

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:

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

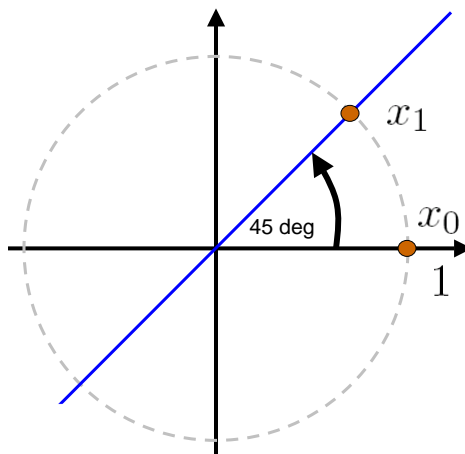
Example: no input

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

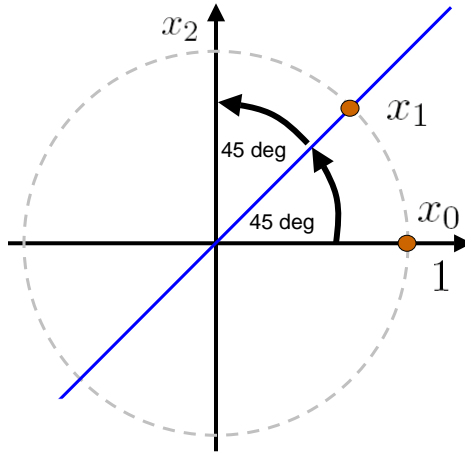
Motion of a system defined by: $x_{k+1} = Ax_k$

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

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

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

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



Example: no input

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

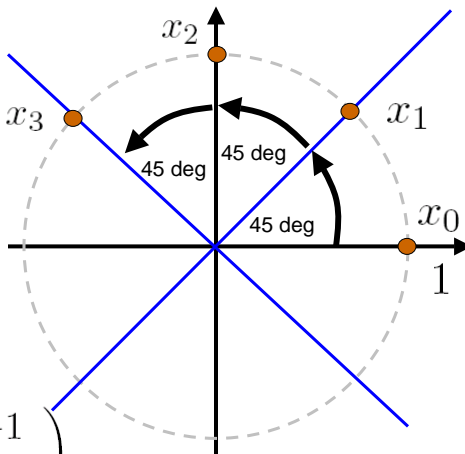
$$A = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$$

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

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

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

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



Example: no input

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

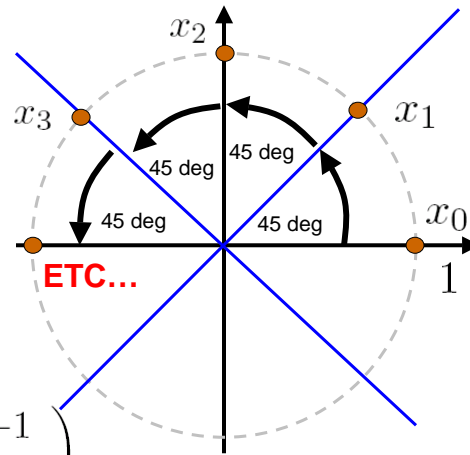
$$A = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$$

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

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

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

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



Example: no input

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

$$x_1 = Ax_0$$

$$x_2 = Ax_1 = A^2x_0$$

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

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

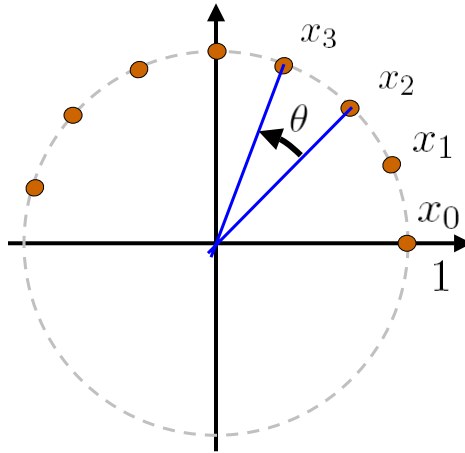
$$A = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$$

Example: no input

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

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

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



Input control

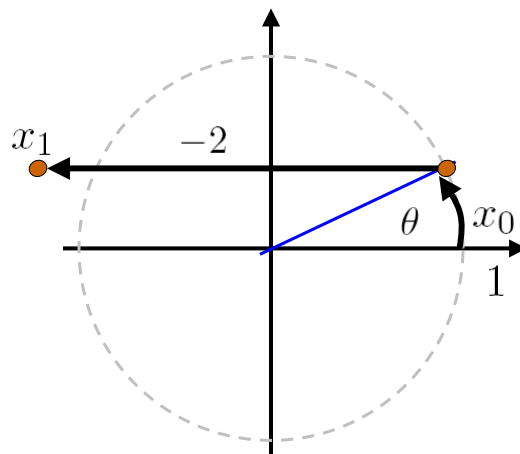
Motion of a system defined by: $x_{k+1} = Ax_k + Bu_k$

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

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

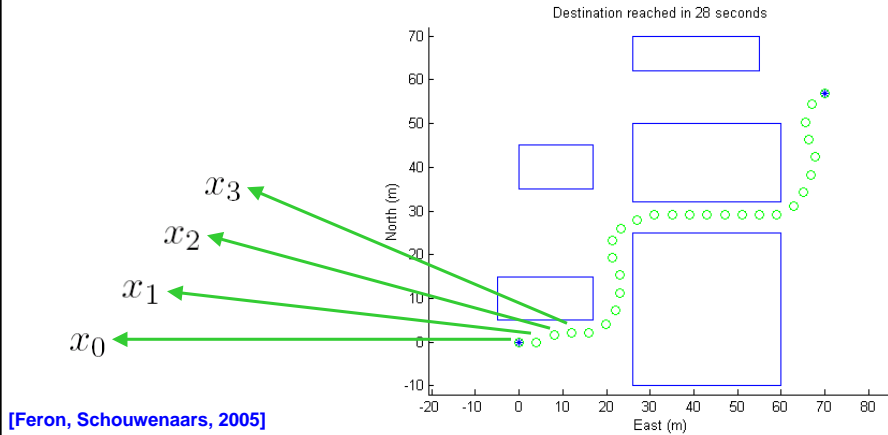
$$x_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$u_0 = -2$$



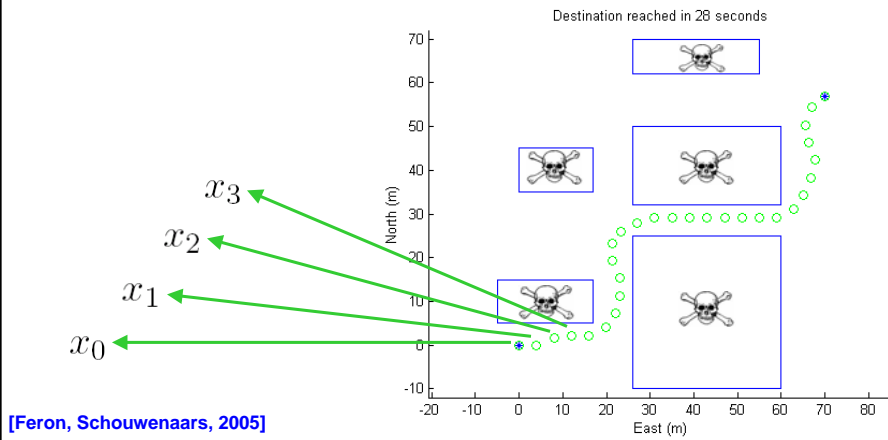
Linear dynamical systems

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.)



But what should we do about obstacles?

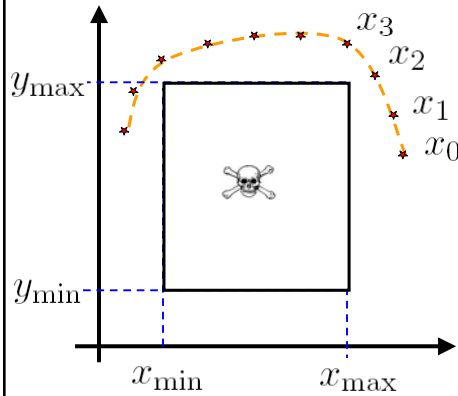
For general matrices A and B, $x_{k+1} = Ax_k + Bu_k$
This framework can be used to model the motion of a general system (aircraft, car, etc.) with input (thrust, etc.)



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.)



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

[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:

$$Mx_k + Nb_k \leq R$$

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

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

MILP formulation

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

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

MILP formulation

Satisfies the dynamics at every step

Steps 1 to T

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

MILP formulation

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

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

MILP formulation

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

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

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

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

MILP formulation

The usual decision variables

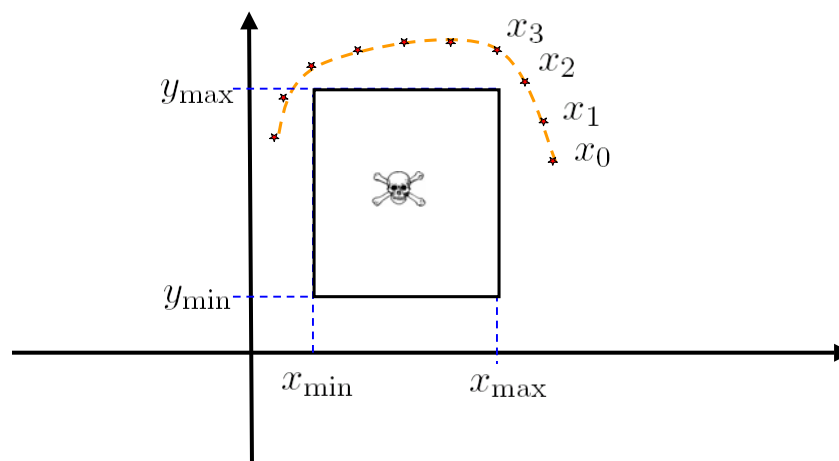
$$\begin{array}{ll} \mathbf{min} & (1, 0) \cdot x_T \\ \mathbf{s.t.} & x_{k+1} = Ax_k + Bu_k \quad \text{for all } k \in \{1, \dots, T\} \\ & Mx_k + Nb_k \leq R \quad \text{for all } k \in \{1, \dots, T\} \\ & x_k \in \mathbb{R}^2 \quad \text{for all } k \in \{1, \dots, T\} \\ & u_k \in U \quad \text{for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

MILP formulation

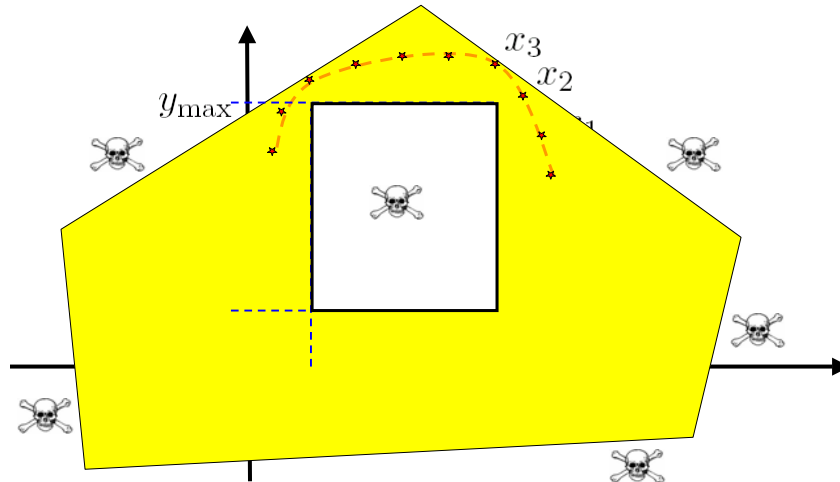
Starting point

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

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

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

MILP formulation

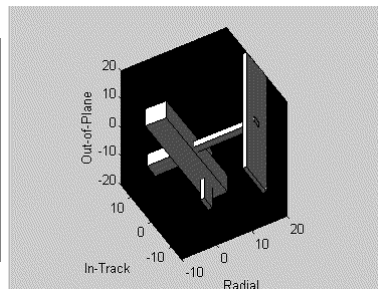
For every time step k between 1 and T , add 5 constraints on the state of the system to force it to stay inside the yellow polygon

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

Outside the square white set (MILP) \rightarrow $x_{k+1} = Ax_k + Bu_k$
 \rightarrow $Mx_k + Nb_k \leq R$
AND
 Inside the yellow set:
 Linear constraints \rightarrow $Cx_k \leq D$

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

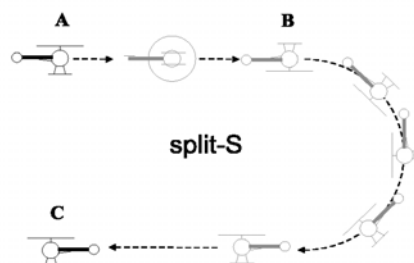
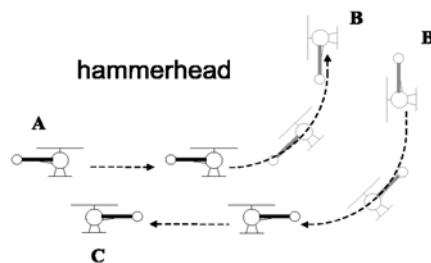
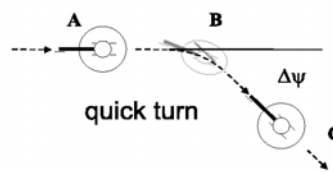
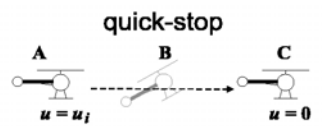
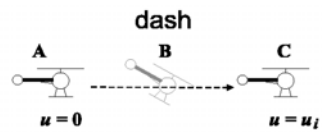
Other vehicles for potential implementations



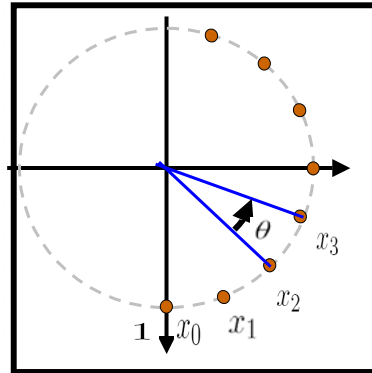
Vision onboard the helicopter



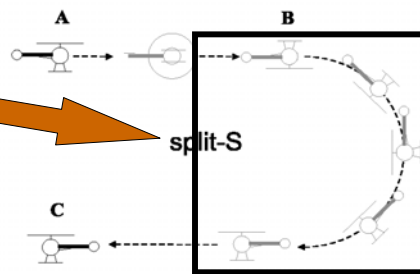
Example of helicopter maneuvers



Example of helicopter maneuvers



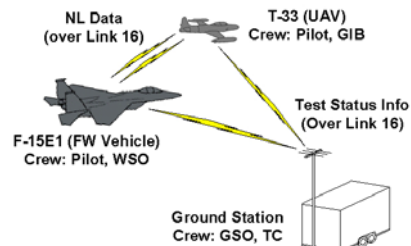
$$\begin{aligned} \text{min: } & (1, 0) \cdot x_T \\ \text{s.t. } & x_{k+1} = Ax_k + Bu_k \quad \text{for all } k \in \{1, \dots, T\} \\ & Mx_k \leq R \quad \text{for all } k \in \{1, \dots, T\} \\ & Nb_k \leq P \quad \text{for all } k \in \{1, \dots, T\} \\ & x_k \in \mathbb{R}^2 \quad \text{for all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



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

UAV State Data				FW State Data				Status Display	GPS Time
ACT	CMD	ACT	CMD	ACT	CMD	ACT		6800.35	
ALT	15,658.7	0	HDG	-162.3	0	G_ALT	21352.0	Alt sep (ft):	
ALT RT	0	0	TURN DIR	SHORTEST	0	B_ALT	20276.0	5693	
GSPD	443	450	TURN RT	-0.01101		HDG	-89.4	Grnd rng (mi):	
WYPI						TASPD	678	19.91	

Status Display
 UAV: Parent, this is UAV 1.
 FW: Go ahead, UAV 1.
 UAV: Completed transit task. Entering location
 FW: Parent, out.

GPS Time: 6800.35
 Alt sep (ft): 5693
 Grnd rng (mi): 19.91

INPT EGPT RVPT
 NFZ 1 NFZ 2
 POBs 1 POBs 2 POBs 3

Lat, Lon (34.902, -117.337) - x, y (561,310) Display Not Connected

Command History

ID	RUN	RESET	REC ON	CMD ON	SND ORST	SND TGTS	ATCR TGT	INR FDL	AlphaOrg	Alpha 1
Alpha 2	UAV Call	Resp2UAV	FW Out	P_CarTsk	UAVStat	UAV RTB	BravoOrg	Bravo 1	Bravo 2	

Results

UAV State Data				FW State Data				Status Display	Mission State
ACT	CMD	ACT	CMD	ACT	CMD	ACT		6800.35	
ALT	14938.3	15000.0	HDG	-162.3	0	G_ALT	21352.0	Alt sep (ft):	
ALT RT	0	0	TURN DIR	SHORTEST	0	B_ALT	20276.0	5693	
GSPD	443	450	TURN RT	-0.01101		HDG	-89.4	Grnd rng (mi):	
WYPI						TASPD	678	19.91	

Status Display
 FW: Wait 15 your status, UAV 1?
 UAV: Current status 8 miles from location base, etc.
 FW: What is your status, UAV 1?
 UAV: Current status 4 miles from location base, etc.
 UAV: 2 miles to task location.
 UAV: Completed transit task. Entering location base to perform pre-defined task.

Mission State: GPS Time: 6800.35, Alt sep (ft): 5693, Grnd rng (mi): 19.91

INPT EGPT RVPT
 NFZ 1 NFZ 2
 POBs 1 POBs 2 POBs 3

Lat, Lon (34.902, -117.337) - x, y (561,310) Connected

Command History

ID	RUN	RESET	REC ON	CMD ON	SND ORST	SND TGTS	ATCR TGT	AlphaOrg	Alpha 1	Alpha 2
UAV Call	Resp2UAV	FW Out	P_CarTsk	UAVStat	UAV RTB	BravoOrg	Bravo 1	Bravo 2		