Task Level Collaboration with Robotic Assistants

ATRV Rover Testbed  Humanoid Simulators

These Robotic Assistants must be able to:

1.)  Interpret task level commands
2.)  Execute the tasks safely and reliably, even under disturbances and execution uncertainties

1.) Interpreting task level commands

Reactive Model-Based Programming Language (Williams et al., ISAIRAS 01)

Apollo LRV Deployment Sequence

Abstract Task Decomposition

(*picture courtesy of NASA)
Task Level Collaboration with Robotic Assistants

ATRV Rover Testbed  Humanoid Simulators

These Robotic Assistants must be able to:

1.) Interpret task level commands
2.) Execute the tasks safely and reliably, even under disturbances and execution uncertainties

Flexible Execution Times

(Dechter, Meiri, Pearl 91)

A Simple STN:

\[ \begin{align*}
    A & \Rightarrow B \\
    & \Rightarrow C \\
    & \Rightarrow D \\
    & \Rightarrow E \\
    & \Rightarrow F
\end{align*} \]

Consistent!

Determine STN consistency:

- Calculate the Single Source Shortest Path (polynomial-time algorithm)

Inconsistent STN!

Determine STN consistency:

- Calculate the Single Source Shortest Path (polynomial-time algorithm)
- A continually looping negative cycle indicates an inconsistency in STN

Two methods to detect a continually looping negative cycle:

1.) Check for any d-value to drop below \(-nC\). (most space efficient)
2.) Keep an acyclic spanning tree of support, and terminate when a self-loop is formed. (Cesta, Oddi 96) (most time efficient)

Incremental Reasoning Algorithm

- Basic Idea:
  1.) Keep dependency information for each shortest-path value in the distance graph. (Cesta, Oddi 96)
  2.) Use incremental update rules to localize necessary changes to the distance graph:
     a.) 3 Update Rules to change a consistent distance graph.
     b.) 3 Update Rules to repair an inconsistent distance graph.

- ITC’s Novel Claims:
  1.) A conflict extraction mechanism to guide plan repair
  2.) Allow multiple arc-changes
  3.) Can repair inconsistent distance graphs incrementally

Performance Improvements

UAV Scenarios

Comparison of Algorithm Runtimes

Randomly Generated Plans
Robustness To Environmental Disturbances and Uncertainty – Andreas Hofmann, PhD Thesis

Nominal Walking
- Allows for linearizing controllers that decouple state variables and makes them directly controllable
  - [Hofmann, et al; 2004]

Walking with constrained foot placement
- Implemented in controller through Lagrangian relaxation
- Orientation goals lower priority than balance goals
- When disturbed, sacrifice tight angular momentum conservation temporarily
  – Until balance restored

Hybrid executive coordinates controllers to sequence biped through qualitative state plan

Plan compilation for efficient execution
- Compiler:
  - Computes tubes of feasible control trajectories from Qualitative State Plan.
- Dispatcher:
  - Dynamically searches for optimal control trajectories within tubes.
  - Dispatcher “pulls springs” of each state variable.
Qualitative State Plan
Goal Regions

Feasible trajectories must go through goal regions

Flow tubes denote all feasible trajectories

Flow tubes denote all feasible trajectories

Flow tubes denote all feasible trajectories

Center of Mass CM tube constrained by foot position tubes:
- Foot positions define support polygon.
- Center of foot Pressure CP constrained to be inside support polygon
- CM coupled to CP.
Center of Mass CM tube constrained by foot position tubes:
- Foot positions define support polygon.
- Center of foot Pressure CP constrained to be inside support polygon.
- CM coupled to CP.

Successful execution of activity
- Execution successful if trajectory stays in state tube.
- If trajectory begins in initial region, and no disturbances, will stay in tube.
- If trajectories of each activity stay in tube, plan executes successfully.

Disturbance displaces trajectory in state space
- If disturbance too large, trajectory pushed outside tube.
- Goal region not achievable at the required time.
- Plan failure detected immediately leaving more room for recovery.

Conclusions
These Robotic Assistants must be able to:
1.) Interpret task level commands with the Reactive Model-based Programming Language
2.) Execute the tasks safely and reliably, even under disturbances and execution uncertainties with Temporal Plan Networks, Qualitative State Plans, and Flow Tubes
Extra Slides

3 Update Rules to Change a Consistent Distance Graph

- Keep track of the support for each shortest path value
- Given a consistent STN, changing Arc(i,j)'s cost can have three possible effects on the shortest path:
  1. Arc(i,j) change does not affect the shortest path to node j.
  2. Arc(i,j) change improves the shortest path to node j.
  3. Arc(i,j) change invalidates the shortest path to node j.

1.) Arc(i,j) change does not affect shortest path

- The cost from node i to node j increases from 2 to 3.
- No changes are needed.

2.) Arc(i,j) change improves shortest path to j

- The cost from node i to node j decreases from 3 to 0.
- Propagate the improved shortest path.
3. **Arc(i,j) change invalidates shortest path to j**

- Increasing Arc(i,j) now invalidates node j's shortest path.
  - Reset node j
  - Recursively reset all nodes dependent upon node j.
  - Insert node j's parents into the queue so that a new path to node j can be found for node j and all other invalidated nodes.

---

**3 Update Rules to repair an Inconsistent distance graph**

- Now ITC must incrementally repair the inconsistency.

  - Three repair steps:
    1. Reset all nodes in negative cycle.
    2. Recursively reset all nodes that depend on the negative cycle nodes.
    3. Put any parent of a reset node that was not also reset on the Q.

---

**Temporal Plan Network**

(Time, Williams, Abrahmson 01)

- Conflict-Directed Plan Repair
  - Generate New Candidate Plan
  - Test Candidate Plan For Temporal Consistency
  - Incremental Updates
  - Inconsistency

- Redundant Methods

(Time, Williams, Abrahmson 01)
Conflict-Directed Plan Repair

![Diagram of a plan repair process with nodes and arcs representing states and transitions.]

**Conclusions**

- ITC is an incremental shortest path algorithm that can repair distance graphs incrementally as the plan changes.
- ITC’s Novel Claims:
  1. A conflict extraction mechanism
  2. Allow multiple arc-changes at once
  3. Can incrementally repair inconsistent distance graphs
- Shows an order of magnitude improvement over non-incremental planning
- Applicable to any plan representation that uses disjunctions of simple temporal constraints.

---

1. Start
2. beHomeBeforeDark()
3. imageTargets()
4. driveTo(p3)
5. driveTo(pL)
6. driveTo(p2)
7. driveTo(pR)
8. driveTo(p2)
9. beHomeBeforeDark()
10. imageTargets()
11. driveTo(p3)
12. driveTo(pL)
13. driveTo(p2)
14. driveTo(pR)
15. driveTo(p2)

Consistent!