Hybrid systems models in the coordination and control of networked vehicle systems

João Tasso de Figueiredo Borges de Sousa

Departamento Engenharia Electrotécnica e Computadores

Faculdade de Engenharia da Universidade do Porto

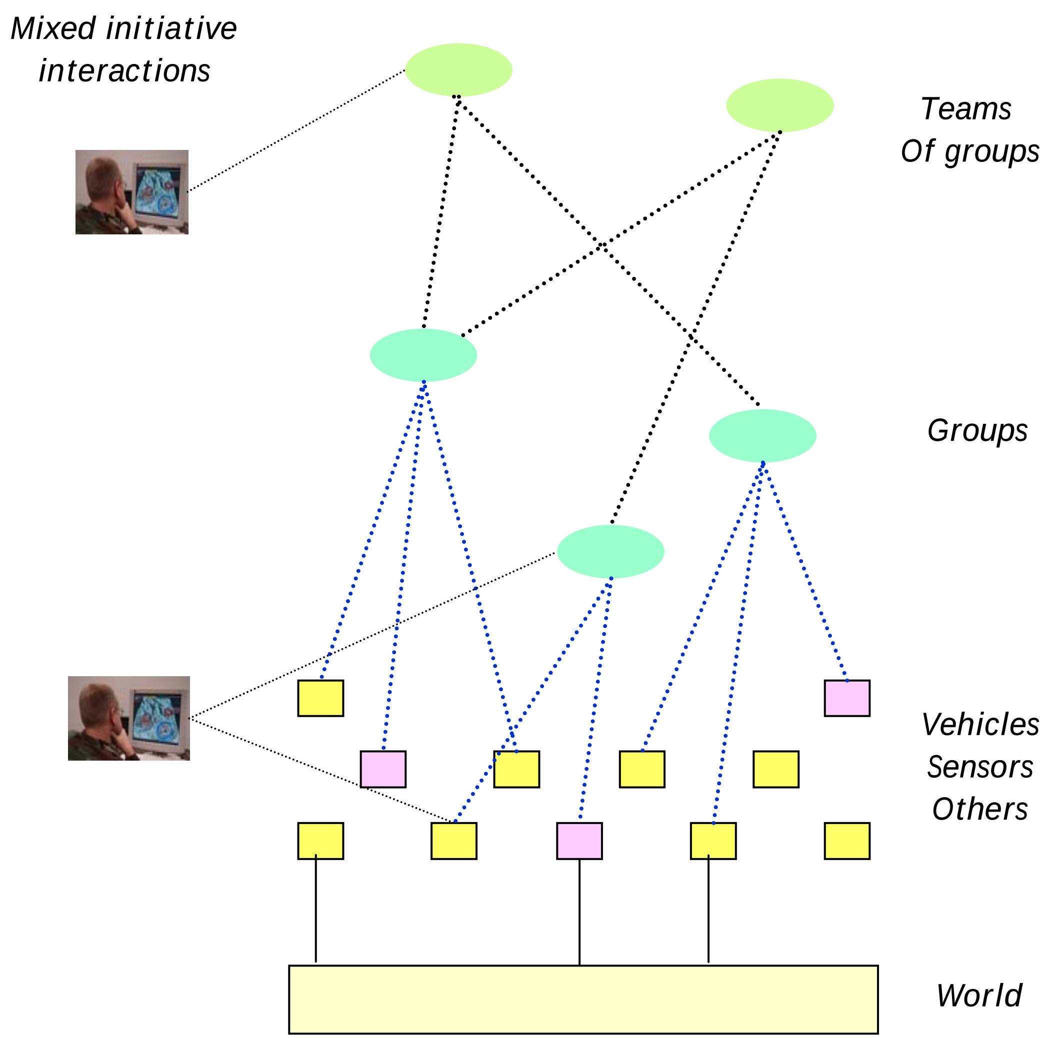
R. Dr. Roberto Frias, 4200 465 Porto

E-mail: jtasso@fe.up.pt

Universidade do Porto
Faculdade de Engenharia
FEUP

LSTS

Mixed initiative interactions

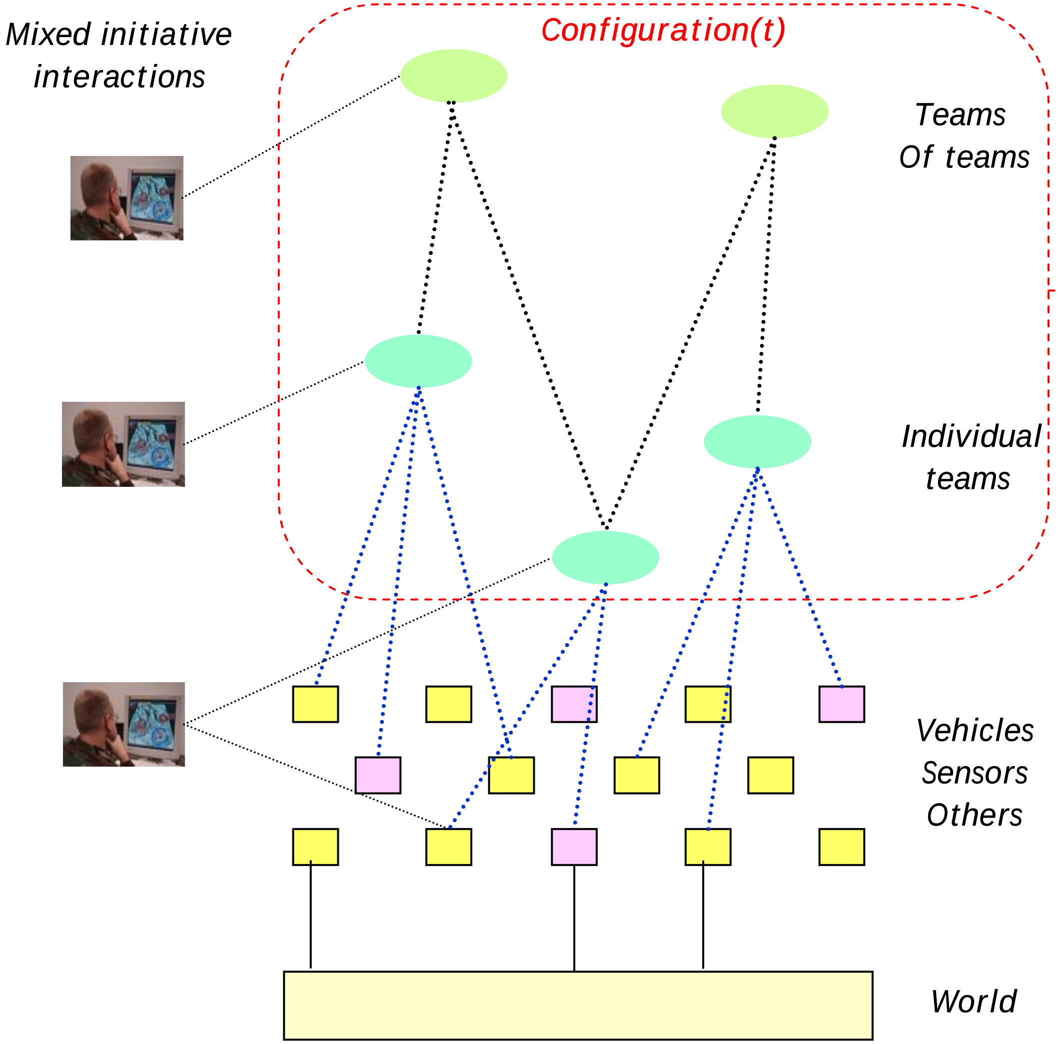


Humans in the planning/control loop

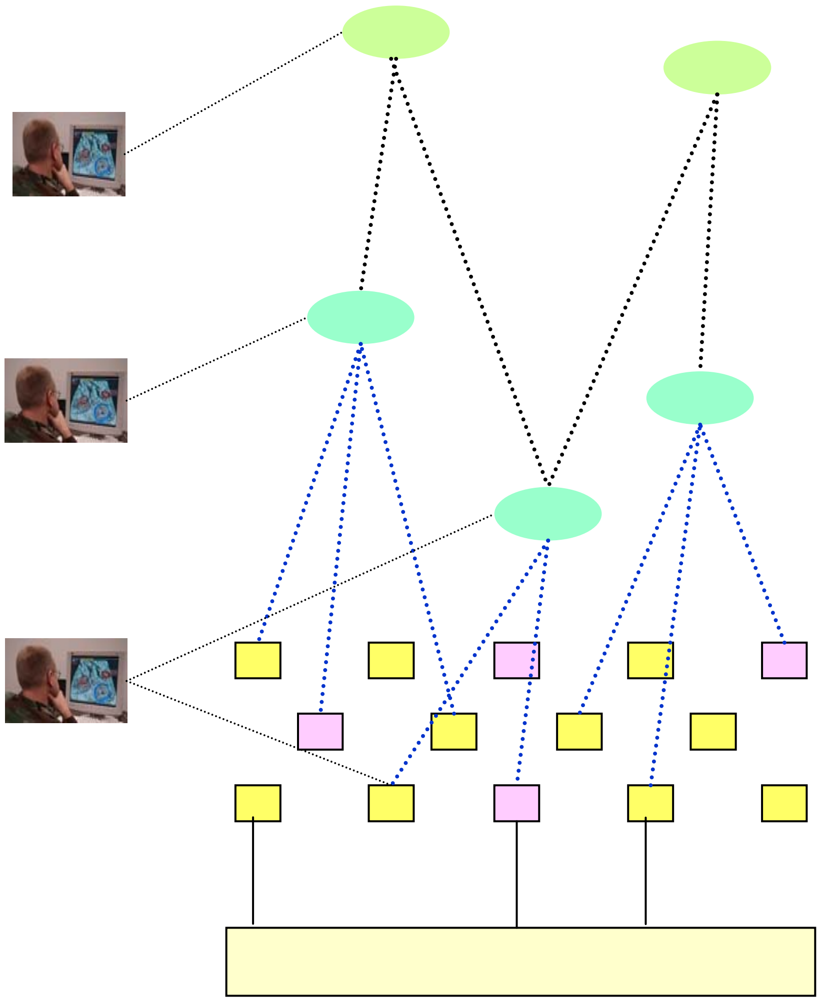
1. Planning & execution control must allow intervention by experienced human operators (experience and operational insight of these operators cannot be reflected in mathematical models)
2. It is impossible to design controllers that can respond satisfactorily to every possible contingency (In unforeseen situations, controllers ask the human operators for direction)

Humans need to understand planning and execution concepts/abstractions

How?



- Control
  - System with evolving structure
  - What is the state of the system?
  - What is the space of controls?
  - How to incorporate mixed initiative interactions?
  - What is a control strategy?
  - Control under communication constraints
- Tools
  - Deployment of dynamic networks hybrid automata
  - Interoperability and standards
  - Real time code generation and distribution



In our opinion, we do not believe that the existing mathematical tools … are powerful enough to define a preferable structure for decentralized and/or hierarchical control. Rather, any future methodology should strive to define sets of distributed information and control structures that are preferable to others.

Sandell, Athans, Varaiya, Safonov, 1978

Dynamic reconfiguration is a common feature of communicating systems. The notion of link, not as a fixed part of the system but as a datum that we can manipulate, is essential for understanding such systems.

What is the mathematics of linkage?

The theories of computation are evolving from notions like value, evaluation and function to those of link, interaction and process.

Milner, 1999

DARPA-MICA

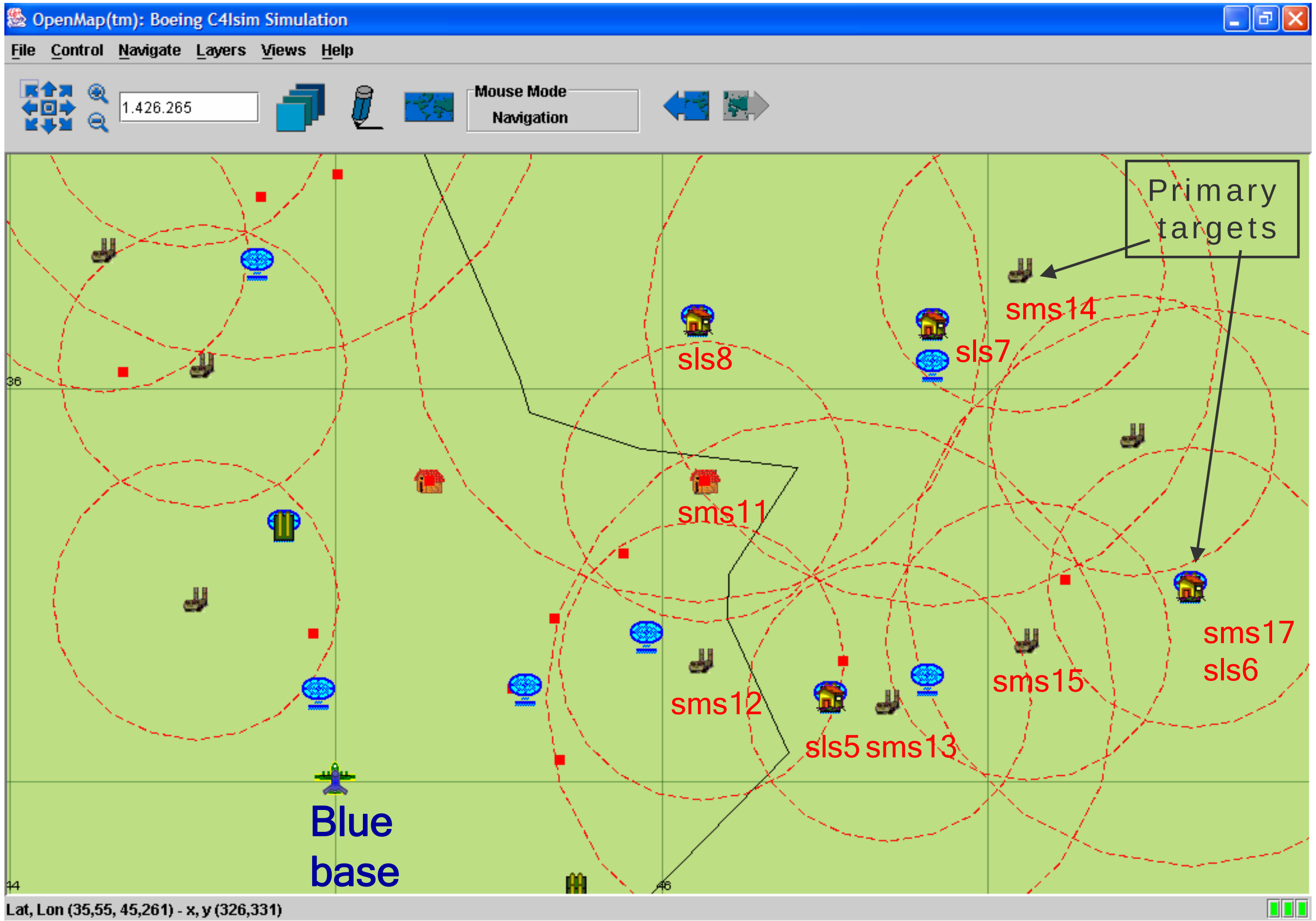
Task planning and execution for UAV teams

- João Sousa*, Pravin Varaiya+ and Tunc Simsek+
  - *Dept. Engenharia Electrotécnica e de Computadores
  - Universidade do Porto, Portugal
  - + Dept. of Electrical Engineering and Computer Sciences
  - University of California, Berkeley, CA 94720
  - {sousa, varaiya, simsek}@eecs.berkeley.edu

  - Research supported by Darpa Contract F33615-01-C-3150.

UAV task planning

- Design the attack of the Blue force of UAV against Red's ground force of SAM sites and radars



J. Borges de Sousa, T. Simsek e P. Varaiya, "Task planning and execution for UAV teams", Proceedings of the Decision and Control Conference, Bahamas, 2004

Threat function and path risk

- Instantaneous threat $r(x,y;P_{A,N}) =$

$$\sum_{j=1}^{k}\sum_{t}\sum_{N_{tj}=0}^{\infty}\sum_{n=1}^{N_{tj}}\left[\int_{A_j} f_t(|(x,y)-(x_n,y_n)|)|A_j|^{-1}dx_n dy_n\right] P(N_{tj})$$

$f_t(d)$ is the instantaneous threat posed at a distance d from target if type d

The integral is the expected value of this threat

- Risk faced by a UAV flying at speed v> 0 along a path γ from $\gamma(0)$ to a destination $\gamma(\tau)$ facing threat $P_{A,N}$

$$\rho(\gamma;P_{A,N}) = \int_{\sigma=0}^{\tau} r(\gamma(\sigma);P_{A,N})\frac{d\sigma}{v},$$

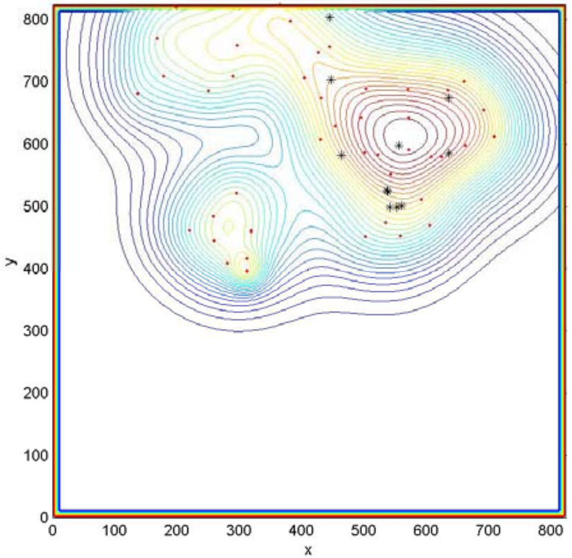
- Value function for threat $P_{A,N}$ with $\gamma(\tau) = (\bar{x},\bar{y})$

$$V((\bar{x},\bar{y});P_{A,N}) = \min_{\gamma} \rho(\gamma;P_{A,N})$$

Initial situation

IPB

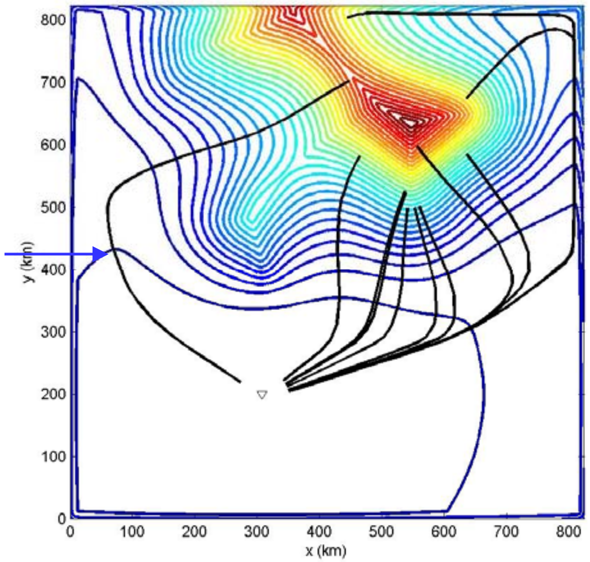
Threat function



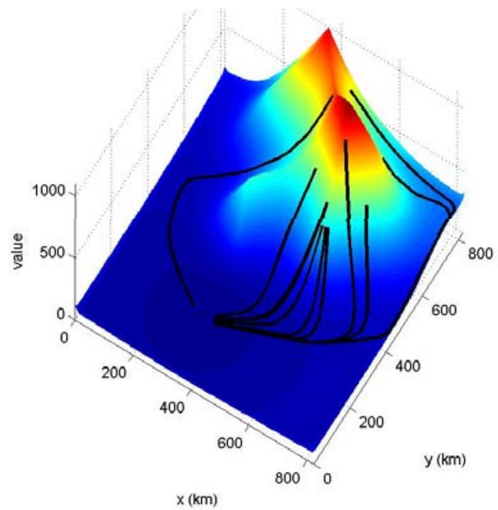
Idea: Phased operation waves

- Resoluti… km
- Number of threats: 91
- Primary targets = {EW1-6,TEL1-4, TELS1-2}

Value function



Minimum risk paths to primary targets



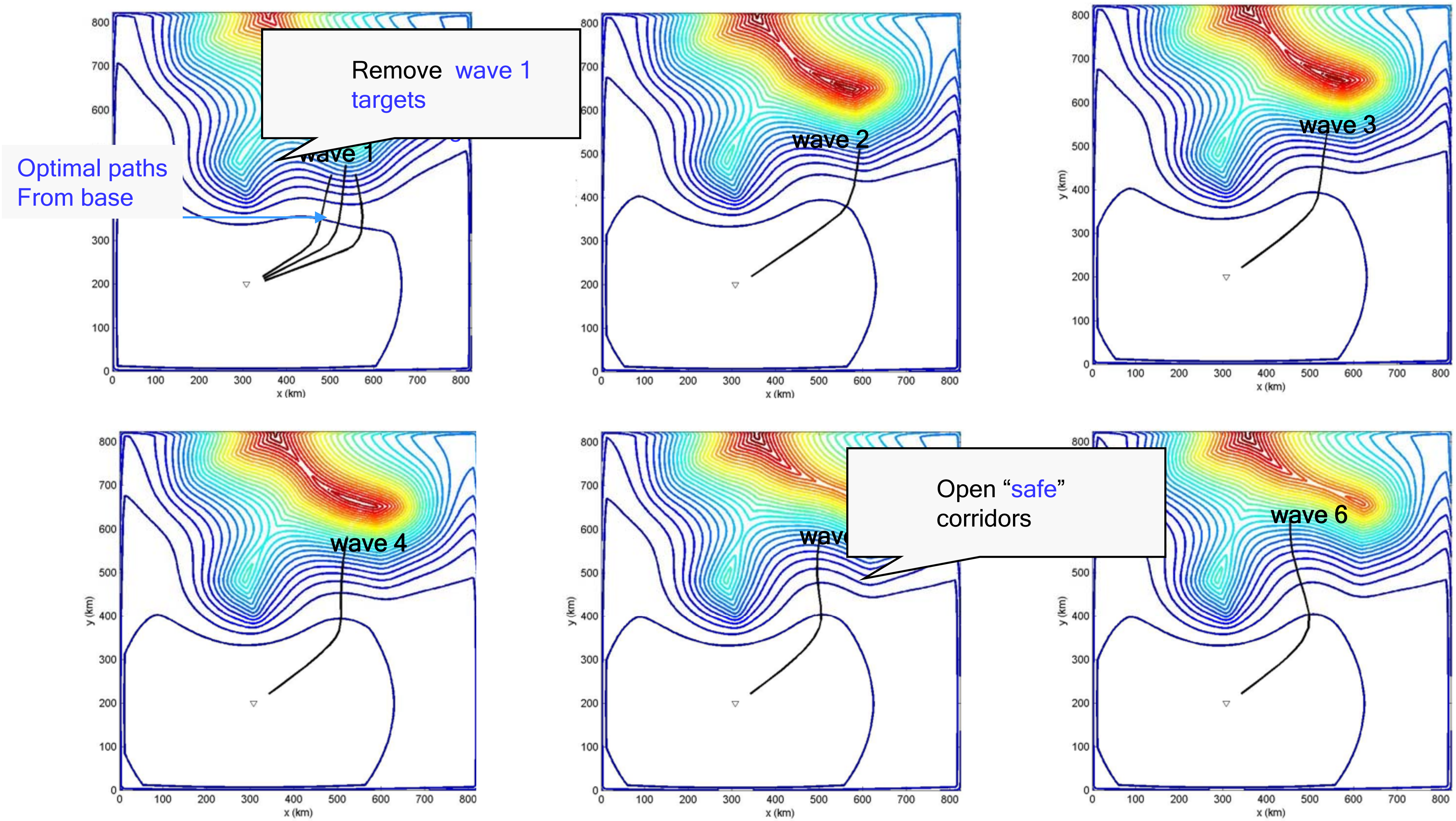
Wave selection

- i = 1
  - Start with initial threat distribution.
  - Determine value function to all targets.
  - Select wave 1 = targets with minimum cost < threshold
- i = 2
  - Remove wave 1 targets.
  - Recalculate value functions for remaining targets.
  - Select wave 2 = targets with minimum cost < threshold
- …
- i = k
  - Remove wave k-1 targets.
  - Recalculate value functions for remaining targets.
  - Select wave k = targets with minimum cost < threshold
- Continue until all primary targets are removed

Heuristic: Add potential targets to reduce risk or create corridors

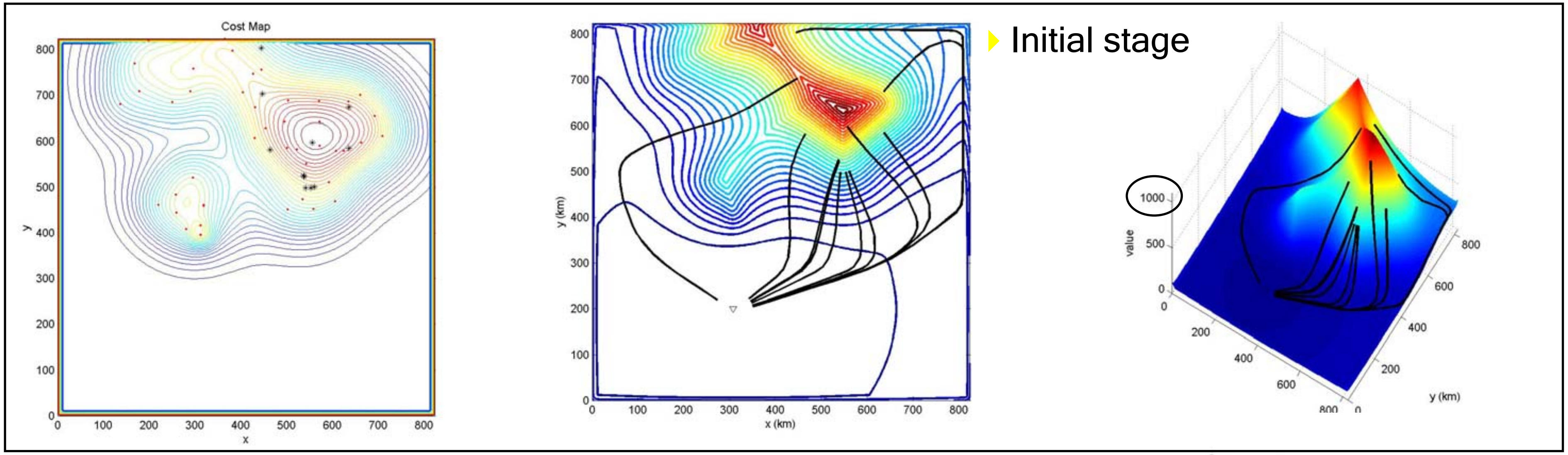
Theorem: Sequence minimizes maximum threat

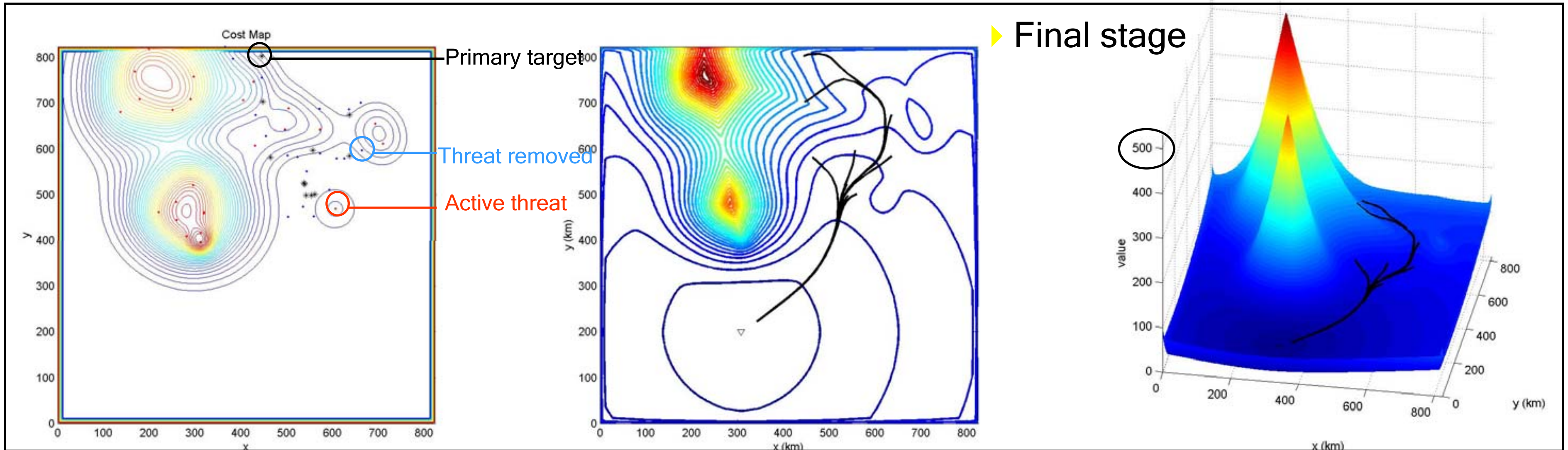
Operator assisted procedure



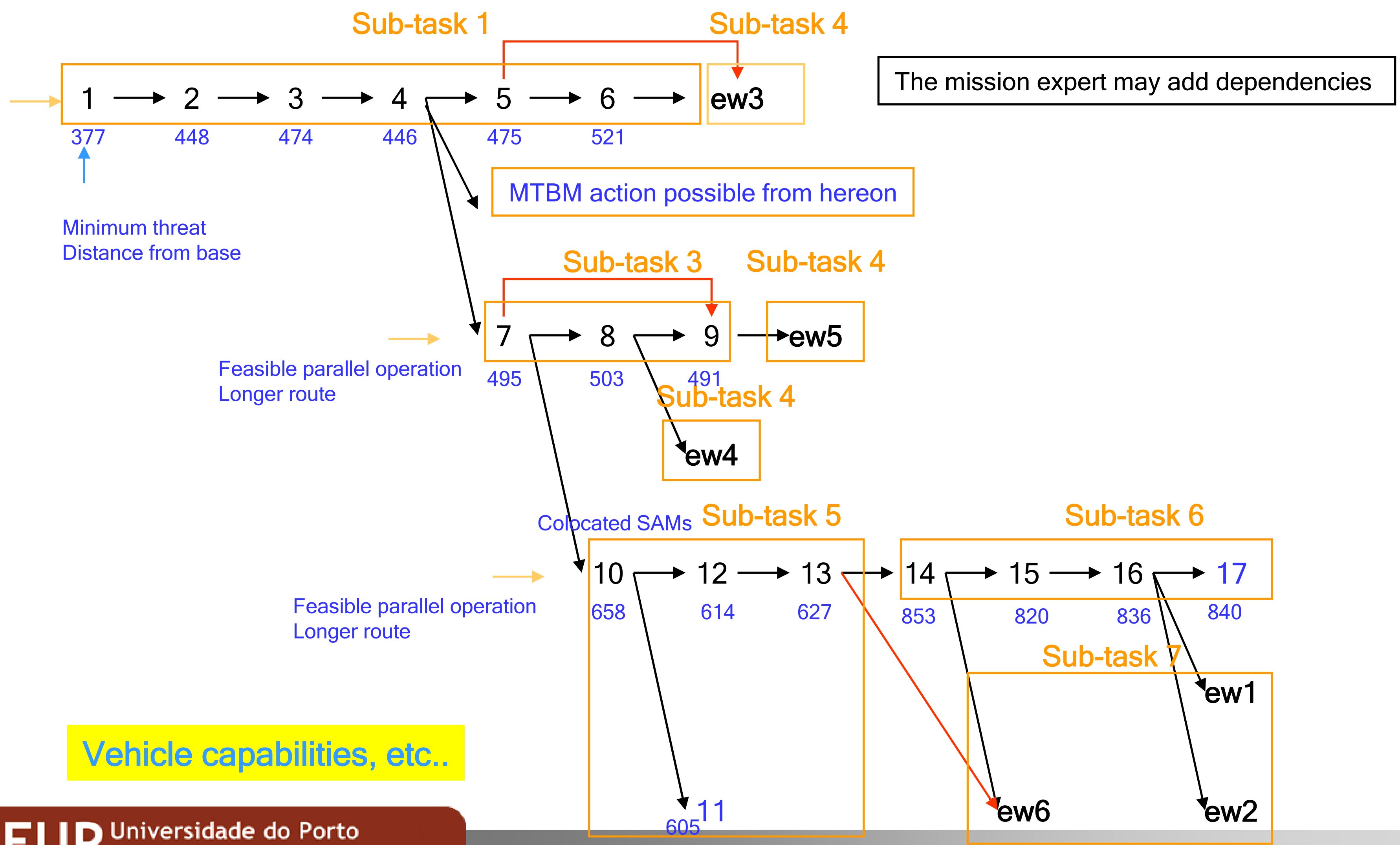
Initial stage versus final stage

$$i = N$$

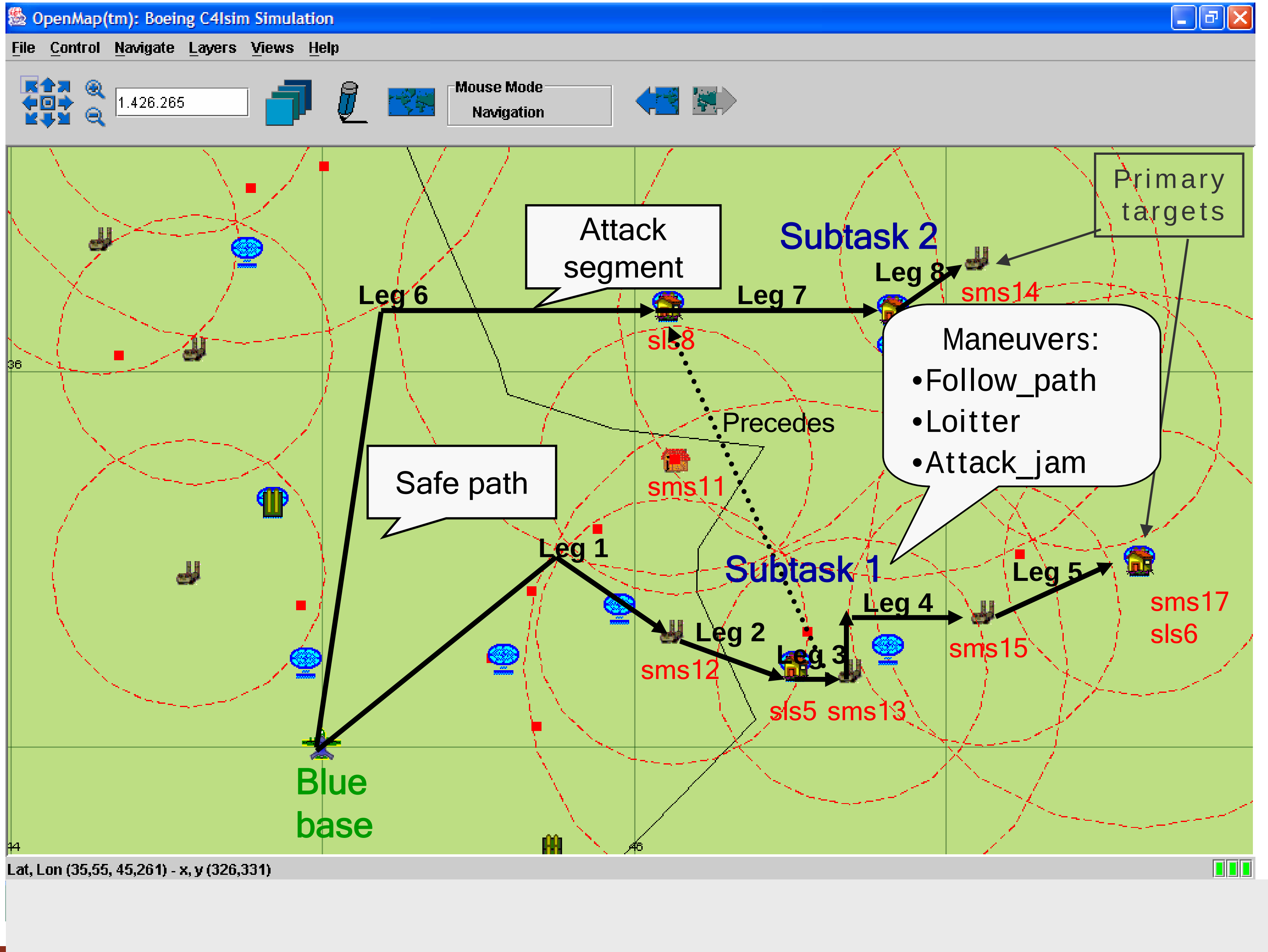




Plan: tasks + subtasks

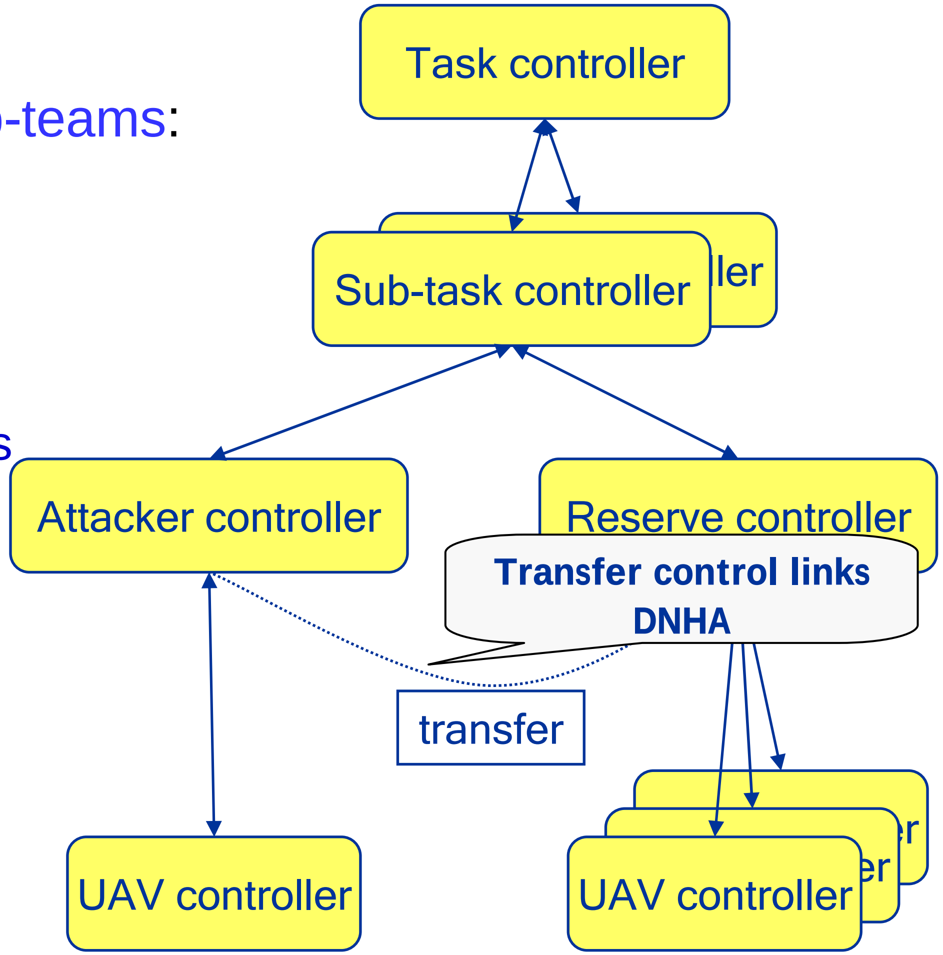


Task specification example



Execution strategy and controllers

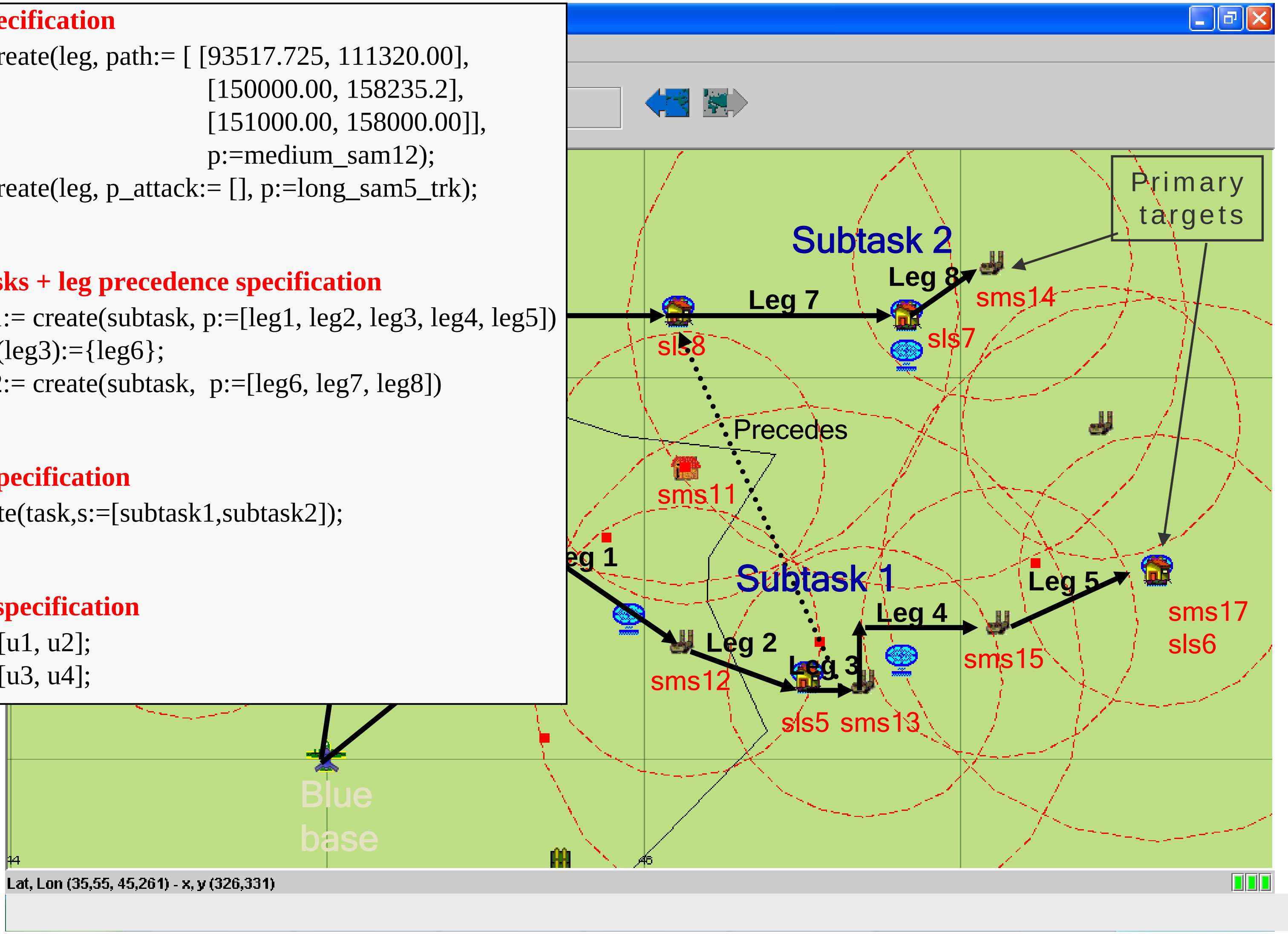
- Task = n sub-tasks + precedence constraints
- One team per sub-task
- Each team is organized as two sub-teams:
  - Attacker
  - Reserve
- Attacker
  - Opens corridor
  - Satisfy task precedence constraints
  - Keep executing legs until
    - Sub-task terminates
    - Eliminated or out of assets
- Reserve
  - Advance while is safe
  - Replace attackers



DARPA-MICA

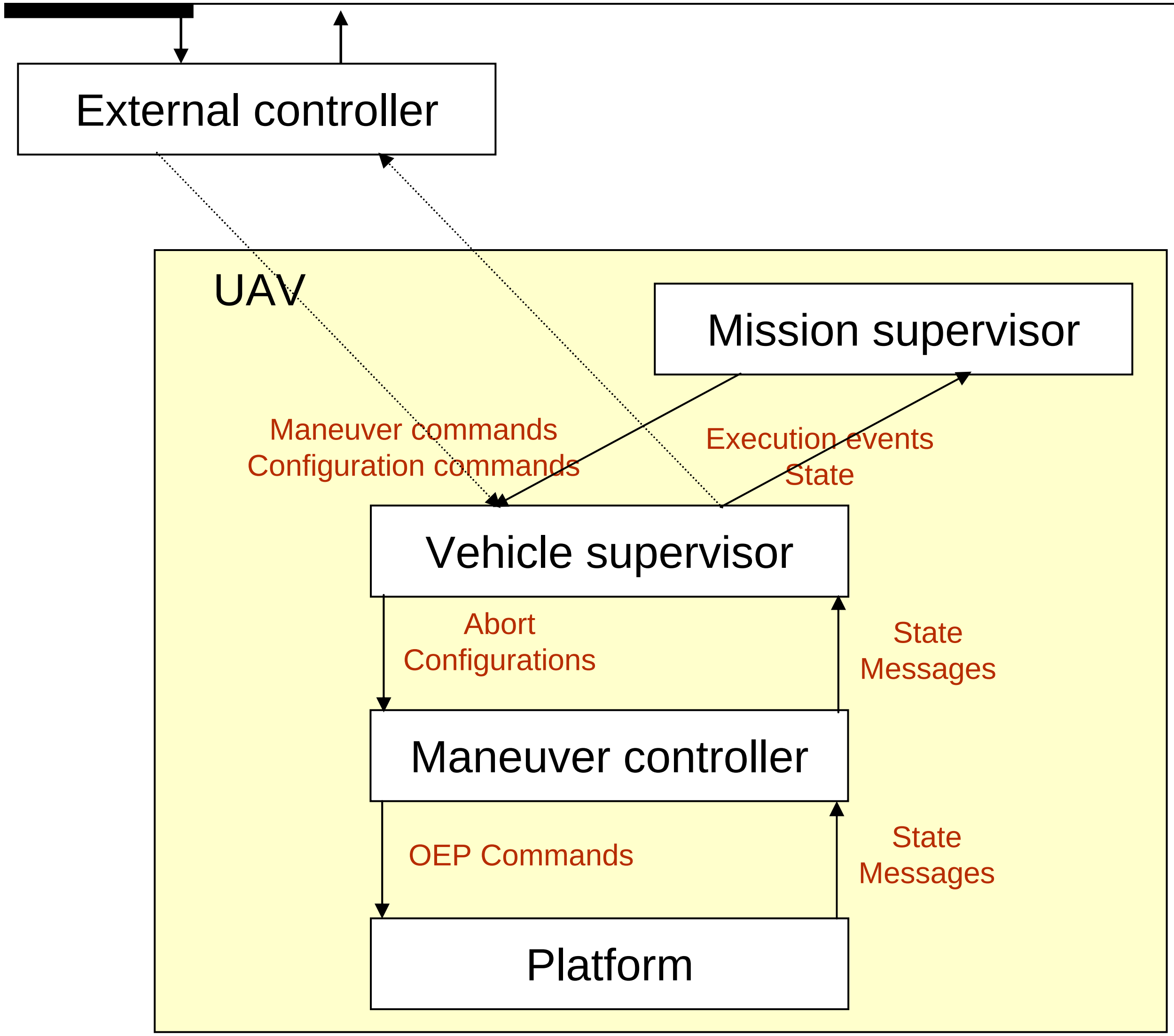
Task specification

```
// leg specification
leg1:= create(leg, path:= [ [93517.725, 111320.00],
                            [150000.00, 158235.2],
                            [151000.00, 158000.00]],
                            p:=medium_sam12);
leg2:= create(leg, p_attack:= [], p:=long_sam5_trk);
    .....

// subtasks + leg precedence specification
subtask1:= create(subtask, p:=[leg1, leg2, leg3, leg4, leg5])
requires(leg3):={leg6};
subtask2:= create(subtask,  p:=[leg6, leg7, leg8])

// task specification
t1:=create(task,s:=[subtask1,subtask2]);

// team specification
team1:=[u1, u2];
team2:=[u3, u4];
```



UAV controllers

DARPA-MICA



Control concepts

Team/task

Individual Mission

Follow_path
Loitter
Attack_jam

Vehicle maneuvers

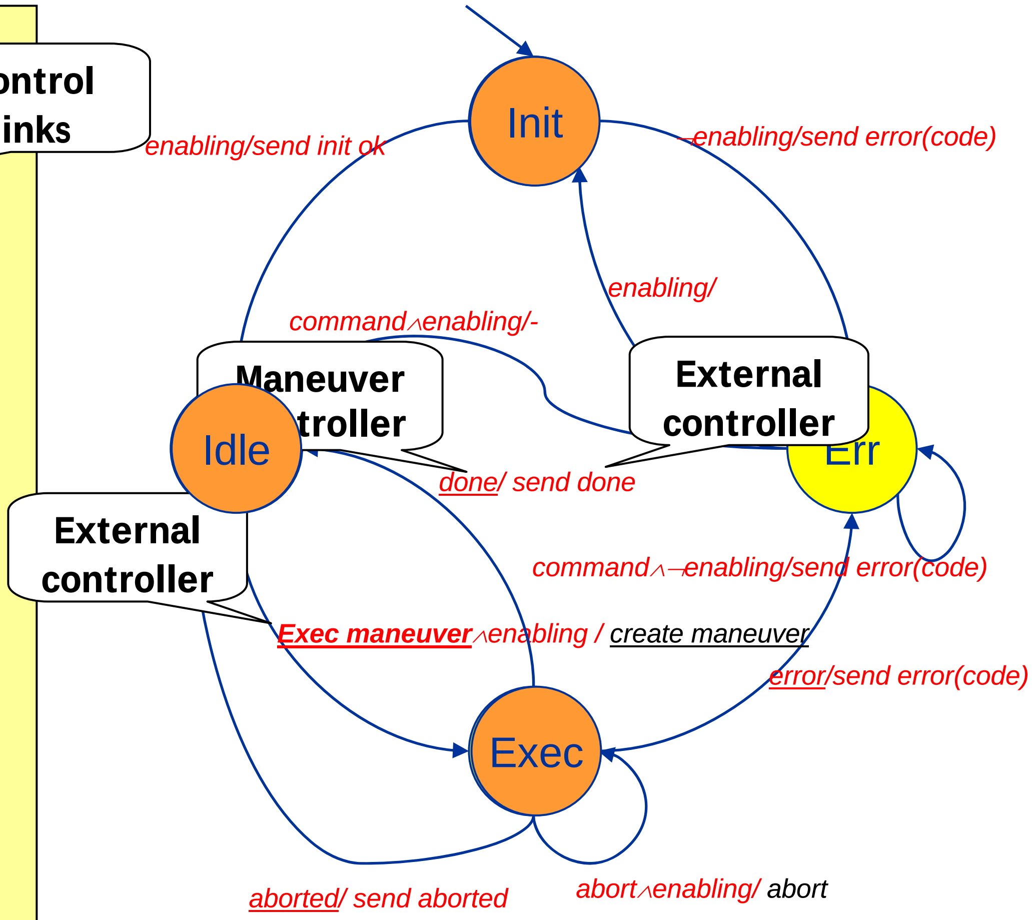
OEP commands

DARPA-MICA

Vehicle supervisor

```
type supervisor  {
  input
    sub_task_controller stc
    mission supervisor  ms;
    set(operator)       op;
    set(UAV)            peers;
    ...
  output
    array[services]     aserv;
    set(maneuver)       am;

  state
    maneuver_controller mc;
    internal_systems    is;
    state_vehicle       vs;
    array(service)      es;
   ...
  discrete // discrete modes
    Init, Exec, Err, Idle;
   ...
}
```

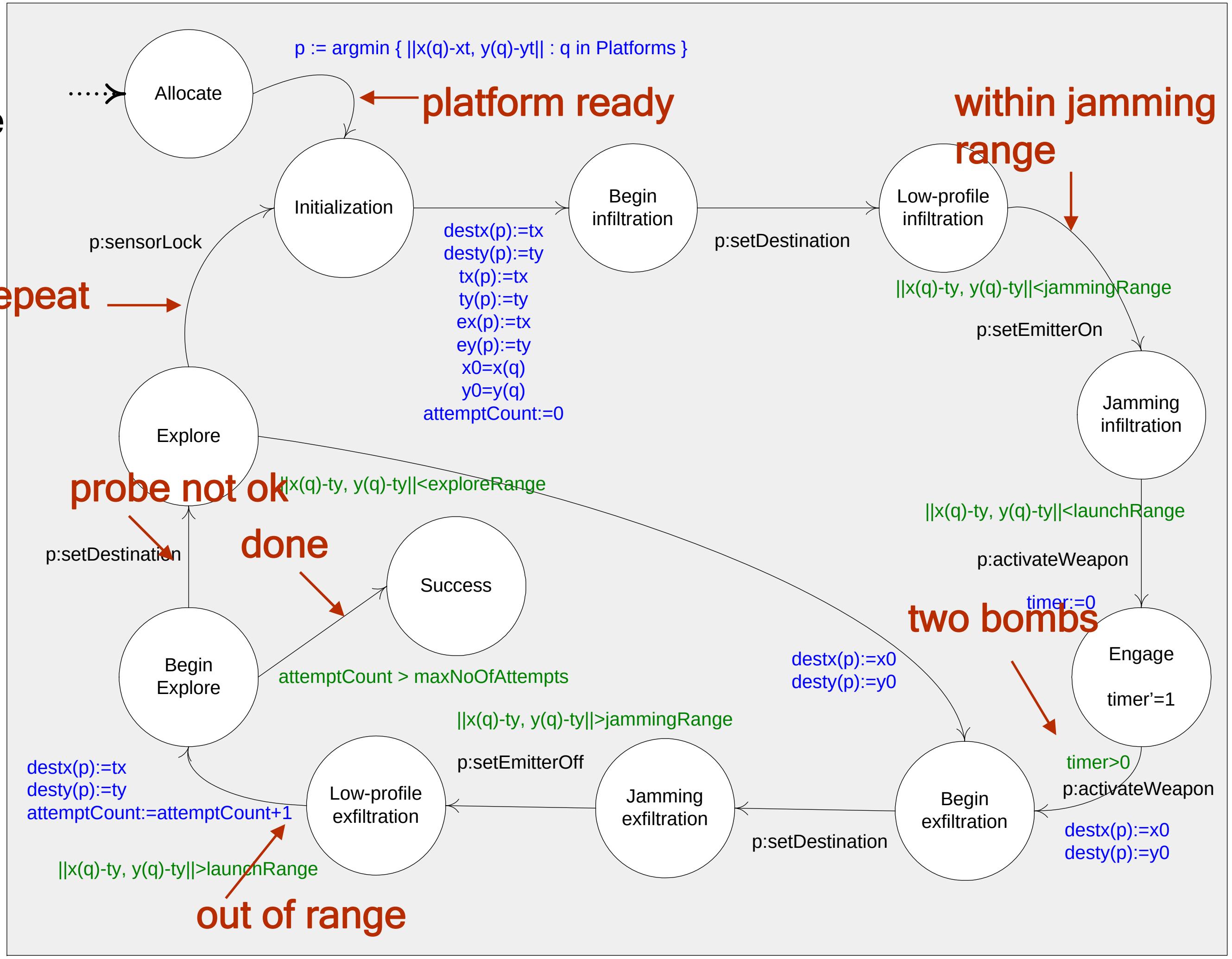
Control links



DARPA-MICA

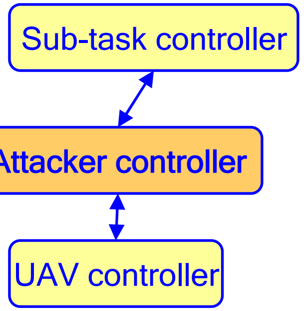
Attack_jam maneuver controller

- Platforms
  - Small combo type
- Controls
  - Velocity
  - Weapons release
  - Jammer
- Guards
  - Time
  - Ranges
    - Weapons
    - Jammer
    - SAMs
  - Locks



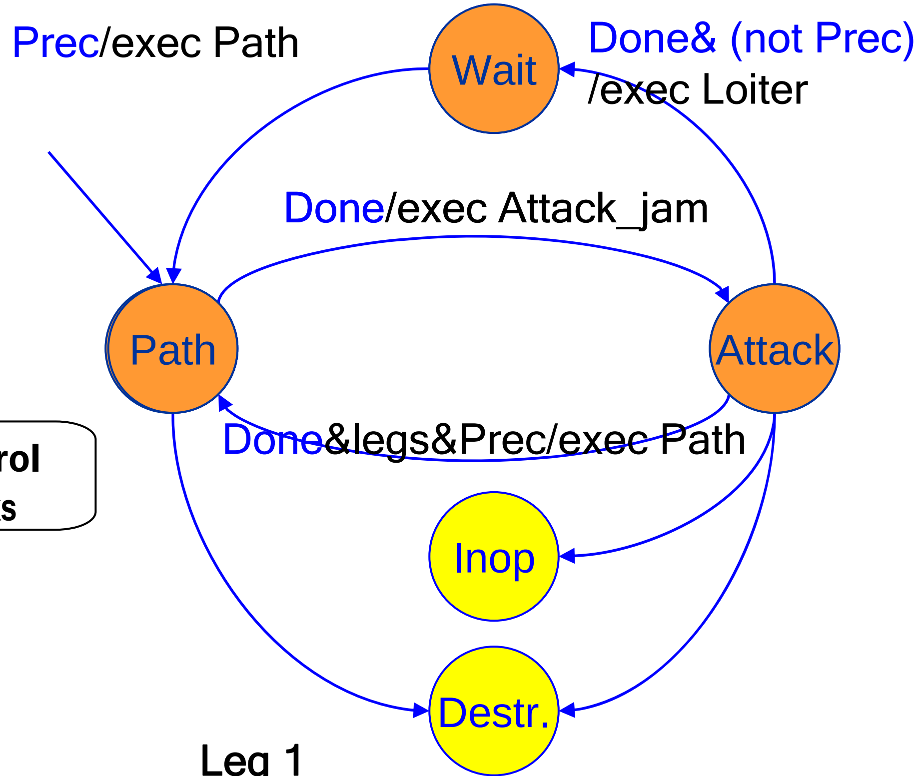
DARPA-MICA

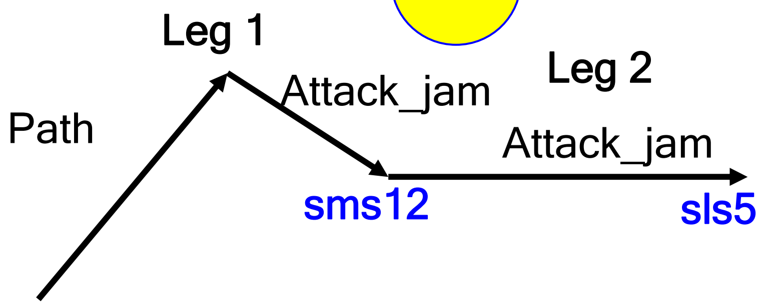
Attacker controller



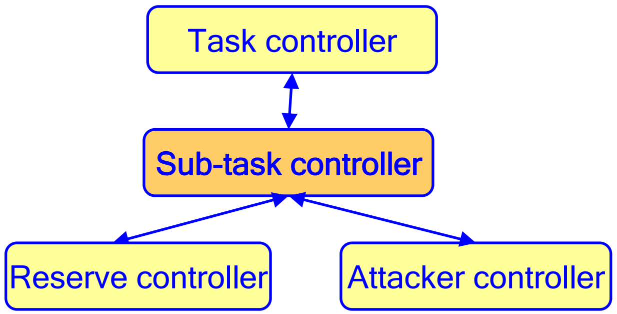

```
type attacker_cont  {
  input
    sub_task_cont         stc;
    set(operator)         op;
    supervisor            s;
    subtask               st;

  state
     leg     current_leg;
     leg     precedence;
   ...
  discrete // discrete modes
    Path, Attack,
    Wait, Destroyed, Inop;
     ...
}
```

Control links

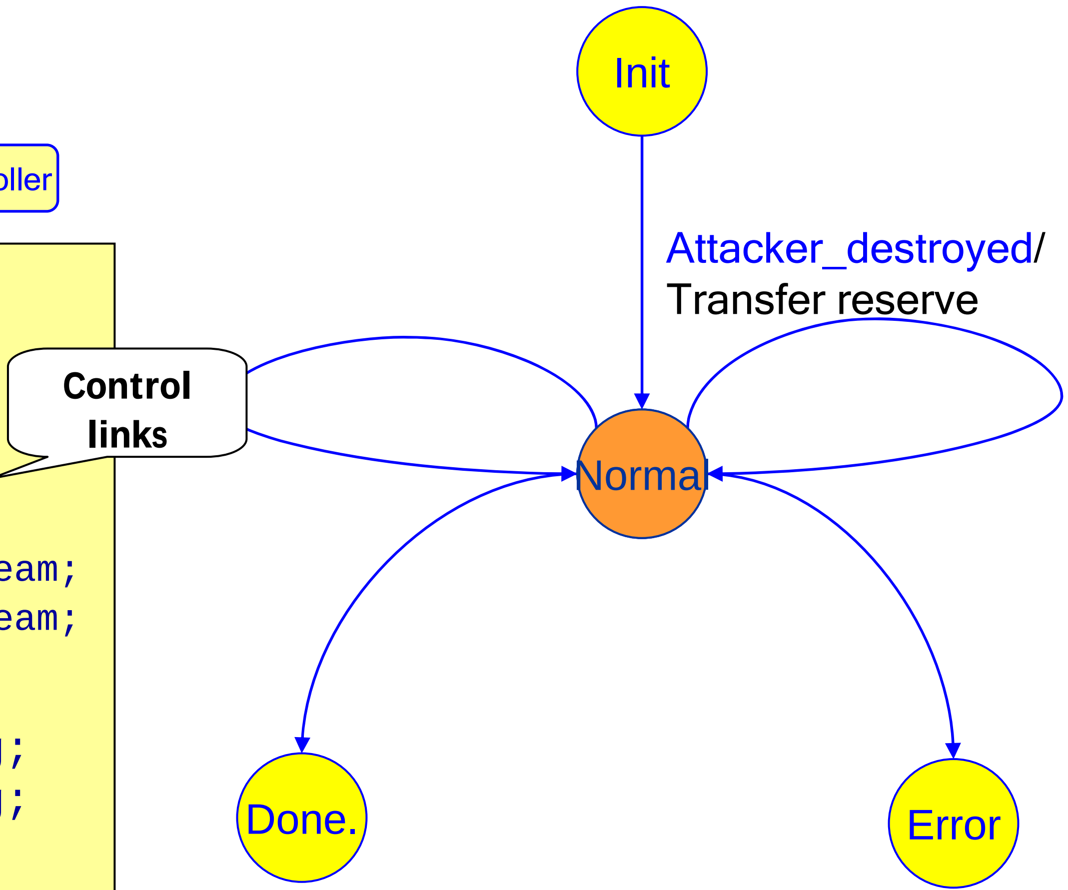




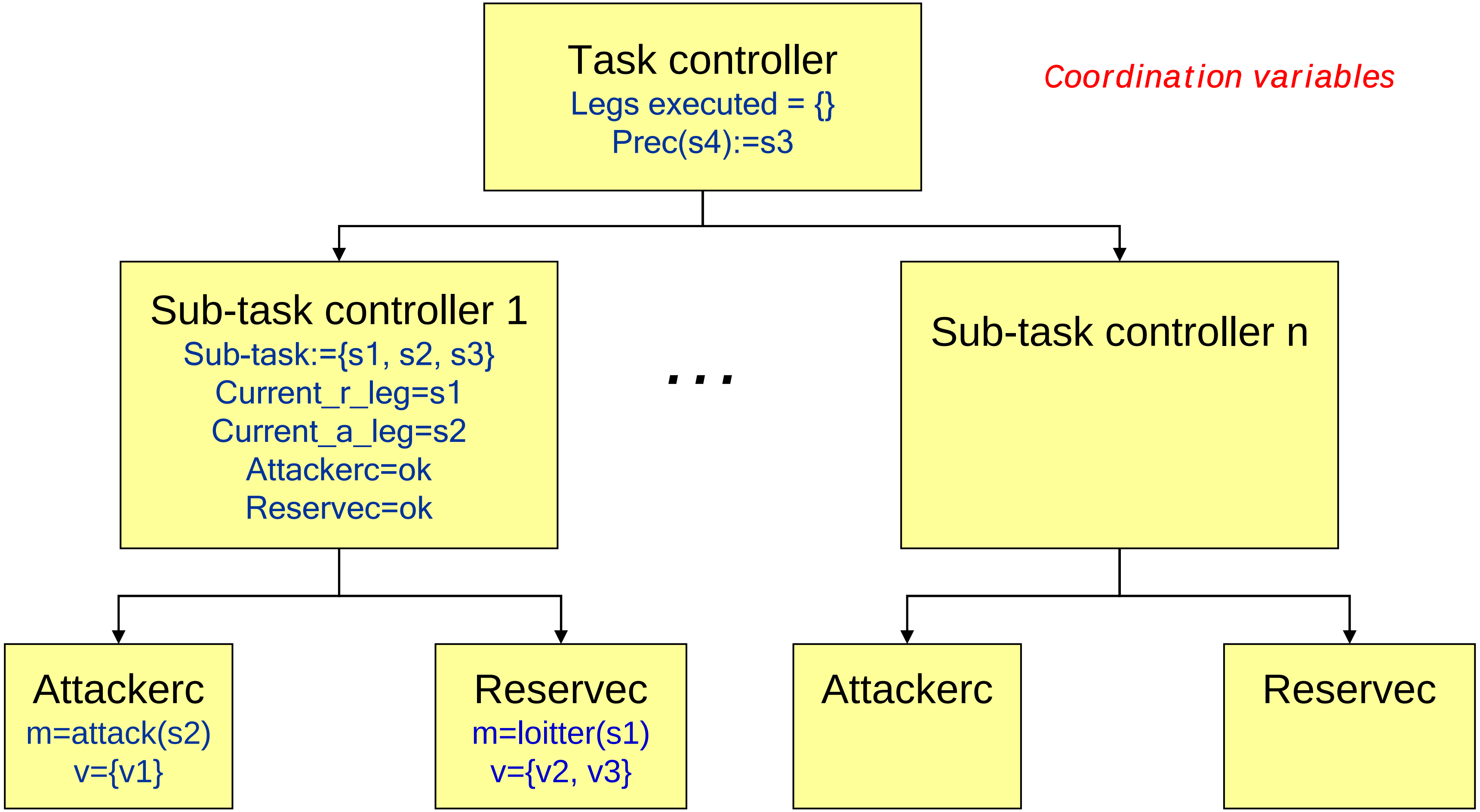
Sub-task controller



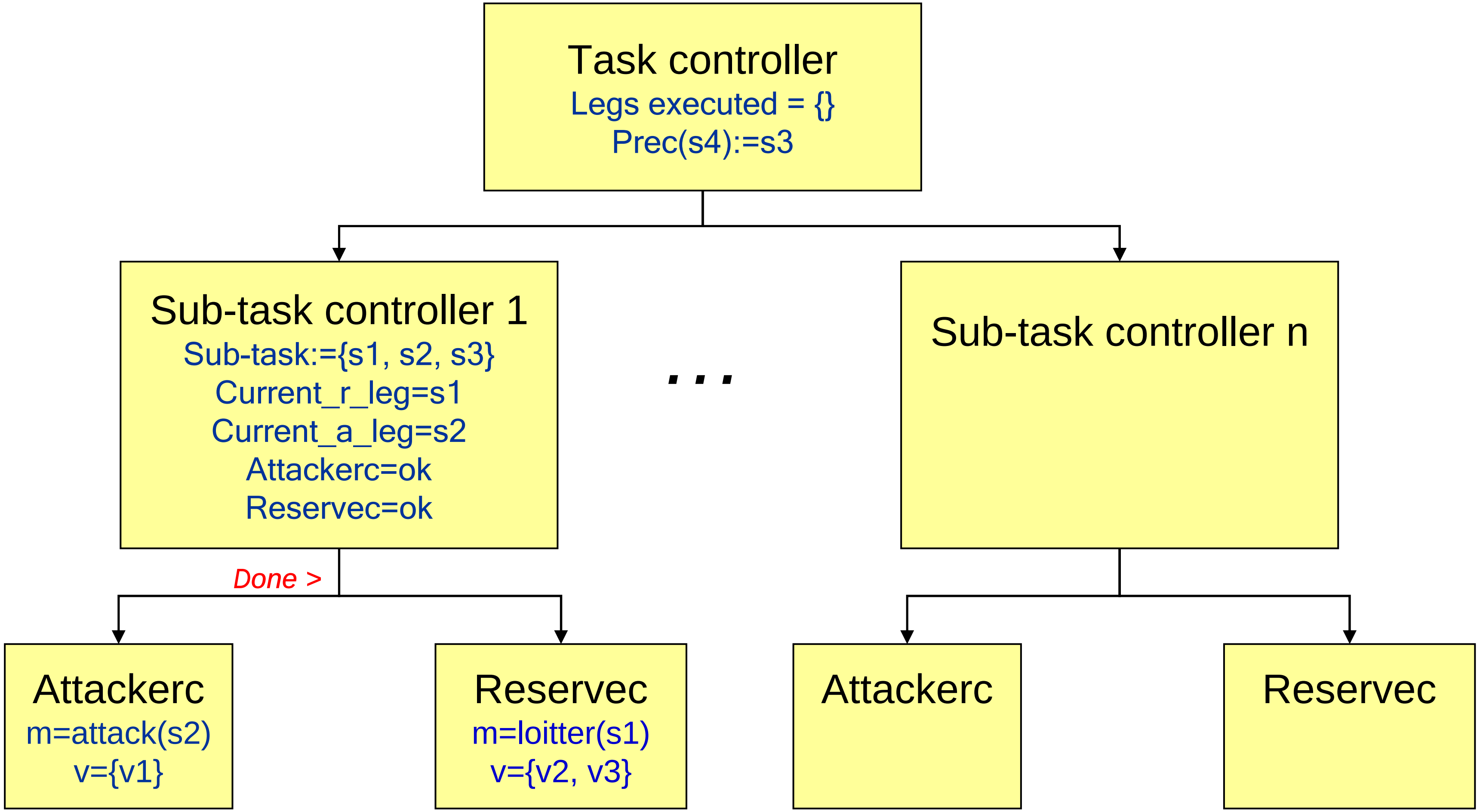

```
type sub_task_cont  {
  input
    task_cont           tc;
    attacker_cont       a;
    reserve_cont        r;
    subtask             st;
    set(UAV)            r_team;
    set(UAV)            a_team;

 state
     leg    current_r_leg;
     leg    current_a_leg;
 ...

 discrete // discrete modes
    Init, Normal, Done, Error
```



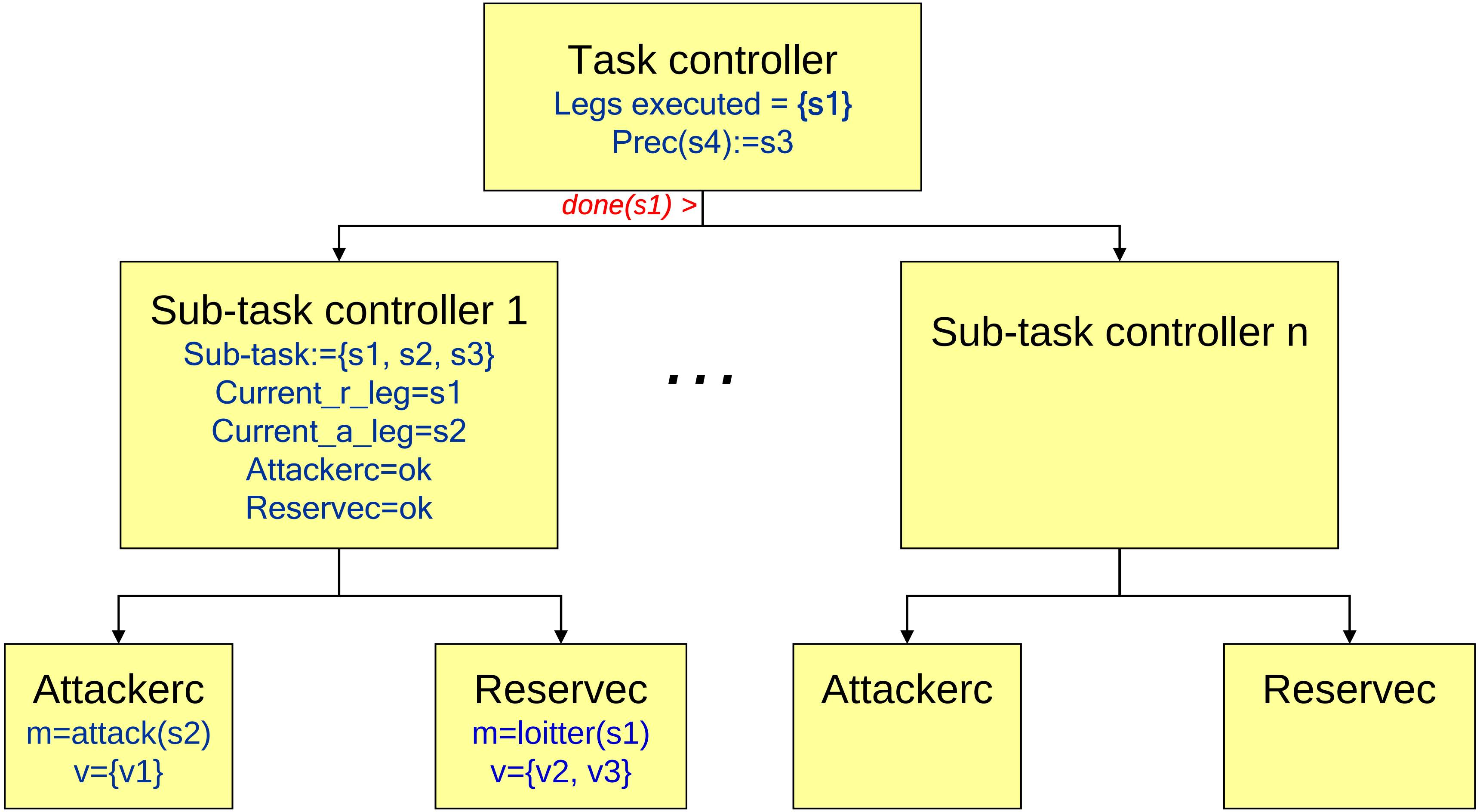
High level coordination



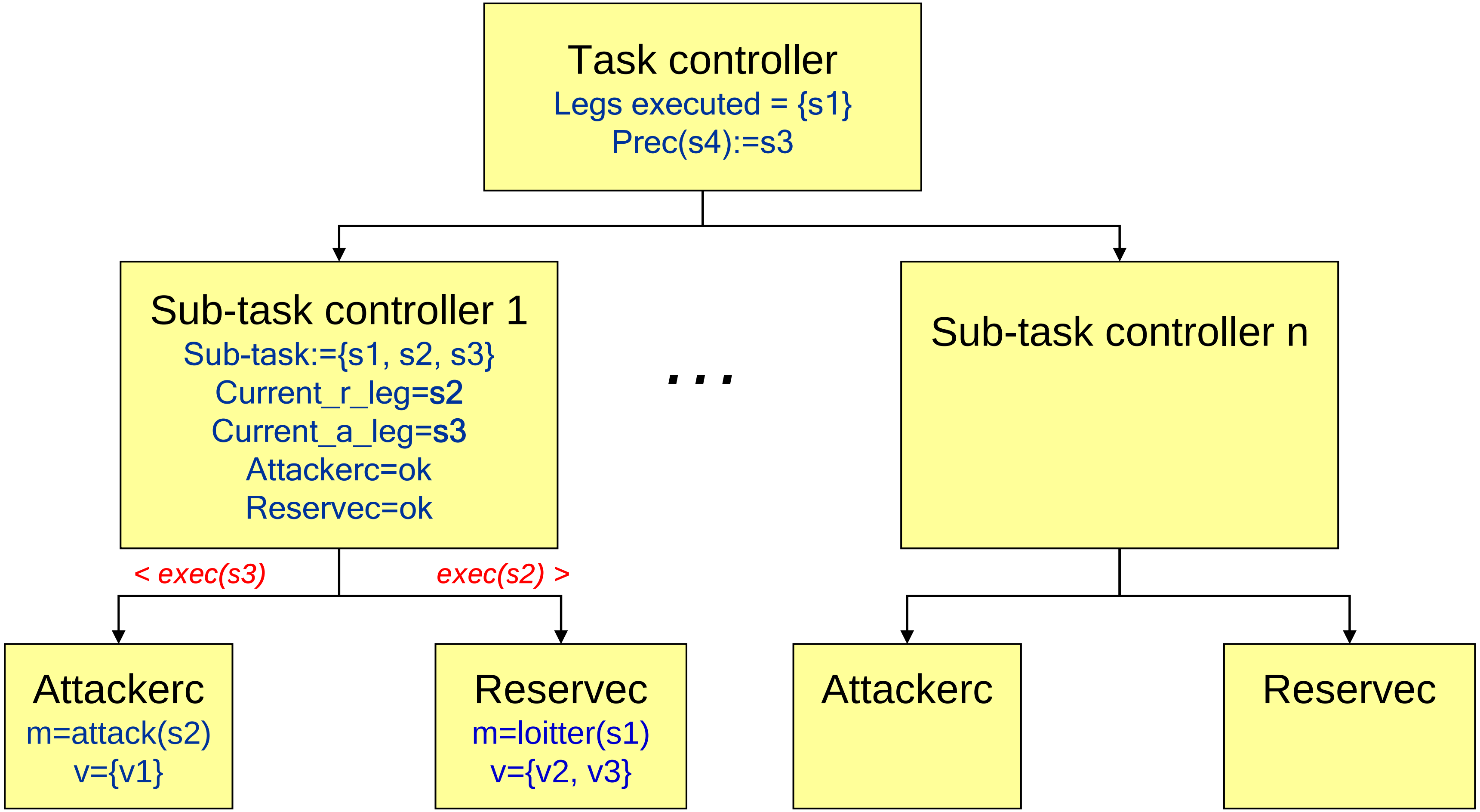
High level coordination



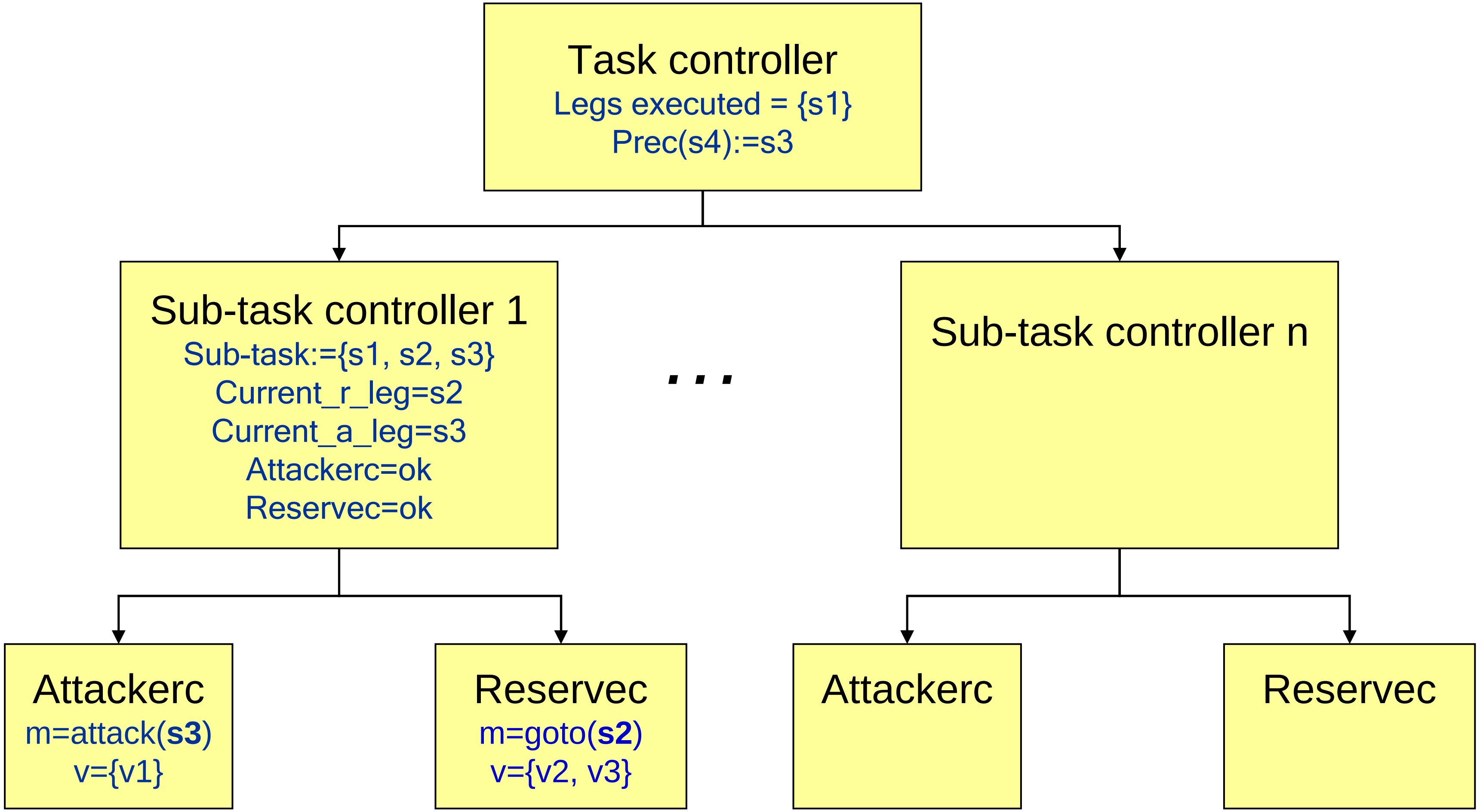
High level coordination



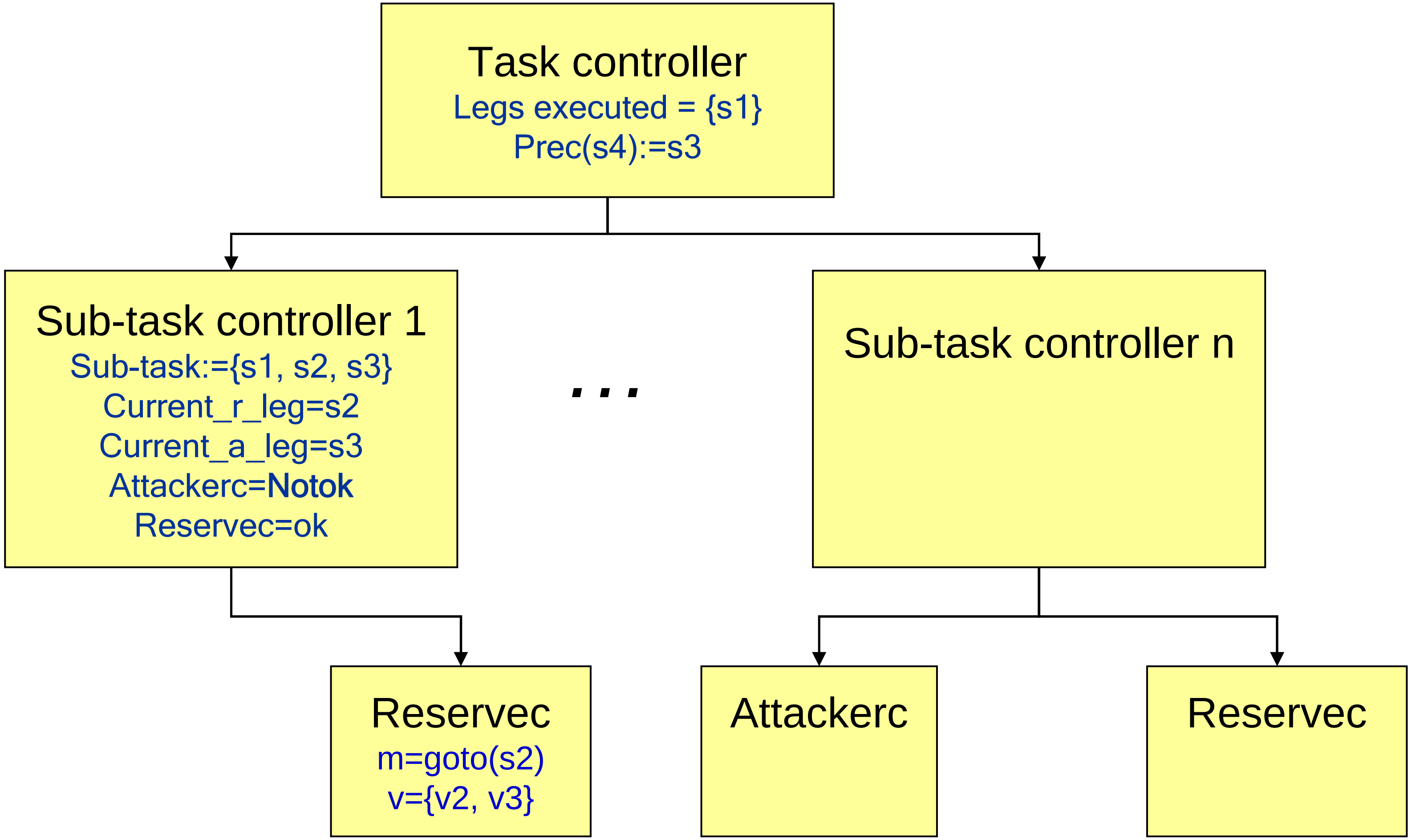
High level coordination



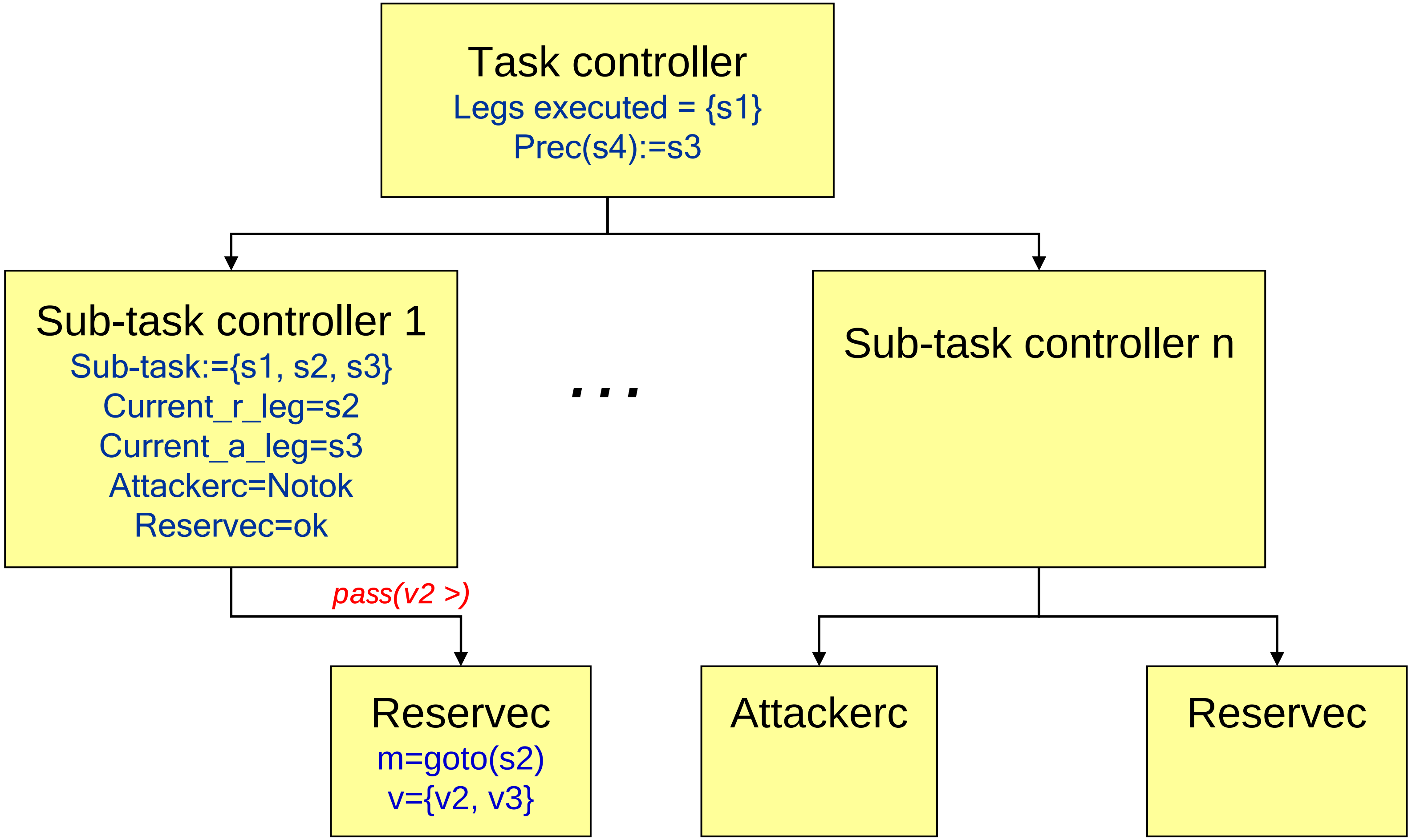
High level coordination



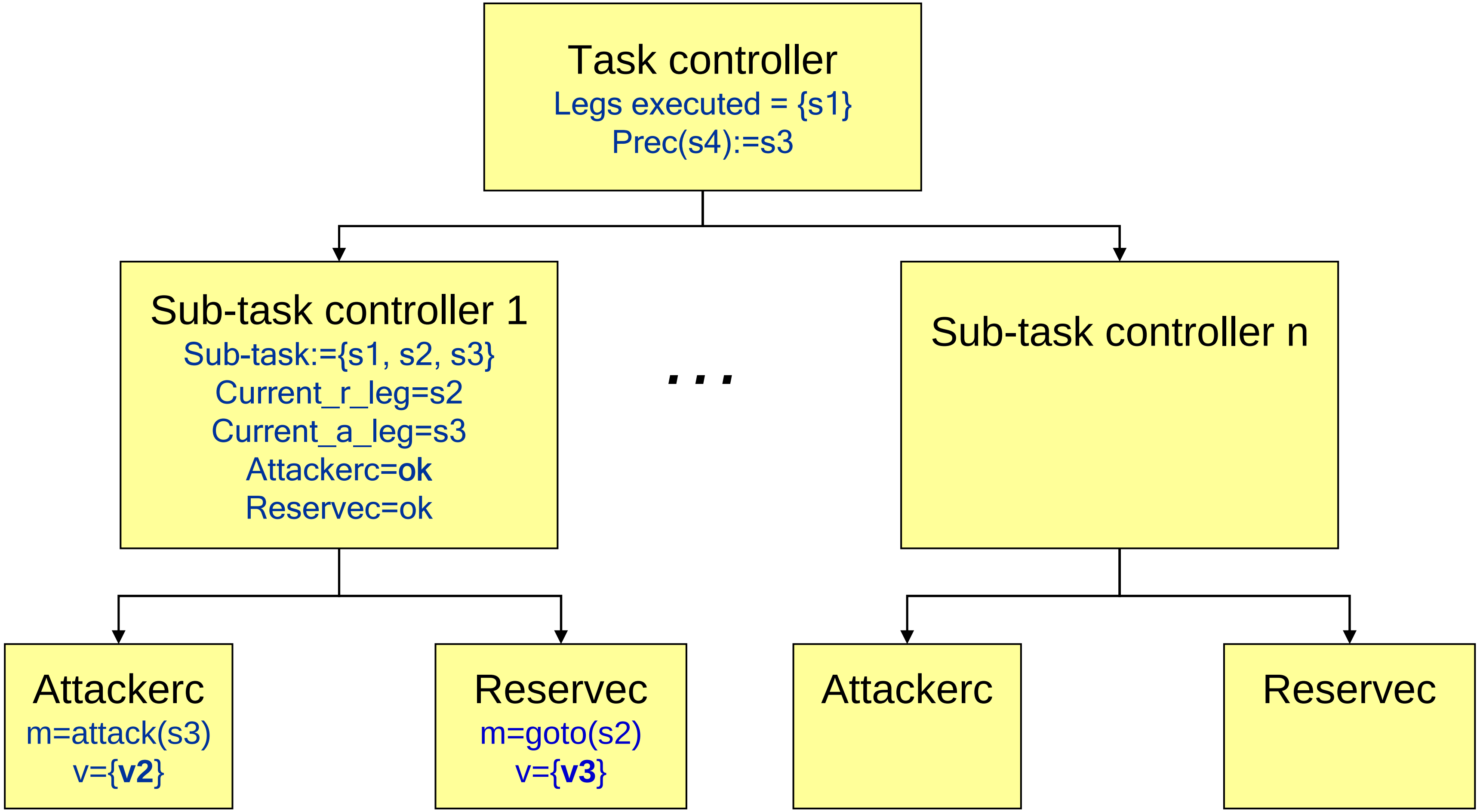
High level coordination



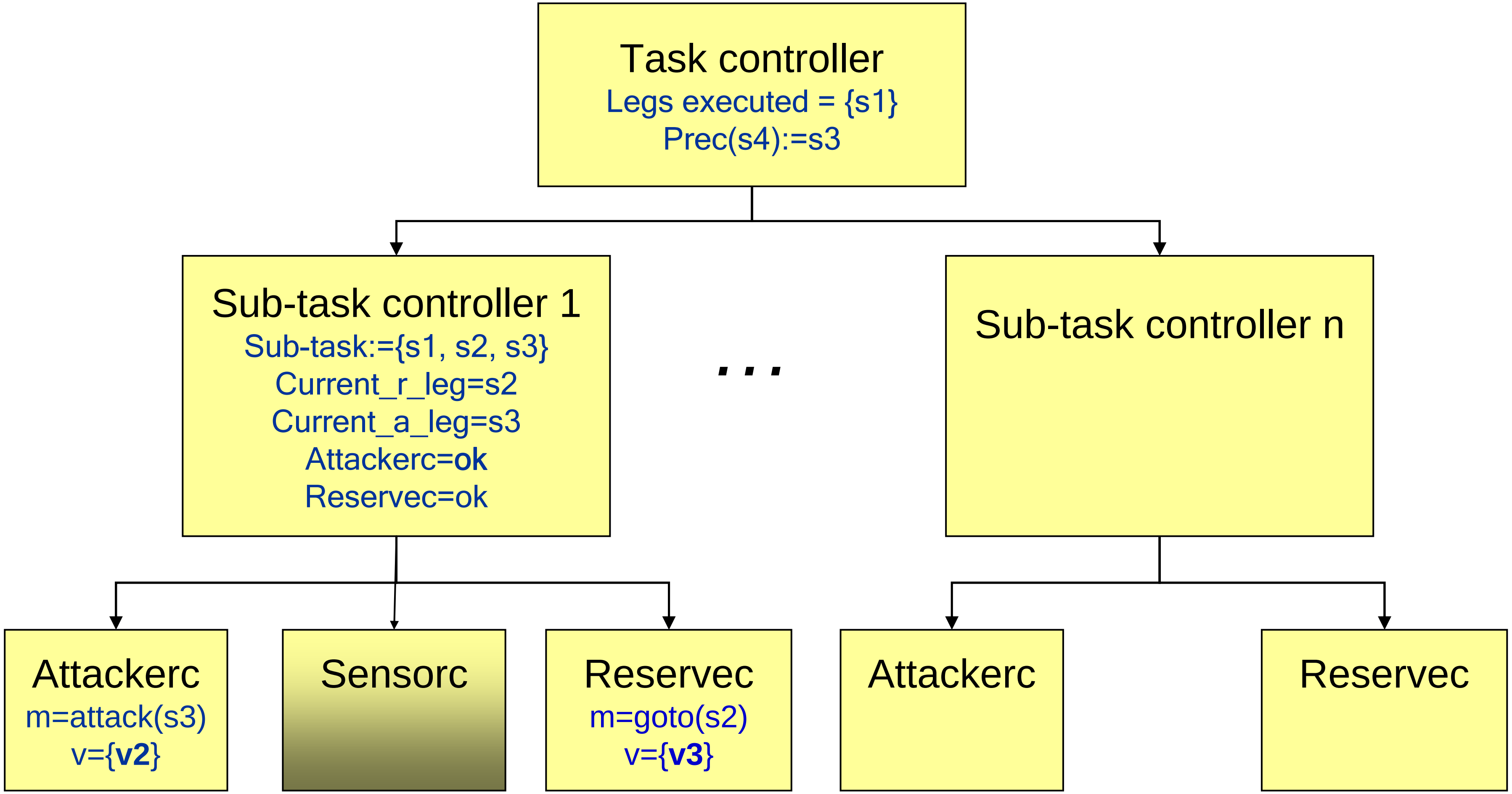
High level coordination



High level coordination



Adding sensor platforms for BDA?



Structure of controllers does not change

A verified hierarchical control architecture for coordinated multi-vehicle operations

João Tasso Borges Sousa*, Karl Henrik Johansson**

Jorge Estrela da Silva*** Alberto Speranzon**,

* Faculdade de Engenharia da Universidade do Porto - Portugal

** Royal Institute of Technology - Sweden

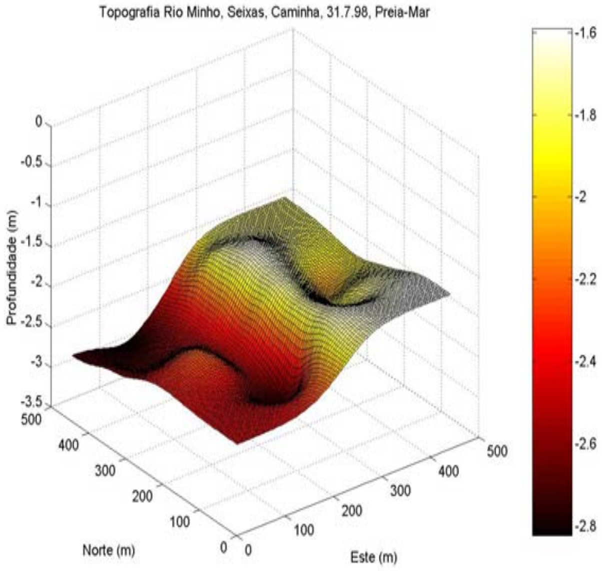
*** Instituto Superior de Engenharia do Porto - Portugal

Multi-vehicle search problem

- Vehicles v_i
  - $V = \{v_1, ..., v_n\}$
  - Each vehicle v_i
    - Limited communication capabilities: bandwidth and range
    - Sensor for local measurements
    - Onboard computer for coordination and control

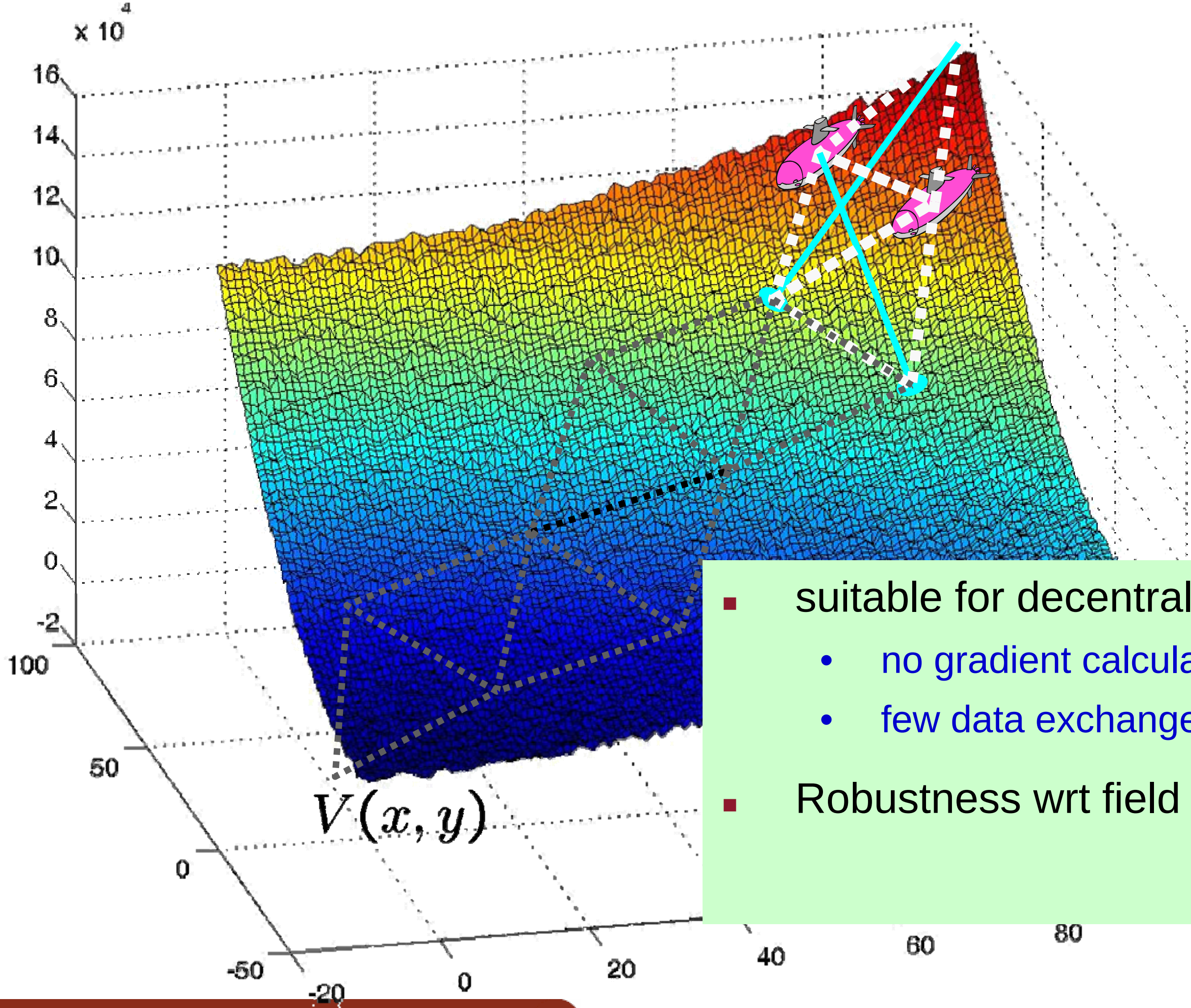
- Scalar field
  - $v = f(x,y,z,t)$

- Search algorithm
  - Repeat until termination
    - Calculate next sampling points
    - Go to sampling points



João Borges de Sousa, Karl H. Johansson, Jorge Silva and Alberto Speranzon, "A verified hierarchical control architecture for co-ordinated multi-vehicle operations", Int. J. Adapt. Control Signal Process. 2006.

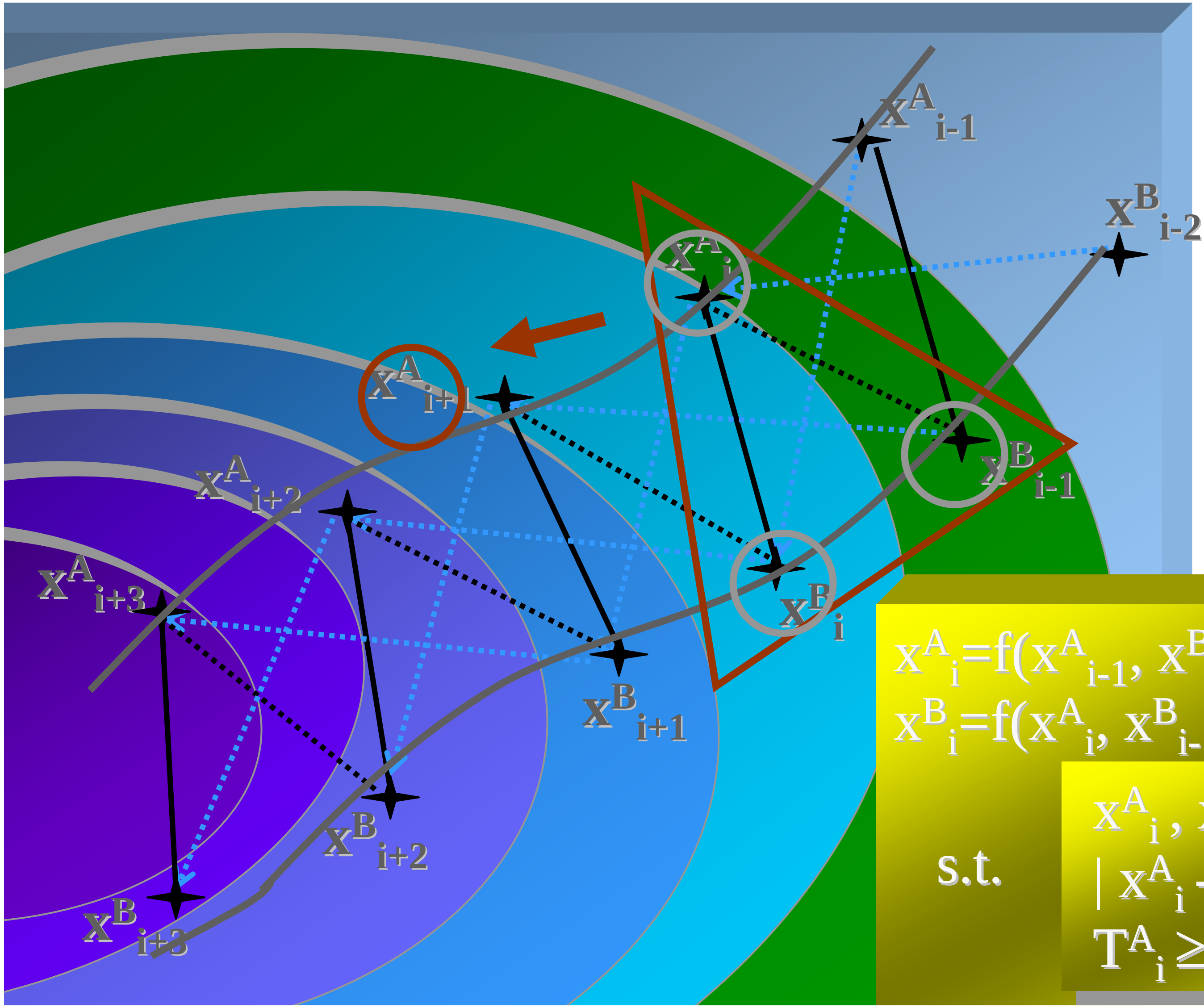
Simplex optimization algorithm



- suitable for decentralization
  - no gradient calculation
  - few data exchanges
- Robustness wrt field noise

Coordinated reachability

Communication constraints



$\{(x^A_i, T^A_i), (x^B_i, T^b_i): i=1,2,\ldots\}$
sequence of measurements
for vehicles A and B

$x^A_i = f(x^A_{i-1}, x^B_{i-1}, x^B_{i-2})$

$x^B_i = f(x^A_i, x^B_{i-1}, x^A_{i-1})$

s.t.

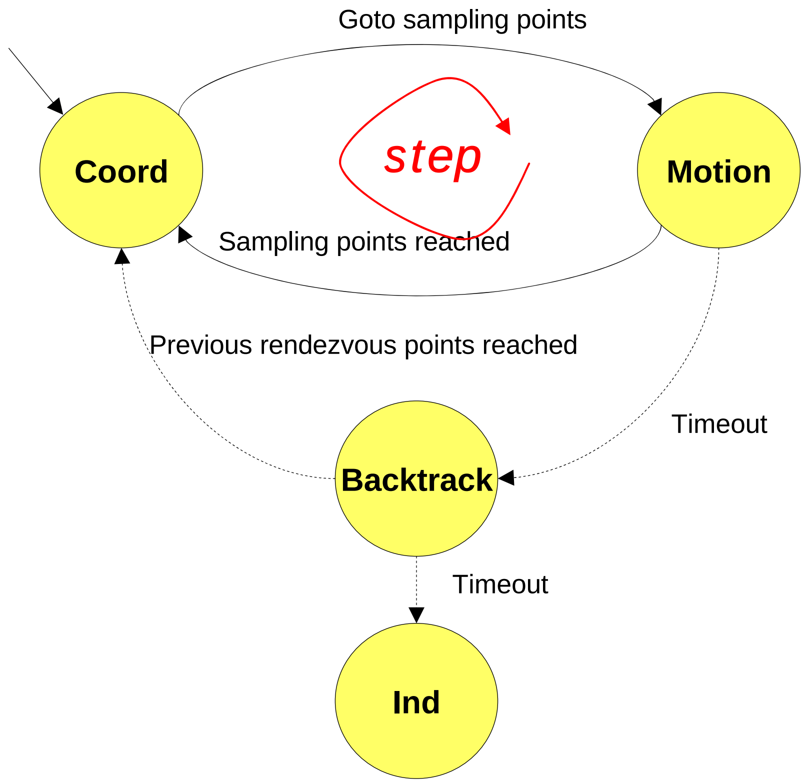
x^A_i, x^B_i are reachable

$| x^A_i - x^B_i | = \text{const}$

$T^A_i \geq T^A_{i-1}, T^B_i \geq T^B_{i-1}$

Problem

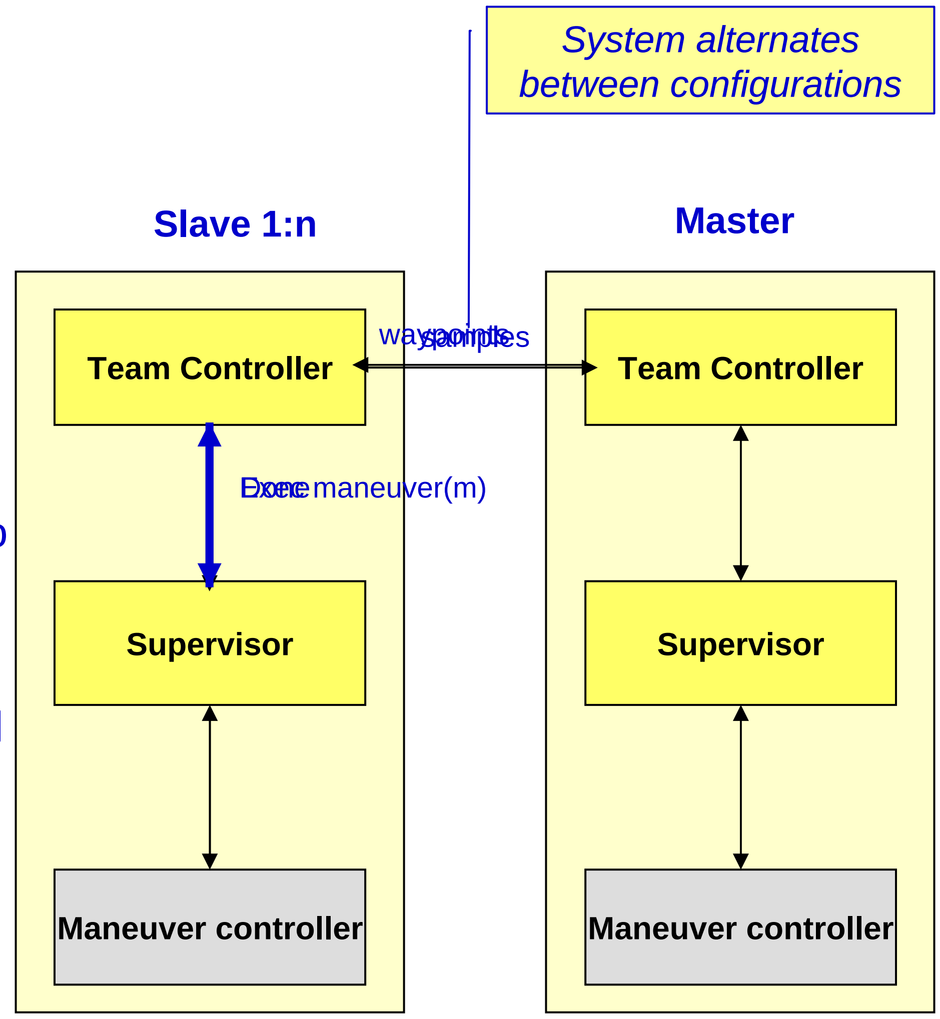
- Given
  - Set of initial locations I
  - Measurement function m
  - Way-point generation function g
  - Termination criteria c
  - Specification S



- Find
  - control architecture = controllers + connections such that
    - Σ = V + control architecture satisfies the specification (simulation relation)

Control architecture

- Team controllers
  - One master; n slaves
  - Run the coordination algorithm
  - Handle structural adaptation and reconfiguration
- Vehicle supervisors
  - Interface with external controllers
  - Makes decisions on what maneuver to execute
- Maneuver controllers
  - Implement elemental feedback control maneuvers for each AUV
  - One active at a time
  - Goto(point)
  - Hold(point)



Formal model: dynamic network of hybrid automata

More formally…

- Formal model for team controllers
  - $T = T_M || T_{S1} || \ldots || T_{SN}$

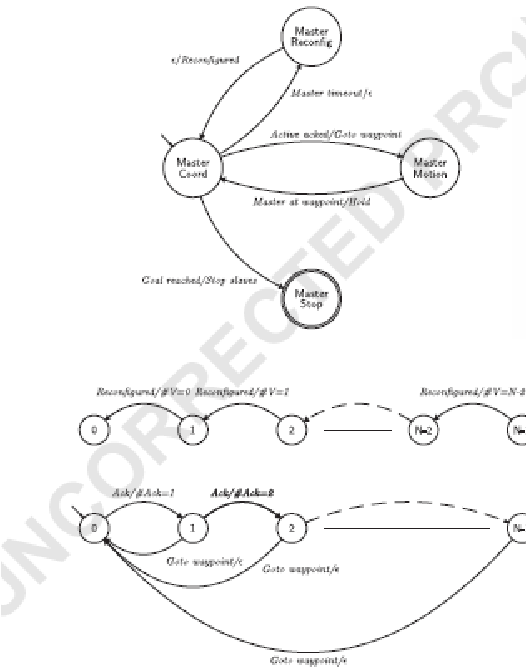


Figure 4. The master team controller is the parallel composition of three transition systems.

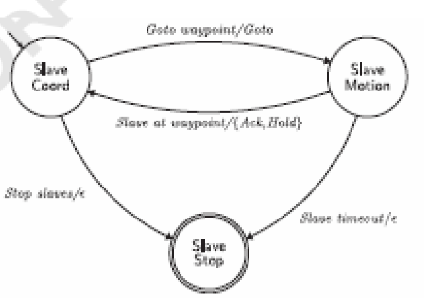
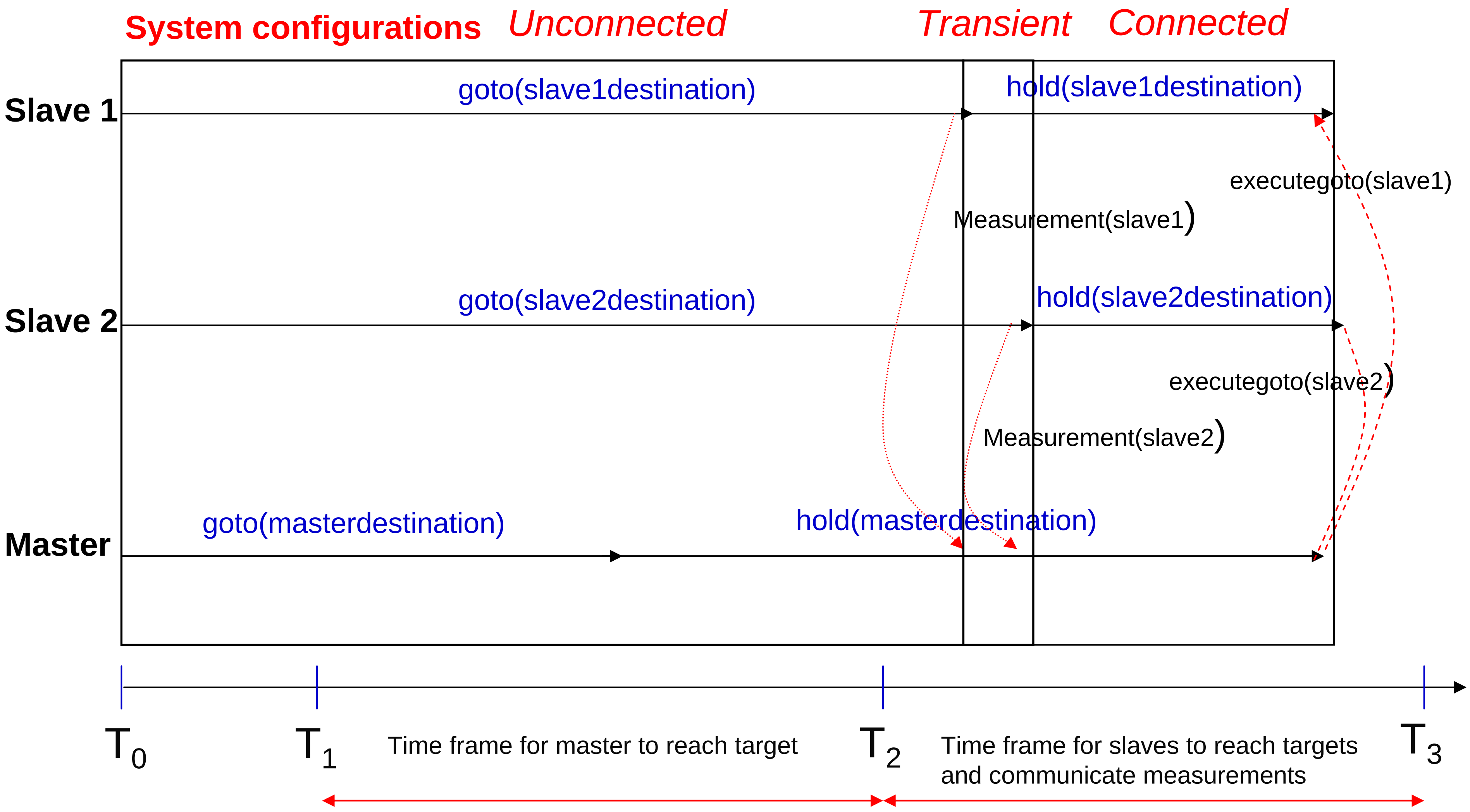


Figure 5. Slave team controller.

- Assumption
  - waypoint generation procedure produces feasible intervals for waypoints

- Theorem: T and S are bi-similar
  - Assumes maneuver controllers produce ensured results
    - Given feasible intervals for waypoints makes sure those are reached within specs
  - Each team controller abstracts the behavior of each vehicle
    - composition with supervisor and maneuver controller

Step execution control logic



Step begins - Vehicles move to their target regions

Vehicle model

```
type AUV  {
 input      /* what we feed to it */
  array(number) tau;   // control settings

 output     /* what we see on the outside */
  supervisor s;       // link to supervisor
  TeamController t;   // link to team controller
  number x,y,z;       // motion state
  array(number) eta; // body fixed velocities
  ...
 flow
  x' = f1(eta,tau,x,y,z,phi,theta,psi);

 transition
  ...
}
```

Event synchronization
Input/output connections

```
v1:= create(AUV, s:=sup1, t:=tc1);
v2:= create(AUV, s:=sup2,t:=tc2);
V := {v1, v2}; // set composed of v1 and v2
```

Create instances
Create set V of AUVs

Maneuver controller

```
type MController  {
 input
  number x,y,z; // motion state
  mspec m;      // maneuver parameters
  supervisor s; // link to supervisor
  ...
 output
  array(number) u;// control for actuators
  ...
 export
  done; // event generated when done
  ...
 transition
   exec -> stop {done} when ...

}
```

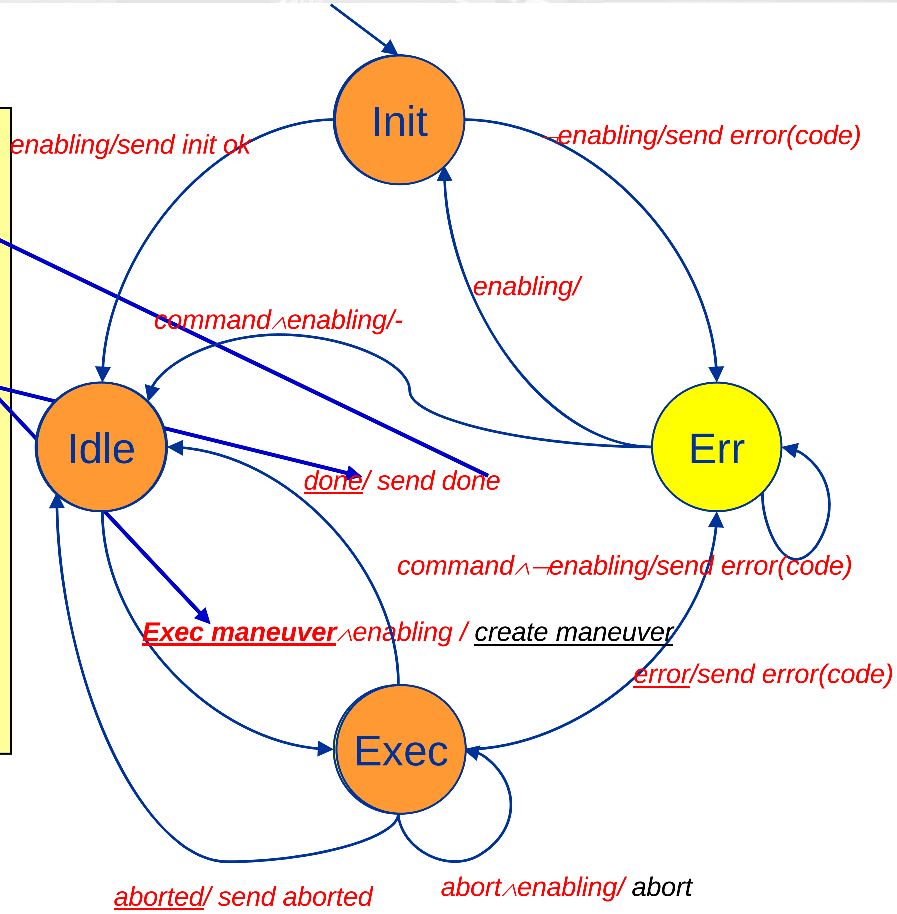
Created on the fly with maneuver parameters *mspec*

Sends event *done* to supervisor when maneuver terminates

- Goto(Location,time interval)
  - Differential game formulation
  - Ensured results
- Hold(location, duration)

Vehicle supervisor

```
type supervisor  {
 input       /* what is fed to it */
  TeamController tc;// link to team controller

 state       /* what is internal */
  MController  mc;// link to maneuver
controller
  mspec         mt;// current maneuver spec
  ...
 discrete   /* discrete modes of behavior */
  Exec, Error, Idle, Err;

 transition
   Idle -> Exec {} ...
}
```



- Abstracts each AUV as a provider of maneuvers

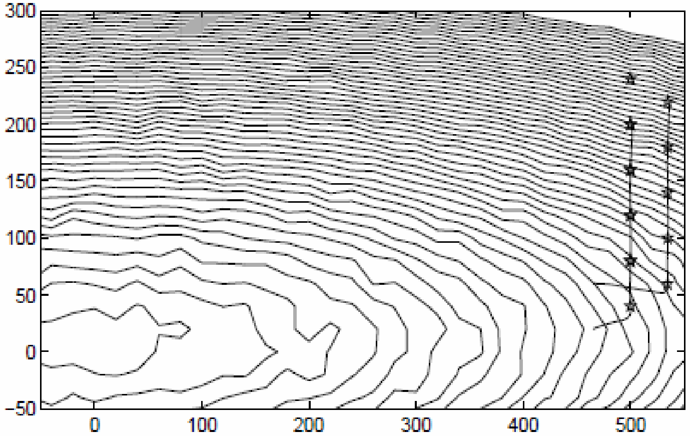
Team controller

```
type TeamController {
 input
  set(AUV) V;            // AUVs in the team
  supervisor s;          // link to its supervisor

 state
  number step;           // last step of algorithm
  number T1, T2, T3;     // coordination times
  number c;              // counts measurements
                         // received
  array(array(number)) L;// visited locations
  number t;              // timer

 output
  TeamController m;// link to master
TeamController
  set(TeamController) tc;// links to
TeamController
  symbol role;      // $master or $slave
  mspec ms;         // maneuver under execution
  set(array(number)) specs; // waypoints

 discrete    /* discrete modes of behavior */
  Init, Error, TMaster, TSlave, SingleN, SingleI;
  ...
}
```

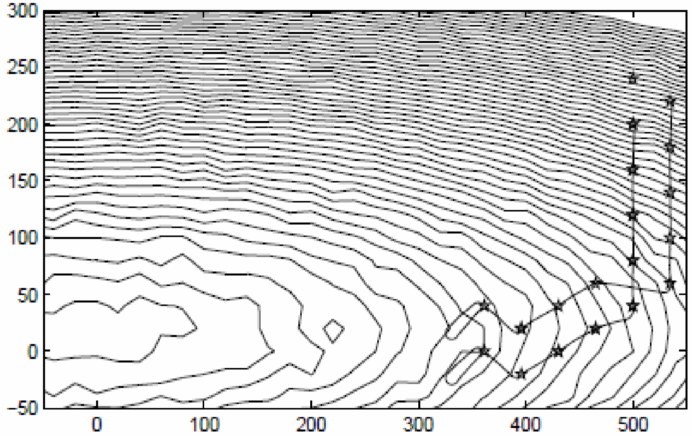
# of steps executed so far
Coordination times
# of measurements received

One Master Team Controller
N-1 Slave Team Controllers

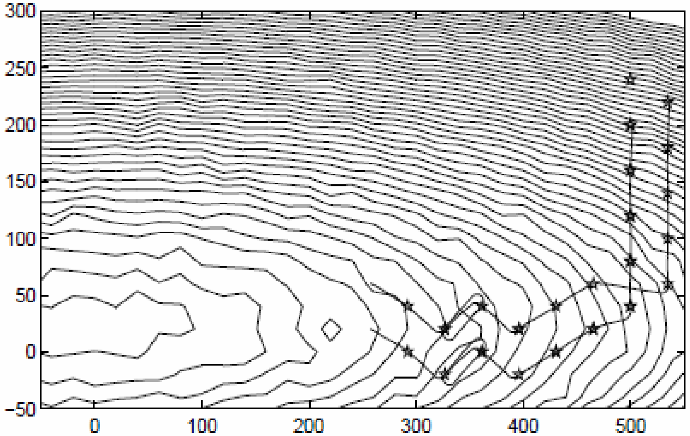
Simulation runs



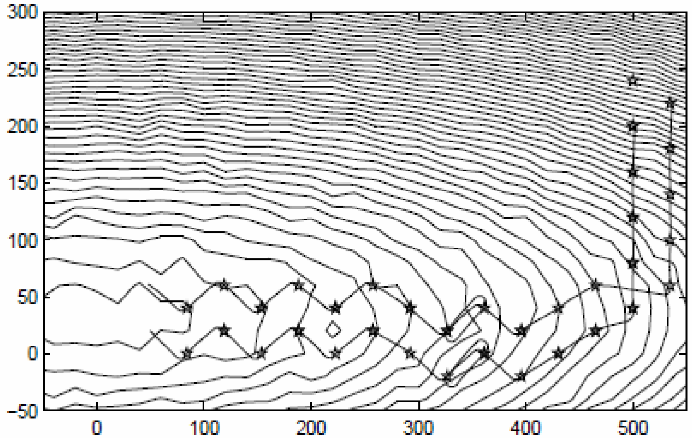
(a) Situation after 6 time steps.



(b) Situation after 9 time steps. Notice the backtrack step.



(c) Situation after 12 time steps.



(d) Situation after 18 time steps.