Decisional issues in multi-UAV systems

Simon Lacroix
Laboratoire d’Analyse et d’Architecture des Systèmes
CNRS, Toulouse

Lakeside Lab research days on multi-UAV systems,
Klagenfurt, Austria (July 2013)
Where do I come from?

Robotics at LAAS/CNRS, Toulouse, France

- Research topics
  - Perception, planning and decision-making, control
  - Plus: control architecture, interactions, ambient intelligence systems, learning

- Research domains
  - Cognitive and interactive Robotics
  - Aerial and Terrestrial Field Robotics
  - Human and anthropomorphic motion
  - Bio-informatics, Molecular motion

- Considered applications: Planetary exploration, Service and personal robotics, virtual worlds and animation, biochemistry, embedded systems, transport, driver assistance, defense, civil safety

A keyword: autonomy

3 research groups:
  12 full time researchers
  10 university researchers
  4 visitors
  50 PhD students
  10 post-docs
Constructive and integrative approach


Open source software tools: www.openrobots.org
autonomous (adj) (of a country or region) having self-government, at least to a significant degree: the federation included sixteen autonomous republics.
- acting independently or having the freedom to do so: an autonomous committee of the school board / autonomous underwater vehicles.
- (in Kantian moral philosophy) acting in accordance with one's moral duty rather than one's desires.

DERIVATIVES
automously (adv)

ORIGIN early 19th cent.: from Greek autonomos
‘having its own laws’ + -ous.
On autonomy

Notion of *dependence*
- Dependance on the humans
  - Command
  - Skilled operators
  - Lambda users
- Dependence on the infrastructure
  - Abandonned sensors
  - Localisation
  - Communication
  - Databases (géographic, semantic, …)
  - …
- Dependence on the other robots

**Autonomies** :
- Power autonomy
- Execution control autonomy (rather “automatic control”)
- Navigation autonomy
- Decisional autonomy
From automatic control to autonomous control

• Automatic control:
  – Well defined task ("regulate variable", "follow trajectory"…)
  – “Direct” link between perception and action
  – Environment well modeled

• Autonomous control:
  – More general task ("reach position", "monitor area"…)
  – Environment mostly “unknown”, variable…
  – Calls for decisional processes

⇒ “perception / Decision / Action” loop

Plus:
  – Processes integration
  – Learning
  – Interaction with humans
  – Interactions with other robots
  – …
E.g. for a drone:

- Regulate heading / speed / altitude
- Follow a list ordered waypoints
- Follow a geometric trajectory
- Follow a road
- Follow a target
- Survey an area while avoiding threats and obstacles

“Decision”: notion of deliberation, planning, prediction and evaluation of the outcomes of an action
On the importance of *models* for Autonomy

Planning = Simulation + Search
  • Simulation of the effects of an action with a predictive model
  • Search over possible organizations of possible actions to meet a goal or to optimize a criteria

Illustration: autonomous rover navigation

Simple instance of a perception / decision / action loop:
  • Gather data on the environment, structure it into a model
  • Plan the trajectory to find the “optimal” one
  • Execute the trajectory
On the importance of *models* for Autonomy

Planning = Simulation + Search
- Simulation of the effects of an action with a predictive model
- Search over possible organizations of possible actions to meet a goal or to optimize a criteria

Illustration: autonomous rover navigation

Simulation = convolution of action and environment models

Environment models:
- at the heart of autonomy
- at the heart of cooperation
Multiple robots call for more autonomy

Main drivers for autonomy

- Dirty, Dull, Dangerous tasks
- Operations in remote areas
- Allows the deployment of complex systems
- Money savings!

Multiple robotics systems

- Are inherently more complex
- Call for new specific processes:
  - Cooperation
  - Task allocation
  - Task coordination
- Implies new decisional architectures
Notion of Autonomy

Multiple UAVs in the sky

Multiple UAV/UGV systems

Current projects
Multiple UAVs in the sky

Environment model? an empty space!
(possibly with a non uniform atmospheric flow field)

→ Allows for “easy” development at the core of decision

Example 1: “Monitoring a set of locations” mission

→ For a fleet of UAVs, mainly a task allocation problem: which UAV will observe which location?
The task allocation problem

The “canonical” task allocation problem:

- Given:
  - A set of robots \( \{R\} \)
  - A set of tasks \( \{T\} \)
  - A cost function \( c : \{R \times T\} \rightarrow \mathbb{R}^+ \cup \{+\infty\} \)
- Find the allocation \( A^* \) that minimizes the cost sum (or the max. of individual costs, or the individual cost repartition, or...)

A well-known and well-posed problem (also name “optimal allocation problem”) – but highly combinatorial

Main approaches:
- Centralized: optimization (MILP), genetic algorithm, simulated annealing
- Distributed:
  - DCOP, distributed protocols
  - Negotiation-based approaches: market-based approaches
Market based task allocation

Auctions (tasks) are published, robots bid, the “best” bidder gets the task

Basic functions required
- Ability to bid: task insertion cost evaluation
- Auctioning strategies: who places auctions?
- Overall objective function to minimize

Many possibilities for each function, *e.g.*:
- Task insertion
  - From a simple cost addition…
  - … to a (complex) plan update
  - Mix costs, risks, utilities…
- Auctioning strategies
  - Centralized vs. bidders can emit auctions
  - When to close the market?
  - Auctions can concern a set of tasks…
- Objective function
  - Sum of individual costs, dispersion of individual costs, max of individual costs…

*B. Dias “Market-Based Multirobot Coordination: A Survey and Analysis” 2006*
Market based task allocation

Illustration 1: the Multiple travelling salesman problem

- White dot = auction token
- Simple task insertion
- The cost includes an “equity” constraint
- All tasks are allocated before moving
- All robots must fly back home
Market based task allocation

Main features of market-based approaches
• A simple protocol, applicable to a wide variety of complex problems
• Can be distributed (can bear with communication constraints)
• Can handle dynamic events:
  • Robot failures
  • Unexpected events
  • New tasks
• No guarantee on any optimality
Satisfying communication constraints

• One single “survey” task (= square pattern)

• The constraint satisfaction yields new tasks (“com relay”)

![Diagram of communication pattern]
Satisfying communication constraints

Illustration: multi TSP + several constrained “survey” tasks

• 4 robots
• 5 survey tasks
• 18 places to visit
Multiple UAVs in the sky

Environment model? an empty space!
(possibly with a non uniform atmospheric flow field)

Allows for “easy” development at the core of decision

Example 2: “Fly a flock of drones amidst threats”

For a fleet of UAVs, again mainly a task allocation problem: which UAV will jam a threat / protect others?
Fly a flock of drones amidst threats

Given:
- A convoy mission planned on a map of known threats (EW radars) – there are unknown threats (TF radars)
- A fleet of *heterogeneous* UAVs
  - Some are equipped with EW jammers
  - Some are equipped with defence against TF jammers

Geometry of EW jammers
Fly a flock of drones amidst threats

Given:

- A convoy mission planned on a map of known threats (EW radars) – there are unknown threats (TF radars)
- A fleet of *heterogeneous* UAVs
  - Some are equipped with EW jammers
  - Some are equipped with defence against TF jammers

Geometry of TF jammers
Fly a flock of drones amidst threats

Given:
- A convoy mission planned on a map of known threats (EW radars) – there are unknown threats (TF radars)
- A fleet of heterogeneous UAVs
  - Some are equipped with EW jammers
  - Some are equipped with defence against TF jammers

Fly safely the fleet (“Formation-less formation flight”) though the route

- Define the optimal configuration (“formation”) of UAVs
- Manage configuration transitions
Fly a flock of drones amidst threats

Fly safely the fleet (“Formation-less formation flight”) though the route

- Define the optimal configuration (“formation”) of UAVs
- Manage configuration transitions
Fly a flock of drones amidst threats
Notion of Autonomy

Multiple UAVs in the sky
  Monitoring a set of locations
  Fly a flock of drones amidst threats

Multiple UAV/UGV systems

Current projects
Context: teams of AGVs/UGVs
Where and what for?

Dozens of heterogeneous robots cooperate to achieve long-lasting missions in large environments

Considered missions:
- observations, scene analyses, situation assessments
- interventions in the environment

In various application contexts:
- Environment monitoring (pollutions, science, …)
- Search and rescue
- Defense applications, Civil security
Where and what for?

Dozens of heterogeneous robots cooperate to achieve long-lasting missions in large environments

Large scale ($km^3$) implies:
• Faster robots, longer missions (“lifelong autonomy”)
• Communication constraints
• Large (multi-scale) environment models
1. Planning a surveillance mission

Given:

- A team of robots
- An environment to monitor
- A set of constraints to satisfy (e.g. communications)

→ Find the (optimal) trajectories to observe the whole environment
1. Planning a surveillance mission

Given:
- A team of robots
- An environment to monitor
- A set of constraints to satisfy (*e.g.* communications)

Actions to plan:
- Observation tasks (hence motion tasks)
- Communications

Approach:
- A task allocation process (distributed market-based approach)
- Large scale: necessity to interleave allocation and decomposition processes
1. Planning a surveillance mission

The overall mission is not necessarily expressed as a set of elementary tasks: it has to be decomposed/refined.

*Decompose then allocate*  
*Allocate then decompose*
1. Planning a surveillance mission

Decomposition made according to a Hierarchical Task Network scheme (HTN)
- Breaks down the planning complexity
- Allows auctions on variable complexity structures
1. Planning a surveillance mission
2. Navigating a rover in an unknown environment

Given:

• A team of robots

• An unknown environment

• A set of constraints to satisfy \((e.g.\,\text{communications})\)

Find the (optimal) trajectory for the rover to reach a given goal
2. Navigating a rover in an unknown environment

Given:
• A team of robots
• An unknown environment
• A set of constraints to satisfy (e.g. communications)

Actions to plan:
• Environment modelling tasks
• Rover Motions
• Communications

Approach:
• The UAV serves the UGV, by providing traversability maps
• Find the areas to perceive relevant for the mission
2. Navigating a rover in an unknown environment

(simulation with [http://morse.openrobots.org](http://morse.openrobots.org))
Decision and environment models

Planning = Simulation + Search
  • Simulation of the effects of an action with a predictive model
  • Search over possible organizations of possible actions to meet a goal or to optimize a criteria
Decision and environment models

Planning = Simulation + Search

- Simulation of the effects of an action with a predictive model
- Search over possible organizations of possible actions to meet a goal or to optimize a criteria

<table>
<thead>
<tr>
<th>Surveillance</th>
<th>Rover navigation</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Environment</td>
<td>• Environment</td>
</tr>
<tr>
<td>observations</td>
<td>modeling</td>
</tr>
<tr>
<td>• Motions</td>
<td>• Motions</td>
</tr>
<tr>
<td>• Communications</td>
<td>• Communications</td>
</tr>
<tr>
<td>Task allocation scheme</td>
<td>Heuristic graph search</td>
</tr>
</tbody>
</table>

Simulation = convolution of action and environment models

Environment models:
- at the heart of autonomy
- at the heart of cooperation
Decision and environment models

Planning = Simulation + Search
  • Simulation of the effects of an action with a predictive model
    ➔ by “convolving” action models with environment models

What are the actions to plan / decide?
  • Motions
  • Environment observations (payload)
  • Communications (within robots, with the control station)

  • Localization
  • Environment perception and modeling
Decision and environment models

Planning motions

- At a coarse level (itinerary)
  - notion of traversability
    (geometry, terrain nature)
Planning motions

• At a coarse level (itinerary)
  ➔ notion of traversability
  (geometry, terrain nature)

• At a fine level
  ➔ geometry, terrain nature
  (Digital Terrain Map)
Decision and environment models

Planning observations

• Need to predict visibilities
  ➔ geometry (2.5D or 3D)
Decision and environment models

Planning observations

• Need to predict visibilities
  ➔ geometry (2.5D or 3D)

Planning communications

• Need to predict radio visibilities
  ➔ geometry, physical properties
Planning localization

• GPS coverage
• INS / Odometry: terrain nature
• Exteroceptive sensors: landmarks or other models (geometry, appearance models, …)
Planning localization

- GPS coverage
- INS / Odometry: terrain nature
- Exteroceptive sensors: landmarks or other models (geometry, appearance models, …)

Planning environment perception & modeling

- Need to predict the *information gain*
  - amount of information in the environment models (uncertainty, entropy…)
A database of environment models
Building envt. models: information flow

Exhaustive environment description
- Geometry
- Semantics
- Physical properties
- Chemical properties
- Temperature, humidity…

Exteroceptive sensor data
- Images
- Point clouds
- Radar echoes
- …

Environment models
- Initial models (GIS)
- Action models

Sensors ➔ Perception ➔ Decision
Building a digital terrain model

With a rover, using point clouds (here stereovision)

Resampling data to obtain a \( z = f(x,y) \) representation on a regular Cartesian grid
Building a digital terrain model

With a rover, using point clouds (here Velodyne Lidar)

Resampling data to obtain a $z=f(x,y)$ representation on a regular Cartesian grid
Building a digital terrain model

With a UAV, using a Lidar

Resampling data to obtain a $z=f(x,y)$ representation on a regular Cartesian grid

[Paul Chavent @ Onera Toulouse]
Building a traversability model

With a rover, using point clouds (here stereo)
Probabilistic labeling (Bayesian supervised learning)

Possibility to introduce luminance / texture attributes
Much more up-to-date classification / learning processes exist
Building a traversability model

With a drone, using vision

img1

img2
Building a traversability model

With a drone, using vision
Building a traversability model

With a drone, using vision

img1

img2
Building a traversability model

With a drone, using vision
Terrain models: data structures

“Raster” models: regular Cartesian grids

“Raster” models: hierarchical Cartesian grids

Graph structures easily derived
Terrain models: data structures

Triangular irregular meshes
Terrain models: data structures

Volumetric representations: octrees
[octomap]

→ Allows 3D visibility computations
Merging air/ground models?

Traversability models

Digital terrain models

Inter-robot spatial consistency required
Terrain models: key points

1. Whatever the encoded information (terrain class, elevation, traversability, ...), it is *essential* maintain its “quality” (confidence, precision, certainty...):

   • To fuse the various sources of information
     • initial model
     • models built by other robots
     • sensor data
   • To drive the decision processes

2. Spatial consistency is crucial
Localization: a classic problem
On the importance of localization

Localization is required to:

- Ensure the spatial consistency of the built models
- Ensure the achievement of the missions, most often defined in localization terms ("goto [goal]", "explore / monitor [area]", ...)
- Ensure the lowest level (locomotion) controls
- Ensure the proper execution of paths / trajectories
Localization solutions

Huge corpus of technological / algorithmic solutions

• Motion / accelerations sensors (dead reckoning):
  Inherently drifts over time and distances

• Absolute localization means (e.g. radioed beacons)
  Hardly reliable, often too coarse

→ Develop solutions relying on the robot exteroceptive sensors
On the importance of localization
But… what localization?

Essential questions to answer:

1. With which precision? From $cm$ to $meters$
2. In which frame? Absolute vs. local
3. At which frequency? From $kHz$ to “sometimes”

- Ensure the lowest level (locomotion) controls
- Ensure the proper execution of paths / trajectories
- Ensure the spatial consistency of the built models
- Ensure the achievement of the missions, most often defined in localization terms (“goto [goal]”, “explore / monitor [area]”, …)

$cm$ accuracy, @ $> 100$ Hz, local frame

$\sim m$ accuracy, “sometimes”, global frame
Localization precision required for a DTM

DTM resolution ~ 10cm, height precision ~ 3cm

- Velodyne lidar provides chunks of 64 points @ 3.5 kHz: 1° error on pitch yields a 17cm elevation error @ 10m

2m/s, GPS RTK @ 20Hz
+ Xsens AHRS @ 100Hz
+ FOG gyro @ 50Hz
Localization precision required for a DTM

- DTM built by an UAV with a Lidar

2m/s, GPS RTK @ 20Hz
+ INS @ x Hz
+ dynamic model
+ compass x Hz
Localization precision required for a DTM

- DTM built by an UAV with a Lidar

During a calm day

With a 10 km/h wind
Visual odometry: principle

1. Stereovision

2. Pixels tracking

3. Motion estimation

Stereovision
Visual odometry on a MAV (+ 3D modelling)
Simultaneous Localization and Mapping

- **Dead reckoning**
  - Monotonic increase of the position uncertainty

- **SLAM**
  - “memory effect” of the mapping
  - Loop closures: position uncertainty decrease
Illustration: 100 Hz vision / low cost INS SLAM
Illustration: 100 Hz vision / low cost INS SLAM
Localization precision required for a DTM

- DTM resolution ~ $10cm$, height precision ~ $3cm$

- Velodyne lidar provides chunks of 64 points @ 3.5 kHz:
  $1^\circ$ error on pitch yields a $17cm$ elevation error @ $10m$

2 m/s, GPS RTK @ 20Hz
+ Xsens AHRS @ 50Hz
+ FOG gyro @ 50Hz

2 m/s, RT-SLAM @ 100Hz
Localization precision required for a DTM

- DTM built by an UAV with a Lidar

2m/s, GPS RTK @ 20Hz + INS @ x Hz + dynamic model + compass x Hz
Localization precision required for a DTM

- DTM built by an UAV with a Lidar

With positions obtained after a global BA (could have been RT-SLAM)
SLAM issues

- SLAM processes complexity grows with the number of landmarks
  - The map size can’t scale up
- The convergence of Kalman filter based solutions can’t be guaranteed
  - The map size can’t scale up, loop closures may lead to inconsistencies
Multi-map hierarchical SLAM

Hierarchical SLAM [Tardos-2005], a graph of “submaps”:
- **Local** maps (EKF) of current vehicle pose and landmarks pose (nodes)
- **Global** map of relative transformations (edges)

Local maps:
- Fully correlated maps (robot and landmark states)
- No information shared between local maps
- Each map is initialized with no uncertainty
Multi-map hierarchical SLAM

Hierarchical SLAM [Tardos-2005], a graph of “submaps”:
- **Local** maps (EKF) of current vehicle pose and landmarks pose (nodes)
- **Global** map of relative transformations (edges)

Global graph of maps:
- Robot’s pose
- The state is the relative transformation between local maps
- Block diagonal covariance before loop closure
Multi-map hierarchical SLAM

Hierarchical SLAM [Tardos-2005], a graph of “submaps”:
- Local maps (EKF) of current vehicle pose and landmarks pose (nodes)
- Global map of relative transformations (edges)

Loop closures in the global graph:
Loop constraint
\[ h(x) = \hat{x}_1 \oplus \hat{x}_2 \cdots \oplus \hat{x}_{n-1} \oplus \hat{x}_n = 0 \]

Minimisation subject to the loop constraint
\[
\min_x f(x) = \min_x \frac{1}{2} (x - \hat{x}_u)^T P_u^{-1} (x - \hat{x}_u) \\
h(x) = 0
\]
Multi-map hierarchical SLAM

Hierarchical SLAM [Tardos-2005], a graph of “submaps”:
- **Local** maps (EKF) of current vehicle pose and landmarks pose (nodes)
- **Global** map of relative transformations (edges)

Loop closures in the global graph:

Loop constraint

\[ h(x) = \hat{x}_1 \oplus \hat{x}_2 \oplus \cdots \oplus \hat{x}_{n-1} \oplus \hat{x}_n = 0 \]

Minimisation subject to the loop constraint

\[
\min_x f(x) = \min_x \frac{1}{2} (x - \hat{x}_u)^T P_u^{-1} (x - \hat{x}_u) \\
h(x) = 0
\]
A distributed multi-robots multi-map approach

- Straightforward extension to hierarchical SLAM

Local level

Global level

A set of fully correlated submaps

A graph of map poses
A distributed multi-robots multi-map approach

Various loop-closing events

“Rendez-vous”: inter-robot pose estimation

Absolute localization (GPS fix / localization wrt. an initial map)

Inter-robot landmark (or map) matches
Detecting loop closures between air/ground robots

Visual point landmarks can’t be exploited

Need to focus on the M of SLAM

Geometry is the key
Points vs. lines in vision
Preliminary multi-robot SLAM results

Experiments with real data
Research perspectives on envt. models

Focus on geometric (3d, vectorized) representations

Integrate existing data (GIS)

Distributed models
Management

Humans in the loop: information sharing (spatial ontologies ?)
Notion of Autonomy

Multiple UAVs in the sky
  Monitoring a set of locations
  Fly a flock of drones amidst threats

Multiple UAV/UGV systems
  Illustrations: need for environments models
  Illustration of environment model building processes
  Importance of localization

Current projects
The ARCAS project

www.arcas-project.eu/ : “development and experimental validation of cooperative UAV systems for assembly and structure construction”
The SkyScanner project

Adaptive synchronous sampling of clouds with a fleet of UAVs

(energy harvesting)
The SkyScanner project

Adaptive synchronous sampling of clouds with a fleet of UAVs
(energy harvesting)

At each At time $t$

1. Collect infos. where ?
2. Who flies where ?
Adaptive synchronous sampling of clouds with a fleet of UAVs (energy harvesting)

À un instant $t$
1. Collect infos. where ?
2. Who flies where ?
Take home messages

• Autonomy calls for specific decisional processes

• Good representations are the foundations of good decisions, and hence of good cooperations

• A *variety* of representations is required

• Geometry is certainly the most important information to represent (but not only)

• Maintaining the quality of information is essential