Spacecraft Docking Benchmark[⋆]

Umberto J. Ravaioli[1], James Cunningham[2], John McCarroll[3], Vardaan Gangal[2], Kyle Dunlap[4], and Kerianne L. Hobbs[5]

[1] Toyon Research Corp., Goleta CA 93117, USA uravaioli@toyon.com
[2] Jacobs Engineering Group, Beavercreek OH 45432, USA
{james.cunningham,vardaan.gangal}@jacobs.com
[3] Matrix Research Inc., Newton VT 02458, USA
john.mccarroll.ext@afresearchlab.com
[4] University of Cincinnati, Cincinnati OH 45219, USA dunlapkp@mail.uc.edu
[5] Air Force Research Laboratory, Wright-Patterson Air Force Base OH 45433, USA
kerianne.hobbs@us.af.mil

1 Benchmark Description

1.1 2D Spacecraft Docking

In the 2D spacecraft docking environment, the state of an active deputy spacecraft is expressed relative to the passive chief spacecraft in Hill's reference frame [1] $\mathcal{F}_\mathrm{H} := (\mathcal{O}_\mathrm{H},\ \hat{i}_\mathrm{H},\ \hat{j}_\mathrm{H})$. The origin of Hill's frame $\mathcal{O}_\mathrm{H}$ is located at the mass center of the chief, the unit vector $\hat{i}_\mathrm{H}$ points away from the Earth along a line connecting the center of Earth to $\mathcal{O}_\mathrm{H}$, and the unit vector $\hat{j}_\mathrm{H}$ is aligned with the orbital velocity vector of the chief. The state of the deputy is defined as $\boldsymbol{x} = [x, y, \dot{x}, \dot{y}]^T \in \mathcal{X} \subset \mathbb{R}^4$, where, $\boldsymbol{r} = x\hat{i}_\mathrm{H} + y\hat{j}_\mathrm{H}$ is the position vector and $\boldsymbol{v} = \dot{x}\hat{i}_\mathrm{H} + \dot{y}\hat{j}_\mathrm{H}$ is the velocity vector of the deputy in Hills Frame. The control for the system is defined by $\boldsymbol{u} = [F_x, F_y]^T{=}[u_1, u_2]^T \in \mathcal{U} \subset \mathbb{R}^2$.

Dynamics A first order approximation of the relative motion dynamics between the deputy and chief spacecraft is given by Clohessy-Wiltshire [2] equations,

$$\begin{aligned} \ddot{x} &= 2n\dot{y} + 3n^2x + \frac{F_x}{m} \\ \ddot{y} &= -2n\dot{x} + \frac{F_y}{m} \end{aligned} \tag{1}$$

where n is spacecraft mean motion and m is the mass of the deputy.

Success Criteria The deputy is considered successfully docked when its distance to the chief is less than a desired distance ρ_d.

$$\varphi_\mathrm{docking} : (\|\boldsymbol{r}_\mathrm{H}\| \leq \rho_d) \tag{2}$$

[⋆] Approved for public release. Distribution is unlimited. Case Number: AFRL-2021-4045.

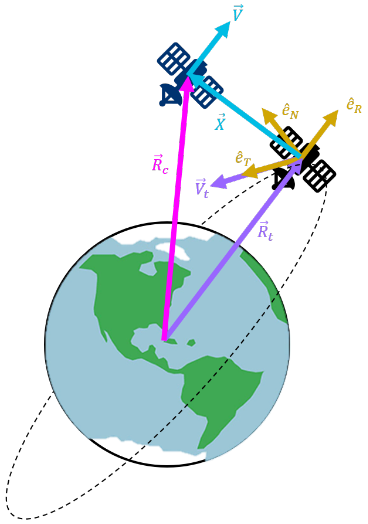

Fig. 1. **Hill's reference frame centered on a chief spacecraft and used to describe the relative motion of a deputy spacecraft conducting proximity operations (not to scale).**

Safety Constraint The RL agent must learn to dock while adhering to a dynamic velocity safety constraint that restricts the relative velocity of the deputy to velocity limit that decreases as it approaches the chief. The system is defined to be safe if it obeys the following safety constraint for all time,

$$\varphi_{s_{2DSC}} := \|\boldsymbol{v}_{\mathrm{H}}\| \leq \nu_0 + \nu_1 \|\boldsymbol{r}_{\mathrm{H}}\| \tag{3}$$

where, ν_0, $\nu_1 \in \mathbb{R}_{\geq 0}$, and

$$\|\boldsymbol{r}_{\mathrm{H}}\| = (x^2 + y^2)^{1/2}, \quad \|\boldsymbol{v}_{\mathrm{H}}\| = (\dot{x}^2 + \dot{y}^2)^{1/2}. \tag{4}$$

The constraint in Eq. (3) enacts a distance-dependent speed limit, with ν_0 defining the maximum allowable docking speed and ν_1 defining the rate at which deputy must slow down as it approaches the chief. The values $\nu_0 = 0.2$ m/s, and $\nu_1 = 2n$ s^{-1} are selected based on elliptical closed natural motion trajectories (eCNMT), and further insight into this choice is given in [3, 4].

Parameters The value of the parameters used in the equations above and the ranges of the control inputs are described in Table 1.

2 Neural Networks

The provided neural network controller operates with full state feedback. Input pre-processing and output post-processing performed during training was included within the network architecture using matrix multiplications and tanh activation as shown in Figure 2. The means that the NN input is simply the state vector and the output is simply the control vector.

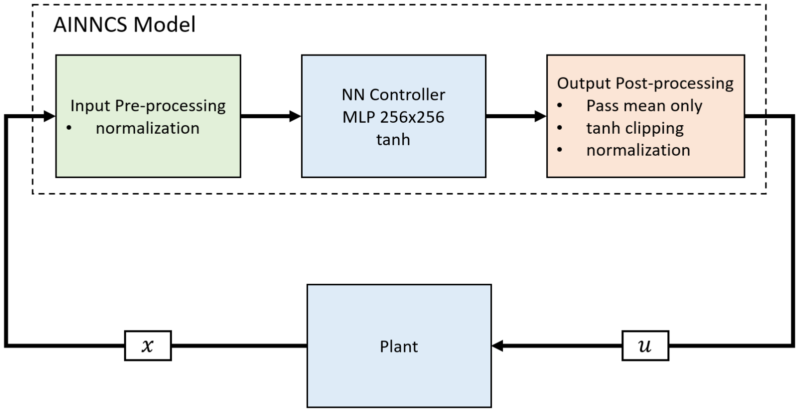

Fig. 2. **AINNCS Model with inlcuded input pre-processing and output post-processing**

The Neural Network controller was trained on the Docking 2d environment with reinforcement learning using the training procedure described in [5]. However, the training procedure differed in providing only the full state (position and velocity) as input and with hard clipping of output actions replaced with soft tanh clipping.

The Neural Network architecture was a shallow Multlayer Perceptron with hidden layer widths of 256 and tanh activation functions.

3 Benchmarks Specifications

3.1 State

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix}$$

Where

- x (m)
  - x-component of position in meters
- y (m)
  - y-component of position in meters

- $\dot{x}$ (m/s)
    - x-component of velocity in meters per second
- $\dot{y}$ (m/s)
    - y-component of velocity in meters per second

3.2 Control

$$u = \begin{bmatrix} F_x \\ F_y \end{bmatrix}$$

Where

- $F_x \in [-1, 1]$ (N)
    - Thrust in x direction in Newtons
- $F_y \in [-1, 1]$ (N)
    - Thrust in y direction in Newtons

3.3 Dynamics

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 2n \\ 0 & 0 & -2n & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \end{bmatrix} \mathbf{u}$$

Where

- $m = 12$ (kg)
    - Spacecraft mass in kilograms
- $n = 0.001027$ (rad/s)
    - Clohessy-Wiltshire reference frame orbital mean motion

3.4 Timestep

The neural network controller was trained with a timestep of 1 second. i.e. Every 1 second, the neural network was polled for a new control value.

$$t = 1 \text{ second} \tag{5}$$

3.5 Safety Constraint

$$\varphi_{s_{2DSC}} := \|\boldsymbol{v}_{\mathrm{H}}\| \leq \nu_0 + \nu_1 \|\boldsymbol{r}_{\mathrm{H}}\|$$

Where

- $\|\boldsymbol{r}_{\mathrm{H}}\| = (x^2 + y^2)^{1/2}$
- $\|\boldsymbol{v}_{\mathrm{H}}\| = (\dot{x}^2 + \dot{y}^2)^{1/2}$
- $\nu_0 = 0.2$ (m/s)
- $\nu_1 = 2n$ (s^{-1})

3.6 Initial Conditions

Training The initial conditions of the system state where determined by randomly selecting a location in polar coordinates parameterized by a radial distance from the origin and an azimuth angle. The ranges of this random sampling are described in Table 1.

Verification For the safety verification task, we will consider a subset of the training initial condition states, a box in the top right corner of the x-y plane.

- $x \in [70, 106]$
- $y \in [70, 106]$
- $\dot{x} \in [-0.28, 0.28]$
- $\dot{y} \in [-0.28, 0.28]$

4 Implementation

This section provides additional details on the specific implementations of the environment.

4.1 Configuration File

The Aerospace SafeRL Framework's modular architecture facilitates editing a single configuration file to change the task, environment platforms and their characteristics, the observation space, rewards, and status. In some cases, the classes in the framework defining components are listed, in others, specific values are assigned such as initial condition ranges, actuator limits, individual reward values.

4.2 Environment Platforms

The environment platforms are defined within the config file in an environment configuration. Each environment platform is given a name, a step size, state limitations, a controller specification, and a range of initial conditions for each state variable.

The controller specification includes the class that defines the agent controller, the name of each actuator, whether the actuator is operating in discrete or continuous space, the bounds on that actuator, and the number of discrete points in the case of discrete spaces (which include the minimum and maximum values in the range, with the remaining points evenly spaced within the range). For continuous actuators, the platform performs automatic rescaling post-processing from an assumed input range of [-1, 1] to the desired actuator range, although this behavior can be disabled.

The default configurations are shown in Table 1. In Table 1, $\mathcal{V}_{\text{safe}} = \big[0,\ 0.2 + 2n\sqrt{x^2 + y^2}\big]$ in 2D and $\mathcal{V}_{\text{safe}} = \big[0,\ 0.2 + 2n\sqrt{x^2 + y^2 + z^2}\big]$ in 3D.

Table 1. **2D Docking Default Environment Configurations**

Chief	
x_0,$\dot{x}_0$, y_0, $\dot{y}_0$, z_0, $\dot{z}_0$	0
Deputy	
Mass, m (kg)	12
Mean Motion, n (rad/s)	0.001027
State Reference	Chief
Velocity @t_0 (m/s)	[0, $\mathcal{V}_{\text{safe}}$]
Rel. Distance @t_0 (m)	[100,150]
Rel. Azimuth @t_0 (rad)	[0,2π]
Thrust X, F_x (N)	[-1,1]
Thrust Y, F_y (N)	[-1,1]
Docking Region	
State Reference	Chief
Shape	Circle
x offset (m)	0
y offset (m)	0
radius (m)	0.5

Table 2. **Docking Environment Observations**

Description	**Expression**	**Normalization Const**	**Clipping**
2D Docking			
Deputy Position	$\boldsymbol{r}_d = [x_d, y_d]$	[100, 100]	$[-\infty, \infty]$
Deputy Velocity	$\boldsymbol{v}_d = [\dot{x}_d, \dot{y}_d]$	[0.5, 0.5]	$[-\infty, \infty]$

4.3 Reward

Table 3 show individual reward component values for the Docking task.

Docking The docking rewards contain both sparse rewards that reward/punish success/failure at the end of the episode and dense rewards that provide immediate feedback during the episode. While the failure rewards are all constant, the success reward includes an episode length component to incentivize quicker solutions. The distance change reward is proportional to the change in an exponential potential function of the distance between the deputy and the chief. Docking also has an additional velocity constraint reward with a constant penalty condition applied whenever the constraint is violated in addition to a scaling penalty that grows proportionally with the degree of constraint violation, i.e. violating the safety constraint is bad, violating it by a lot is very bad.

Table 3. **Docking Reward**

Description	**Expression**
Distance Change	$R_t^d = 2\left(e^{-ad_t} - e^{-ad_{t-1}}\right)$ $d_i = \lVert \boldsymbol{r}_{deputy,i} - r^i_{chief,i} \rVert$ $a = \dfrac{\ln(2)}{100}$
Velocity Constraint	$R_t^{vc} = \begin{cases} -0.01 - 0.01(\lVert \boldsymbol{v}_{deputy} \rVert - v_{limit}) & \text{, if } \lVert \boldsymbol{v}_{deputy} \rVert > v_{limit} \\ 0 & \text{, if } \lVert \boldsymbol{v}_{deputy} \rVert \leq v_{limit} \end{cases}$
Δv	$-0.01(\lVert \frac{\boldsymbol{u}}{m} \rVert)$
Success	$+2 - \frac{t}{t_{max}}$
Failure	
Crash	-1
Distance	-1
Timeout	-1
Vel Constr. Reward Limit	0

4.4 Terminal State

Table 4 show the terminal states for the docking problem. The top row shows the agent's success condition and the subsequent rows show the various failure condition. Of particular note, the docking problem has a velocity constraint reward bound failure condition that terminates the episode when its reward reaches a predetermined lower bound of -5. This terminal condition creates a hard limit on the soft velocity constraint, preventing reward from growing negatively unbounded and pruning episodes that are clearly unable to respect this soft constraint at all. The reward is used as a proxy to measure degree of constraint compliance failure, although this terminal state does not necessarily need to be directly coupled to reward and is done so here for simplicity. The soft velocity constraint can be tightened and even made hard by decreasing the reward lower to zero.

Table 4. **Docking Terminal States**

Description	**Condition**
Success	$\|\boldsymbol{r}_d - \boldsymbol{r}_c\| \leq 0.5$, $\|\boldsymbol{v}_d\| \leq v_{limit}$
Crash	$\|\boldsymbol{r}_d - \boldsymbol{r}_c\| \leq 0.5$, $\|\boldsymbol{v}_d\| > v_{limit}$
Distance	$\|\boldsymbol{r}_d - \boldsymbol{r}_c\| > 40000$
Timeout	
2D/3D	$t > 2000$s
2D Oriented	$t > 3000$s
Vel Constr. Reward Limit	$\sum_{i=0}^{t} R_i^{vc} < -5$

Acknowledgments

This material is based upon work supported by the Air Force Research Laboratory Innovation Pipeline Fund. In particular, the authors would like to thank Sean Mahoney, whose support contributed to the success of this project. 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.

References

1. G. W. Hill, "Researches in the lunar theory," *American journal of Mathematics*, vol. 1, no. 1, pp. 5–26, 1878.
2. W. Clohessy and R. Wiltshire, "Terminal guidance system for satellite rendezvous," *Journal of the Aerospace Sciences*, vol. 27, no. 9, pp. 653–658, 1960.
3. M. L. Mote, C. W. Hays, A. Collins, E. Feron, and K. L. Hobbs, "Natural motion-based trajectories for automatic spacecraft collision avoidance during proximity operations," in *2021 IEEE Aerospace Conference*. Institute of Electrical and Electronics Engineers (IEEE), 2021, pp. 1–12.
4. K. Dunlap, M. Mote, K. Delsing, and K. L. Hobbs, "Run Time Assured Reinforcement Learning for Safe Satellite Docking," *AIAA SciTech Forum*, , Submitted.

5. U. Ravaioli, J. Cunningham, J. McCarroll, K. Dunlap, V. Gangal, and K. L. Hobbs, “Safe reinforcement learning benchmark environments for aerospace control systems,” *IEEE Aerospace*, 2022.