Examples of Supervisory Interaction with Route Optimizers
Oliver Turnbull and Arthur Richards
oliver.turnbull@bristol.ac.uk
arthur.richards@bristol.ac.uk
Overview

• The SESAR Concept of Operations calls for:
  “Extensive use of automation support to reduce operator task load, but in which controllers remain in control as managers”

• Much work has been performed on trajectory optimization

• Rarely are humans included in the trajectory design process

• Part of SUPEROPT project whose goal is to:
  “Develop tools to facilitate interactions between humans and trajectory optimizers”

• Trajectory optimization can play a key role in automation support
Overview

• How do we facilitate supervisor interaction?
Overview

• **How** do we facilitate supervisor interaction?

  **Sense Constraints**

• **What** are sense constraints?

• **Why** are they useful?
Outline

1. MILP
   – Fast
   – Global optimum
   – Linearized
   – 3D dynamics model
   – Extension of sense constraints to 3D

2. Collocation with Polar Sets
   – Nonlinear model
   – More general problems, eg noise as cost
   – Explicit modelling of time
   – 4D obstacles

3. Conclusions
Assumptions

• 4-D trajectories (RBTs)
• Trajectories updated via data-link
• Free routing
MILP Obstacle Avoidance

- Approximate obstacle with multiple avoidance constraints

\[
\begin{align*}
 r_x(a_2, k_1) - r_x(a_1, k_1) & \geq D_x \\
r_x(a_2, k_1) - r_x(a_1, k_1) & \geq -D_x \\
r_y(a_2, k_1) - r_y(a_1, k_1) & \geq D_y \\
r_y(a_2, k_1) - r_y(a_1, k_1) & \geq -D_y
\end{align*}
\]

\[\forall k_1 \in \{1, \ldots, N_t\}, \forall k_2 \in \{1, \ldots, N_t\}: (k_1 \geq k_2)\]
MILP Obstacle Avoidance

- Approximate obstacle with multiple avoidance constraints
- Define “binary” variables that enable each avoidance constraint to be relaxed

\[ r_x(a_2, k_1) - r_x(a_1, k_1) \geq D_x + M b_a(a_1, a_2, k_1, k_2, 1) \]
\[ r_x(a_2, k_1) - r_x(a_1, k_1) \geq -D_x - M b_a(a_1, a_2, k_1, k_2, 2) \]
\[ r_y(a_2, k_1) - r_y(a_1, k_1) \geq D_y + M b_a(a_1, a_2, k_1, k_2, 3) \]
\[ r_y(a_2, k_1) - r_y(a_1, k_1) \geq -D_y - M b_a(a_1, a_2, k_1, k_2, 4) \]

\[ \forall k_1 \in \{1, \ldots, N_t\}, k_2 \in \{1, \ldots, N_t\}: (k_1 \geq k_2) \]
MILP Obstacle Avoidance

- Approximate obstacle with multiple avoidance constraints
- Define “binary” variables that enable each avoidance constraint to be relaxed
- Require at least one of the constraints to be enforced

\[
\begin{align*}
    r_x(a_2, k_1) - r_x(a_1, k_1) & \geq D_x + Mb_a(a_1, a_2, k_1, k_2, 1) \\
    r_x(a_2, k_1) - r_x(a_1, k_1) & \geq -D_x - Mb_a(a_1, a_2, k_1, k_2, 2) \\
    r_y(a_2, k_1) - r_y(a_1, k_1) & \geq D_y + Mb_a(a_1, a_2, k_1, k_2, 3) \\
    r_y(a_2, k_1) - r_y(a_1, k_1) & \geq -D_y - Mb_a(a_1, a_2, k_1, k_2, 4) \\
    \sum_{i=1}^{4} b_a(a_1, a_2, k_1, k_2, i) & \leq 3 \\
    \forall k_1 \in \{1, \ldots, N_t\}, k_2 \in \{1, \ldots, N_t\}: (k_1 \geq k_2)
\end{align*}
\]
MILP Sense Constraints

- We can force a trajectory to pass to one side of an obstacle by “freezing” the appropriate binary:

\[
\begin{align*}
    r_x(a_2, k_1) - r_x(a_1, k_1) & \geq D_x + Mb_a(a_1, a_2, k_1, k_2, 1) \\
    r_x(a_2, k_1) - r_x(a_1, k_1) & \geq -D_x - Mb_a(a_1, a_2, k_1, k_2, 2) \\
    r_y(a_2, k_1) - r_y(a_1, k_1) & \geq D_y + Mb_a(a_1, a_2, k_1, k_2, 3) \\
    r_y(a_2, k_1) - r_y(a_1, k_1) & \geq -D_y - Mb_a(a_1, a_2, k_1, k_2, 4)
\end{align*}
\]

\[
\sum_{i=1}^{4} b_a(a_1, a_2, k_1, k_2, i) \leq 3
\]

\[
b_a(a_1, a_2, k_1, k_2, 3) = 1
\]

\[\forall k_1 \in \{1, \ldots, N_t\}, k_2 \in \{1, \ldots, N_t\}: (k_1 \geq k_2)\]
Sense Constraints in ATC

Three Requirements:

1. **Binaries** to define relative position
2. **3-D dynamics model**: derived from BADA
3. **Resolve class of problem**: Fix in 1 or more dimensions.
   - The problem is to resolve **only** in a certain way
   - Other dimensions should not change
Sense Constraints in ATC

F002 over F001
(Unconstrained)

F001 over F002
Sense Constraints in ATC

- Vertical:
  - Common notion of *up/down*
- Horizontal
  - Direction (left/right) relative to heading
  - Define as *ahead/behind*
  - Enforced by applying the constraints at the present time and all future time-steps
Sense Constraints in ATC

F002 ahead of F001

(Unconstrained; 2D)

F002 behind F001
Sense Constraints in ATC

F002 ahead of F001
(Unconstrained)

F002 behind F001
Sense Constraints in ATC

F002 ahead of F001
(Unconstrained)

F002 behind F001
Sense Constraints in ATC

F002 ahead of F001
(Unconstrained)

F002 behind F001
Sense Constraints in ATC

F002 ahead of F001

(Unconstrained)

F002 behind F001
Multi-Sector Controller (MSC)
MSC - Input
MSC - Input

- 3 pairs of conflicting aircraft
MSC - Input

- 3 pairs of conflicting aircraft
MSC - Input

- 3 pairs of conflicting aircraft
- Select constraints
MSC – Input

- 3 pairs of conflicting aircraft
- Select constraints
- Relative Cost history
MSC – Input

- 3 pairs of conflicting aircraft
- Select constraints
- Relative Cost history
- Highlight specific aircraft
MSC - Input

- 3 pairs of conflicting aircraft
- Select constraints
- Relative Cost history
- Highlight specific aircraft
- Generate Plans
MSC – Step 1

- Conflict free trajectories in green
- Original trajectories in red
MSC – Step 1

- Highlight specific aircraft
- Alternative cost functions
MSC – Step 1

Trajectories now reach destinations earlier
MSC – Step 2

• Request horizontal resolution
 MSC – Step 2

• Request horizontal resolution
MSC – Step 2

- Request horizontal resolution
  - Increased cost (expected)
MSC – Step 3

• Vertical resolution
MSC – Step 3

- Vertical resolution
  - F042 over F036
MSC – Step 3

• Vertical resolution
  • F042 over F036
MSC – Step 4

• F036 over F042
MSC – Step 4

- F036 over F042
  - Large increased cost
MSC – Step 5

• F036 over F042
  • Large increased cost
• F039 over F062
  • Small increased cost
MSC – Step 5

- F036 over F042
  - Large increased cost
- F039 over F062
  - Small increased cost
MSC – Step 5

- F036 over F042
  - Large increased cost
- F039 over F062
  - Small increased cost
MSC – Step 6

• F036 over F042
  • Large increased cost
• F039 over F062
  • Small increased cost
MILP Summary

• Well established for trajectory optimization
  – Fast (typically < 1 sec)
  – Robust
  – Globally optimal

• Extended to allow input of user preference for conflict resolution

• Assumptions
  – Linearized dynamics/constraints/cost
Collocation with Polar Sets

• Generalization to a \textit{nonlinear} model is a logical step
  – Collocation method to model aircraft dynamics
  – Polar sets for obstacle avoidance
Collocation with Polar Sets

• First we find a point, $y$, that lies within the polar set of the obstacle:
  \[ x \notin R \iff y^T x \geq 1 \]
  \[ y \in R^0 \]

• Where $y$ becomes a decision variable
Collocation with Polar Sets

• First we find a point, $y$, that lies within the polar set of the obstacle:  
  \[ x \notin R \iff y^T x \geq 1 \]  
  \[ y \in R^0 \]

• Next we ensure that the aircraft remains outside of the obstacle (within the polar set) at all time-steps:
  \[
y(t)^T \begin{pmatrix} r(t) - r_{obs}(t) \\ t - t_{obs} \end{pmatrix} \geq 1 \quad \forall \ t \in \{2, \ldots, N_t\}
  \]
Polar Set Sense Constraints

- Equivalent to fixing MILP binaries
- Sense constraints imply removing vertices from the polar set:

\[
\{ t \mid N_a(y, 2, 0, 3, 1, 1, \ldots) \leq \tau \quad \forall \tau \in \{2, \ldots, N_t\} \}
\]

- Alternatively we can restrict the location of the point \( y_e(t) \) within the polar set, e.g., to cause \( a_1 \) to pass under an obstacle we would require:
4-D Obstacle

- 2 closed sectors
4-D Obstacle

- 2 closed sectors
- Trajectory avoids closed sector
4-D Obstacle

- 2 closed sectors
- Trajectory avoids closed sector
- Sector re-opened
- Trajectory resumes shortest path (through sector)
Collision Avoidance

- Planned for F001 (cyan line) – represented by series of temporal obstacles
- Planned for F002 (cyan line)
- Add F003 (blue-dotted line)
- Allows planning over independent time-scales
Collision Avoidance

- Planned for first aircraft (cyan line) – represented by series of temporal obstacles
- Planned for F002 (cyan line)
- Add third aircraft (F003)
- Require F003 to pass under F002
- Horizontal separation of F001
Collision Avoidance

- Planned for first aircraft (cyan line) – represented by series of temporal obstacles
- Planned for F002 (cyan line)
- Add third aircraft (F003)
- Require F003 to pass under F002
- Horizontal separation of F001
Collision Avoidance

• Planned for first aircraft (cyan line) – represented by series of temporal obstacles
• Planned for F002 (cyan line)
• Add third aircraft (F003)
• Require F003 to pass under F002
• Horizontal separation of F001
• Vertical separation (F003 under F002)
Collision Avoidance

- Planned for first aircraft (cyan line) – represented by series of temporal obstacles
- Planned for F002 (cyan line)
- Add third aircraft (F003)
- Require F003 to pass under other aircraft
- Horizontal separation of F001
- Vertical separation (F003 under F002)
Collision Avoidance

- Planned for first aircraft (cyan line) – represented by series of temporal obstacles
- Planned for F002 (cyan line)
- Add third aircraft (F003)
- Require F003 to pass under other aircraft
- Horizontal separation of F001
- Vertical separation (F003 under F002)
- Alternative sense (F003 over F002)
Conclusion

- Demonstrated **two models** that incorporate sense constraints
- Allows **intuitive human input** in terms of high-level decision making
- While still enabling the optimizer to do what it does best: **design efficient 4D trajectories** subject to avoidance constraints
Thanks!