DARTSim: An Exemplar for Evaluation and Comparison of Self-Adaptation Approaches for Smart Cyber-Physical Systems

Gabriel Moreno, Cody Kinneer*, Ashutosh Pandey*, David Garlan*

*Institute for Software Research, Carnegie Mellon University

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM19-0541

Self-Adaptation for Smart Cyber-Physical Systems

Self-adaptation capabilities are used to allow cyber-physical systems to operate in dynamic and uncertain environments.

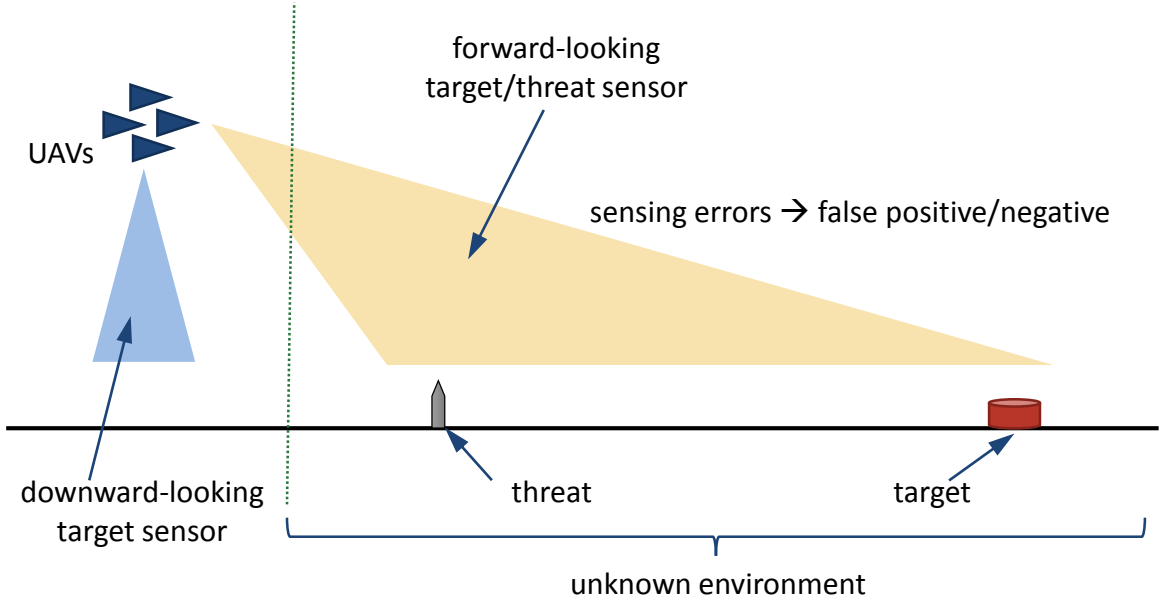
Self-adaptation for sCPS face particular challenges:

- environment monitoring subject to sensing errors
  - e.g., noisy sensors
- adaptation actions that take time
  - e.g., due to physical movement, sensors that recalibrate on power-up
- dire consequences for not adapting in a timely manner
  - e.g., physical damage to system or third party
- incomparable objectives that cannot be conflated into a single utility metric
  - e.g., avoiding an accident vs. providing good service

DARTSim

Simulated autonomous team of unmanned aerial vehicles (UAVs)

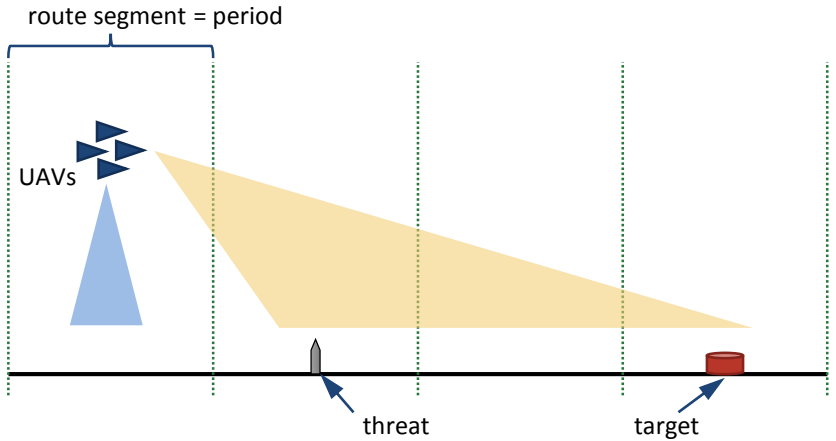
Reconnaissance mission in hostile and unknown environment: *fly predefined route to detect targets on the ground while avoiding threats*



*DART = Distributed Adaptive Real-Time

Adaptation Tactics

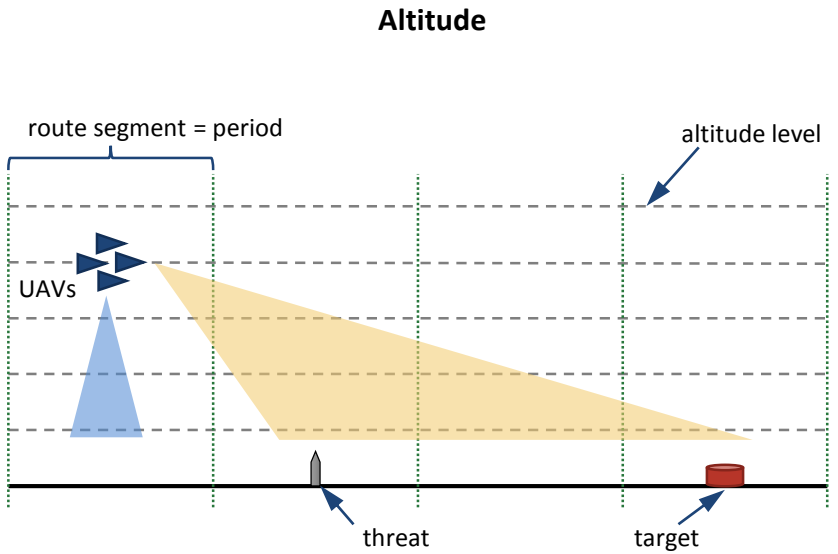
Tactic Code	Description	Latency
IncAlt	Climb one altitude level	1 period*
DecAlt	Descend one altitude level	1 period*
IncAlt2	Climb two altitude levels	1 period*
DecAlt2	Descend two altitude levels	1 period*
GoTight	Change to tight formation	immediate
GoLoose	Change to loose formation	immediate
EcmOn	Turn ECM on	immediate
EcmOff	Turn ECM off	immediate



*default latency, configurable

Adaptation Tactics

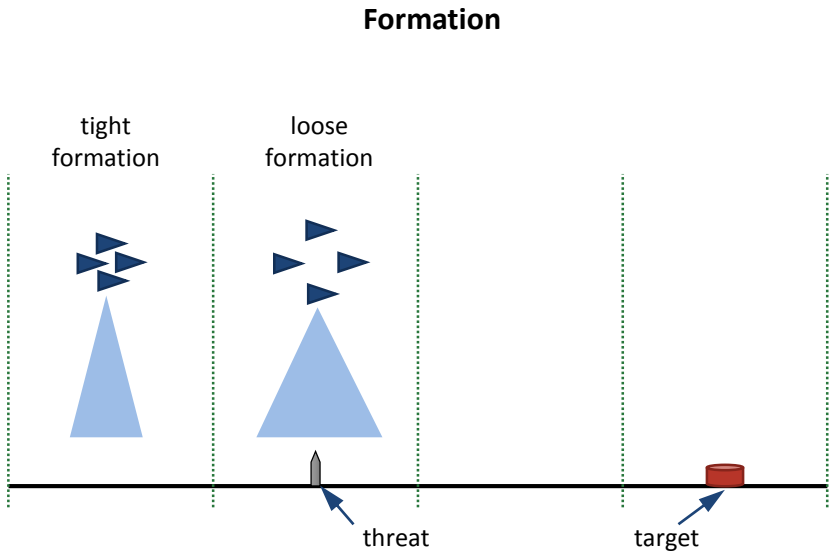
Tactic Code	Description	Latency
IncAlt	**Climb one altitude level**	**1 period***
DecAlt	**Descend one altitude level**	**1 period***
IncAlt2	**Climb two altitude levels**	**1 period***
DecAlt2	**Descend two altitude levels**	**1 period***
GoTight	Change to tight formation	immediate
GoLoose	Change to loose formation	immediate
EcmOn	Turn ECM on	immediate
EcmOff	Turn ECM off	immediate



*default latency, configurable

Adaptation Tactics

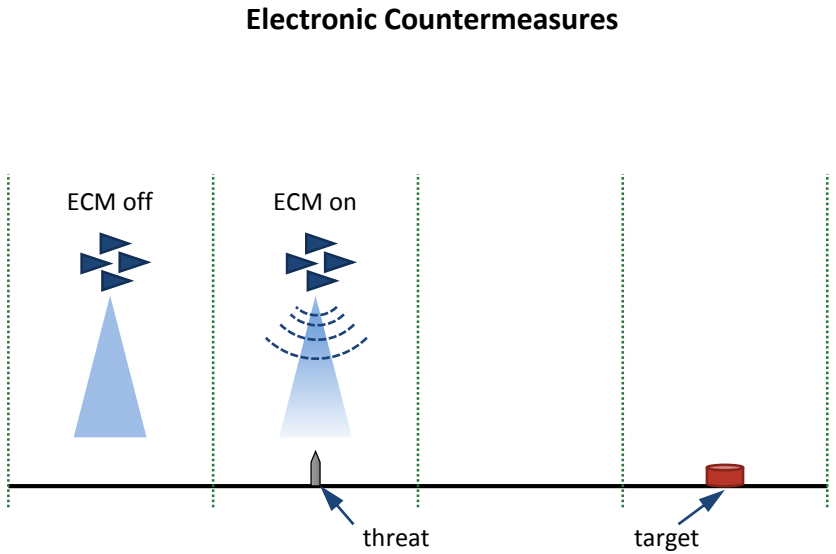
Tactic Code	Description	Latency
IncAlt	Climb one altitude level	1 period*
DecAlt	Descend one altitude level	1 period*
IncAlt2	Climb two altitude levels	1 period*
DecAlt2	Descend two altitude levels	1 period*
GoTight	**Change to tight formation**	**immediate**
GoLoose	**Change to loose formation**	**immediate**
EcmOn	Turn ECM on	immediate
EcmOff	Turn ECM off	immediate



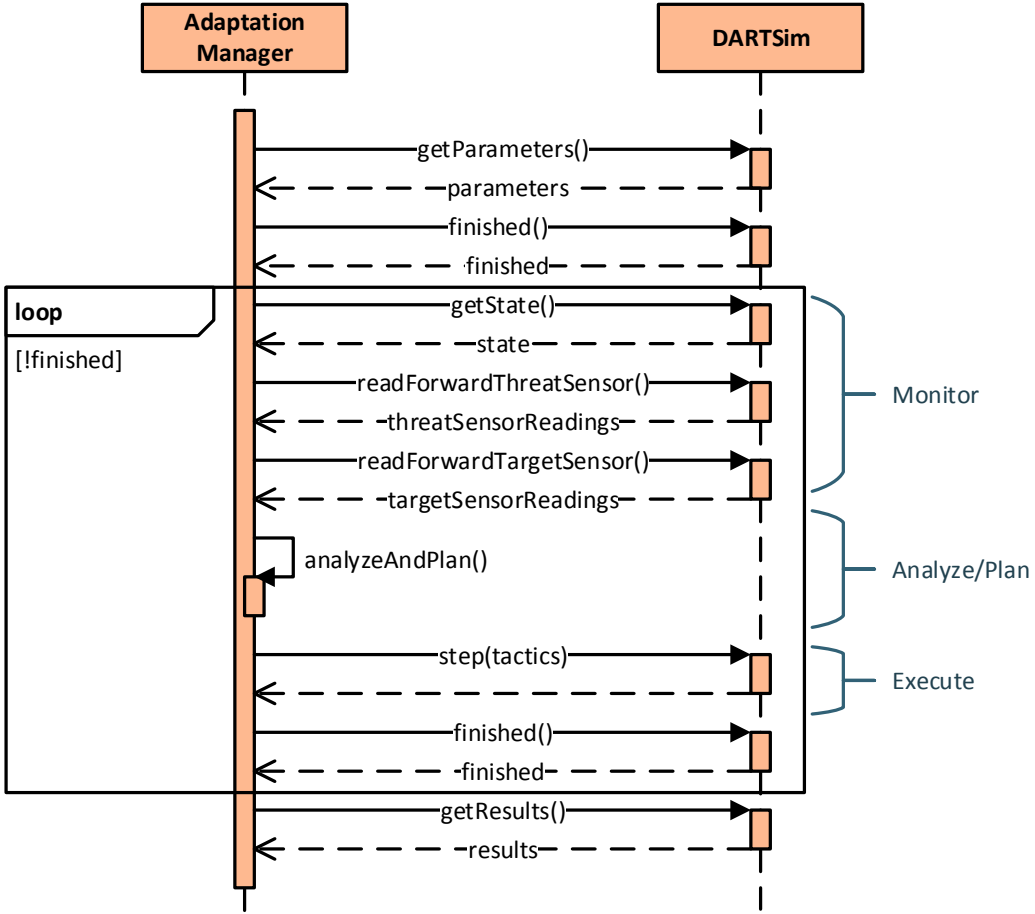
*default latency, configurable

Adaptation Tactics

Tactic Code	Description	Latency
IncAlt	Climb one altitude level	1 period*
DecAlt	Descend one altitude level	1 period*
IncAlt2	Climb two altitude levels	1 period*
DecAlt2	Descend two altitude levels	1 period*
GoTight	Change to tight formation	immediate
GoLoose	Change to loose formation	immediate
EcmOn	**Turn ECM on**	**immediate**
EcmOff	**Turn ECM off**	**immediate**



*default latency, configurable



Interfacing with DARTSim

Two options:

- DARTSim as separate process
  - TCP-based interface
  - JSON-encoded parameters and responses
  - Java client library implementing this interface included
    - Simple integration with adaptation managers in Java
    - Reference implementation

- DARTSim as a library
  - linked with adaptation manager
  - simple method calls from C++
  - no inter-process communication overhead

Demonstration

Get DARTSim

Docker image:

https://hub.docker.com/r/gabrielmoreno/dartsim

Source code:

https://github.com/cps-sei/dartsim