Intelligent perception and control for space robotics

Autonomous Satellite Rendezvous and Docking

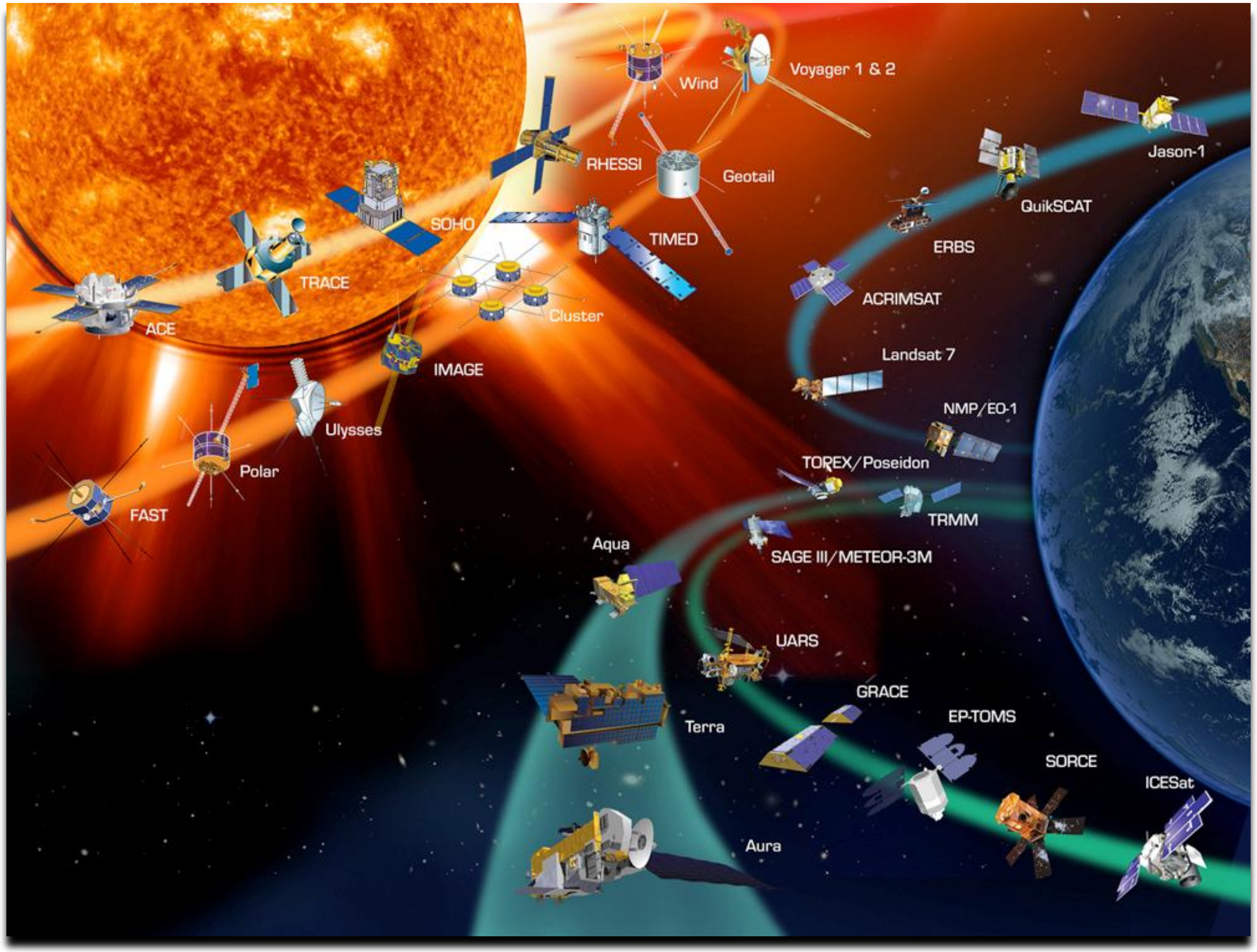
paper by
Faisal Qureshi & Demetri Terzopoulos

Oleksandr Murov
s1372009(at)student.rug.nl

Outline

- Introduction
- Do it autonomous
- Steps involved
- Phases of AR&D
- Developed system
- CoCo control framework
- Priorities of modules
- CoCo system architecture
- Modules
- Results
- Conclusion

Introduction



Goal : System capable of capturing a free flying satellite for the purpose of on-orbit satellite servicing

Why ?

- Now :
  - Manual(astronauts)
    - costly
    - dangerous
  - Discrete-event scripted controllers (pre-programmed)
    - tedious
    - brittle
- Ground-controlled missions are infeasible.

So why do we do it?

- “It extends the operational life of the satellite, mitigates technical risks, reduces on-orbit losses, and helps manage orbital debris.”

=

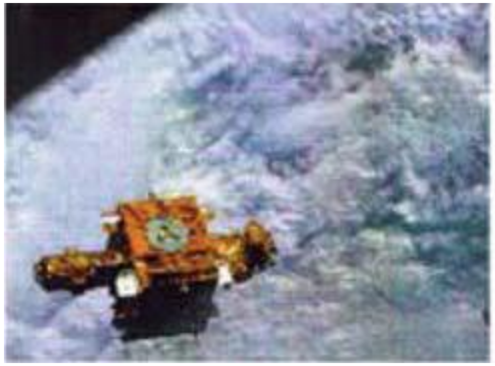
More money is saved on more expensive or one-of-a-kind systems

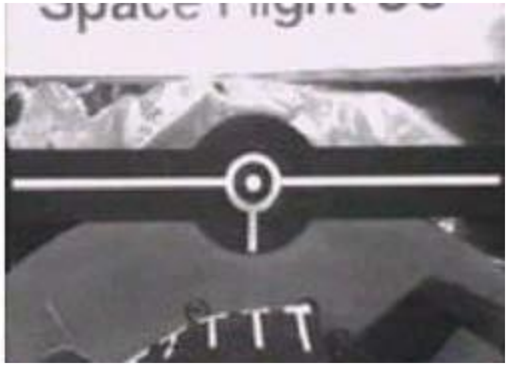
Do it autonomous

- No danger for astronauts
- Can be done with less astronauts on board
- Excludes human error
- Can reach places that manned missions incapable of (GEO)
- But do it with sliding autonomy
  - human operator can take over the manual operation of the robotic system at any level of the task hierarchy.

Steps involved

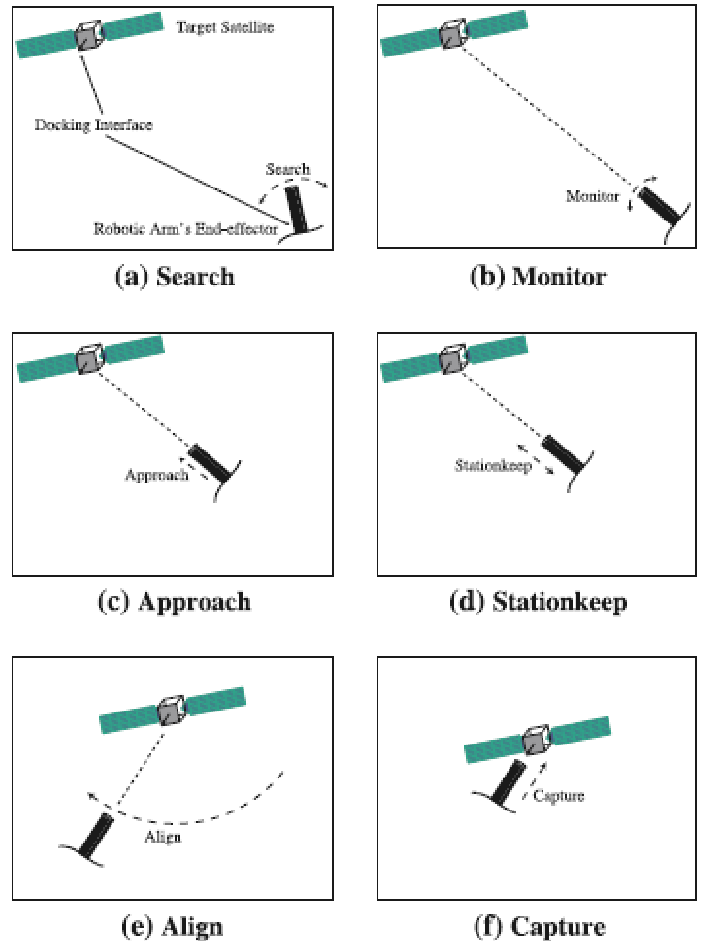
- Rendezvous
  - most interesting and challenging
- Docking
  - connecting to the target satellite





Phases of AR&D

- Six phases for satellite capture :
  - Visual search
    - Images processed to compute an initial estimate of the position and orientation of the satellite
  - Monitor
    - Fixates on the satellite and maintaining distance
  - Approach
    - reduces distance and keeps cameras focused on the target
  - Stationkeep
    - distance is preserved
  - Align
    - aligning with the docking interface of the target satellite
  - Capture
    - end-effector moves in to dock with the satellite.



(a) Search (b) Monitor (c) Approach (d) Stationkeep (e) Align (f) Capture

Developed system

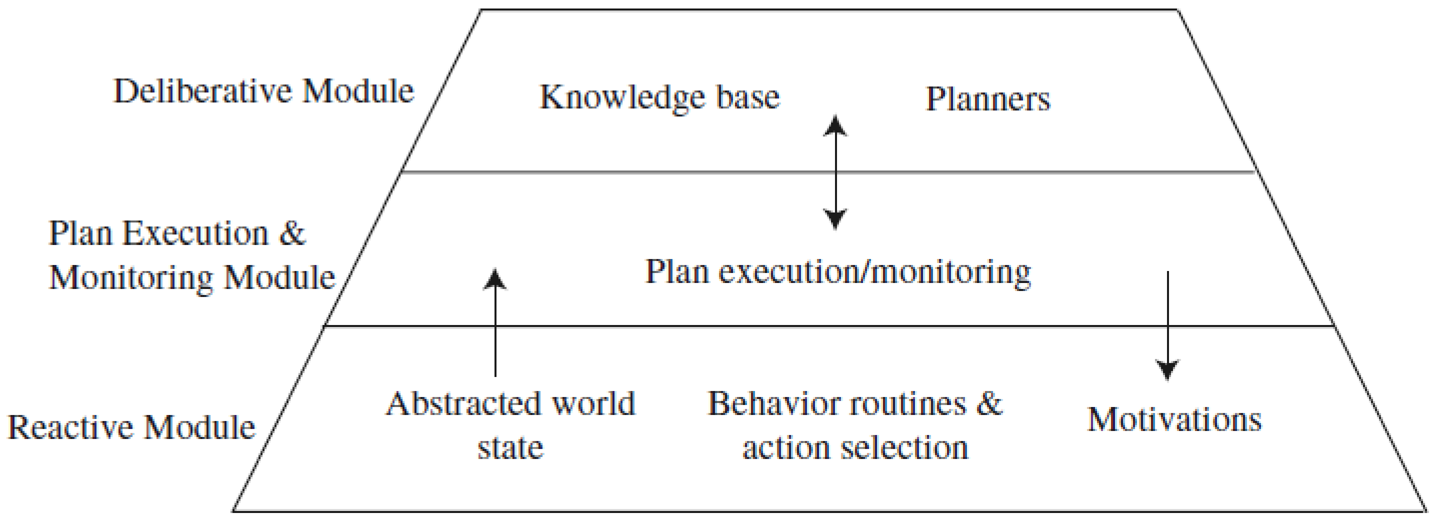
- Visual-guided AR&D system that uses vision as its primary sensory modality
  - Validated in a realistic laboratory environment that emulates on-orbit lighting conditions and target satellite drift
- Features CoCo control framework
  - combines a behavior-based reactive component and a logic-based deliberative component

Developed system

1. images are processed to estimate the pose
2. behavior-based perception and memory units use contextual information to construct a symbolic description of the scene.
3. the cognitive module uses knowledge about scene dynamics encoded using the *situation calculus to construct* a scene interpretation.
4. the cognitive module formulates a plan to achieve the current goal.

The scene description constructed in the third step provides a mechanism to verify the findings of the vision system. Its ability to plan enables the system to handle unforeseen situations.

CoCo control framework



CoCo is a three-tiered control framework that consists of **deliberative, reactive,** and **plan execution and monitoring** modules

CoCo framework

- The **deliberative module** encodes a knowledge-based domain model and implements a high-level symbolic reasoning system
  - can support multiple specialized planners and cooperation between more than one planner

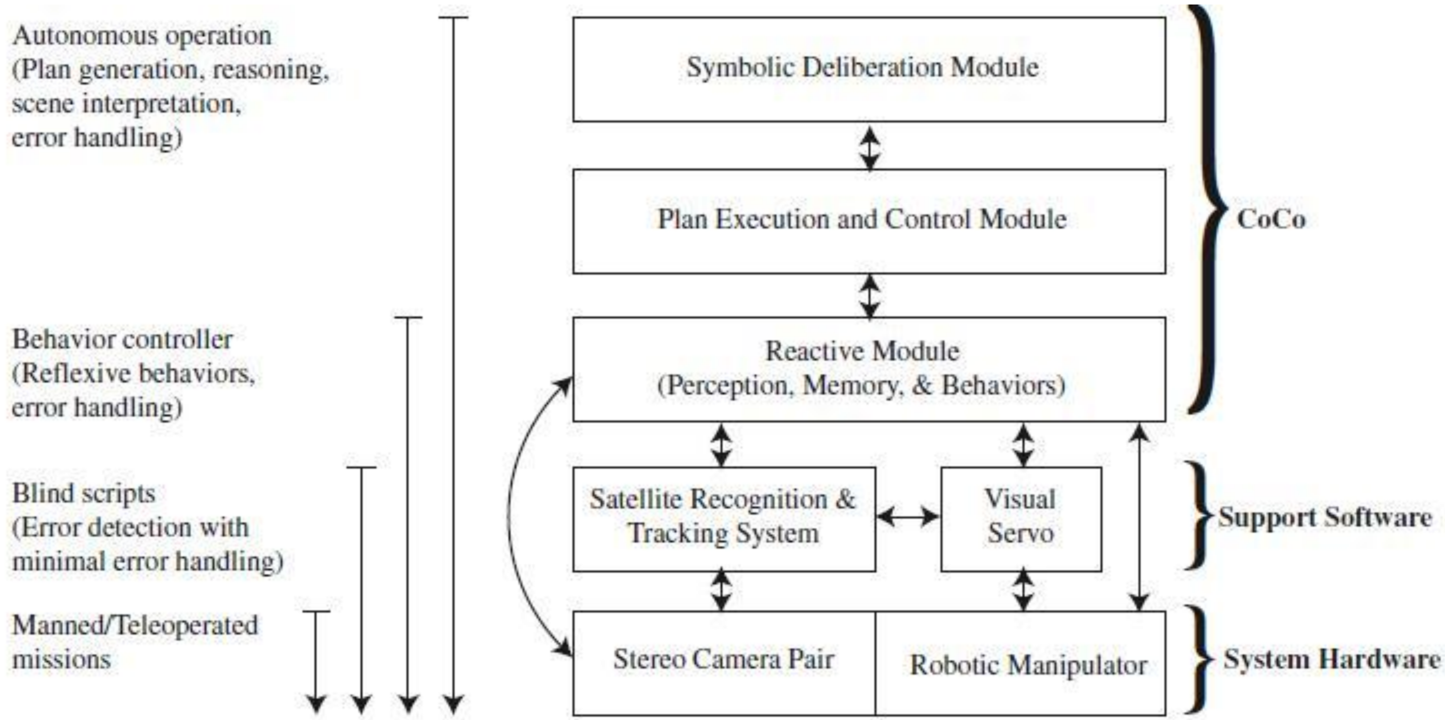
- The **reactive module** implements a low-level behavior-based controller with supporting perception and memory subsystems
  - constructs and maintains the abstracted world state in real-time using contextual and temporal information

- The **plan execution and monitoring module** establishes an adviser–client relationship between the deliberative and reactive modules
  - non-intrusive scheme for combining deliberation and reactivity

Priorities of modules

- The reactive module functions completely on its own and runs at the highest level of priority
  - responsible for immediate safety of the agent
- The deliberative module advises the reactive module on a particular coarse of action through motivational variables, thus it runs on the back-ground level.

CoCo system architecture

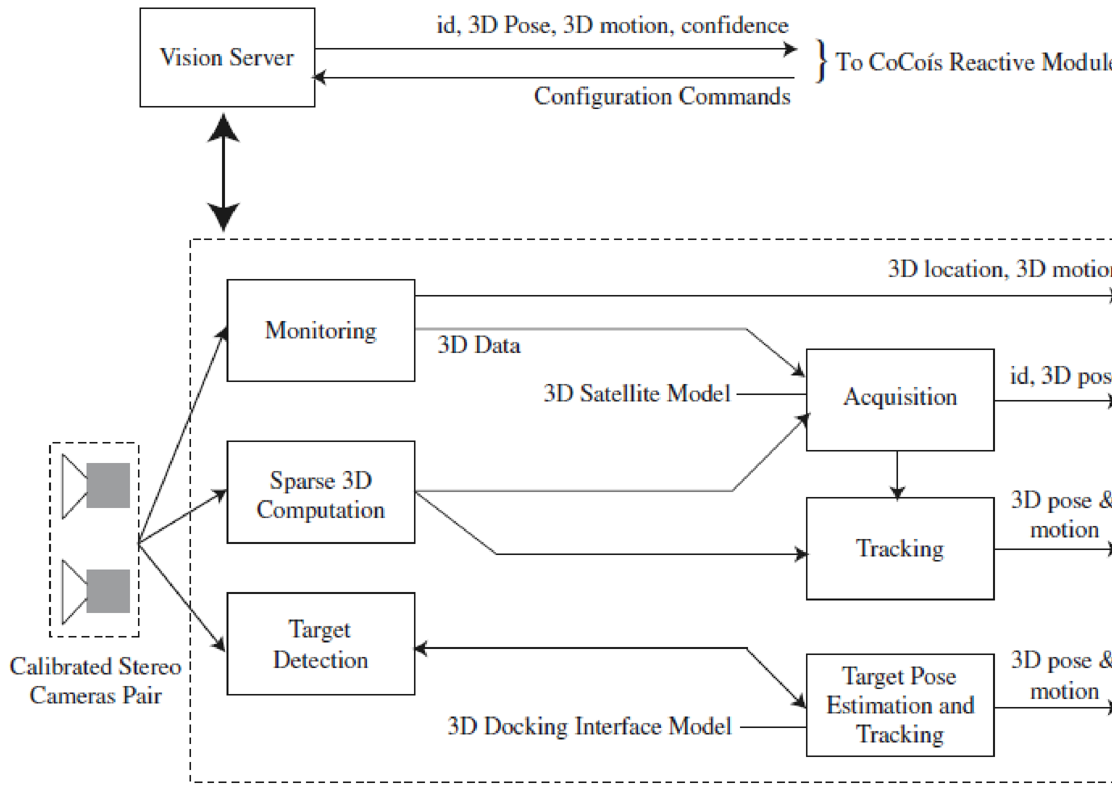


On the left is the list of the mission capabilities corresponding to different levels of control

Total modules overview

- Recognition and tracking module
- Reactive module
- Deliberative module
- Plan execution and monitoring module

Recognition and tracking module



Processes images from a calibrated passive video camera-pair mounted on the end-effector of the robotic manipulator and estimates the relative position and orientation of the target satellite. It supports medium (6m) and short (0.2m) range.

The medium range (≤6m)

- The system relies on natural features observed in stereo images to estimate the motion and pose of the satellite
- There are three configurations:
  1. Model-free motion estimation
  2. Motion-based pose acquisition
  3. Model-based pose tracking

(left image at 5m; right at 0.2m)

Model-free motion estimation

- The vision system combines stereo and structure-from-motion to indirectly estimate the satellite motion in the camera reference frame by solving for the camera motion, which is the opposite of the satellite motion

Motion-based pose acquisition

- Performs binary template matching to estimate the pose of the satellite without using prior information. It matches a model of the observed satellite with the 3D-data produced by the last phase and computes a six degree of freedom (DOF) rigid transformation that represents the relative pose of the satellite.

Model-based pose tracking

- Tracks the satellite with high precision and update rate by iteratively matching the 3D-data with the model using a version of the iterative closest point algorithm (returns an estimate of the satellite’s pose at 2Hz with an accuracy on the order of a few centimeters and a few degrees).

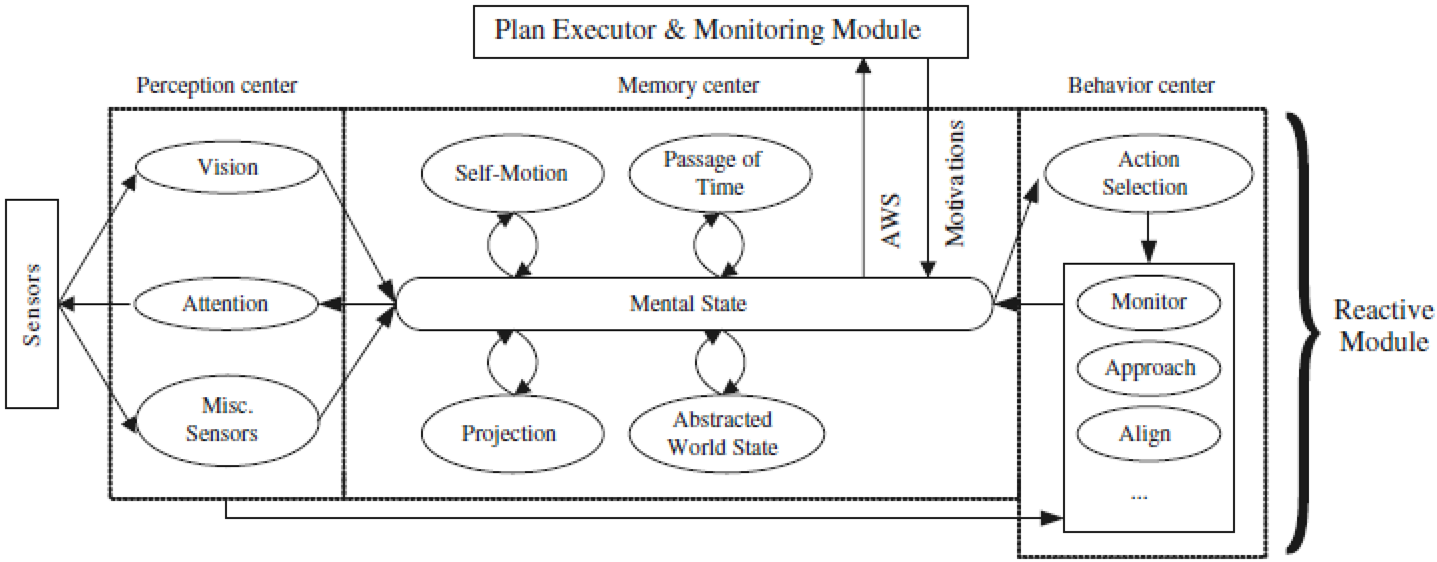
Short range (0.2m)

- One Configuration*: Visual target based pose acquisition and tracking*
  - Due to close distance only monocular images are processed
  - Markers on the docking interface are used to determine the pose and attitude of the satellite efficiently and reliably at close range (need at least 4 points, when there are more the group with best pose information is chosen).
    - Accuracy is in order of a fraction of a degree and 1mm

Output of the visual system

- The vision system returns a 4×4 matrix that specifies the relative pose of the satellite, a value between 0 and 1 quantifying the confidence in that estimate, and various flags that describe the state of the vision system.

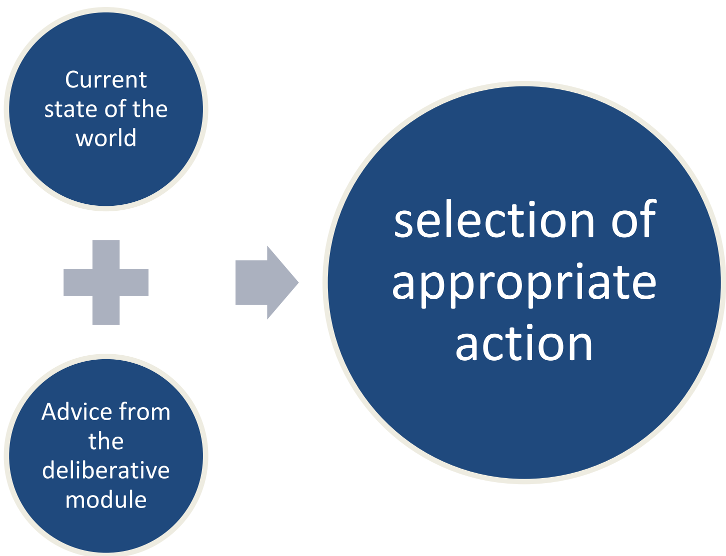
The reactive module



Functional decomposition of the reactive module, which is realized as a set of asynchronous processes and comprises three functional units: perception, memory, and behavior.

Recapitulation

- a behavior-based controller that is responsible for the immediate safety of the agent.
- it functions competently on its own and runs at the highest priority.



Recapitulation

- Present a tractable discrete representation of reality within which the deliberative module can effectively formulate plans.

Perception center

- Two extremes
  - single process computes every feature of interest(longest time)
  - every feature has its own sensing process(highest process management)
- Second approach was chosen
  - higher order features
  - immediately available for subsequential processing

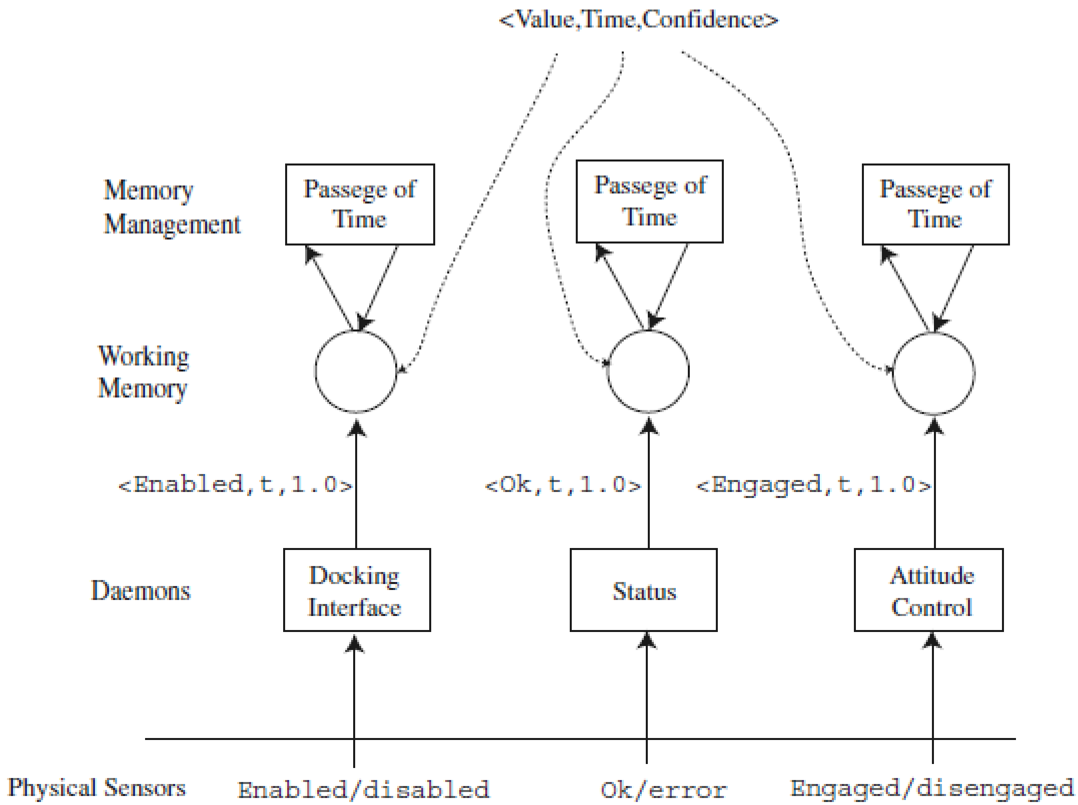
Perception center

- Manages the vision system :
  - activation of different vision modules
  - combination of the information (processing time, operational ranges, and noise)
  - constructs appropriate perceptual features and stores them in the memory center for later use (behavior center, behavior execution)

Memory center

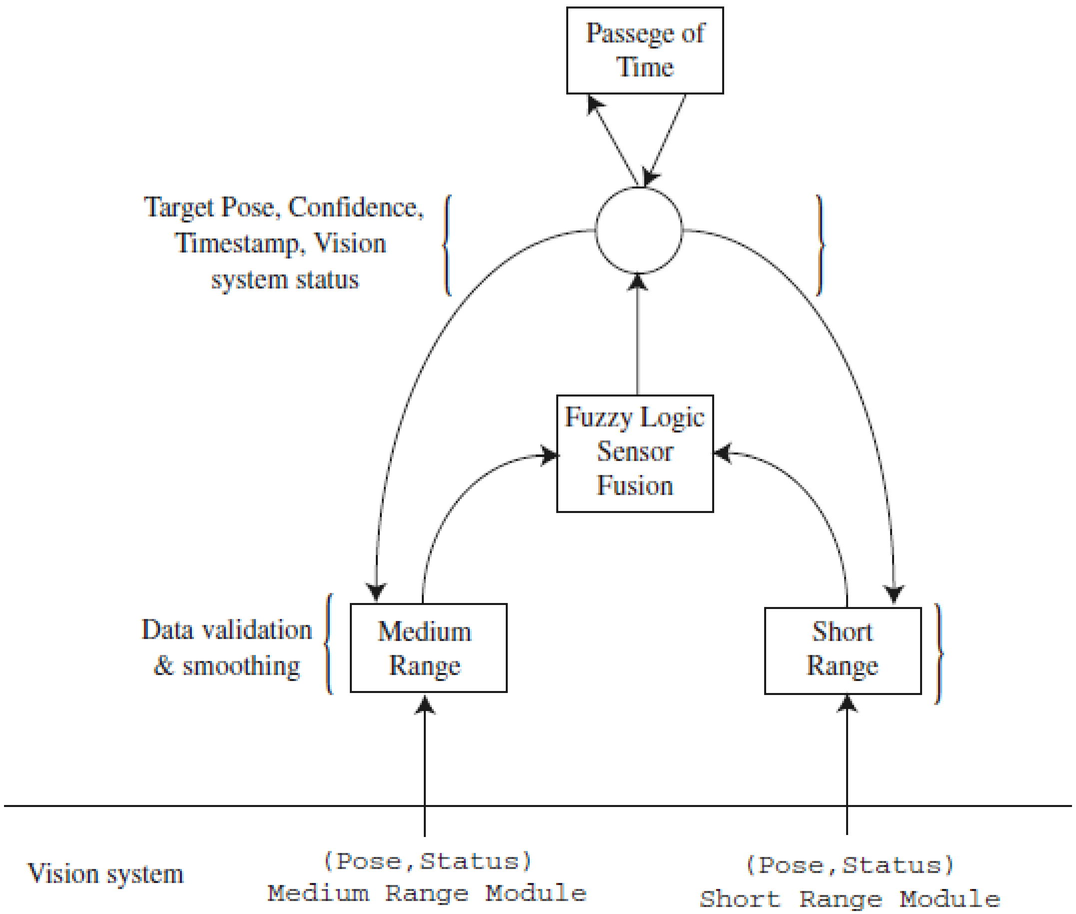
- Perceptual features are assigned a timestamp and confidence value between 0 and 1, and they are stored in memory center
- Responsible for use of relevant and correct information.
- More later…

Cooperation



Daemon processes for monitoring and reading satellite attitude control, docking interface, and robotic arm status sensors. The daemons that collect information from the sensors are associated with the perception center (*bottom row), whereas those that operate upon the working* memory belong to the memory center (*top two rows)*

Communicating with the vision module



The perception center is in charge of the vision system that implements satellite identification, recognition, and tracking routines.

Decision about acceptance are thresholded (medium 0.3, short 0.6).
Threshold reflects the performance of the vision system.

Medium to short and back

- The perception center is responsible for transitioning the visual tracking task from the medium to the short range module and back if necessary.
  - estimated distance
  - confidence values

Justification of transition

- Operational ranges of different vision modules
  - The operational range of a vision module and the estimated distance of the target satellite determines the suitability of the vision module
  - one module is more successful then another one
- But avoid unnecessary hand-overs

Fuzzy Logic Sensor Fusion

- Fuzz logic based sensor fusion scheme computes a weighted sum of the pose estimates from active vision modules
- The position **p** of the satellite is given by:

$$\mathbf{p} = w\mathbf{p}_s + (1 - w)\mathbf{p}_m \qquad (1)$$

Where $0 \le w \le 1$ is the weight determined by the fuzzy logic based controller for short-range module. $\mathbf{p}_s$ and $\mathbf{p}_m$ position estimates.

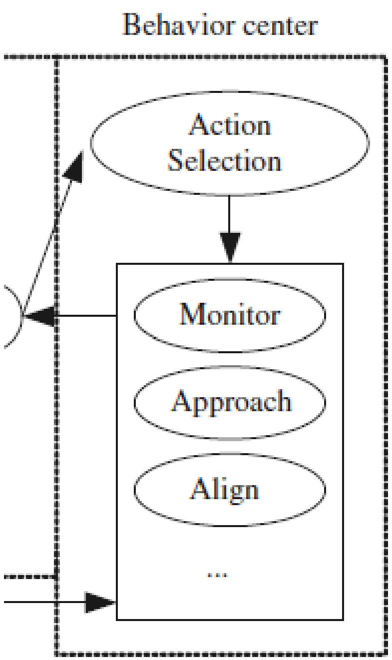
Fuzzy Logic Sensor Fusion

- Orientation estimation done also with combination of short- and medium-range modules

  $q=(q_s q_m^{-1})^w q_m$ (2)

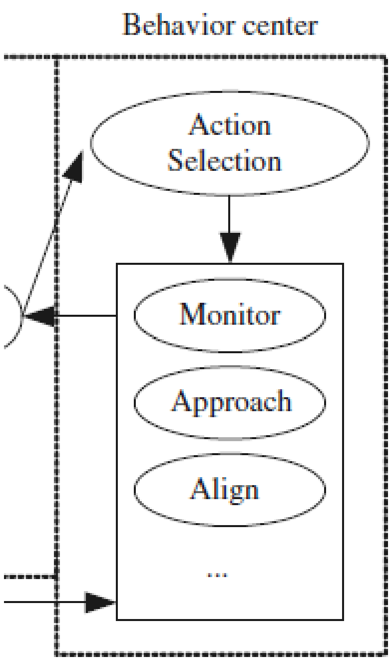
q_s and q_m – the rotation estimates.

Behavior center



- Manages the reactive module’s behavioral repertoire
  - The reactivemodule supports multiple concurrent processes, and arbitrates between them so that the emergent behavior is the desired one
- At each instant the appropriate high level behavior is chosen according to current state of the world and the motivations

Behavior center



- When conditions are favorable the reactive module will follow the advice from the deliberative module
- Otherwise one that ensures the safety of the agent
- Action is chosen by the motivational variables

Motivational variables

- a set of internal mental state variables, which encode the motivations of the robotic arm: (1) search, (2) monitor, (3) approach, (4) align, (5) contact, (6) depart, (7) park, (8) switch, (9) latch, (10) sensor, and (11) attitude control
- Values between 0 and 1
- The behavior with highest variable is selected.
- Prioritizing resolves behavior selection conflict
- After goal is fulfilled the variable decreases to 0

Motivational variables

- Level-of-interest –prevents infinite pursued of an unattainable goal
  - Cutoff time for each motivational variable
- Mutual inhibition –to avoid behavior where the action selection keeps alternating between two goals without ever satisfying either of them.

Motivational variables

- The values of the motivational variables are calculated as follows

$$m_t = \begin{cases} max(0, m_{t-1} - d_a \Delta t(1 - d_b m_{t-1}^2)) \; when \; t > t_c \; or \; the \; associated \; behavior \; achives \; its \; goal, \\ min(1, m_{t-1} + g_a \Delta t(e^{m_{t-1}^2} - g_b)) \; when \; the \; associated \; behavior \; is \; first \; initiated \end{cases}$$

- where $0 \leq g_a, g_b, d_a, d_b \leq 1$ are the coefficients that control the rate of change (here 0.5,0.99,0.05,0.99)
- Δt is time step
- t_c is cutoff time that depends on the motivation and other motivational variables (>0).

$$t_c = \{0 > t_1 > t_2 > \infty\}$$

Memory center

- Manages the short-term memory of the agent
  - It holds the relevant sensory information, motivations, state of the behavior controller, and the abstracted world state.
  - is responsible for ensuring that this information is valid
- Thus, the behavior center need not wait for new sensory information; it can simply use the information stored in the memory center.

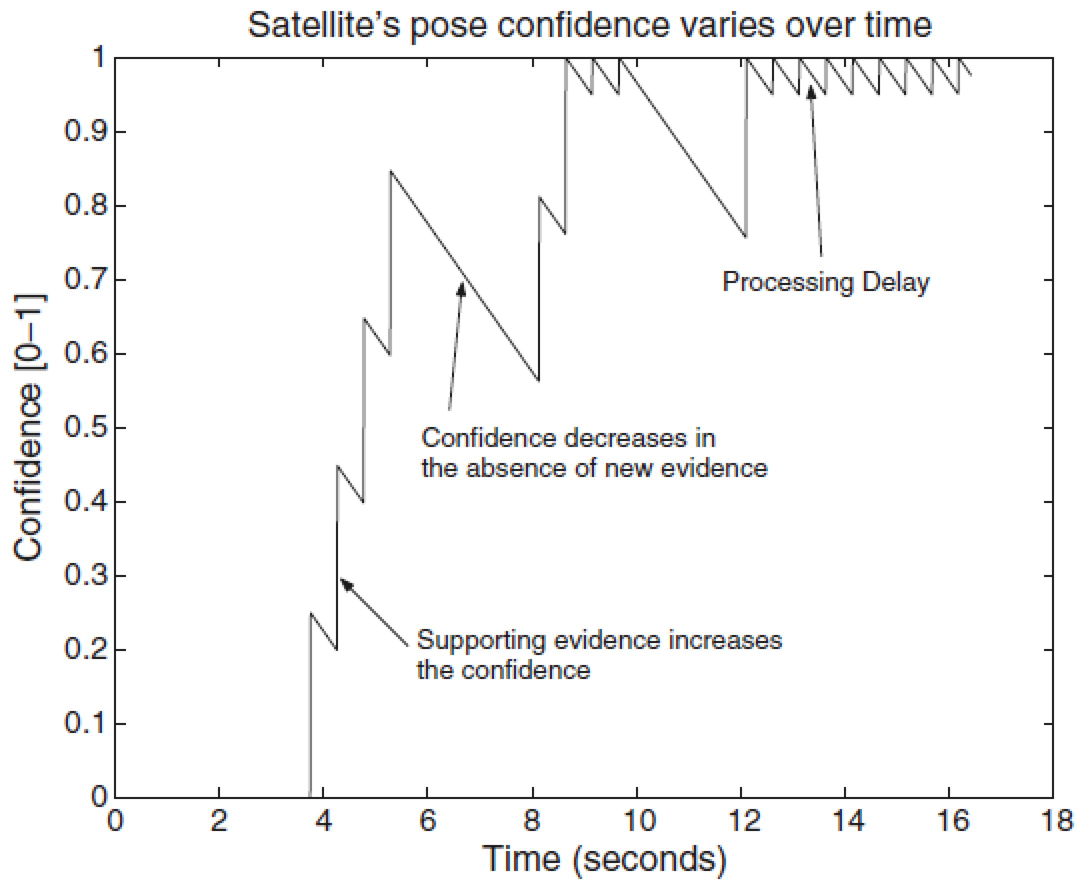
Memory center

- Uses two behavior routines per feature

1. Self-motion
   – constantly updates the internal world representation of position, heading and speed
2. Passage-of-time
   – ensures the currency and coherence of the information

Representation

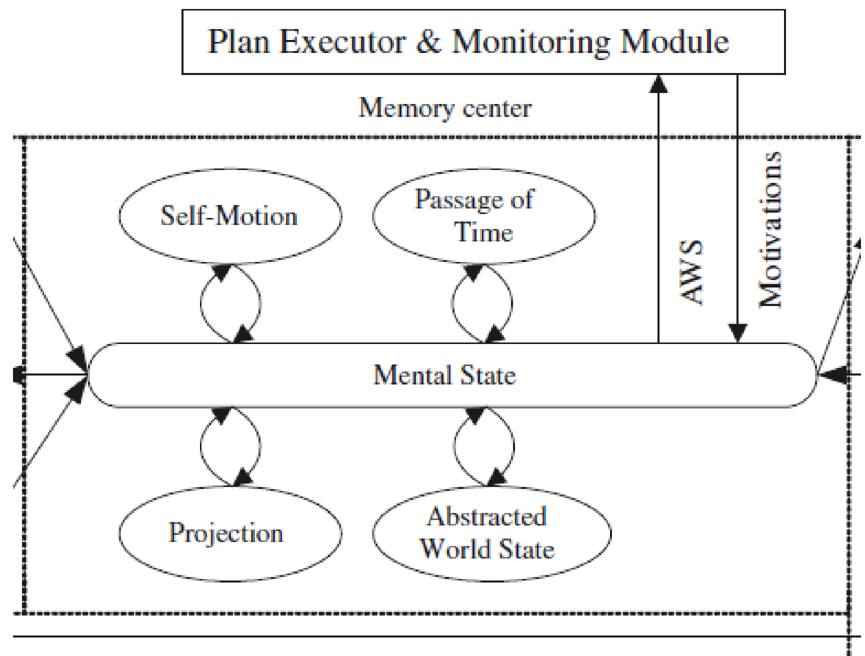
- Each perceptual feature is represented as a tuple <Value, Timestamp, Confidence>
  - With absence of new readings Confidence should decrease with time
- The ability to forget
  - prevents the reactive or the deliberative modules from operating upon inaccurate, or worse, incorrect information
  - detecting sensor malfunctioning



The confidence in the satellite's pose decreases in the absence of supporting evidence from the vision system

Connection between reactive and deliberative modules

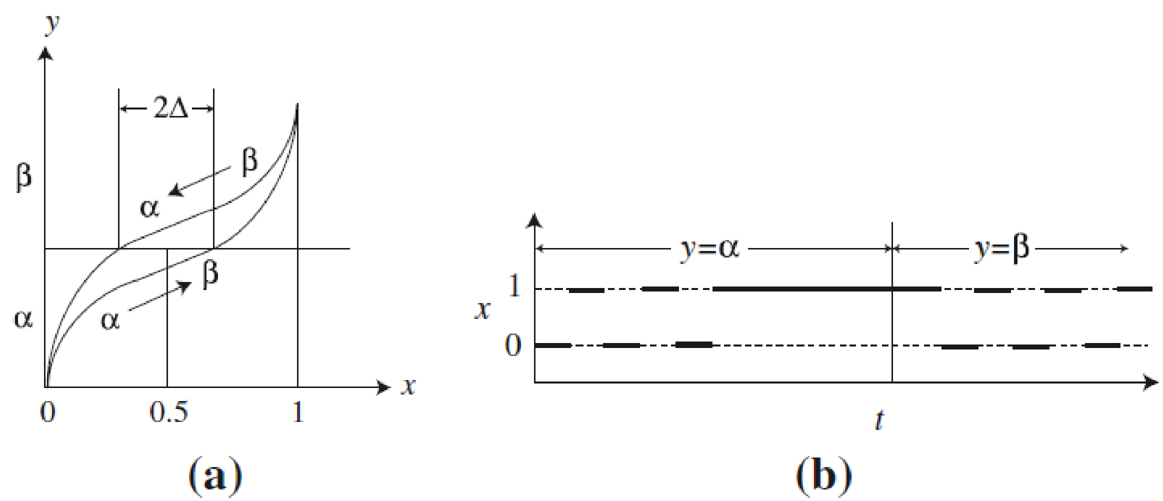
- Reactive module – detailed sensory information
- Deliberative module – abstract information about the world
- Memory center filters out information and generates symbolical representation of the world –Abstract world state (AWS)



AWS

- The abstracted world state is a discrete, multi-valued representation of an underlying continuous reality
- Discretization involves dividing a continuous variable into ranges of values and assigning the same discrete value to all values of the continuous variable that fall within a certain range.

Problems of discretization



Emulating hysteresis during discretization. **a** y is a discretization of x that takes values between 0 and 1. If y is α and the value of $x > 0.5 + \Delta$, y becomes β. Otherwise, if y is β and the value of $x < 0.5 - \Delta$, then y becomes α. **b** $x \in [0, 1]$ is mapped to $y \in [\alpha, \beta]$. The state $y = \alpha$ indicates that $x = 0$ and $y = \beta$ indicates that $x = 1$. The variable y resists changing its value from α to β and vice-versa, which allows y to exhibit more stable behavior by ignoring spurious changes in x

Still to come

- The deliberative module
- Plan execution and monitoring module
- Results
- Conclusion

The deliberative module

- The deliberative module endows our agent with the ability to plan its actions, so that it can accomplish high-level tasks that are too difficult to carry out without “thinking ahead.”
- It maintains a set of planners, each with its own knowledge base and planning strategy
- GOLOG was used to develop planners

Planners

- Planner A
  - responsible for generating plans to achieve the goal of capturing the target satellite

Primitive actions available to the planner that creates plans to accomplish the goal of safely capturing the target satellite

Actions/#args	Arguments' values	Description
aTurnon/1	On/off	Turns on the servicer
aLatch/1	Arm/disarm	Enables/disables the latching mechanism
aErrorHandle/1		Informs the operator of an error condition
aSensor/2	Medium/near, On/off	Configures the vision system
aSearch/1	Medium/near	Initiates medium/short visual search sequence
aMonitor/0		Initiates monitor phase
aAlign/0		Initiates align phase
aContact/0		Moves in to make contact
aGo/3	Park/medium/near, Park/medium/near, Vis/mem	Moves to a particular location using either current information from the vision system (if vision system is working satisfactorily) or relying upon the mental state
aSatAttCtrl/1	Off/on	Asks ground station to turn off the satellite attitude control
aCorrectSatSpeed/0		Informs the operator that the satellite is behaving erratically

Planners

- Planner B
  - attempts to explain the changes in AWS

Primitive actions available to the planner that constructs abstract, high-level interpretations of the scene by explaining how the AWS is evolving

Actions/#args	Arguments' values	Description
aBadCamera/0		Camera failure
aSelfShadow/0		Self-shadowing phenomenon
aGlare/0		Solar glare phenomenon
Sun/1	Front/behind	The relative position of the Sun
aRange/1	Near/medium	Distance from the satellite

Planners world

The abstracted world state for the satellite servicing task

Fluents/arity	Values	Description
fStatus/1	On/off	Status of the servicer
fSatPosConf/1	Yes/no	Confidence in the estimated pose of the satellite
fSatPos/1	Near/medium/far/contact	Distance from the satellite
fSatSpeed/1	Yes/no	Whether the satellite's relative speed is within the acceptable limits
fLatch/1	Unarmed/armed	Status of the latch (docking interface)
fSatCenter/1	Yes/no	Whether the satellite is in the center of the field of view
fSatAlign/1	Yes/no	Whether servicer is aligned with docking interface of the satellite
fSensor/2	Short/medium, on/off	Current configuration of the vision system
fError/1	Sensor/shadow/any/no	Error status
fSatContact/1	False/true	Whether satellite is already docked
fSatAttCtrl/1	On/off	Whether or not the satellite's attitude control is active
fSun/1	Front/behind	Location of the Sun relative to the servicer
fRange/1	Near/far	Distance from the satellite

The choice of fluents describing the abstracted world state depends upon the target application

Plan

- Sequence of zero or more actions
  - Each action contains execution extractions
  - Preconditions
  - Postconditions

A linear plan generated by the GOLOG program to capture the target

---

Starting world state:

fStatus(off) $\wedge$ fLatch(unarmed) $\wedge$ fSensor(all,off) $\wedge$ fSatPos(medium) $\wedge$ fSatPosConf(no) $\wedge$ fSatCenter(no) $\wedge$ fAlign(no) $\wedge$ fSatAttCtrl(on) $\wedge$ fSatContact(no) $\wedge$ fSatSpeed(yes) $\wedge$ fError(no)

Execution result:

aTurnon(on) $\rightarrow$ aSensor(medium,on) $\rightarrow$ aSearch(medium) $\rightarrow$ aMonitor $\rightarrow$ aGo(medium,near,vis) $\rightarrow$ aSensor(short,on) $\rightarrow$ aSensor(medium,off) $\rightarrow$ aAlign $\rightarrow$ aLatch(arm) $\rightarrow$ aSatAttCtrl(off) $\rightarrow$ aContact

Scene interpretation

- Progress in monitored by examining AWS
- Upon encountering an undesirable situation the interpretations is attempted
- Success –appropriate corrective steps are suggested
- Failure –safely abort the mission

Cooperation between active planners

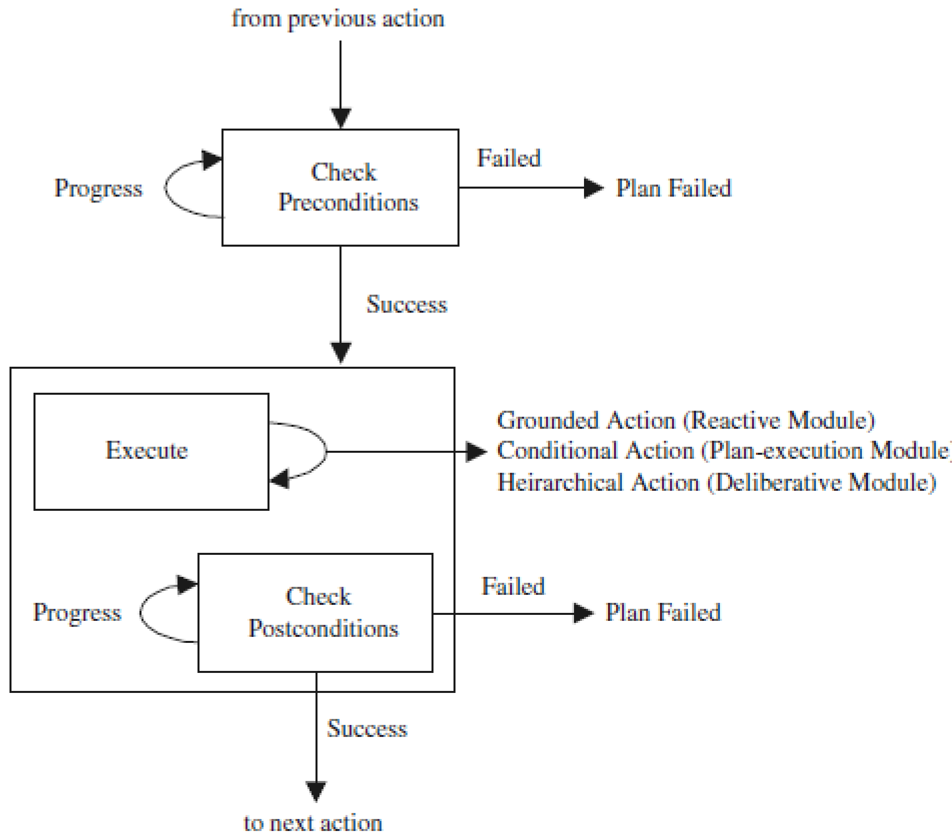
- The planners cooperate to achieve the goal—safely capturing the satellite
- Planner A generates a plan that transforms the current state of the world to the goal state
- Planner B is only activated when problem is detected and suggests possible corrections
  - Abort mission
  - Retry immediately
  - Retry after a random interval of time
- If Planner B successful Planner A resumes

Plan Execution and Monitoring (PEM) module

- Acts as an intermediate for deliberative and reactive modules
  - It initiates the planning activity in the deliberative module
    - when the user has requested the agent to perform some task,
    - when the current plan execution has failed,
    - when the reactive module is stuck, or
    - when it encounters a non-grounded action that requires further elaboration

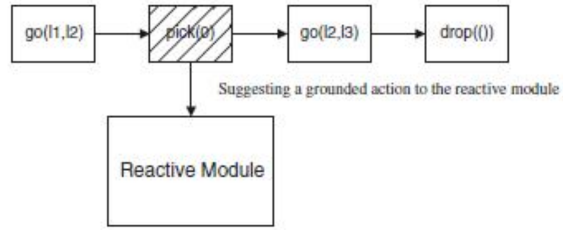
PEM

- Can handle
  - linear plans –deliberative module
  - conditional, and hierarchical plans –scripts uploaded by human operators
- Can execute multiple actions concurrently
  - However, it assumes that plan execution control knowledge for these planes will prevent any undesirable side affects of concurrent execution



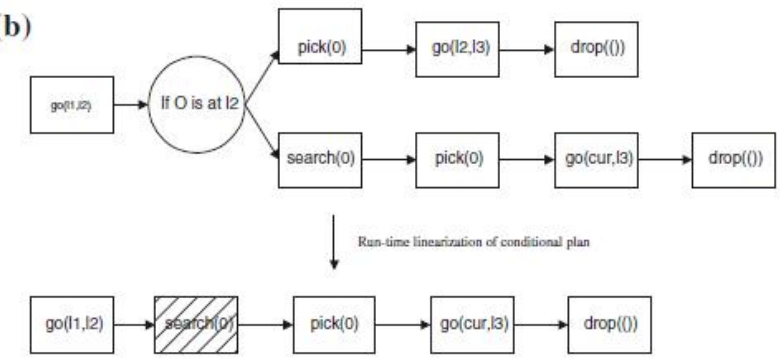
The plan execution and monitoring module sequentially executes each action. It checks the current action's preconditions until they succeed or fail. If they succeed, it enters the current action's execution/postcondition-check loop, wherein it activates the current action's execution code until the postconditions either succeed or fail. Upon success, it proceeds to the next action

(a)



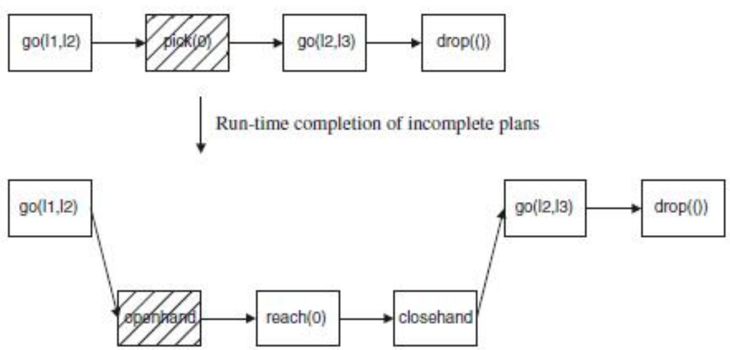
PEM suggests a grounded action to the reactive module

(b)



PEM evaluates the condition, and the out-come determines which actions are chosen

(c)



At run-time, the PEM expands the non-grounded action as a linear plan

PEM module executing linear, conditional, and hierarchical plans

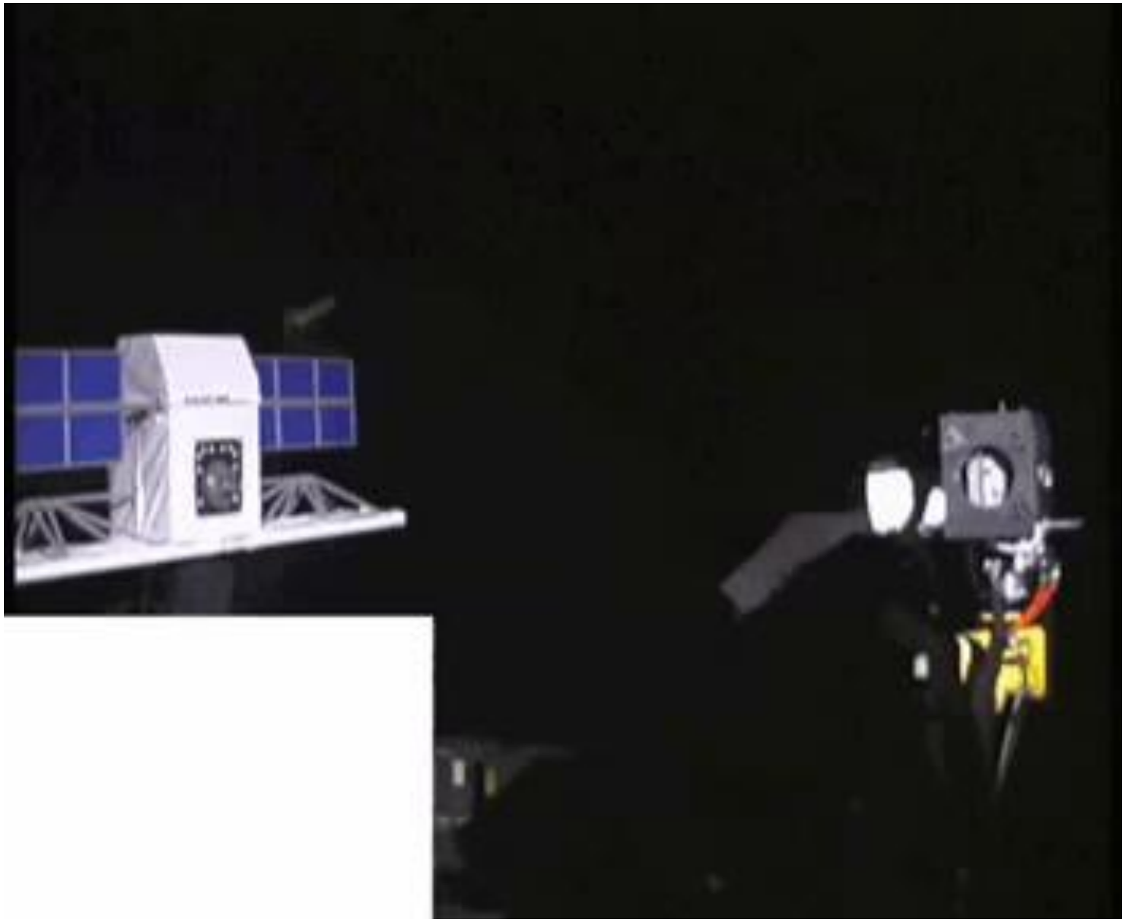
Plan execution control knowledge

- The PEM relies upon it to properly execute a plan
- Defined over the abstracted world state, and it consists of conditions that must hold before, during, or after an action (or a plan).
  - Span over entire plan
  - Action-dependant
- Questions that are vital for the correct execution of a plan:
  - Plan validity (a plan might become irrelevant due to some occurrence in the world).
  - Action execution start time.
    - Now.
    - Later.
    - Never; the plan has failed.
  - Action execution stop time.
    - Now; the action has either successfully completed or failed.
    - Later; the action is progressing satisfactorily.

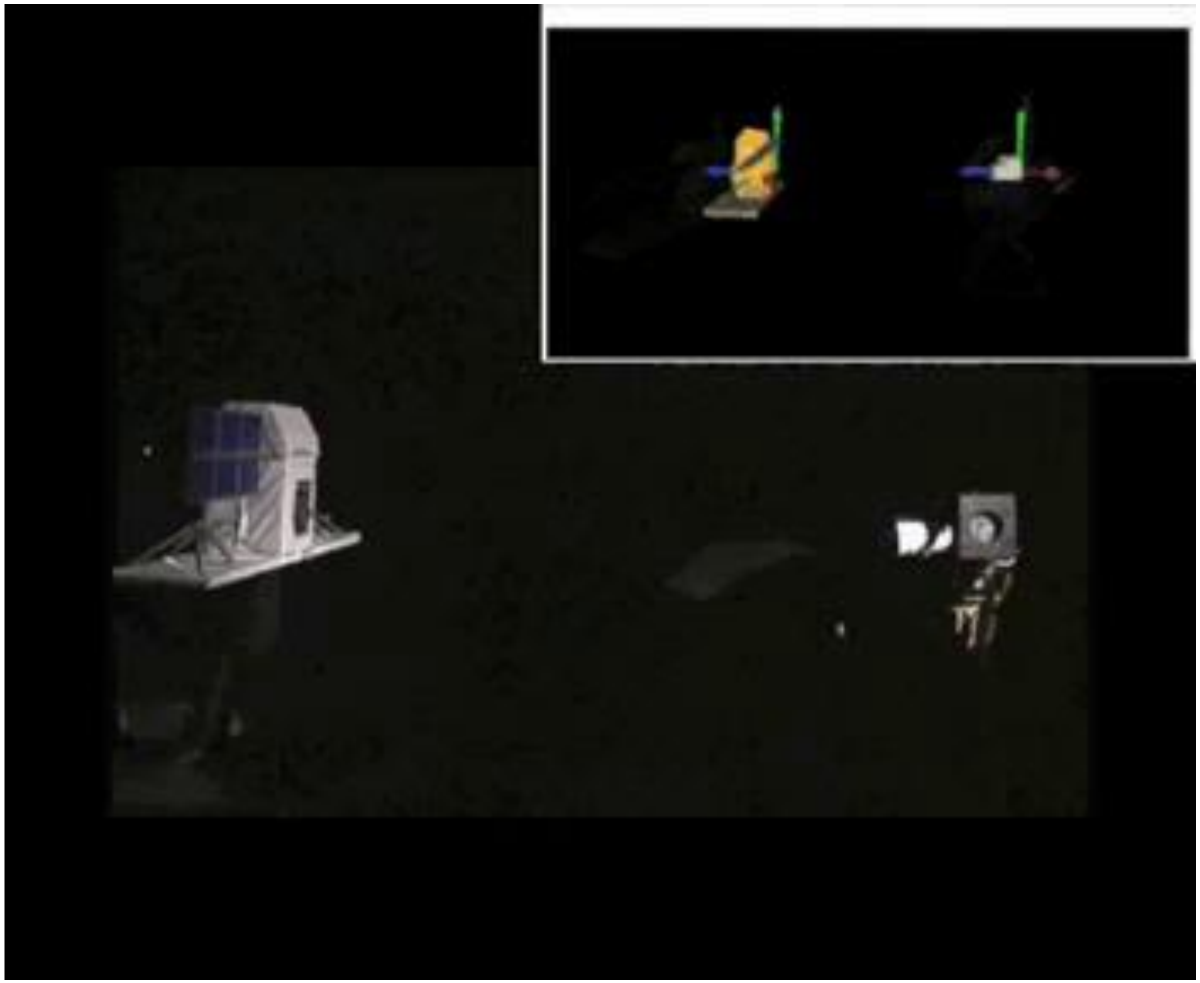
Results

- In the following movie despite a simulated vision system failure, the servicer robot captures the satellite using vision by making use of its cognitive abilities

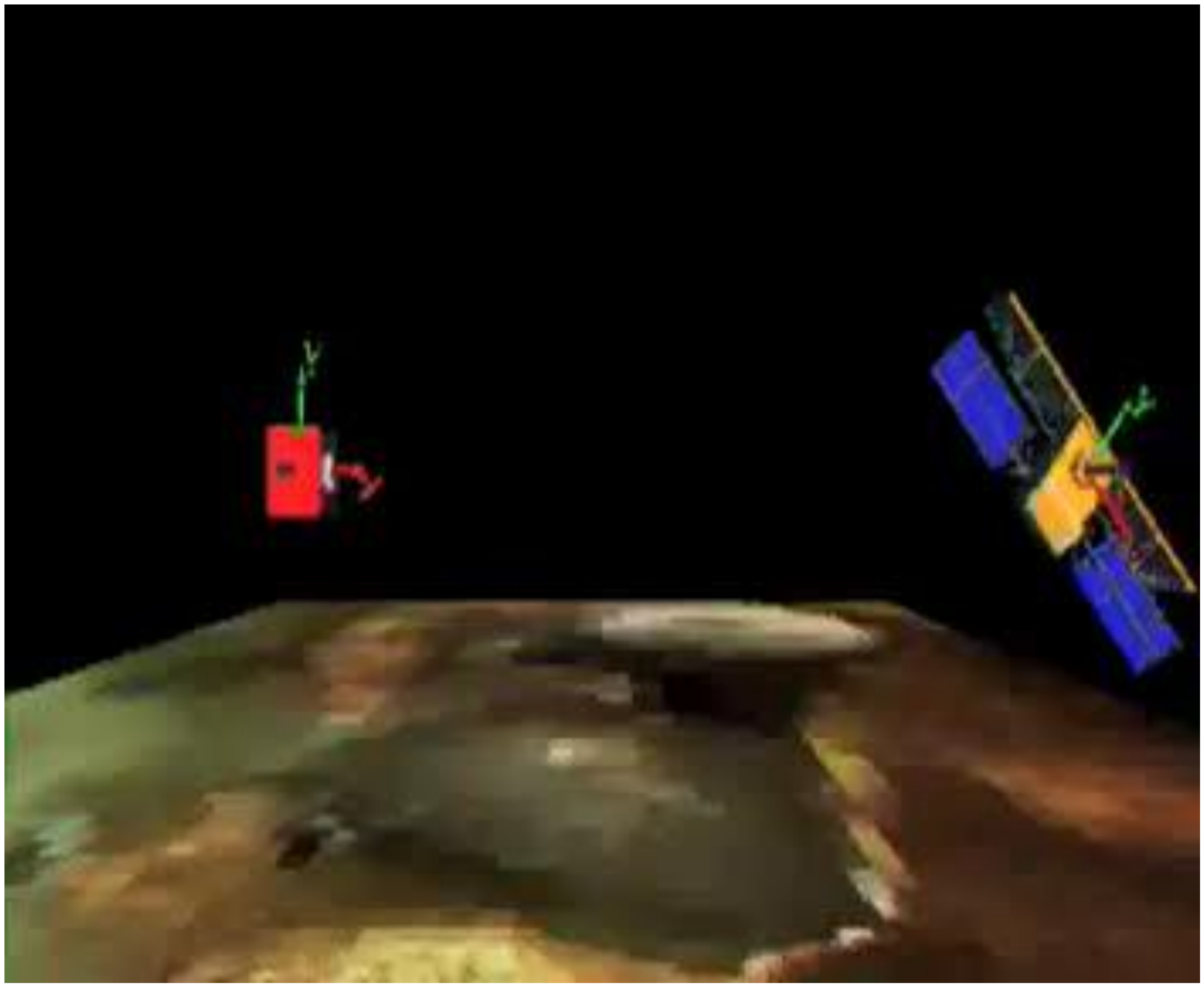
Results (failure)



Results (no failure)



Results (computer simulation)



Conclusion

- Combination of high-level AI components and low-level reactive components resulted in system capable of successful reasoning and performance on the complex task of autonomously capturing a free-orbiting satellite.

Questions

?