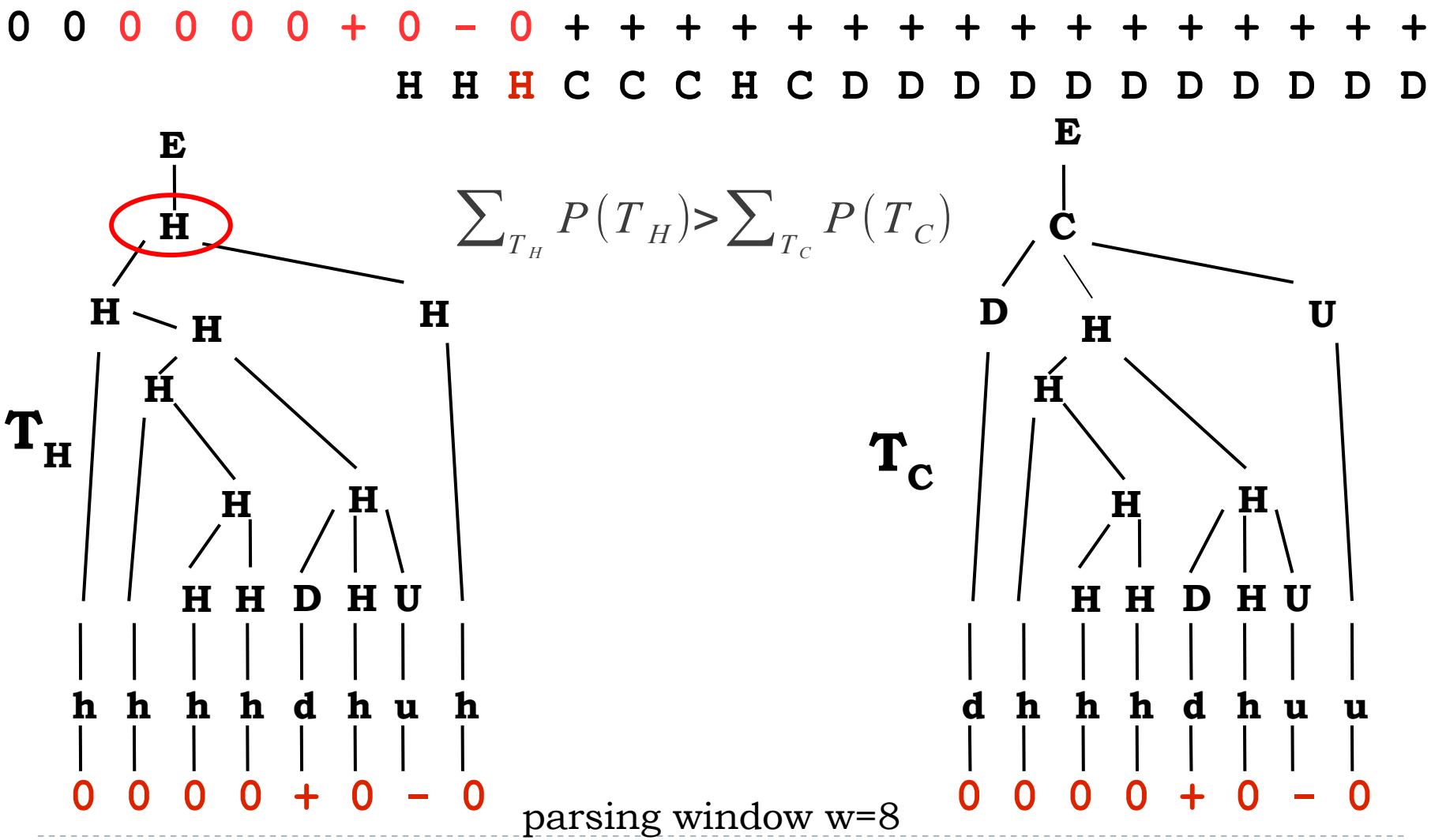
Grammar for Depth Events

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]

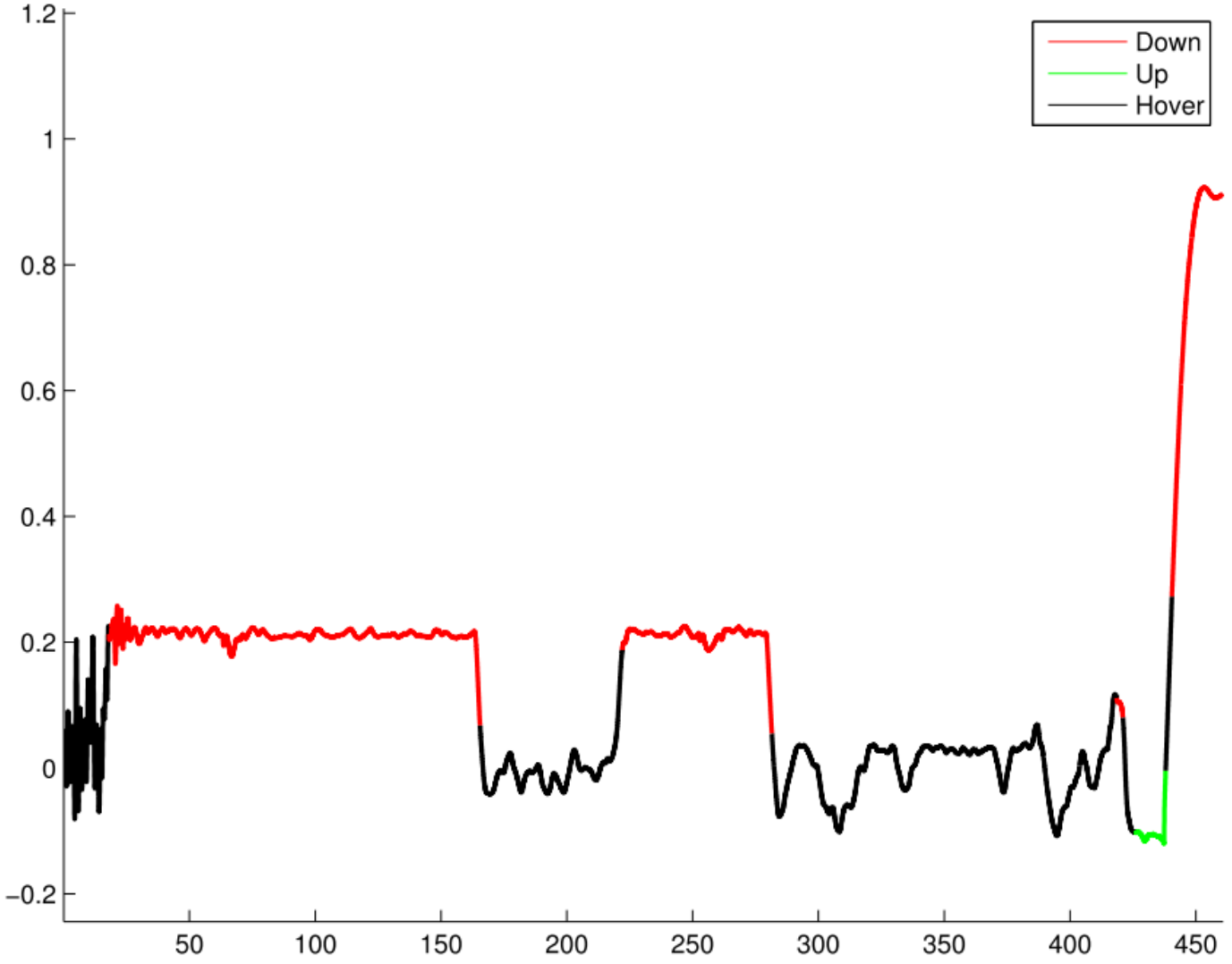
Recognize event E by choosing the derivation that maximizes production probability

Level-1 Parsing: Depth



Level-1 Annotated Depth

Level-1 Annotated Pitch



Level-2 Event Recognition

- **Level-2 Observations**
  - combined events from Level-1
  - DD, DU, DH, UD, UU, UH, HD, HU, HH, CD, CU, CH
- **Level-2 events**
  - **OK**, **Co**llision, **UpD**raft, **DownD**raft
- **Level-2 parsing**
  - input: synchronized Level-1 outputs
  - output: most probable diving event
- **Patterns**
  - the two Level-1 parsers “disagree” for a significant time period

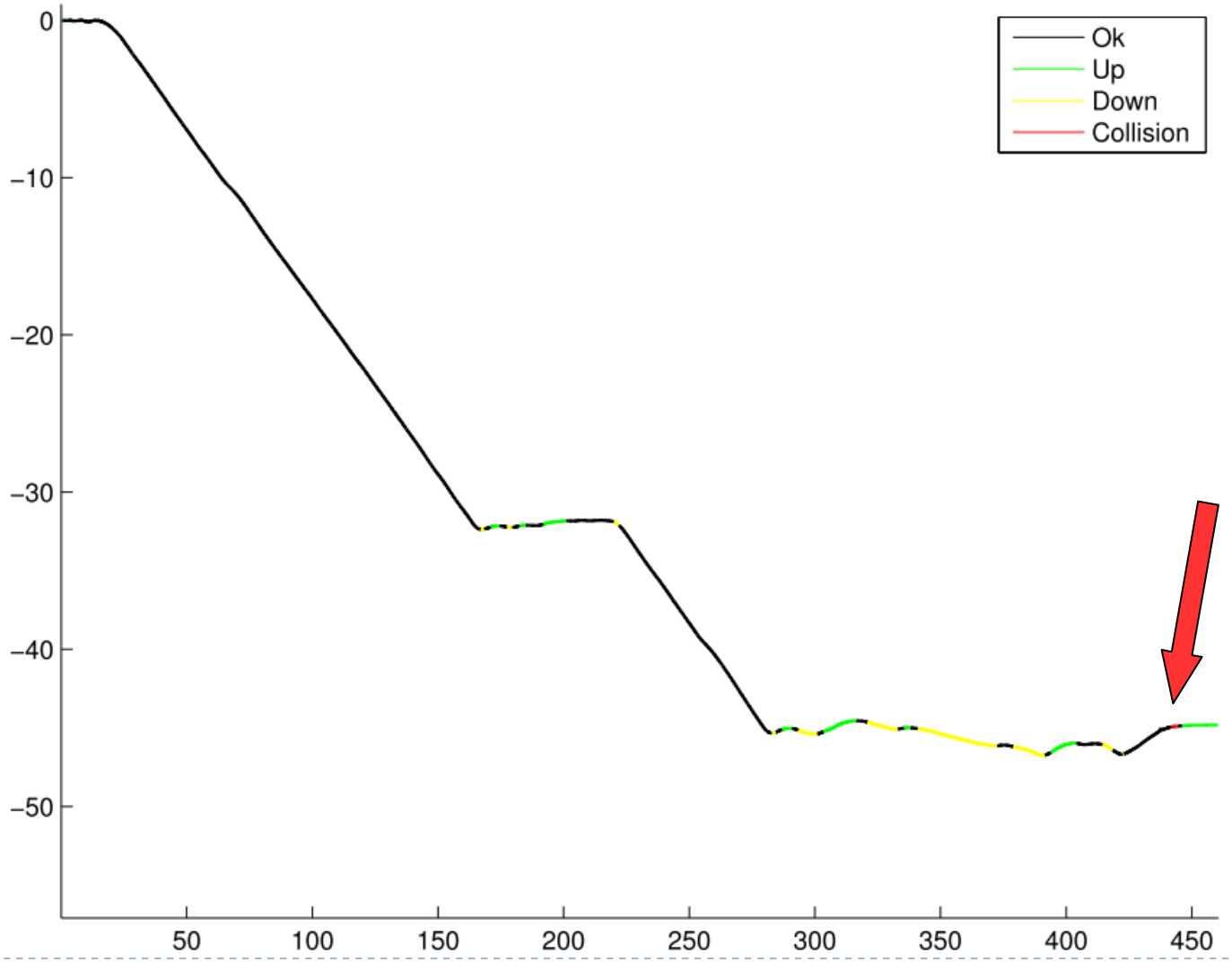
E → OK **[0.889]**
E → Co **[0.001]**
E → UpD **[0.050]**
E → DownD **[0.050]**

Co → Co co [0.500]
Co → co [0.500]

co → o [0.01]
co → 'UD' **[0.90]**

o → 'DD' […]
.
.
.
o → 'UU' [...]

Level-2 Annotated Depth

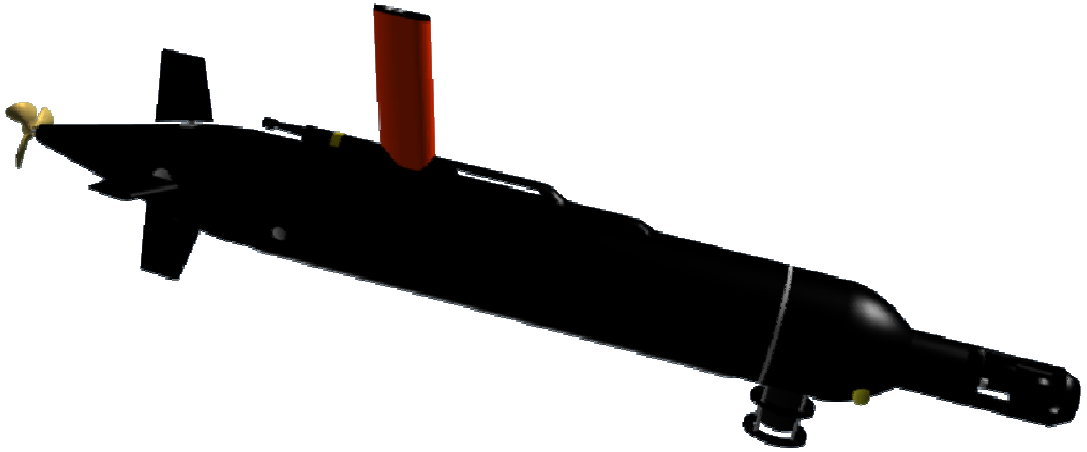


PCFG Event Recognition

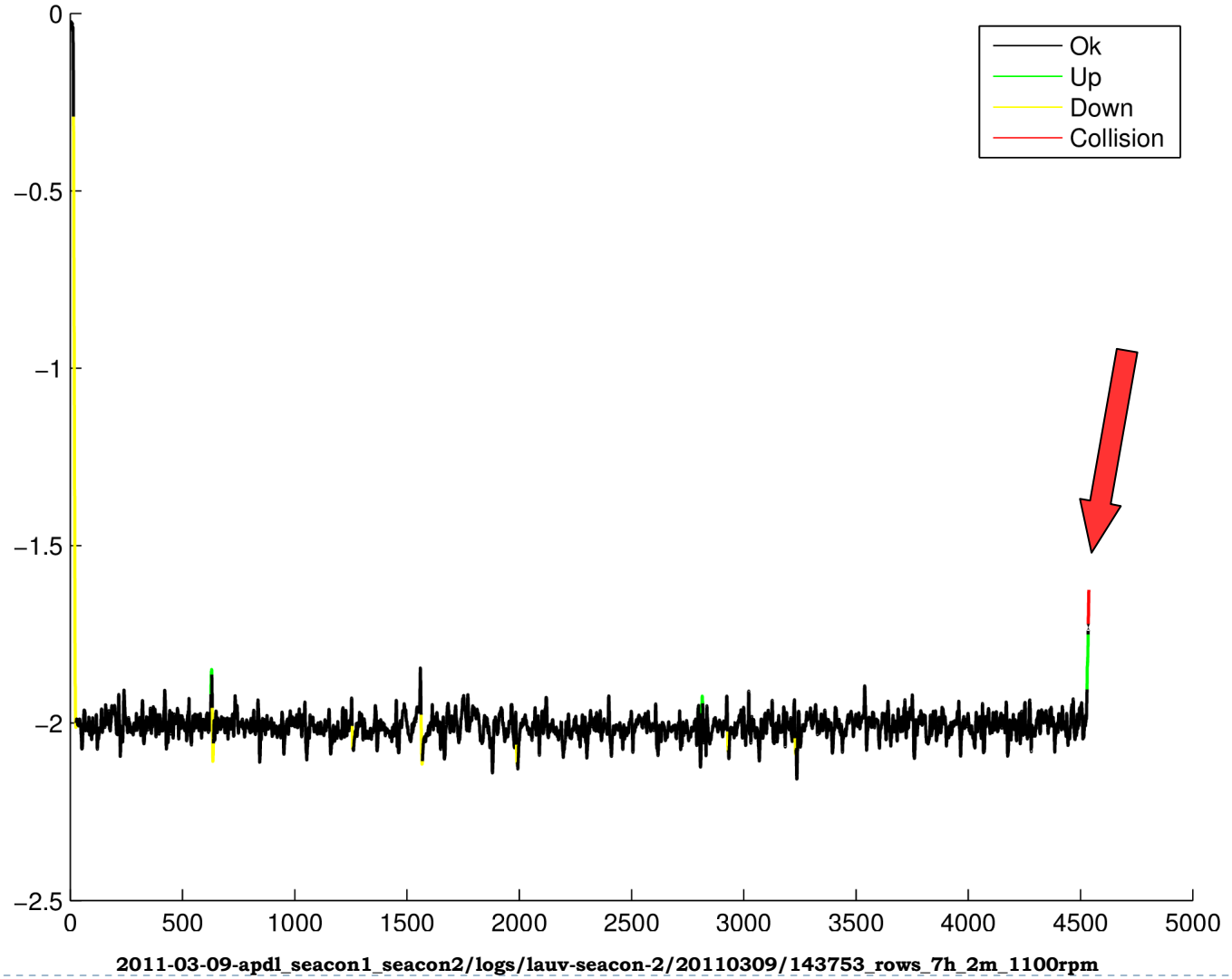
Multiple Experiments

Mission Data

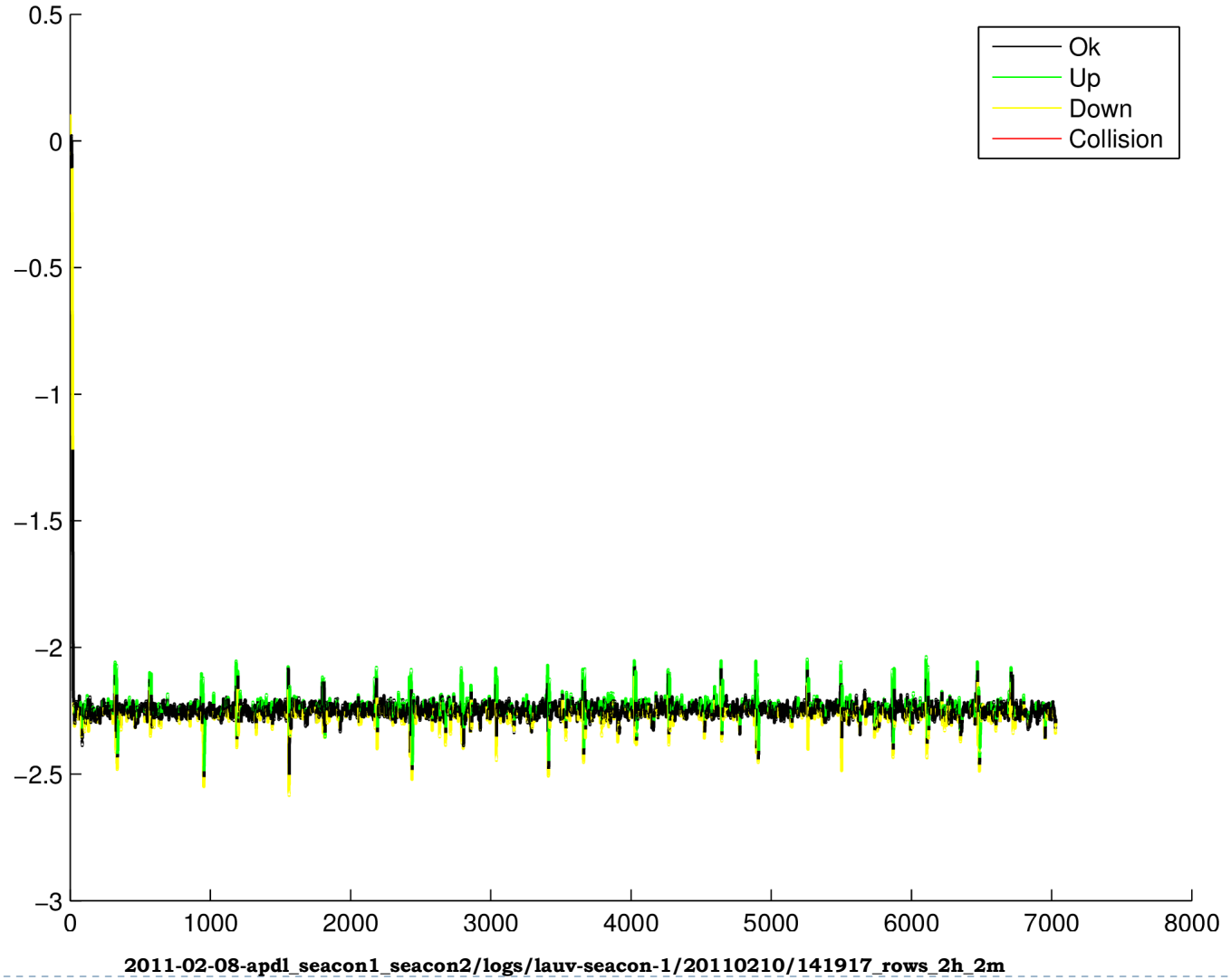
- **Data logs**
  - seacon-1, seacon-2, swordfish
  - more than 20 hours of mission time
- **Human expertise**
  - descriptive annotation of events
  - related sensor streams



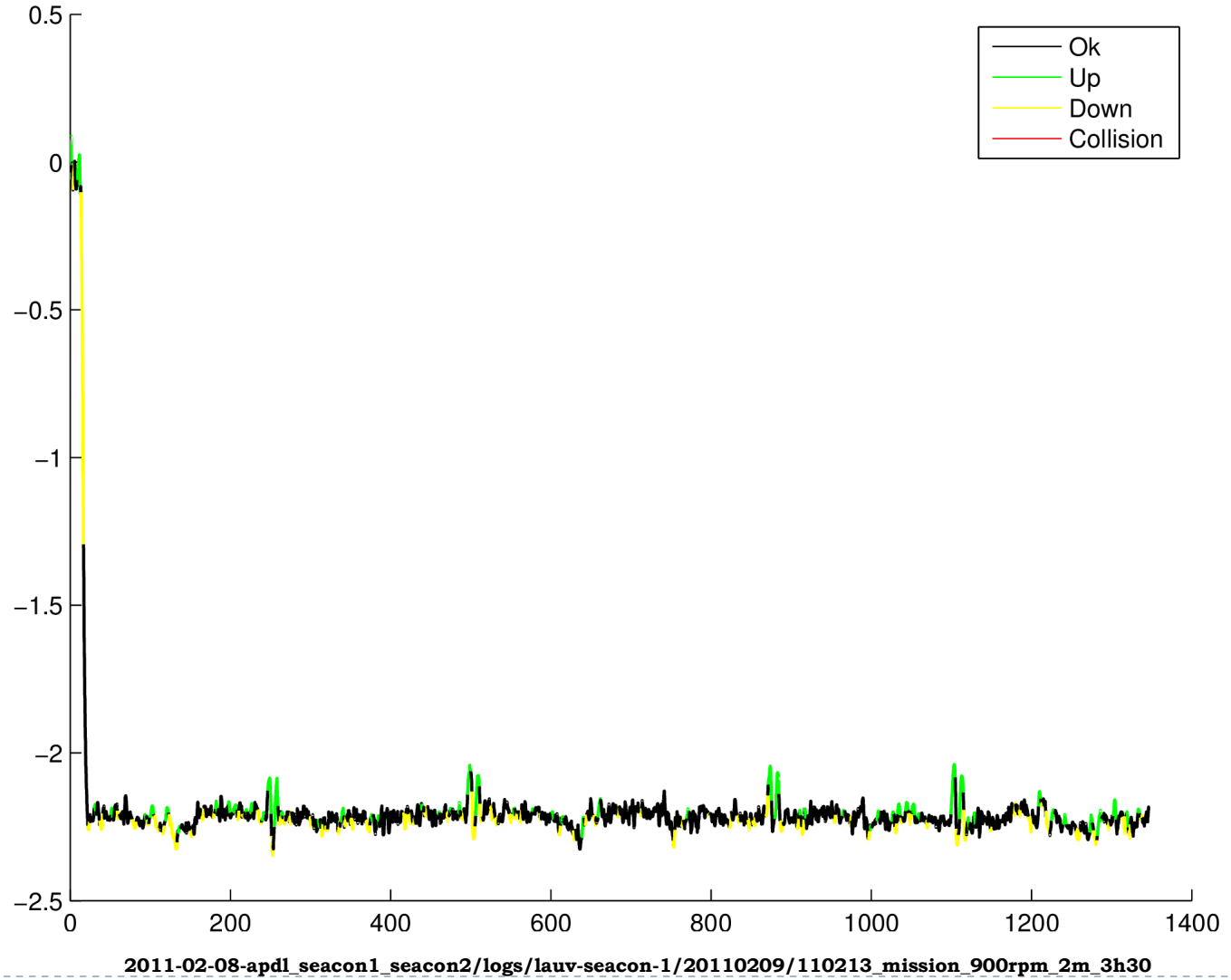
New Mission #1 Depth Annotation



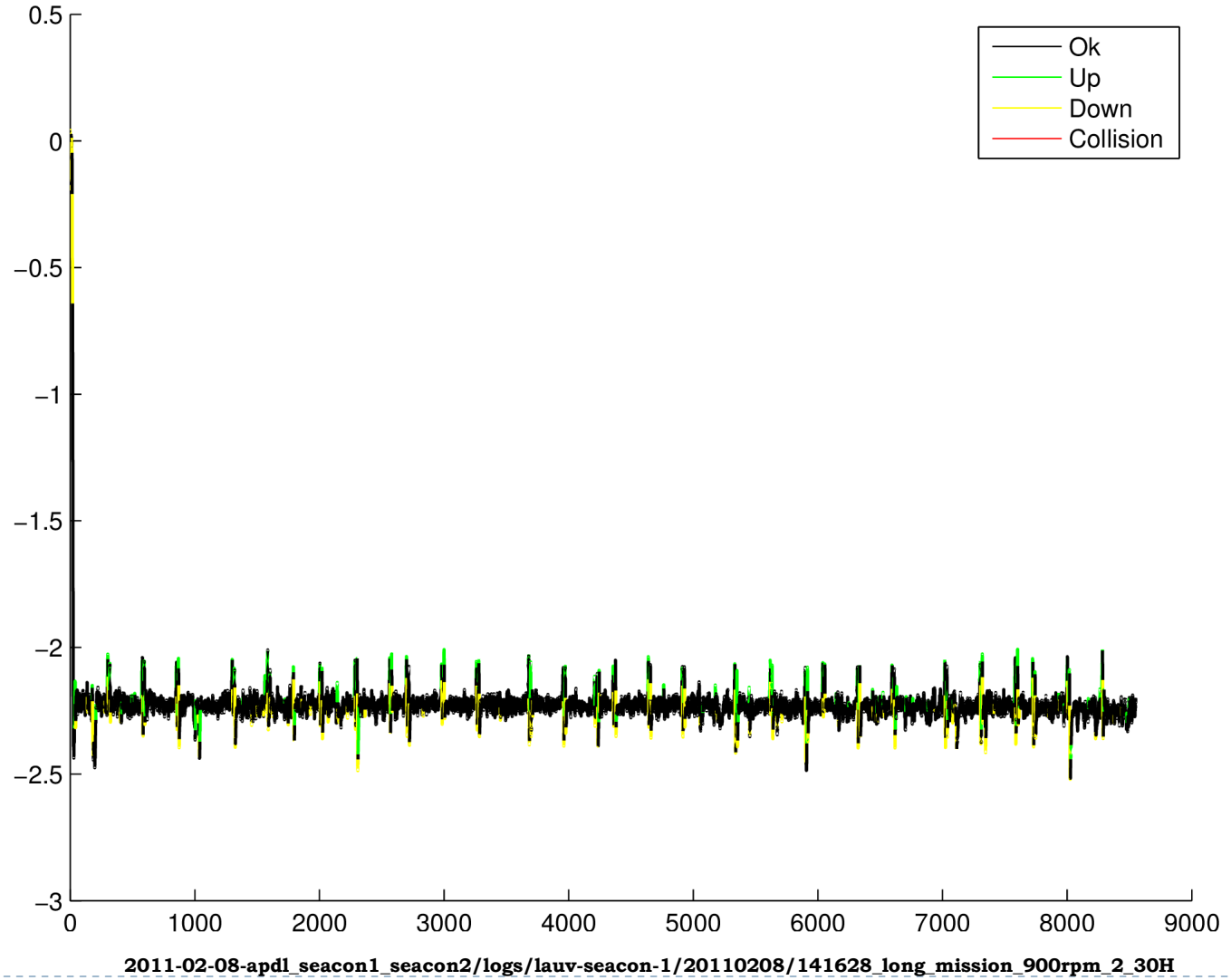
New Mission #2 Depth Annotation



New Mission #3 Depth Annotation



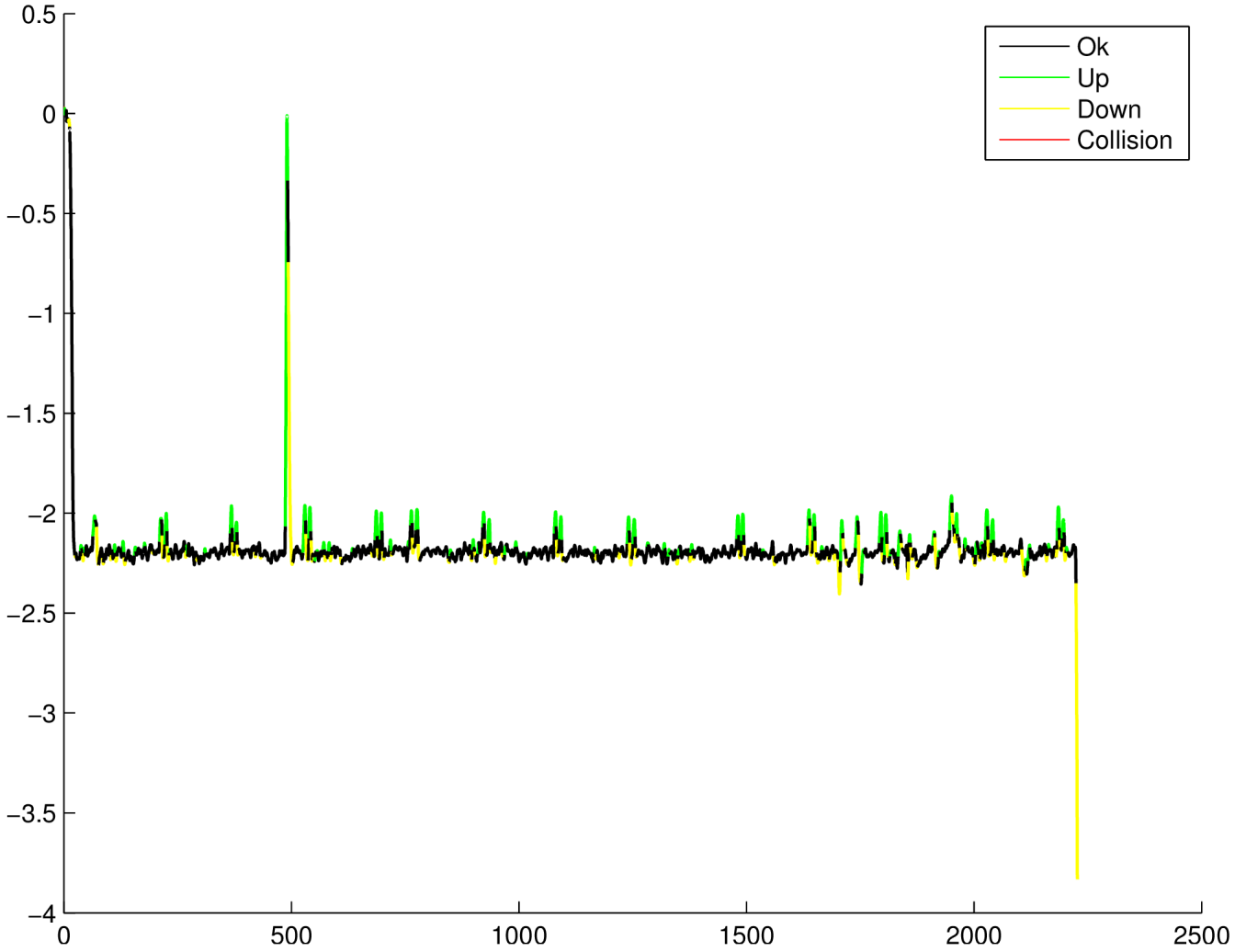
New Mission #4 Depth Annotation



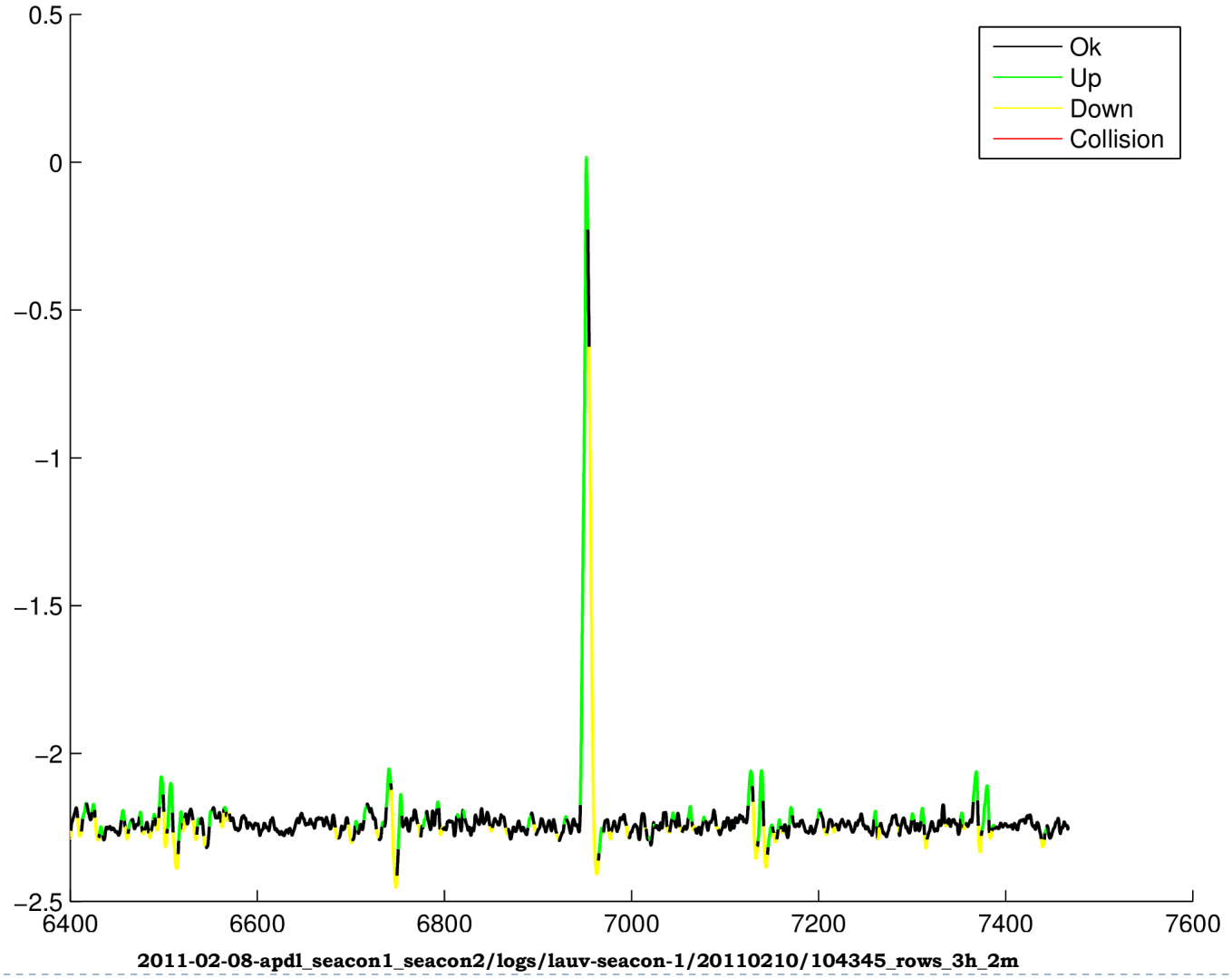
Experimental Results

- **Grammars**
  - minor refinement of the grammars
  - plain tuning of probabilities
- **Event recognition**
  - collision is correctly recognized, where present
  - collision is correctly not recognized, where absent
  - initial grammar not an overfit to the initial log!
- **Concern**
  - lateral collisions were not recognized
  - current grammar unable to detect such events

Unrecognized Lateral Collision #1



Unrecognized Lateral Collision #2



WP6 Progress and Plans

What's there? What's next?

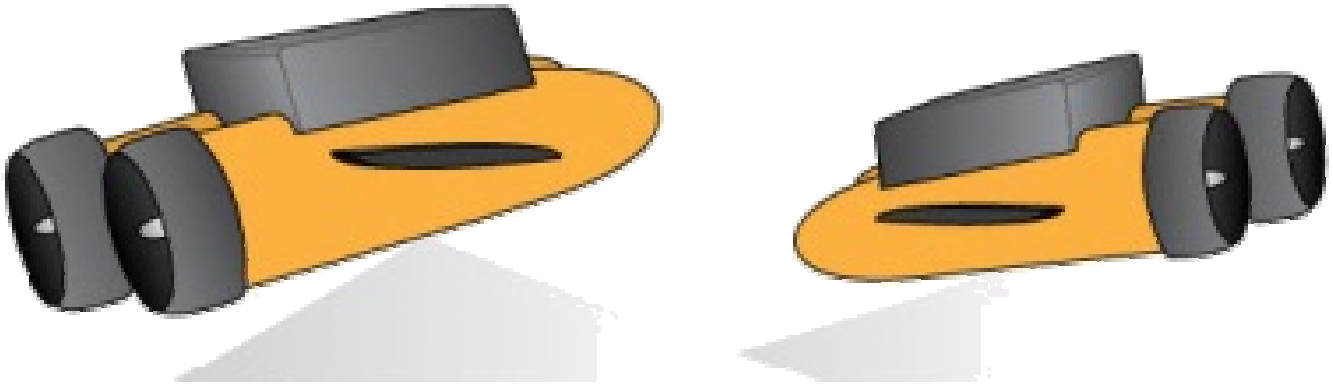
WP6 Progress

- **Preprocessing**
  - abstraction from raw data to observations
- **Event recognition**
  - *events*: diving behavior
  - *technology*: probabilistic context-free grammars
  - successful recognition on several past mission logs
  - demonstration of feasibility and potential
- **Software**
  - optimized parser for event recognition in Matlab

WP6 Plans

- **Software**
  - optimized parser for real-time event recognition in Dune
- **Event recognition**
  - *new events*: getting caught in net, vehicle faults, …
- **Rethinking events**
  - normal vs. abnormal AUV operation
  - *idea*: instead of looking for the abnormal and rare, …
  - … why not look after the normal and frequent?
- **Structured prediction**
  - automatic construction of grammars
  - learning rules and probabilities from past missions

Thank you!



NOPTILUS