A Sampling-Based Approach to Spacecraft Autonomous Maneuvering with Safety Specifications

Joseph A. Starek*, Brent W. Barbee†, Marco Pavone*

*Autonomous Systems Lab
Dept. of Aeronautics & Astronautics
Stanford University

†Navigation and Mission Design Branch
Goddard Space Flight Center
NASA

AAS-GNC 2015
February 3rd, 2015
Outline

- Autonomous Vehicle Safety
Outline

- Autonomous Vehicle Safety
- Spacecraft Safety
Outline

- Autonomous Vehicle Safety
- Spacecraft Safety
- Safety in CWH Dynamics
Outline

- Autonomous Vehicle Safety
- Spacecraft Safety
- Safety in CWH Dynamics
- Numerical Experiments
Outline

• Autonomous Vehicle Safety
• Spacecraft Safety
• Safety in CWH Dynamics
• Numerical Experiments
• Conclusions and Future Work
The Need for Safe Autonomy
The Need for Safe Autonomy

- Satellite servicing (DARPA Phoenix Mission)
The Need for Safe Autonomy

- Satellite servicing (DARPA Phoenix Mission)
- Automated rendezvous
The Need for Safe Autonomy

- Satellite servicing (DARPA Phoenix Mission)
- Automated rendezvous

Key Question
How do we implement a general, automated spacecraft planning framework with hard safety specifications?
Original Contribution

Our work:

1. Establishes a **provably-correct framework** for the *systematic* encoding of safety specifications into the spacecraft trajectory generation process

2. Derives an efficient **one-burn escape maneuver policy** for proximity operations near circular orbit
Previous Work

Spacecraft rendezvous approaches with explicit characterizations of safety:

- Kinematic path optimization \cite{Jacobsen2002}
- Artificial potential functions \cite{Roger2000}
- MILP formulations \cite{Breger2008}
- Safety ellipses \cite{Gaylor2007,Naasz2005}
- Motion planning \cite{Frazzoli2003}
- Robust Model-Predictive Control \cite{Carson2008}
- Forced equilibria \cite{Weiss2013}
Types of Spacecraft Rendezvous Safety

- **Passive Trajectory Protection**: Constrain coasting trajectories to avoid collisions up to a given horizon time
- **Active Trajectory Protection**: Implement an *actuated* escape maneuver to save/abort a mission

Design Choice

We emphasize *active safety* as it is the less-conservative approach
Definition (Trajectory Safety Problem)

For all possible failure times $t_{\text{fail}} \in T_{\text{fail}}$ and failure modes $U_{\text{fail}}(x(t_{\text{fail}}))$, we seek a sequence of admissible actions $u(\tau) \in U_{\text{fail}}(x(t_{\text{fail}}))$ from $x(t_{\text{fail}})$ such that the remaining trajectory is safe.

Examples:

- **Rovers/Land vehicles**: Come to a complete stop
- **Manipulators**: Return to previous configuration, disengage, or execute emergency plan
- **UAV’s**: Enter a safe loiter pattern
- **Spacecraft**: Less straightforward; generally require mission-specific solutions (with human oversight)
Definition (Trajectory Safety Problem)

For all possible failure times $t_{\text{fail}} \in T_{\text{fail}}$ and failure modes $U_{\text{fail}}(x(t_{\text{fail}}))$, we seek a sequence of admissible actions $u(\tau) \in U_{\text{fail}}(x(t_{\text{fail}}))$ from $x(t_{\text{fail}})$ such that the remaining trajectory is safe.
Definition (Trajectory Safety Problem)

For all possible failure times $t_{fail} \in T_{fail}$ and failure modes $U_{fail}(x(t_{fail}))$, we seek a sequence of admissible actions $u(\tau) \in U_{fail}(x(t_{fail}))$ from $x(t_{fail})$ such that the remaining trajectory is safe.
Definition (Trajectory Safety Problem)

For all possible failure times $t_{\text{fail}} \in T_{\text{fail}}$ and failure modes $U_{\text{fail}}(x(t_{\text{fail}}))$, we seek a sequence of admissible actions $u(\tau) \in U_{\text{fail}}(x(t_{\text{fail}}))$ from $x(t_{\text{fail}})$ such that the remaining trajectory is safe.
Definition (Trajectory Safety Problem)

For all possible failure times $t_{\text{fail}} \in \mathcal{T}_{\text{fail}}$ and failure modes $\mathcal{U}_{\text{fail}}(x(t_{\text{fail}}))$, we seek a sequence of admissible actions $u(\tau) \in \mathcal{U}_{\text{fail}}(x(t_{\text{fail}}))$ from $x(t_{\text{fail}})$ such that the remaining trajectory is safe.
Definition (Trajectory Safety Problem)

For all possible failure times $t_{\text{fail}} \in \mathcal{T}_{\text{fail}}$ and failure modes $\mathcal{U}_{\text{fail}}(x(t_{\text{fail}}))$, we seek a sequence of admissible actions $u(\tau) \in \mathcal{U}_{\text{fail}}(x(t_{\text{fail}}))$ from $x(t_{\text{fail}})$ such that the remaining trajectory is safe.

\[ \mathcal{T}_{\text{fail}} = \bigcup \{ t_{\text{fail}} \} \]
Definition (Trajectory Safety Problem)

For all possible failure times $t_{\text{fail}} \in \mathcal{T}_{\text{fail}}$ and failure modes $\mathcal{U}_{\text{fail}}(x(t_{\text{fail}}))$, we seek a sequence of admissible actions $u(\tau) \in \mathcal{U}_{\text{fail}}(x(t_{\text{fail}}))$ from $x(t_{\text{fail}})$ such that the remaining trajectory is safe.
Finite-horizon safety guarantees can ultimately violate constraints:

\[ \mathcal{X}_{\text{obs}} \]

\[ \mathcal{X}_{\text{init}} \rightarrow x(t_{\text{fail}}) \rightarrow x(T) \rightarrow x_{\text{goal}} \]

Failure

\[ u(\tau) = 0, \quad t_{\text{fail}} \leq \tau \leq T \]
Idea: Positively-Invariant Sets

Definition (Positively-Invariant Set)

A set $\mathcal{X}_{\text{invariant}}$ is positively invariant with respect to $\dot{x} = f(x)$ if and only if

$$x(t_0) \in \mathcal{X}_{\text{invariant}} \implies x(t) \in \mathcal{X}_{\text{invariant}}, \; t \geq t_0$$
Idea: Positively-Invariant Sets

Definition (Vehicle State Safety)

A state is safe if and only if there exists, under all failure conditions, a safe, dynamically-feasible trajectory that *navigates the vehicle to a safe, stable positively-invariant set.*
Finite-Time Trajectory Safety

\[
\begin{align*}
\text{minimize} & \quad J(x, u, t) \\
\text{subject to} & \quad \dot{x}(t) = f(x(t), u(t), t) \quad \text{(Dynamics)} \\
& \quad x(t_0) = x_0 \quad \text{(Initial Condition)} \\
& \quad x(t_f) \in X_{\text{invariant}} \quad \text{(Invariant Termination)} \\
& \quad u(t) \in U_{\text{fail}}(x_0) \quad \text{(Control Admissibility)} \\
& \quad g_i(x, u) \leq 0, \quad i = [1, \ldots, p] \quad \text{(Inequality Constraints)} \\
& \quad h_j(x, u) = 0, \quad j = [1, \ldots, q] \quad \text{(Equality Constraints)}
\end{align*}
\]
Challenge: Solving the Finite-Time Safety Problem under Failures

For a $K$-fault tolerant spacecraft with $N$ control components (thrusters, momentum wheels, CMG’s, etc), this yields:

$$N_{\text{fail}} = \sum_{k=0}^{K} \binom{N}{k} = \sum_{k=0}^{K} \frac{N!}{k!(N-k)!}$$

total optimization problems (one for each $U_{\text{fail}}$) for each failure time $t_{\text{fail}}$. 
Idea: Simplify the Finite-Time Safety Problem

Theorem (Sufficient Fault-Tolerant Active Safety)

1. From each \( x(t_{\text{fail}}) \), prescribe a Collision-Avoidance Maneuver \( \Pi_{\text{CAM}}(x) \) that gives a horizon \( T \) and escape sequence \( u \) that satisfies \( x(T) \in \mathcal{X}_{\text{invariant}} \) and \( u(\tau) \subset U \) for all \( t_{\text{fail}} \leq \tau \leq T \).

2. For each failure mode \( U_{\text{fail}}(x(t_{\text{fail}})) \subset U(x(t_{\text{fail}})) \) up to tolerance \( K \), check if \( u = \Pi_{\text{CAM}}(x) \subset U_{\text{fail}} \).
Idea: Simplify the Finite-Time Safety Problem

Theorem (Sufficient Fault-Tolerant Active Safety)

1. From each $x(t_{fail})$, prescribe a Collision-Avoidance Maneuver $\Pi_{CAM}(x)$ that gives a horizon $T$ and escape sequence $u$ that satisfies $x(T) \in \mathcal{X}_{\text{invariant}}$ and $u(\tau) \subset \mathcal{U}$ for all $t_{fail} \leq \tau \leq T$.

2. For each failure mode $\mathcal{U}_{\text{fail}}(x(t_{fail})) \subset \mathcal{U}(x(t_{fail}))$ up to tolerance $K$, check if $u = \Pi_{CAM}(x) \subset \mathcal{U}_{\text{fail}}$.

Key Simplifications

Removes decision variables $u$, reducing to:

- a test of escape control feasibility under failure(s)
- numerical integration for satisfaction of dynamics
- an \textit{a posteriori} check of constraints $g_i$ and $h_j$
Safe Sampling-Based Spacecraft Planning

Solution is in exact form required for sampling-based motion planning.

- Restrict planning to actively-safe nodes
- Restrict burns to nodes

Actively-safe Sampling-based Spacecraft Planning

Incorporating Safety Constraints:
- Add CAM policy generation to sampling algorithm
- Include CAM-trajectory collision-checking in tests of sample feasibility
Example: CAM Policy Design
Using CWH Set Invariance for CAMs

Circular Clohessy-Wiltshire-Hill (CWH) CAM policy:
1. Coast from $x(t)$ to some new $T > t$ such that $x(T-t)$ lies at a position in $X$ invariant.
2. Circularize the orbit at $x(T)$ such that $x(T+t) \in X$ invariant.
3. Coast along the new orbit (horizontal drift along the in-track axis) in $X$ invariant.
Example: CAM Policy Design
Using CWH Set Invariance for CAMs

Circular Clohessy-Wiltshire-Hill (CWH) CAM policy:

1. Coast from \( x(t) \) to some new \( T > t \) such that \( x(T^-) \) lies at a position in \( \mathcal{X}_{\text{invariant}} \).

2. Circularize the orbit at \( x(T) \) such that \( x(T^+) \in \mathcal{X}_{\text{invariant}} \).

3. Coast along the new orbit (horizontal drift along the in-track axis) in \( \mathcal{X}_{\text{invariant}} \).
Example: CAM Policy Design
Choosing the Circularization Time, $T$

**CWH Finite-Time Safety Problem:**

Given: $x(t), u(\tau) = 0, t \leq \tau < T$

minimize $\Delta v_{circ}^2(T)$

subject to $\dot{x}(\tau) = f(x(\tau), 0, \tau)$ (Dynamics)

$x(\tau) \notin X_{KOZ}$ (KOZ Avoidance)

$x(T^+) \in X_{\text{invariant}}$ (Invariant Termination)

**Key Result**

Can be reduced to an analytical expression that is solvable in milliseconds
Scenario

- Simulates an automated approach to LandSat-7 (e.g., for servicing) between pre-specified waypoints
- Calls on the Fast Marching Tree (FMT\textsuperscript{*}) algorithm for implementation
Scenario

- Simulates an automated approach to LandSat-7 (e.g., for servicing) between pre-specified waypoints
- Calls on the Fast Marching Tree (FMT*) algorithm for implementation

Assumptions:

- Begins at insertion into a coplanar circular orbit sufficiently close to the target
- The target is nadir-pointing
- The chaser is nominally nadir-pointing, or executes a “turn-burn-turn” along CAMs
Scenario

- Simulates an automated approach to LandSat-7 (e.g., for servicing) between pre-specified waypoints
- Calls on the Fast Marching Tree (FMT*) algorithm for implementation

Constraints:

- **Plume impingement:** No exhaust plume impingement
- **Collision avoidance:** Clearance of an elliptic Keep-Out Zone (KOZ)
- **Target communication:** Target comm lobe avoidance
- **Safety:** Two-fault tolerance to stuck-off failures
Motion Planning Problem

Motion planning query:

\[ \delta x \] (Radial)

\[ \delta y \] (In-Track)

\[ x_{\text{init}} \]

\[ x_{\text{goal}} \]

Waypts

KOZ
Motion Plan Comparison

Motion planning solutions:

\[ \delta x \quad \text{(Radial)} \]

\[ \delta y \quad \text{(In-Track)} \]

- \( x_{\text{init}} \)
- \( x_{\text{goal}} \)
- \( \text{Waypts} \)
- \( \text{KOZ} \)
Motion Plan Comparison

Motion planning solutions:

\[ \delta x \text{ (Radial)} \]
\[ \delta y \text{ (In-Track)} \]

- init
- goal
- Waypts
- KOZ
Success Rate Comparison

Success comparison as a function of thruster failure probability, computed over 50 trials:
Success Rate Comparison

Success comparison as a function of thruster failure probability, computed over 50 trials:
Conclusions

Key Ideas

1. Use termination constraints inside safe, stable, positively-invariant sets for infinite-horizon maneuver safety
2. Embed invariant-set constraints into sampling-based algorithms for safety-constrained planning

Synopsis

- Demonstrated the idea for failure-tolerant circular CWH planning
- CAM policies can be precomputed offline for more efficient online computation
Future Goals

- Extend to thruster stuck-on and mis-allocation failures
- Account for localization uncertainty
- Apply these notions to small-body proximity operations
Thank you!

Joseph A. Starek, Brent W. Barbee, and Marco Pavone

Aeronautics & Astronautics   Navigation and Mission Design
Stanford University           NASA GSFC
jstarek@stanford.edu
Clohessy-Wiltshire-Hill (CWH) Equations

- Motion is linearized about a moving reference point in circular orbit:
  \[
  \mathbf{x} = [\delta x, \delta y, \delta z, \dot{\delta x}, \dot{\delta y}, \dot{\delta z}]^T
  \]
  \[
  \mathbf{u} = \frac{1}{m}[F_x, F_y, F_z]^T
  \]

- Yields LTI dynamics:
  \[
  \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}
  \]

\[
\mathbf{A} = \begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
3n_{\text{ref}}^2 & 0 & 0 & 0 & 2n_{\text{ref}} & 0 \\
0 & 0 & 0 & -2n_{\text{ref}} & 0 & 0 \\
0 & 0 & -n_{\text{ref}}^2 & 0 & 0 & 0
\end{bmatrix}
\]

\[
\mathbf{B} = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]
Definition (Optimal Motion Planning Problem)

Given $\mathcal{X}$, $\mathcal{X}_{\text{obs}}$, $\mathcal{X}_{\text{free}}$, and $J$, find an action trajectory $u : [0, T] \rightarrow \mathcal{U}$ yielding a feasible path $x(t) \in \mathcal{X}_{\text{free}}$ over time horizon $t \in [0, T]$, which reaches the goal region $x(T) \in \mathcal{X}_{\text{goal}}$ and minimizes the cost functional $J = \int_0^T c(x(t), u(t)) \, dt$.

Characteristics:

- PSPACE-hard (and therefore NP-hard)
- Requires kinodynamic motion planning
- Almost certainly requires approximate algorithms, tailored to the particular application
Generalized Mover’s Problem

\[ \dot{x}(t) = f(x(t), u(t)) \]
Generalized Mover’s Problem

\[ \dot{x}(t) = f(x(t), u(t)) \]

\( C \)

\( C_{\text{obs}} \)

\( C_{\text{goal}} \)
Generalized Mover's Problem

\[ \dot{x}(t) = f(x(t), u(t)) \]
The FMT* Algorithm
The FMT* Algorithm

Initial State $x_I$

Goal State $x_G$

Goal Region

$C_{obs}$

$C$

Dynamics

Sampling-Based Motion Planning

Optimal Motion Planning

FMT*
The FMT* Algorithm
The FMT* Algorithm

- = Unexplored, \( \mathcal{W} \)
- = Interior
- = Frontier, \( \mathcal{H} \)

\( x_I \), \( x_G \), \( C \), \( C_{obs} \)
The FMT* Algorithm

- Unexplored, $\mathcal{W}$
- Interior
- Frontier, $\mathcal{H}$

$x_G$

$z$

$x_I$

$C_{obs}$

$C$
The FMT* Algorithm
The FMT* Algorithm

- = Unexplored, $\mathcal{W}$
- = Interior
○ = Frontier, $\mathcal{H}$

$Z_{near}$
$z$
$x_I$

$C_{obs}$

$C$

$x_G$
The FMT* Algorithm

- = Unexplored, $\mathcal{W}$
- = Interior
- = Frontier, $\mathcal{H}$

$\mathcal{V}$

$Z_{near}$

$z$

$x_l$

$C_{obs}$

$x_G$

$C$

$C_{obs}$

FMT*

Optimal Motion Planning

Dynamics

Sampling-Based

Spacecraft Safety

J. Starek, B. Barbee, M. Pavone
The FMT* Algorithm

\[ x_I, Z_{\text{near}}, x_G, C, C_{\text{obs}} \]

- \( \text{Unexplored, } \mathcal{W} \)
- \( \text{Interior} \)
- \( \text{Frontier, } \mathcal{H} \)
The FMT* Algorithm
The FMT\(^*\) Algorithm
The FMT* Algorithm
The FMT* Algorithm
The FMT* Algorithm

- Unexplored, $\mathcal{W}$
- Interior
- Frontier, $\mathcal{H}$

$z$, $z_{near}$, $x_I$, $x_G$, $C_{obs}$, $C$
The FMT* Algorithm
The FMT* Algorithm
The FMT* Algorithm

- Unexplored, $W$
- Interior
- Frontier, $H$

$Z_{near}$

$C_{obs}$

$C$
The FMT* Algorithm

- = Unexplored, $W$
- = Interior
- = Frontier, $H$

$X_I$

$Z_{near}$

$C_{obs}$

$C_{obs}$
The FMT* Algorithm
The FMT* Algorithm

The FMT* Algorithm

- Unexplored, $\mathcal{W}$
- Interior
- Frontier, $\mathcal{H}$

$C_{obs}$

$x_I$

$x_G$