Mixed-initiative Task Planning for Autonomous Underwater Vehicles

In collaboration with LSTS lab, University of Porto
[Chrpa et al., 2015;2017]

Automated Planning

- We have **Domain Definition languages** (e.g. PDDL)
- We have **Planning Engines** (e.g., LAMA, LPG)
- So, we can generate **Plans** (quite easily)

- But what about their **execution** ?

Task Planning for AUVs [Chrpa et al., 2015]

- Necessity to control multiple heterogeneous Autonomous Underwater Vehicles (AUVs)
- An operator (human) specifies high-level tasks (e.g. “sample an object with ctd camera”)
- Task assignment to each AUV should be automatized

How task assignment can be automatized ?

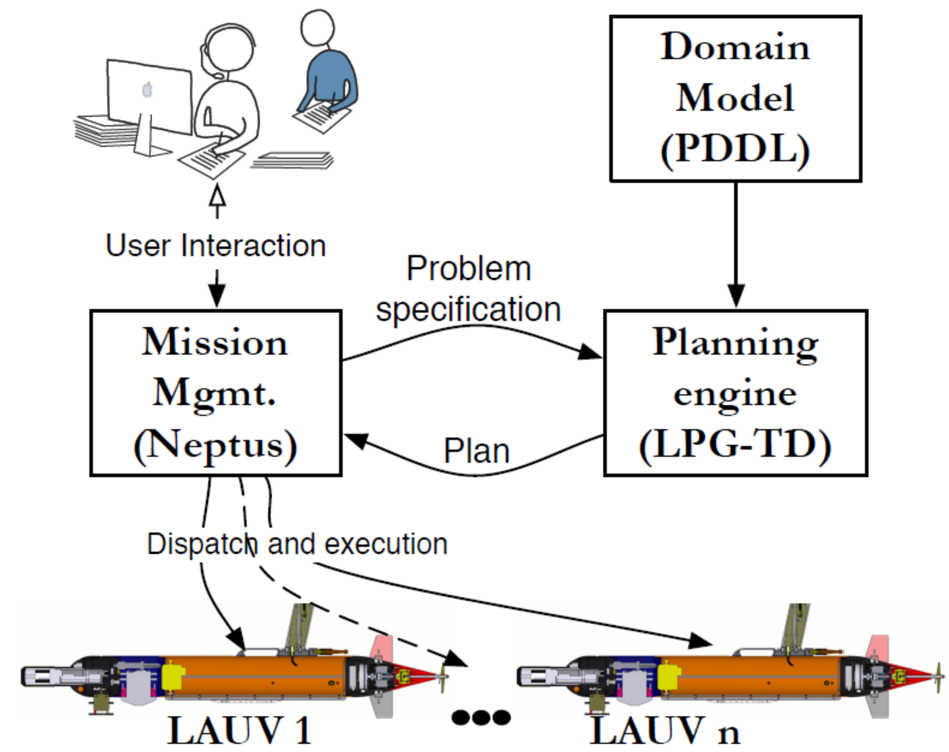
- Each **task** has specific **requirements**
- Each **vehicle** has specific **capabilities**
- For completing tasks AUVs have to perform certain sequences of **actions**
- Hence, we need to find **a plan** that if executed, the AUVs will complete all given tasks

Available “Machinery”

- In LSTS, AUVs are controlled via **NEPTUS** (a decision support tool with GUI) and **DUNE** (onboard vehicle control) → **“low-level” control**
- Domain-independent **AI planning** (i.e., finding a sequence of actions that achieves a defined goal) → **“high-level” task planning**
  - **PDDL**, a language for specifying planning domain models and problem instances
  - **LPG-td**, a planning engine accepting domain and problem descriptions in PDDL and returning a plan (if exists)

Modular Architecture

- **User specifies tasks** in NEPTUS
- NEPTUS **generates a planning problem** and sends it to LPG-td
- LPG-td **returns a plan** to NEPTUS
- NEPTUS **distributes the plan** to each of the vehicles



“High-level” Specification

- Each **AUV** has certain **payloads** attached to it
- Each task must be completed by using a certain **payload** (e.g. camera, sidescan)
- Each AUV has a limited amount of **energy** that is consumed by executing actions
- Collected data can be **communicated** while an AUV is in its “depot” (a “safe spot” close to shore/ship)
- Two (or more) AUVs cannot be at the same **location** or perform the same **task** simultaneously

Formal conceptualization - objects

- Vehicles (*V*)
- Payloads (*P*)
- Phenomenons (*X*)
- Tasks (*T*)
- Locations (*L*)

Formal Conceptualization – predicates

- *at* $\subseteq V \times L$ (vehicle’s location)
- *base* $\subseteq V \times L$ (vehicle’s “depot”)
- *has* $\subseteq V \times P$ (attached payloads to the vehicle)
- *at-phen* $\subseteq X \times L$ (phenomenon’s location)
- *task* $\subseteq T \times X \times P$ (task description)
- *sampled* $\subseteq T \times V$ (acquired task data by vehicle)
- *data* $\subseteq T$ (acquired task data by the control centre)

Formal Conceptualization – (numeric) fluents

- *dist:* $L \times L \rightarrow \mathbb{R}^+$ (distance between locations)
- *survey-dist:* $L \times L \rightarrow \mathbb{R}^+$ (length of survey)
- *speed:* $V \rightarrow \mathbb{R}^+$ (vehicle's speed)
- *battery-level:* $V \rightarrow \mathbb{R}^+$ (vehicle's battery level)
- *battery-use:* $V \cup P \rightarrow \mathbb{R}^+$ (vehicle's or payload's energy consumption)

Formal Conceptualization - actions

Move (v,l1,l2)

Duration: $d=dist(l1,l2)/speed(v)$

Precondition:

At start: $(v,l1) \in at$, $battery\text{-}level(v) \geq d*battery\text{-}use(v)$

At end: $\nexists v' \neq v: (v',l2) \in at$

Effects:

At start: $(v,l1) \notin at$, $battery\text{-}level(v)=battery\text{-}level(v)-d*battery\text{-}use(v)$

At end: $(v,l2) \in at$

Formal Conceptualization - actions

Sample (v,t,x,p,l)

Duration: *d=60* (constant duration)

Precondition:

At start: $battery\text{-}level(v) \geq d*battery\text{-}use(p)$

Overall: $(v,l) \in at$, $(x,l) \in at\text{-}phen$, $(v,p) \in has$, $(t,x,v) \in task$

Effects:

At start: $battery\text{-}level(v)=battery\text{-}level(v)-d*battery\text{-}use(p)$

At end: $(t,v) \in sampled$

Formal Conceptualization - actions

Survey (v,t,x,p,l1,l2)

Duration: *d=survey-dist(l1,l2)*

Precondition:

At start: *(v,l1)∈at, battery-level(v)≥ d*(battery-use(v)+battery-use(p))*

Overall: *(x,l1)∈at-phen, (x,l2)∈at-phen, (v,p)∈has, (t,x,v)∈task*

Effects:

At start: *(v,l1)∉at, battery-level(v)=battery-level(v)-d*(battery-use(v)+battery-use(p))*

At end: *(v,l2)∈at, (t,v)∈sampled*

No concurrent survey action can be executed over *x*

Formal Conceptualization - actions

Collect-data (v,t,l)

Duration: $d=60$ (constant duration)

Precondition:

Overall: $(v,l) \in at$, $(v,l) \in base$, $(t,v) \in sampled$

Effects:

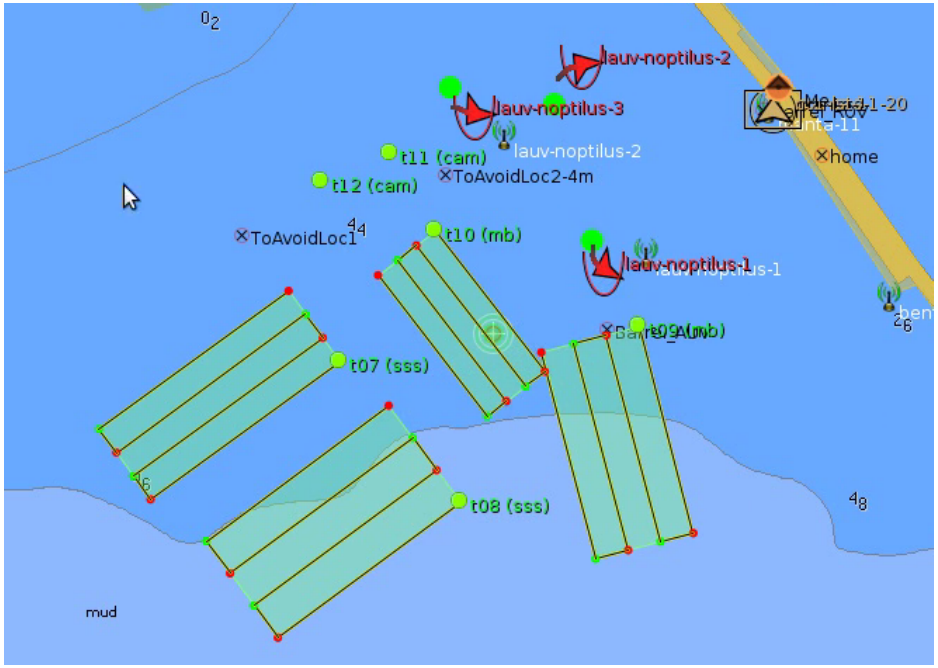
At end: $t \in data$

PDDL model of the Sample action

```
(:durative-action sample
 :parameters (?v - vehicle ?l – location ?t -task
              ?o - phenomenon ?p - payload)
 :duration (= ?duration 60)
 :condition (and (over all (at-phen ?o ?l))
                 (over all (task ?t ?o ?p))
                 (over all (at ?v ?l))
                 (over all (has ?p ?v))
                 (at start (>= (battery-level ?v)
                               (* (battery-use ?p) 60))))
 :effect (and (at end (sampled ?t ?v))
              (at start (decrease (battery-level ?v)
              (* (battery-use ?p) 60)))) )
```

Execution of the model: Settings

- Evaluated in Leixões Harbour, Porto
- 3 light AUVs carrying different payloads
- In phase one, areas of interest were surveyed
- In phase two, contacts identified in phase one were explored



Planned vs. Execution time

- The plans were **executable**
- **High discrepancies**, especially for move and survey actions
- **Rough time predictions** that were done only on distance and type of vehicle

Vehicle	Action	Time Difference
Noptilus-1	move	47.80 ± 49.11
	survey	23.15 ± 23.26
	sample	1.33 ± 0.58
	communicate	0.16 ± 0.17
Noptilus-2	move	39.57 ± 35.66
	survey	107.88 ± 141.10
	sample	N/A
	communicate	0.25 ± 0.07
Noptilus-3	move	59.90 ± 57.05
	survey	24.00 ± 0.00
	sample	9.57 ± 13.64
	communicate	0.11 ± 0.16

Additional Assumptions [Chrpa et al., 2017]

1) Users can add, remove or modify tasks during the mission

2) Vehicles might fail to execute an action

3) Communication with the control center is possible only when a vehicle is in its “depot”

Additional Requirements for the System

- System has to be **flexible** (e.g. a user can add a new task) and **robust** (e.g. handling vehicles' failures)
- Dynamic Planning, Execution and Re-planning
  - Automatized response on task changes by user and/or exceptional circumstances during plan execution
- How the "one shot" model has to be changed ?

Model Amendments

- Removed *battery constraints*
  - vehicles' battery levels were much higher than duration of operations
- Added *maximum "away" time constraints*
  - Vehicles have to come to their depots to establish communication (if they are "away" communication might not be possible)
- Split the *move* action into *move-to-sample, move-to-survey, move-to-base,* the former two must be succeeded by *sample* and *survey* action respectively
  - Optimizing plans (vehicles cannot go to locations they do not have anything to do)
- Modified representation of *phenomenons* (objects and areas of interests are explicitly distinguished)

Maximum "away" time constraints

- Numeric fluents
  - *from-base:* $V \rightarrow \mathbb{R}^+$ (how long the vehicle is "away")
  - *max-to-base:* $V \rightarrow \mathbb{R}^+$ (maximum "away"time)
- Preconditions (at start) of the *move, sample, survey* actions contain *(d* – action duration)*:*
  - *from-depot(v) ≤ max-to-depot(v) – d*
- Effects (at end) of the *move, sample, survey* actions contain *(d* – action duration)*:*
  - *from-depot(v) = from-depot(v) + d*
- Effects (at end) of the *move-to-base* action contain:
  - *from-depot(v)=0*

PDDL model of amended sample action

```
(:durative-action sample
:parameters (?v - vehicle ?l - location ?t -task ?o - oi ?p - payload)
:duration (= ?duration 60)
:condition (and (over all (at-oi ?o ?l))
                (over all (task ?t ?o ?p))
                (over all (at ?v ?l))
                (over all (has ?p ?v))
                (at start (<= (from-base ?v) (- (max-to-base ?v) 60)))
           )
:effect (and (at end (sampled ?t ?v))
             (at end (can-move ?v))
             (at start (increase (from-base ?v) 60))
        )
)
```

Considered Models

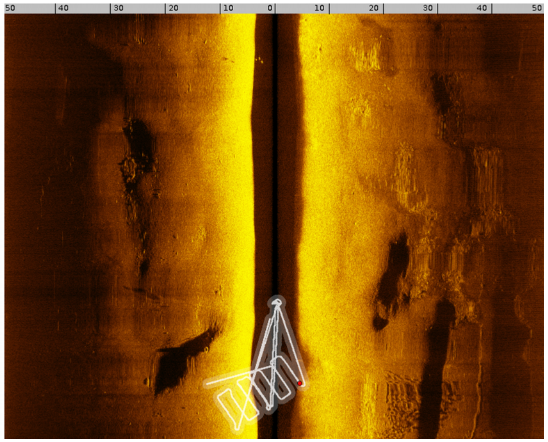
- **All Tasks**
  - Allocates all specified tasks to the vehicles
  - Minimizes the plan execution time and the number of vehicles’ returns to their depots
- **One Round**
  - Allocates only tasks for the next “round” (i.e., after vehicles return to their depots they cannot move)
  - Maximizes the number of completed tasks

Execution

- Preprocessing
  - Splitting large surveillance areas into smaller ones
- Planning
  - NEPTUS generates a problem specification in PDDL, runs LPG-td, then processes and distributes the plan among the vehicles
- Execution
  - Each vehicle is responsible for executing its actions
  - Move actions are translated into timed-waypoints for mitigating the differences between planned and actual times
  - When in depots vehicles communicate status of completed tasks (success/failure) – failed tasks are “re-inserted”
- Replanning
  - If a new planning request comes (e.g. a user added a new task), vehicles continue to execute their current plans until they come back to their depots, then they receive new plans

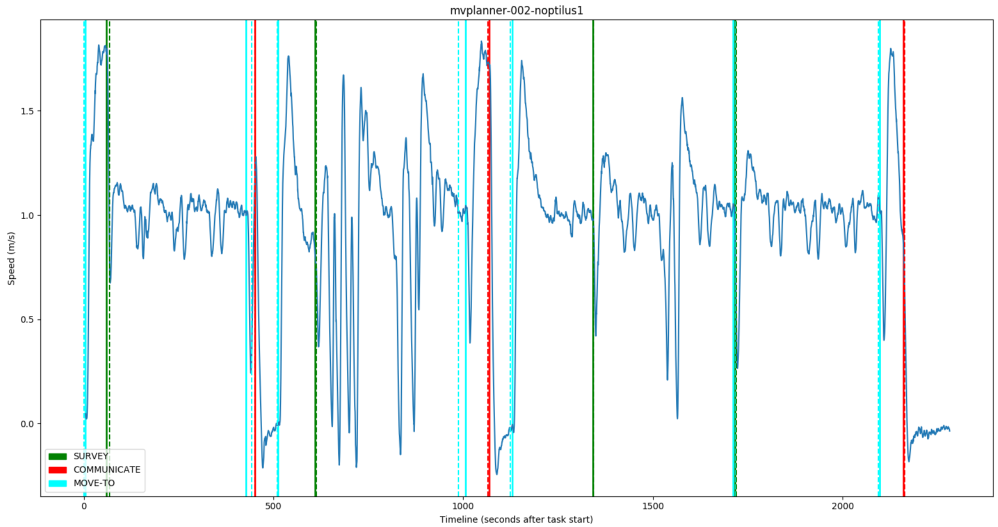
Execution of the models: Settings

- Evaluated in Leixões Harbour, Porto
- Mine-hunting scenario was used
- 3 light AUVs, 2 carried sidescan, one carried camera
- In phase one, areas of interest were surveyed
- In phase two, contacts identified in phase one sampled to identify them as mines, or false positives



Results of the Models Execution

- Both models produced correct plans that were successfully executed
- During one of the executions one AUV (Noptilus 3) failed (depth sensor fault) – tasks were automatically re-inserted and allocated to a different AUV, which completed them
- All Tasks model produces better quality plans (for larger scenarios, however, One Round model might be more efficient)



Most planned/actual differences are quite small (less than 3 seconds).

Around time 1000 a noticeable difference occurred (vehicle had to ascend during the survey). The delay was eliminated by accelerating during the following move action.