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: \( x_{k+1} = Ax_k \)

\[
\begin{align*}
x_1 &= Ax_0 \\
x_2 &= Ax_1 = A^2 x_0 \\
x_3 &= Ax_2 = A^3 x_0, \text{ etc...}
\end{align*}
\]

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: \( x_{k+1} = Ax_k \)

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

\[
x_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}
\]

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

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

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

ETC...

Example: no input

Motion of a system defined by: \( x_{k+1} = Ax_k \)

\[
x_1 = Ax_0
\]

\[
x_2 = Ax_1 = A^2x_0
\]

\[
x_3 = Ax_2 = A^3x_0, \text{ etc}...
\]

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: \( x_{k+1} = Ax_k \)

\[
x_1 = Ax_0
\]

\[
x_2 = Ax_1 = A^2x_0
\]

\[
x_3 = Ax_2 = A^3x_0, \text{ etc}...
\]

Input control

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

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

\[
B = \begin{pmatrix} 1 \\ 0 \end{pmatrix}
\]

\[
x_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}
\]

\[
a_0 = -2
\]

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.)

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.)
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.)

Expression of these constraints as a MILP

Introduce vector containing variables (real and integer)

\[
X_k = \begin{pmatrix} x_k \\ b_k \end{pmatrix}, \quad x_k \in \mathbb{R}^2, \quad b_{k,i} \in \{0, 1\}^3
\]

These constraints are linear:

\[
M x_k + N b_k \leq R
\]

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)

\[
\begin{aligned}
\text{min} & \quad (1, 0) \cdot x_T \\
\text{s.t.} & \quad (1, 0) \cdot x_k = Ax_k + Bu_k \quad \text{for all } k \in \{1, \ldots, T\} \\
& \quad M x_k + N b_k \leq R \quad \text{for all } k \in \{1, \ldots, T\} \\
& \quad x_k \in \mathbb{R}^2 \quad \text{for all } k \in \{1, \ldots, T\} \\
& \quad u_k \in U \quad \text{for all } k \in \{1, \ldots, T\} \\
& \quad b_{k,i} \in \{0, 1\}^4 \quad \text{for all } k \in \{1, \ldots, T\} \\
& \quad x_0 = x_{\text{start}} \quad \text{given}
\end{aligned}
\]

Satisfies the dynamics at every step

\[
\begin{aligned}
\text{min} & \quad (1, 0) \cdot x_T \\
\text{s.t.} & \quad (1, 0) \cdot x_k = Ax_k + Bu_k \quad \text{for all } k \in \{1, \ldots, T\} \\
& \quad M x_k + N b_k \leq R \quad \text{for all } k \in \{1, \ldots, T\} \\
& \quad x_k \in \mathbb{R}^2 \quad \text{for all } k \in \{1, \ldots, T\} \\
& \quad u_k \in U \quad \text{for all } k \in \{1, \ldots, T\} \\
& \quad b_{k,i} \in \{0, 1\}^4 \quad \text{for all } k \in \{1, \ldots, T\} \\
& \quad x_0 = x_{\text{start}} \quad \text{given}
\end{aligned}
\]

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

\[
\begin{aligned}
\text{min} & \quad (1, 0) \cdot x_T \\
\text{s.t.} & \quad (1, 0) \cdot x_k = Ax_k + Bu_k \quad \text{for all } k \in \{1, \ldots, T\} \\
& \quad M x_k + N b_k \leq R \quad \text{for all } k \in \{1, \ldots, T\} \\
& \quad x_k \in \mathbb{R}^2 \quad \text{for all } k \in \{1, \ldots, T\} \\
& \quad u_k \in U \quad \text{for all } k \in \{1, \ldots, T\} \\
& \quad b_{k,i} \in \{0, 1\}^4 \quad \text{for all } k \in \{1, \ldots, T\} \\
& \quad x_0 = x_{\text{start}} \quad \text{given}
\end{aligned}
\]

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

\[
\begin{aligned}
\text{min} & \quad (1, 0) \cdot x_T \\
\text{s.t.} & \quad (1, 0) \cdot x_k = Ax_k + Bu_k \quad \text{for all } k \in \{1, \ldots, T\} \\
& \quad M x_k + N b_k \leq R \quad \text{for all } k \in \{1, \ldots, T\} \\
& \quad x_k \in \mathbb{R}^2 \quad \text{for all } k \in \{1, \ldots, T\} \\
& \quad u_k \in U \quad \text{for all } k \in \{1, \ldots, T\} \\
& \quad b_{k,i} \in \{0, 1\}^4 \quad \text{for all } k \in \{1, \ldots, T\} \\
& \quad x_0 = x_{\text{start}} \quad \text{given}
\end{aligned}
\]
The control evolves in a set $U$. For example, if the control is bounded (limited input), $U$ is bounded. $U$ can be a polygon.

The usual decision variables:

\[
\begin{align*}
\text{min:} & \quad (1, 0) \cdot x^T \\
\text{s.t.:} & \quad x_{k+1} = Ax_k + Bu_k \quad \text{for all } k \in [1, \ldots, T] \\
& \quad Mx_k + Nb_k \leq R \quad \text{for all } k \in [1, \ldots, T] \\
& \quad x_k \in \mathbb{R}^2 \quad \text{for all } k \in [1, \ldots, T] \\
& \quad u_k \in U \quad \text{for all } k \in [1, \ldots, T] \\
& \quad b_k \in \{0, 1\}^1 \quad \text{for all } k \in [1, \ldots, T] \\
& \quad x_0 = x_{\text{start}} \quad \text{given}
\end{align*}
\]

How about adding a constraint set?

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.

\[
\begin{align*}
\text{min:} & \quad (1, 0) \cdot x^T \\
\text{s.t.:} & \quad x_{k+1} = Ax_k + Bu_k \quad \text{for all } k \in [1, \ldots, T] \\
& \quad Mx_k + Nb_k \leq R \quad \text{for all } k \in [1, \ldots, T] \\
& \quad x_k \in \mathbb{R}^2 \quad \text{for all } k \in [1, \ldots, T] \\
& \quad u_k \in U \quad \text{for all } k \in [1, \ldots, T] \\
& \quad b_k \in \{0, 1\}^1 \quad \text{for all } k \in [1, \ldots, T] \\
& \quad x_0 = x_{\text{start}} \quad \text{given} \\
& \quad x_k \leq D \quad \text{for all } k \in [1, \ldots, T]
\end{align*}
\]

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:

\[
\begin{align*}
\text{minimize:} & \quad (1.10) \cdot x_f \\
\text{subject to:} & \quad x_{k+1} = A(x_k + B) u_k, \quad \text{for all } k \in [1, \ldots, T] \\
& \quad M_k + N_k u_k \preceq P, \quad \text{for all } k \in [1, \ldots, T] \\
& \quad x_k \in \mathcal{K}, \quad \text{for all } k \in [1, \ldots, T] \\
& \quad u_k \in \mathcal{U}, \quad \text{for all } k \in [1, \ldots, T] \\
& \quad \beta_k \in [0,1]^k, \quad \text{given for all } k \in [1, \ldots, T]
\end{align*}
\]

Outside the square white set (MILP)

AND

Inside the yellow set: Linear constraints

Application (MIT / Georgia Tech: Eric Feron)

- **Rovers:**
  - Mars/Moon exploration,
  - inspection of nuclear waste sites,
  - automated highways
- **Autonomous Underwater Vehicles:**
  - 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

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]