AFRL-RY-WP-TP-2022-0175

1D SPACECRAFT DOCKING EXAMPLE (Preprint)

Kerianne L. Hobbs
Autonomous Capability Team 3 Branch
Sensors Plans & Advanced Programs Division

James Cunningham
Jacobs Engineering Group

John McCarroll
Matrix Research Inc.

Umberto J. Ravaioli
Toyon Research Corp.

Kyle Dunlap
University of Cincinnati

Xin Chen
University of Dayton

MAY 2022
Final Report

AIR FORCE RESEARCH LABORATORY
SENSORS DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7320
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED START DATE	END DATE
May 2022	Journal Article Preprint	23 May 2022	23 May 2022

4. TITLE AND SUBTITLE
1D SPACECRAFT DOCKING EXAMPLE (Preprint)

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
In-House	N/A	N/A
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**
N/A	N/A	N/A

6. AUTHOR(S)
Kerianne L. Hobbs (AFRL/RYZA)
James Cunningham (Jacobs Engineering Group)
John McCarroll (Matrix Research Inc.)
Umberto J. Ravaioli (Toyon Research Corp.)
Kyle Dunlap (University of Cincinnati)
Xin Chen (University of Dayton)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Autonomous Capability Team 3 Branch
Sensors Plans & Advanced Programs Division
Air Force Research Laboratory, Sensors Directorate
Wright-Patterson Air Force Base, OH 45433-7320
Air Force Materiel Command, United States Air Forces

Jacobs Engineering Group 1415 Research Park Dr. Beavercreek, OH 45432

Matrix Research Inc. 607 Washington St. Newton, VT 02458

Toyon Research Corp. 6800 Cortona Dr. Goleta, CA, 93117

University of Cincinnati 2901 Woodside Drive Cincinnati, OH, 45219

University of Dayton 300 College Park Dayton, OH 45469

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory, Sensors Directorate
Wright-Patterson Air Force Base, OH 45433-7320
Air Force Materiel Command, United States Air Forces

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RYZA

11. SPONSOR/MONITOR'SREPORT NUMBER(S)
AFRL-RY-WP-TR-2022-0175

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
PAO case number AFRL-2022-2438, Clearance Date 23 May 22. The U.S. Government is joint author of this work and has the right to use, modify, reproduce, release, perform, display, or disclose the work. Report contains color.

14. ABSTRACT

This document explains the dynamics, objective, safety constraints, pre-processing, and post-processing for a 1D spacecraft docking reinforcement learning and neural network verification challenge problem.

15. SUBJECT TERMS
verification

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	C. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
Unclassified	Unclassified	Unclassified	SAR	5

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Kerianne Hobbs	

PREVIOUS EDITION IS OBSOLETE.

STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

1D Spacecraft Docking Example*

James Cunningham
Jacobs Engineering Group
1415 Research Park Dr.
Beavercreek, OH 45432
james.cunningham@jacobs.com

John McCarroll
Matrix Research Inc.
607 Washington St.
Newton, VT 02458
john.mccarroll.ext@afresearchlab.com

Umberto J. Ravaioli
Toyon Research Corp.
6800 Cortona Dr.
Goleta, CA, 93117
uravaioli@toyon.com

Kyle Dunlap
University of Cincinnati
2901 Woodside Drive
Cincinnati, OH, 45219
dunlapkp@mail.uc.edu

Xin Chen
University of Dayton
300 College Park
Dayton, OH 45469
xchen4@udayton.edu

Kerianne L. Hobbs
Air Force Research Laboratory
2241 Avionics Circle
Wright-Patterson Air Force Base, OH, 45433
kerianne.hobbs@us.af.mil

***Abstract*—This document explains the dynamics, objective, safety constraints, pre-processing, and post-processing for a 1D spacecraft docking reinforcement learning and neural network verification challenge problem.**

I. Dynamics

In this example, a spacecraft is represented with a position $x \in \mathbb{R}$ (bounded along the x-axis with no y component) with a bounded acceleration control variable $\ddot{x} = u, u \in [u_{min}, u_{max}]$ (negative acceleration is braking). Let us assume that thrust $u \in [-1, 1]$, and mass $m = 1$ kg.

Our system state is the position x and velocity $\dot{x}$ of the spacecraft:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}. \tag{1}$$

This results in dynamics described by these state space equations:

$$\dot{x}_1 = x_2 \tag{2}$$

$$\dot{x}_2 = \frac{1}{m} u \tag{3}$$

In matrix form $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$, this becomes:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} \mathbf{u} \tag{4}$$

II. Discrete Dynamics

The earlier dynamics described the continuous system dynamics, but for some of the examples, we are interested in discrete system dynamics of the form $\mathbf{x}_{k+1} = A_{dt}\mathbf{x}_k + B_{dt}\mathbf{u}_k$.

The linearized dynamics for the system are:

This work was supported by the Air Force Research Laboratory Innovation Pipeline Fund. The views expressed are those of the authors and do not reflect the official guidance or position of the United States Government, the Department of Defense or of the United States Air Force.

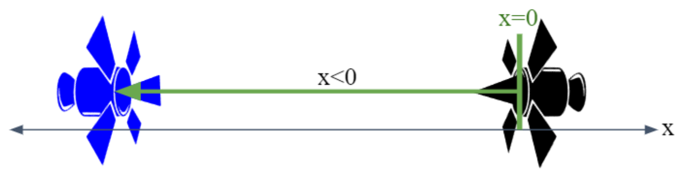

Fig. 1. 1D spacecraft docking concept.

$$\begin{aligned} x_{k+1} &= x_k + v_k \Delta t + 1/2 a_k \Delta t^2 \\ v_{k+1} &= v_k + a_k \Delta t \end{aligned} \tag{5}$$

Considering just the change between states, and that $a_k = u$, we get:

$$\begin{aligned} x_{1,k+1} &= x_{2,k}\Delta t + \frac{1}{2m} u_k \Delta t^2 \\ x_{2,k+1} &= \frac{1}{m} u \Delta t \end{aligned} \tag{6}$$

$$\mathbf{x}_k = \begin{bmatrix} 0 & \Delta t \\ 0 & 0 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} 0.5\Delta t^2 \\ \Delta t \end{bmatrix} \frac{1}{m} \mathbf{u}_k \tag{7}$$

III. Objective

Let's assume the origin of the coordinate system is the perfect spot to dock the center of mass of the spacecraft. Consider that the objective of the primary controller (neural network, PID, bang-bang, etc.) is find the optimal control u^* to dock the spacecraft at the origin (so that $\dot{x} = 0, x = 0$) in minimum time.

We specify this as a successful terminal condition as the following:

- $x \leq 0 \land x \geq -0.5$
- $\dot{x} \leq \sqrt{-2x}$

IV. Initial Conditions

@t=0

- $x \in [-125, -75]$
- $\dot{x} = 0$

V. Safety Constraint and Run Time Assurance Objective

The objective of the run time assurance (RTA) is to ensure that the satellite does not crash into the satellite it is docking with

$$\varphi_{RTA} : (x_1 \leq 0) \tag{8}$$

or alternatively,

$$\varphi_{RTA} = -x_1 \geq 0 \tag{9}$$

However, there are several different interpretations of this constraint that may be better suited for different run time assurance approaches.

In general there are two approaches to defining safety - explicitly and implicitly. Explicit safety definitions rely on whether the state of the system is in a safe set, while implicit approaches check for whether a simulated future state is safe. There are also two ways to approach the control solution: Simplex methods that feature a switch from the primary controller to a backup controller, and Active Set Invariance Filtering (ASIF) that feature a smooth filter of the desired control signal to ensure safety. This results in a total of 4 RTA approaches that we'll talk about in this section: Explicit Simplex, Implicit Simplex, Explicit ASIF, and Implicit ASIF. Let's start by explicitly defining what is safe.

A. Explicit Safe Set

In many cases, we can explicitly describe the set of safe states with

$$\mathcal{C}_{\mathrm{S}} := \{\mathbf{x} \in \mathbb{R}^n \mid h(\mathbf{x}) \geq 0\}. \tag{10}$$

where $h : \mathbb{R}^n \rightarrow \mathbb{R}$ is the control barrier function safety constraint. Checking whether the state is in the safe set ($\mathbf{x} \in \mathcal{C}_{\mathrm{S}}$) is then equivalent to checking whether $h(\mathbf{x}) \geq 0$.

For the double integrator spacecraft, the constraint set is

$$\mathcal{C}_{\mathrm{A}} := \{\mathbf{x} \in \mathbb{R}^2 \mid -x_1 \geq 0\}. \tag{11}$$

The maximum distance that a particle (our spacecraft) can travel before stopping D_{dec} is computed using a deceleration of $\bar{a}$ in a variation of the standard braking or stopping equation, described in Eq.:

$$D_{dec} = \frac{x_2^2}{2\bar{a}}. \tag{12}$$

Assuming that our backup control action is to apply a maximum deceleration of $u = \bar{a} = -1$, and solving for the maximum velocity x_2 for any required required $x_1 = D_{dec}$, gives

$$\begin{aligned} x_1 &= \frac{x_2^2}{2(-1)} \\ x_2^2 &= -2x_1 \\ x_2 &= \sqrt{-2x_1} \end{aligned} \tag{13}$$

Intuitively, $x_2 = \sqrt{-2x_1}$ represents the maximum velocity the spacecraft can travel at a given distance while safely approaching the obstacle at the origin. If this approach speed is exceeded, then there will not be sufficient distance to stop before a collision occurs.

This result is made apparent by considering the flow (trajectory) of the system under a recovery maneuver $u = -1$, which applies maximum thrust force away from the obstacle. $\mathcal{C}_{\mathrm{S}}$ is a forward invariant set under this control law. This is depicted in Fig. 2, where *unsafe states (the complement of the constraint set) ($-x_1 < 0$) are shaded in red, states within the constraint set but not in the safe set ($\mathcal{C}_{\mathrm{A}} \notin \mathcal{C}_{\mathrm{S}}$) ($-x_1 \geq 0$ and $-2x_1 - x_2^2 < 0$) are shaded in purple, states within the safe set $\mathcal{C}_S$ ($-x_1 \geq 0$ and $-2x_1 - x_2^2 \geq 0$) where the spacecraft is moving towards the docking point ($x_2 > 0$) are shaded in green, and states within the safe set $\mathcal{C}_S$ where the spacecraft is stopped or moving away from the docking point ($x_2 \leq 0$) are shaded in cyan.

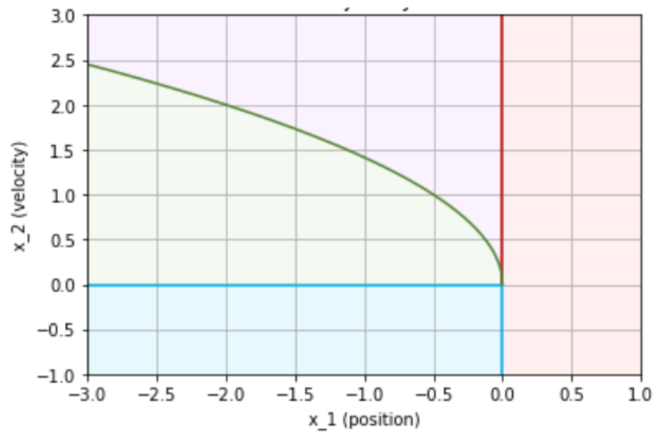

Fig. 2. Depiction of the safety constraint where x_1 is the position of the spacecraft and x_2 is the closing velocity. The region in green is the safe set, the region in red represents the st of collision states, the region in purple is inevitable collision states, and the region in cyan represents areas where the spacecraft is stopped or moving away from the docking point.

B. Explicit Simplex RTA

Explicit Simplex RTAs decide when to switch to a backup controller based on location in an explicitly designed set. In both the Explicit Simplex RTA and the Explicit ASIF RTA, the primary controller and backup controller are identical. For this example, the difference is in the decision to switch. The *primary controller* is the "lead foot" ($u = 1$). The *backup controller* is a maximum braking ($u = -1$). Next we need to define an explicitly barrier constraint to know when to switch to the backup controller.

For cases when the spacecraft is moving towards the docking point ($x_2 > 0$), the barrier function is the square of the stopping distance equation: $h(\mathbf{x}) = -2x_1 - x_2^2$. This is represented by the green line in the plots.

For cases when the car is moving away from the wall, any position $x_1 < 0$ is safe, resuting in a second barrier function $h(\mathbf{x}) = -x_1$. This is represented by the teal region in the plots.

Therefore, the largest control invariant subset (i.e. the viability kernel) of $\mathcal{C}_{\rm A}$ is $\mathcal{C}_{\rm S} = \{x \in \mathbb{R}^2 \,|\, h(x) \geq 0\}$ where

$$h(\mathbf{x}) = \begin{cases} -2x_1 - x_2^2 & \text{if} \quad x_2 > 0 \\ -x_1 & \text{if} \quad x_2 \leq 0. \end{cases} \tag{14}$$

The unsafe states in the constraint set $\mathcal{C}_{\rm A} \setminus \mathcal{C}_{\rm S}$ represent states for which a future collision is inevitable, as represented by the purple section of the plots.

This `explicitsimplex` RTA function receives the environment object and the desired control signal u_{des}. It sets the backup control signal to max deceleration ($u_b = -1$).

Using the state and discrete time dynamics, the predicted state x_p under the desired dynamics is computed.

Next, using the explicit control barrier function, the RTA decides whether to let the desired signal pass or to pass the backup control signal instead using the following logic (on the predicted state):

$$h(\mathbf{x}) = \begin{cases} u_{des} & \text{if } (x_2 > 0) \wedge (-2x_1 - x_2^2 \geq 0) \\ u_{des} & \text{if } (x_2 \leq 0) \wedge (-x_1 \geq 0) \\ u_b & \text{otherwise.} \end{cases} \tag{15}$$

VI. Neural Network Control System

A neural network control system (NNCS) provides the primary control for the spacecraft. In order to train the NNCS, pre-processing and post-processing are performed.

A. Input Pre-Processing

Pre-processing of the neural network input, $\mathbf{x}_{obs}$, is a simply application of individual normalization factors to the two components of the state vector. $\frac{1}{100}$ for x and $\frac{1}{10}$ for $\dot{x}$.

$$\mathbf{x}_{obs} = \begin{bmatrix} \frac{1}{100} & 0 \\ 0 & \frac{1}{10} \end{bmatrix} \mathbf{x} \tag{16}$$

B. Output Post-Processing

After receiving the output of neural network, we must transform it into the control vector u.

Note that the network provided is a combined actor-critic model. That means that it has 2 outputs, the policy output and the value network output. We are only interested in the policy network output.

Additionally, the policy network output contains two elements, the action mean and action std dev. Since we are using the policy in deterministic configuration, we can simply keep the action mean and throw away the action std dev.

Finally, we apply $\tanh$ to the action mean to squash the values continuously between [-1, 1]

$$\begin{aligned} \mathbf{y}_1, \mathbf{y}_2 &= \pi_\theta(\mathbf{x}_{obs}) \\ \mathbf{u} &= \tanh\left(\begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{y}_1\right) \end{aligned} \tag{17}$$

C. Network Architecture

The neural network used in this challenge problem is defined as follows:

1) Policy:

- Input Layer: Fully-Connected
  - input dim: [None, 2]
  - output dim: [None, 256]
  - activation: $\tanh$
- Hidden Layers: x2
  - input dim: [None, 256]
  - output dim: [None, 256]
  - activation: $\tanh$
- Output Layer: Fully-Connected
  - input dim: [None, 256]
  - output dim: [None, 2]
  - activation: linear
  - output: [thrust mean, thrust std dev]

2) Value:

- Input Layer: Fully-Connected
  - input dim: [None, 2]
  - output dim: [None, 256]
  - activation: $\tanh$
- Hidden Layers: x2
  - input dim: [None, 256]
  - output dim: [None, 256]
  - activation: $\tanh$
- Output Layer: Fully-Connected
  - input dim: [None, 256]
  - output dim: [None, 1]
  - activation: linear
  - output: [[state value]]