Lecture 7¾, more on MILP

- New example: fundamental equations of control
- Example: no input
- Input control
- Linear dynamical systems
- Obstacles
- MILP formulation of the control problem
- Adding a constraint set
- MILP formulation with constraint set
- Applications: Eric Feron’s work at MIT / Georgia Tech

New example: fundamental equations of control

Motion of a system defined by: $x_{k+1} = Ax_k + Bu_k$

$x_k \in \mathbb{R}^2$	State of the system (for example position)
$A \in \mathbb{R}^{2\times 2}$	Dynamics of the system
$u_k \in \mathbb{R}$	Input of the system (your control)
$B \in \mathbb{R}^{2\times 1}$	Input matrix

Example: no input

Motion of a system defined by: $x_{k+1} = Ax_k$

$$x_1 = Ax_0$$
$$x_2 = Ax_1 = A^2x_0$$
$$x_3 = Ax_2 = A^3x_0, \text{ etc...}$$

For example if matrix A is a 45 degree rotation matrix:

$$A = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$$

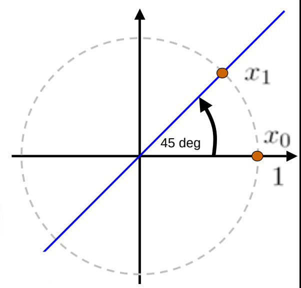
Example: no input

Motion of a system defined by: $x_{k+1} = Ax_k$

$$A = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$$

$$x_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$x_1 = Ax_0 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$



Example: no input

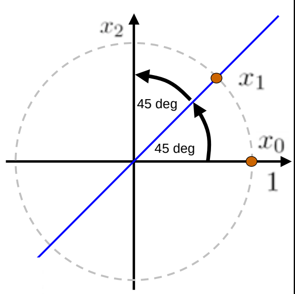
Motion of a system defined by: $x_{k+1} = Ax_k$

$$A = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$$

$$x_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$x_1 = Ax_0 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$x_2 = Ax_1 = A^2x_0 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$



Example: no input

Motion of a system defined by: $x_{k+1} = Ax_k$

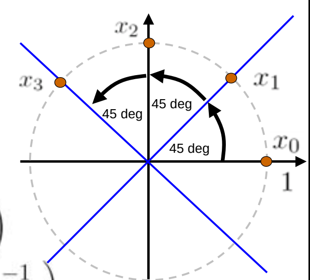
$$A = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$$

$$x_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$x_1 = Ax_0 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$x_2 = Ax_1 = A^2x_0 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$x_3 = Ax_2 = A^3x_0 = \frac{1}{\sqrt{2}}\begin{pmatrix} -1 \\ 1 \end{pmatrix}$$



Example: no input

Motion of a system defined by: $x_{k+1} = Ax_k$

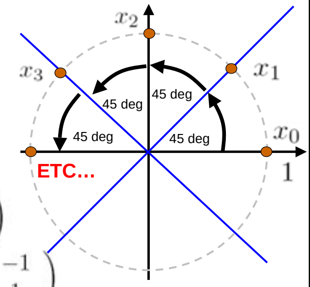
$$A = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$$

$$x_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$x_1 = Ax_0 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$x_2 = Ax_1 = A^2 x_0 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$x_3 = Ax_2 = A^3 x_0 = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$



Example: no input

Motion of a system defined by: $x_{k+1} = Ax_k$

$$x_1 = Ax_0$$
$$x_2 = Ax_1 = A^2 x_0$$
$$x_3 = Ax_2 = A^3 x_0, \text{ etc...}$$

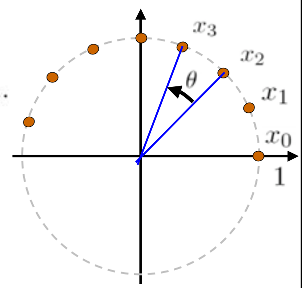
More generally, A can be a rotation matrix (angle theta)

$$A = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

Example: no input

Motion of a system defined by: $x_{k+1} = Ax_k$

$$x_1 = Ax_0$$
$$x_2 = Ax_1 = A^2 x_0$$
$$x_3 = Ax_2 = A^3 x_0, \text{ etc...}$$



$$A = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

Input control

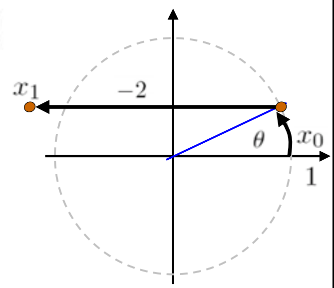
Motion of a system defined by: $x_{k+1} = Ax_k + Bu_k$

$$A = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

$$B = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$x_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

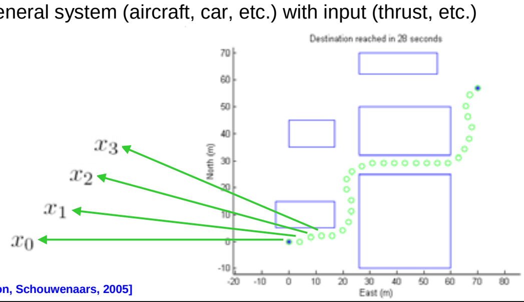
$$u_0 = -2$$



Linear dynamical systems

For general matrices A and B, $x_{k+1} = Ax_k + Bu_k$

This framework can be used to model the motion of a general system (aircraft, car, etc.) with input (thrust, etc.)

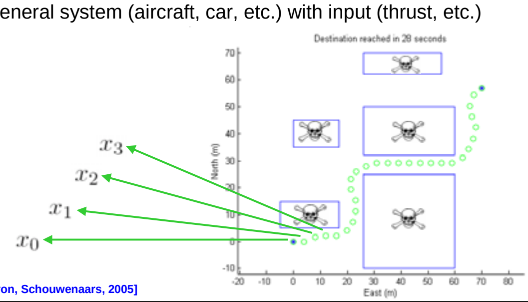


[Feron, Schouwenaars, 2005]

But what should we do about obstacles?

For general matrices A and B, $x_{k+1} = Ax_k + Bu_k$

This framework can be used to model the motion of a general system (aircraft, car, etc.) with input (thrust, etc.)

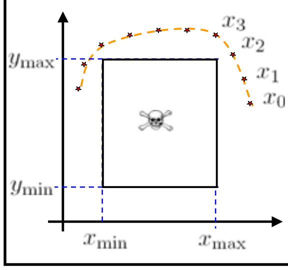


[Feron, Schouwenaars, 2005]

But what should we do about obstacles?

For general matrices A and B, $x_{k+1} = Ax_k + Bu_k$

This framework can be used to model the motion of a general system (aircraft, car, etc.) with input (thrust, etc.)



$$\begin{aligned} x_k^1 &\le x_{\min} + Kb_{k,1} \\ -x_k^1 &\le -x_{\max} + Kb_{k,2} \\ x_k^2 &\le y_{\min} + Kb_{k,3} \\ -x_k^2 &\le -y_{\max} + Kb_{k,4} \\ \sum_{i=1}^4 b_{k,i} &\le 3 \\ b_{k,i} &\in \{0,1\} \end{aligned}$$

[Feron, Schouwenaars, 2005]

Expression of these constraints as a MILP

Introduce vector containing variables (real and integer)

$$X_k = \begin{pmatrix} x_k \\ b_k \end{pmatrix}, \quad x_k \in \mathbb{R}^2, \quad b_{k,i} \in \{0,1\}^2$$

These constraints are linear:

$$Mx_k + Nb_k \le R \quad \left\{ \begin{aligned} x_k^1 &\le x_{\min} + Kb_{k,1} \\ -x_k^1 &\le -x_{\max} + Kb_{k,2} \\ x_k^2 &\le y_{\min} + Kb_{k,3} \\ -x_k^2 &\le -y_{\max} + Kb_{k,4} \\ \sum_{i=1}^4 b_{k,i} &\le 3 \\ b_{k,i} &\in \{0,1\} \end{aligned} \right.$$

They involve real and integer numbers: they are MILP !!!

MILP formulation

As much on the left as possible: minimizes the first component of the vector at the last step (step T)

$$\begin{array}{lll} \textbf{min:} & (1,0)\cdot x_T & \\ \textbf{s.t.} & x_{k+1} = Ax_k + Bu_k & \text{for all } k \in \{1,\cdots T\} \\ & Mx_k + Nb_k \le R & \text{for all } k \in \{1,\cdots T\} \\ & x_k \in \mathbb{R}^2 & \text{for all } k \in \{1,\cdots T\} \\ & u_k \in U & \text{for all } k \in \{1,\cdots T\} \\ & b_k \in \{0,1\}^4 & \text{for all } k \in \{1,\cdots T\} \\ & x_0 = x_{\text{start}} & \text{given} \end{array}$$

MILP formulation

Satisfies the dynamics at every step

Steps 1 to T

$$\begin{array}{lll} \textbf{min:} & (1,0)\cdot x_T & \\ \textbf{s.t.} & x_{k+1} = Ax_k + Bu_k & \text{for all } k \in \{1,\cdots T\} \\ & Mx_k + Nb_k \le R & \text{for all } k \in \{1,\cdots T\} \\ & x_k \in \mathbb{R}^2 & \text{for all } k \in \{1,\cdots T\} \\ & u_k \in U & \text{for all } k \in \{1,\cdots T\} \\ & b_k \in \{0,1\}^4 & \text{for all } k \in \{1,\cdots T\} \\ & x_0 = x_{\text{start}} & \text{given} \end{array}$$

MILP formulation

At every step, the system avoids the obstacle (MILP)

$$\begin{array}{lll} \textbf{min:} & (1,0)\cdot x_T & \\ \textbf{s.t.} & x_{k+1} = Ax_k + Bu_k & \text{for all } k \in \{1,\cdots T\} \\ & Mx_k + Nb_k \le R & \text{for all } k \in \{1,\cdots T\} \\ & x_k \in \mathbb{R}^2 & \text{for all } k \in \{1,\cdots T\} \\ & u_k \in U & \text{for all } k \in \{1,\cdots T\} \\ & b_k \in \{0,1\}^4 & \text{for all } k \in \{1,\cdots T\} \\ & x_0 = x_{\text{start}} & \text{given} \end{array}$$

MILP formulation

The system lives in two dimensional space (could be three dimensional space or different space)

$$\begin{array}{lll} \textbf{min:} & (1,0)\cdot x_T & \\ \textbf{s.t.} & x_{k+1} = Ax_k + Bu_k & \text{for all } k \in \{1,\cdots T\} \\ & Mx_k + Nb_k \le R & \text{for all } k \in \{1,\cdots T\} \\ & x_k \in \mathbb{R}^2 & \text{for all } k \in \{1,\cdots T\} \\ & u_k \in U & \text{for all } k \in \{1,\cdots T\} \\ & b_k \in \{0,1\}^4 & \text{for all } k \in \{1,\cdots T\} \\ & x_0 = x_{\text{start}} & \text{given} \end{array}$$

MILP formulation

The control evolves in a set U. For example, if the control is bounded (limited input), U is bounded. U can be a polygon

$$\begin{array}{lll}
\textbf{min:} & (1,0)\cdot x_T & \\
\textbf{s.t.} & x_{k+1} = Ax_k + Bu_k & \text{for all } k \in \{1, \cdots T\} \\
& Mx_k + Nb_k \leq R & \text{for all } k \in \{1, \cdots T\} \\
& x_k \in \mathbb{R}^2 & \text{for all } k \in \{1, \cdots T\} \\
& u_k \in U & \text{for all } k \in \{1, \cdots T\} \\
& b_k \in \{0,1\}^4 & \text{for all } k \in \{1, \cdots T\} \\
& x_0 = x_{\text{start}} & \text{given}
\end{array}$$

MILP formulation

The usual decision variables

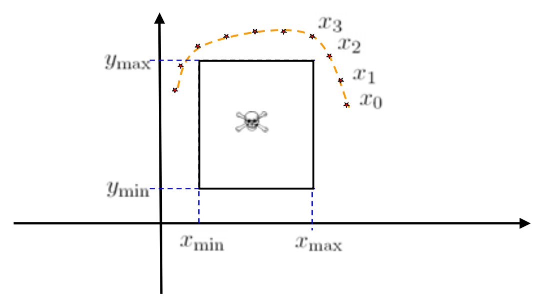
$$\begin{array}{lll}
\textbf{min:} & (1,0)\cdot x_T & \\
\textbf{s.t.} & x_{k+1} = Ax_k + Bu_k & \text{for all } k \in \{1, \cdots T\} \\
& Mx_k + Nb_k \leq R & \text{for all } k \in \{1, \cdots T\} \\
& x_k \in \mathbb{R}^2 & \text{for all } k \in \{1, \cdots T\} \\
& u_k \in U & \text{for all } k \in \{1, \cdots T\} \\
& b_k \in \{0,1\}^4 & \text{for all } k \in \{1, \cdots T\} \\
& x_0 = x_{\text{start}} & \text{given}
\end{array}$$

MILP formulation

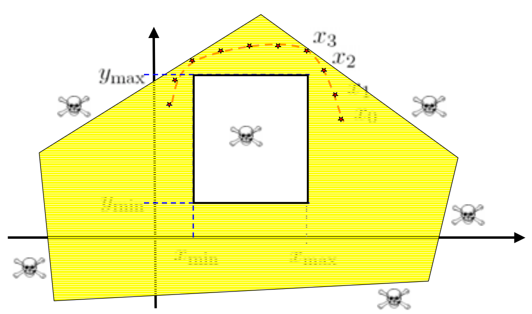
Starting point

$$\begin{array}{lll}
\textbf{min:} & (1,0)\cdot x_T & \\
\textbf{s.t.} & x_{k+1} = Ax_k + Bu_k & \text{for all } k \in \{1, \cdots T\} \\
& Mx_k + Nb_k \leq R & \text{for all } k \in \{1, \cdots T\} \\
& x_k \in \mathbb{R}^2 & \text{for all } k \in \{1, \cdots T\} \\
& u_k \in U & \text{for all } k \in \{1, \cdots T\} \\
& b_k \in \{0,1\}^4 & \text{for all } k \in \{1, \cdots T\} \\
& x_0 = x_{\text{start}} & \text{given}
\end{array}$$

How about adding a constraint set?



How about adding a constraint set?



Requires that at every step, the system is inside the yellow polygon: add 5 more inequality constraints at every time step!!!!

MILP formulation

For every time step k between 1 and T, add 5 constraints on the state of the system to force it to stay inside the yellow polygon

$$\begin{array}{lll}
\textbf{min:} & (1,0)\cdot x_T & \\
\textbf{s.t.} & x_{k+1} = Ax_k + Bu_k & \text{for all } k \in \{1, \cdots T\} \\
& Mx_k + Nb_k \leq R & \text{for all } k \in \{1, \cdots T\} \\
& x_k \in \mathbb{R}^2 & \text{for all } k \in \{1, \cdots T\} \\
& u_k \in U & \text{for all } k \in \{1, \cdots T\} \\
& b_k \in \{0,1\}^4 & \text{for all } k \in \{1, \cdots T\} \\
& x_0 = x_{\text{start}} & \text{given} \\
& Cx_k \leq D & \text{for all } k \in \{1, \cdots T\}
\end{array}$$

MILP formulation

For every time step k between 1 and T, add 5 constraints on the state of the system to force it to stay inside the yellow polygon

Outside the square white set (MILP)

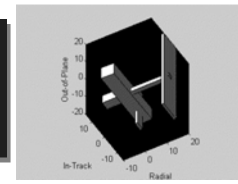
AND

Inside the yellow set: Linear constraints

$$
\begin{array}{lll}
\textbf{min:} & (1,0)\cdot x_T & \\
\textbf{s.t.} & x_{k+1} = Ax_k + Bu_k & \text{for all } k \in \{1,\cdots T\} \\
& Mx_k + Nb_k \le R & \text{for all } k \in \{1,\cdots T\} \\
& x_k \in \mathbb{R}^2 & \text{for all } k \in \{1,\cdots T\} \\
& u_k \in U & \text{for all } k \in \{1,\cdots T\} \\
& b_k \in \{0,1\}^4 & \text{for all } k \in \{1,\cdots T\} \\
& x_0 = x_{start} & \text{given} \\
& Cx_k \le D & \text{for all } k \in \{1,\cdots T\}
\end{array}
$$

Application (MIT / Georgia Tech: Eric Feron)

- Rovers:
  - Mars/Moon exploration,
  - inspection of nuclear waste sides,
  - automated highways …
- **A**utonomous **U**nderwater **V**ehicles:
  - coast guard support,
  - oceanographic research …
- Spacecraft:
  - ISS inspection camera,
  - distributed satellites
  - autonomous docking …
- Air Traffic Control

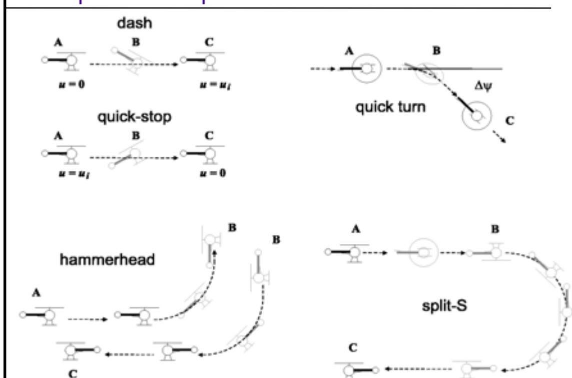


→ All require some form of **Trajectory Optimization**

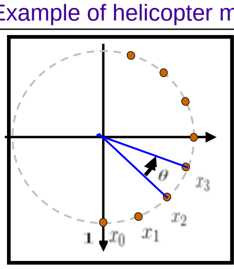
Other vehicles for potential implementations

Vision onboard the helicopter

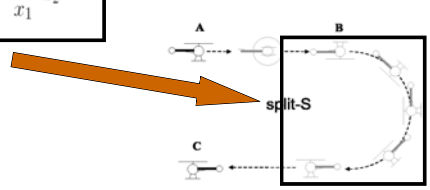
Example of helicopter maneuvers



Example of helicopter maneuvers

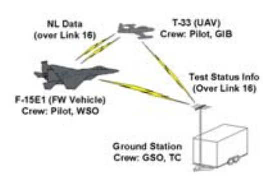


$$
\begin{array}{lll}
\textbf{min:} & (1,0)\cdot x_T & \\
\textbf{s.t.} & x_{k+1} = Ax_k + Bu_k & \text{for all } k \in \{1,\cdots T\} \\
& Mx_k \le R & \text{for all } k \in \{1,\cdots T\} \\
& Nb_k \le P & \text{for all } k \in \{1,\cdots T\} \\
& x_k \in \mathbb{R}^2 & \text{for all } k \in \{1,\cdots T\} \\
& u_k \in U & \text{for all } k \in \{1,\cdots T\} \\
& b_k \in \{0,1\}^4 & \text{for all } k \in \{1,\cdots T\} \\
& x_0 = x_{start} & \text{given}
\end{array}
$$



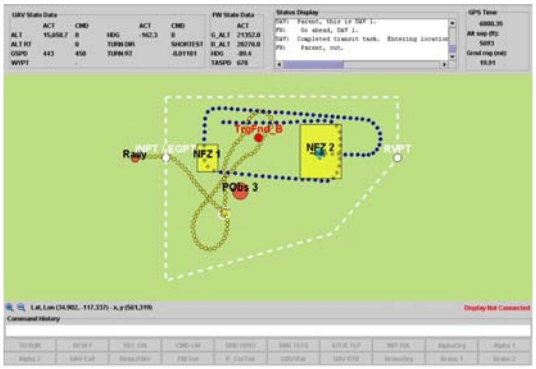
Was actually implemented on a T33 and F15

- **MILP module** integrated with Boeing's **OCP platform**:
  - Runs on laptop installed in T33 (Pentium 4, 2.4 GHz)
  - Send and receive user-defined data between F15 and T33 using Link-16 communications interface
  - Receive current vehicle state data
  - Send set of pre-defined commands to the T33
    - Set and Hold Speed
    - Set and Hold Turn Rate
    - Set and Hold Heading …
  - Hard real-time execution



[IEEE Aerospace, Submitted to JGCD]

Results



Results