Coordinating Agile Systems through the Model-based Execution of Temporal Plans

Thomas Léauté,
Brian C. Williams

July 11, 2005

Overview of the Presentation

1. **Introduction**
   - **Objective and Challenges**
   - **Previous Work and Innovations**
   - **Problem Statement**
2. Approach
3. Discussion

1. Introduction

Objective and Challenges

Objective: **task-directed coordinated control of agile dynamic systems**

- Challenges to address:
  - Under-actuated systems
  - Tight synchronization
  - Robustness to disturbances

1. Introduction

Previous Work

- Challenges to address:
  - Deal with under-actuation ⇒ reason in terms of state
  - Handle tight synchronization
  - Provide robustness
- Previous work:
  - **Model-based programming** (*Williams et al. 03*): State-level control of under-actuated discrete plants.

1. Introduction

Previous Work

- Challenges to address:
  - Deal with under-actuation
  - Handle tight synchronization ⇒ execute temporal plans
  - Provide robustness ⇒ use temporal flexibility & replan when necessary
- Previous work:
  - **Dispatchable plan execution** (*Vidal & Ghallab 96, Morris & Muscettola 98, Tsamardinos & Ramakrishnan 03*): Scheduling and execution of temporally flexible plans
  - **Continuous planning and execution** (*Ambros-Ingerson & Steel 88, Wilkins & Myers 95, Chien et al. 00*): Robust interleaved planning and execution of temporal plans; inspired by **Model Predictive Control**

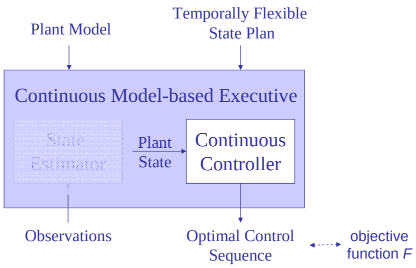
1. Introduction

Innovative Claim

- Model-based execution of temporally flexible state plans for continuous, under-actuated systems
- Technical Innovations:
  - Responds to disturbances by framing temporal state plan execution as **Model Predictive Control** (*Propoi 63, Richalet 76, How et al. 02*)
  - Achieves real-time performance through novel **constraint pruning policies**

1. Introduction

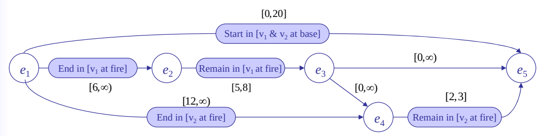
Continuous Model-based Execution (***CMEx***)



- Hofbaur, M. W. and Williams, B. C., Hybrid Estimation of Complex Systems, in *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 2004
- Blackmore, L., Funiak, S. and Williams, B.C., Combining Stochastic and Greedy Search in Hybrid Estimation, in *Proceedings of the 20th National Conference on Artificial Intelligence*, 2005

1. Introduction

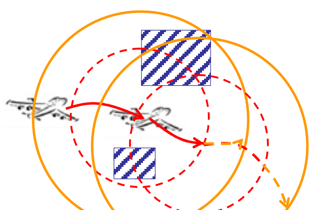
Temporally Flexible State Plan



2. Approach

Overall Approach

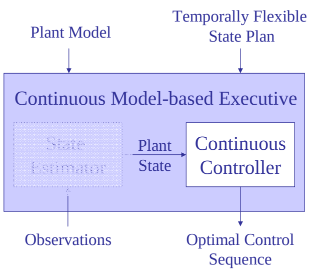
- ***Receding Horizon CMEx***:
  - Solving the full CMEx problem is intractable
  - Iteratively solve smaller versions of the problem
- Plan up to a small planning horizon N_t (e.g. 25 sec)
- Execute only up to an execution horizon n_t (e.g. 18 sec) and replan



Richalet, J. *et al*, Algorithmic control of industrial processes, in *Proceedings of the 4th IFAC Symposium on Identification and System Parameter Estimation*, 1976
Bellingham, J., Richards, A., How, J., Receding Horizon Control of Autonomous Aerial Vehicles, in *Proceedings of the American Control Conference*, 2002

2. Approach

Continuous Model-based Controller



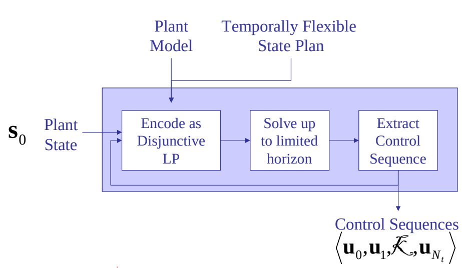
2. Approach

Continuous Model-based Controller

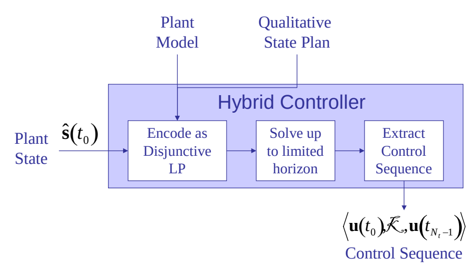
- Iteratively solve Receding Horizon CMEx

Formulate the problem as a **Disjunctive Linear Program** (DLP)

Egon Balas, Disjunctive Programming, in *Annals of Discrete Mathematics*, 1979



2. Approach



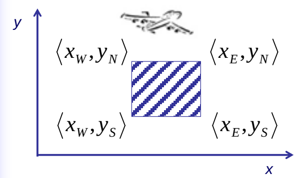
2. Approach

Disjunctive Linear Programming (DLP)

- In Conjunctive Normal Form (CNF):

$$\text{Minimize} \quad f(\mathbf{x})$$
$$\text{Subject to} \quad \bigwedge_{i=1..n} \left\{ \bigvee_{j=1..m} g_{i,j}(\mathbf{x}) \le c_{i,j} \right\}$$

Example in CNF:



$$\bigwedge_{t=0..N_t} \left\{ \begin{array}{l} x(t) \ge x_E \\ \vee \; x(t) \le x_W \\ \vee \; y(t) \ge y_N \\ \vee \; y(t) \le y_S \end{array} \right\}$$

Schouwenaars, T., De Moor, B., Féron, E. and How, J., Mixed Integer Programming for Multi-Vehicle Path Planning, *ECC*, 2001

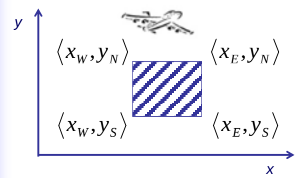
2. Approach

Disjunctive Linear Programming (DLP)

- In Conjunctive Normal Form (CNF):

$$\text{Minimize} \quad f(\mathbf{x})$$
$$\text{Subject to} \quad \bigwedge_{i=1..n} \left\{ \bigvee_{j=1..m} g_{i,j}(\mathbf{x}) \le c_{i,j} \right\}$$

Example in CNF:



$$\bigwedge_{t=0..N_t} \left\{ \begin{array}{l} x(t) \ge x_E \\ \vee \; x(t) \le x_W \\ \vee \; y(t) \ge y_N \\ \vee \; y(t) \le y_S \end{array} \right\}$$

Li, H. and Williams, B. C., Efficiently Solving Hybrid Logic/Optimization Problems through Generalized Conflict Learning, *ICAPS Workshop "Plan Execution: A Reality Check"*, 2005

2. Approach

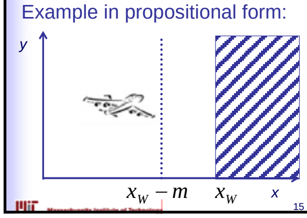
Disjunctive Linear Programming (DLP)

- In general propositional form:

$$\text{Minimize} \quad f(\mathbf{x})$$
$$\text{Subject to} \quad \Phi(\mathbf{x})$$

where: $\Phi(\mathbf{x}) := \Phi(\mathbf{x}) \wedge \Phi(\mathbf{x}) \mid \Phi(\mathbf{x}) \vee \Phi(\mathbf{x}) \mid \Phi(\mathbf{x}) \Rightarrow \Phi(\mathbf{x}) \mid \Phi(\mathbf{x}) \Leftrightarrow \Phi(\mathbf{x}) \mid \neg\Phi(\mathbf{x}) \mid g(\mathbf{x}) \le c$

Example in propositional form:

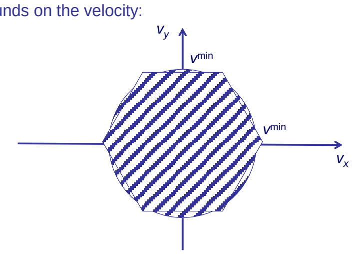


$$\bigwedge_{t=0..N_t} \left\{ \begin{array}{l} x(t) \le x_W - m \\ \Rightarrow v_x(t) \le v_x^{\max} \end{array} \right\}$$

2. Approach

DLP Encodings

- Plant model encodings (cont.):
  - Forbidden regions in the state space (cont.):
    - Bounds on the velocity:

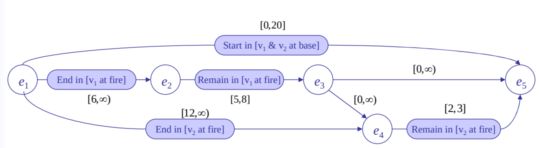


Schouwenaars, T., De Moor, B., Féron, E. and How, J., Mixed Integer Programming for Multi-Vehicle Path Planning, *ECC*, 2001

2. Approach

DLP Encodings

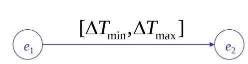
- State plan encodings:



2. Approach

DLP Encodings

- State plan encodings (cont.):
  - Time constraint between two events e_1 and e_2:

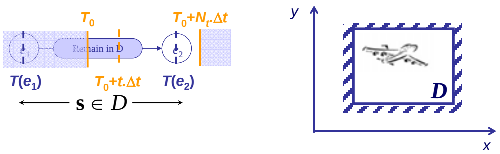


$$T(e_2) - T(e_1) \ge \Delta T_{\min}$$
$$\wedge \quad T(e_2) - T(e_1) \le \Delta T_{\max}$$

2. Approach

DLP Encodings

- State plan encodings (cont.):
  - Constraint associated with a *Remain in* activity:

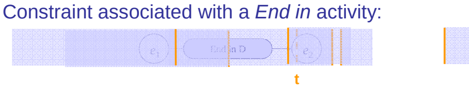


$$\bigwedge_{t=0..N_t} \left\{ \begin{array}{l} T(e_1) \le T_0 + t\cdot\Delta t \\ \wedge\ T(e_2) \ge T_0 + t\cdot \Delta t \end{array}\right\} \Rightarrow \mathbf{s}(t) \in D$$

2. Approach

DLP Encodings

- State plan encodings (cont.):
  - Constraint associated with a *End in* activity:



$$\bigvee_{t=0..N_t} \left\{ \begin{array}{l} T(e_2) \ge T_0 + (t-1/2)\cdot\Delta t \\ \wedge\ T(e_2)\le T_0+(t+1/2)\cdot\Delta t \\ \wedge\ \mathbf{s}_t \in D\end{array}\right\}$$

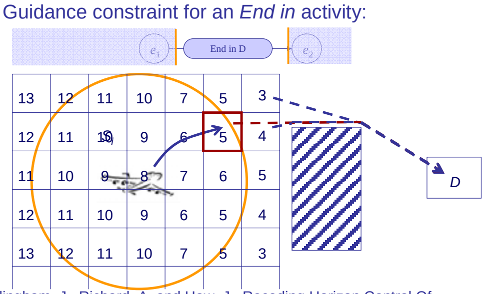
$$\vee\ T(e_2) \le T_0 - \Delta t/2$$

$$\vee\ T(e_2)\ge T_0+(N_t+1/2)\cdot\Delta t$$

2. Approach

DLP Encodings

- State plan encodings (cont.):
  - Guidance constraint for an *End in* activity:

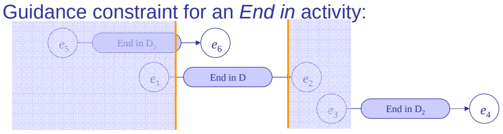


Bellingham, J., Richard, A. and How, J., Receding Horizon Control Of Autonomous Aerial Vehicles, *ACC*, 2002

2. Approach

DLP Encodings

- State plan encodings (cont.):
  - Guidance constraint for an *End in* activity:



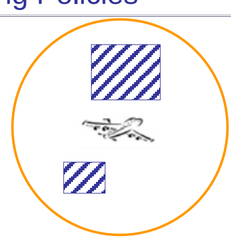
Minimize h

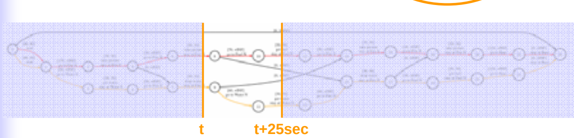
$$\left\{\begin{array}{l} T(e_1) < T_0 + n_t\cdot\Delta t \\ \wedge\ T(e_2) \ge T_0 + n_t \cdot \Delta t\end{array}\right\} \Rightarrow \bigvee_{S_i\subset S}\left\{\begin{array}{l} h = h_D(S_i) \\ \wedge\ \mathbf{s}(n_t)\in S_i\end{array}\right\}$$

2. Approach

Constraint Pruning Policies

- The DLPs can have a very large number of constraints
- Prune part of the search space to reduce the scope of the problem:
  - Spatial search space
  - Temporal search space

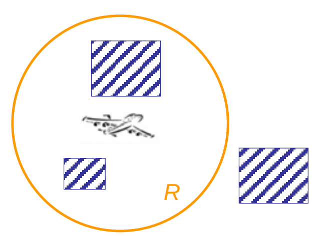




2. Approach

Constraint Pruning Policies

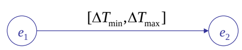
- Plant model constraint pruning:
  - Obstacle avoidance constraint pruning



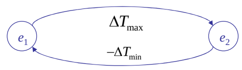
2. Approach

Constraint Pruning Policies

- State plan constraint pruning:
  - Initial graph corresponding to the state plan:



  - Corresponding *distance graph*:



- Run shortest path algorithms to infer absolute time bounds on any event: $T_e^{min} \le T(e) \le T_e^{max}$

Dechter, R., Meiri, I. and Pearl, J., Temporal Constraint Networks, *ACC*, 1991

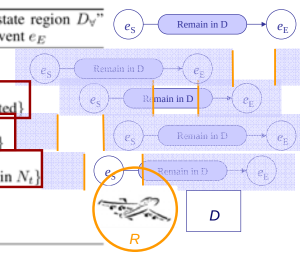
2. Approach

Constraint Pruning Policies

- State plan constraint pruning (cont.):
  - Pruning policy for the constraint on a *Remain in* activity:

Alg. 4 Pruning policy for a "Remain in state region $D_\forall$" activity starting at event e_S and ending at event e_E

```
if T_{e_E}^max < T_0 then
    prune {activity is completed}
else if T_{e_S}^max < T_0 then
    do not prune{activity is being executed}
else if T_{e_S}^min > T_0 + N_t · ΔT then
    prune {activity will start beyond N_t}
else if T_{e_S}^max < T_0 + N_t · ΔT then
    do not prune {activity will start within N_t}
else if R ∩ D_∀ = ∅ then
    prune; POSTPONE(e_S)
end if
```

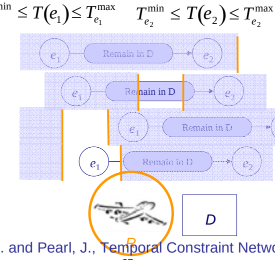


2. Approach

Constraint Pruning Policies

- State plan constraint pruning (cont.):
  - Pruning policy for the constraint on a *Remain in* activity:

$T_{e_1}^{min} \le T(e_1) \le T_{e_1}^{max}$ $\quad T_{e_2}^{min} \le T(e_2) \le T_{e_2}^{max}$



- $T_{e_2}^{max} < T_0$ — PRUNE
- $T_{e_1}^{max} < T_0$ — DOT NOT PRUNE
- $T_{e_1}^{min} > T_0 + N_t \cdot \Delta t$ — PRUNE
- $T_{e_2}^{max} < T_0 + N_t \cdot \Delta t$ — DOT NOT PRUNE
- PRUNE

Dechter, R., Meiri, I. and Pearl, J., Temporal Constraint Networks, *ACC*, 1991

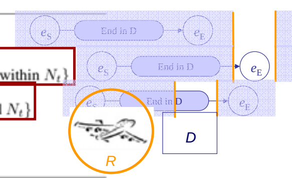
2. Approach

Constraint Pruning Policies

- State plan constraint pruning (cont.):
  - Pruning policy for the constraint on an *End in* activity:

Alg. 5 Pruning policy for a "End in state region D_E" activity ending at event e_E

```
if T_{e_E}^max < T_0 then
    prune {e_E has already occurred}
else if T_{e_E}^max ≤ T_0 + N_t · ΔT then
    do not prune {e_E will be scheduled within N_t}
else if T_{e_E}^min > T_0 + N_t · ΔT then
    prune {e_E will be scheduled beyond N_t}
else if R ∩ D_E = ∅ then
    prune; POSTPONE(e_E)
end if
```



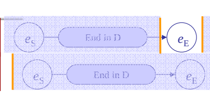
2. Approach

Constraint Pruning Policies

- State plan constraint pruning (cont.):
  - Pruning policy for the guidance constraint for an *End in* activity:

Alg. 6 Pruning policy for the guidance constraint for an "End in state region D_E" activity ending at event e_E

```
if T_{e_E}^max < T_0 + n_t · ΔT then
    prune {e_E will be scheduled within the horizon}
else if T_{e_S}^min ≥ T_0 + n_t · ΔT then
    prune {e_S will be scheduled beyond the horizon}
end if
```



Overview of the Presentation

1. Introduction
2. Approach
3. **Discussion**
   - **Fire-fighting UAV Demonstration**
   - **Other examples of Agile Systems**

3. Conclusion

Fire-fighting UAV Demonstration

Lake N

Fire N

1. Go to Lake S
2. Fill up water tank
3. Go to Fire S
4. Drop water over fire
5. Go to Lake N
6. Fill up water tank
7. Go to Fuel Station
8. Fill up fuel tank
9. Go to Fire N
10. Drop water over fire
11. Go back to Base

Fire S

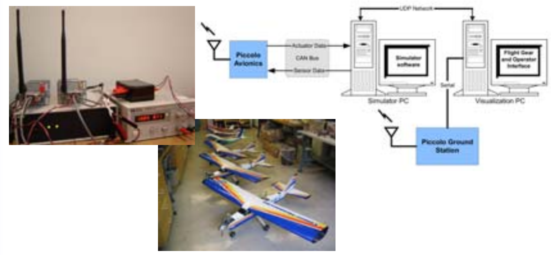
Lake S

31

3. Conclusion

Fire-fighting UAV Demonstration

- *CloudCap Simulator*: a real-time hardware-in-the-loop UAV simulation



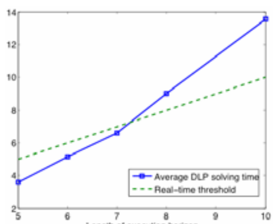
CloudCap Technologies (*www.cloudcaptech.com*)

32

3. Conclusion

Performance Analysis

- Input state plan:
  - 2 vehicles, 2 obstacles,
  - 26 activities,
  - Total execution time of 1300s
- Maintained a planning buffer of 10s



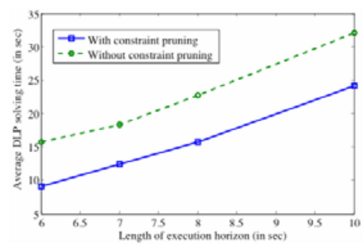
The model-based executive designs optimal control sequences in real-time for horizons < 7.3s

Above 7.3s, the control sequences are sub-optimal

33

3. Conclusion

Performance Analysis



34

3. Conclusion

Other Examples of Agile Systems

- Demonstrate the executive on other agile systems:
  - Wheeled exploratory ATRV rovers

  - Arm manipulators performing coordinated assembly tasks

35

3. Conclusion

Conclusion

- Model-based **execution of temporally flexible state plans** enables coordination of agile systems.
- **Real-time execution** is obtained by Model Predictive Control and pruning policies.
- Our executive has been demonstrated on a real-time hardware-in-the-loop UAV testbed.

36