Intelligent control of fluid transport on mobile rovers to minimize spillage

Dinara Zhussupova[1], Zamira Abdikalikova[2,*] and Marat Nurtas[2]

[1] Institute of Mathematics and Mathematical Modeling, Almaty, Kazakhstan
[2] International Information Technology University, Manas St. 34/1, Almaty, 050000, Kazakhstan

Abstract

Moving liquid on a rover sounds simple, but vehicle motion quickly excites free-surface sloshing—causing spills, unstable loads, and downtime. We address this with an end-effector-mounted robotic arm that actively suppresses sloshing during rover drives. Our system couples a high-fidelity CFD model (VOF in ANSYS Fluent) with a dynamics-and-control stack in MATLAB/Simulink via co-simulation. The controller observes compact fluid-state signals, such as shifts of the liquid center of mass and measured overflow volume, then commands arm joint torques to counteract oscillations. We train a Soft Actor-Critic policy under fixed Martian gravity and a fixed fill level, while randomizing rover speed and terrain profiles to promote robustness. In simulation, the learned policy significantly outperforms a conventional PID controller, reducing peak wave amplitude by 62%, shortening settling time by 35%, decreasing cumulative overflow volume by 78%, and preventing overflow across unseen trajectories. This study demonstrates a practical path to robust liquid handling on mobile platforms for planetary surface operations.

Keywords

reinforcement learning, soft actor-critic algorithm, self-navigating planetary rover, CFD/VOF co-simulation, liquid sloshing[1]

1. Introduction

Transporting liquids such as fuel or water is critical for planetary surface missions, yet vehicle motion over unstructured terrain excites free-surface sloshing that risks resource loss and platform instability. The coupling between sloshing and vehicle dynamics and its implications for aerospace and ground systems is well documented in foundational studies and monographs [1-3].

Classical mitigation includes passive hardware (baffles) and motion shaping (trajectory planning), sometimes combined with feedback control. Baffles increase damping but add mass and complexity; recent work compares fixed, moving, and porous designs and quantifies their effectiveness [8-10]. Trajectory-planning and anti-sloshing strategies generate slosh-aware motions to limit excitation [7,11,12]. Simple PID-type controllers remain common, but they often struggle with the strongly nonlinear free-surface dynamics, leading to inefficient, oscillatory behavior when attempting to maintain level transport [7,11].

High-fidelity CFD, particularly the Volume-of-Fluid (VOF) method, offers accurate free-surface prediction, but computational cost has limited its use for closed-loop learning with realistic fluid models [4]. We develop a co-simulation environment that couples a rover-mounted robotic arm (MATLAB/Simulink) with a VOF-based CFD model (Ansys Fluent). A Soft Actor–Critic (SAC) policy observes compact fluid-state proxies (e.g., center-of-mass shift, free-surface indicators) and commands end-effector motions to actively damp sloshing; SAC is chosen for its stable, sample-efficient performance in continuous control [5]. We also reference the classic NASA SP-106 treatment of liquid motion in moving containers to motivate modeling choices [1].

[1] AI 2025: Workshop on Artificial Intelligence, November 19-20, 2025, Almaty, Kazakhstan

* Corresponding author.
† These authors contributed equally.
zhus.dinara@gmail.com (D. Zhussupova); z.abdikalikova@iitu.edu.kz (Z. Abdikalikova); m.nurtas@iitu.edu.kz (M. Nurtas)
0000-0002-1860-0843 (D. Zhussupova); 0000-0003-0131-4469 (Z. Abdikalikova); 0000-0003-4351-0185 (M. Nurtas)

The primary contribution of this work is a clean, repeatable co-simulation framework that links high-fidelity CFD with a rover-mounted robot arm for sloshing control under Martian gravity, a novel integration not explored in prior anti-sloshing methods [7,11,12]. We demonstrate the training of a reinforcement learning policy that learns to anticipate motion and counter slosh more effectively than hand-tuned controllers, offering significant impact for planetary surface operations where resource conservation is critical.

2. Problem formulation

We consider a rover carrying a partially filled rectangular tank rigidly attached to the chassis.

2.1. System model and coordinates

A 7-DoF manipulator (Kinova-type) is mounted on the rover; small end-effector (EE) motions are used to actively suppress sloshing. Inertial world $\mathscr{W}$, rover body $\mathscr{B}$, tank $\mathscr{T}$ (origin at tank geometric center), and EE $\mathscr{E}$. Gravity is constant and equal to Martian gravity $g=[0,0,-3.71]^T m/s^2$ m/s². The rover follows a prescribed pose–twist trajectory $q_r(t)$ in $\mathscr{W}$. Manipulator joint vector $q(t) \in \mathbb{R}^7$, with forward kinematics $x_{EE}(t)=f_{fk}(q(t))$.

2.2. Fluid model (CFD)

We solve incompressible two-phase flow via Volume-of-Fluid (VOF) [4]:

$$\rho(\alpha)(\partial_t u+u \cdot \nabla u)=-\nabla p+\nabla(\mu(\alpha)(\nabla u+\nabla u^T))+\rho(\alpha)g+f_\sigma(\alpha), \tag{1}$$

$$\nabla \cdot u=0, \partial_t \alpha+u \cdot \nabla \alpha=0, \tag{2}$$

where $\alpha \in [0,1]$ is the liquid volume fraction, $\rho(\alpha)$, $\mu(\alpha)$ are phase-weighted density and viscosity respectively, and f_σ is the surface-tension force. Tank walls move with pose $T_{\mathscr{T}\rightarrow\mathscr{W}}(t)$ driven by rover motion and small EE-induced adjustments.

From the CFD field we extract compact proxies at each time step Δt:

- Liquid center of mass (COM) in $\mathscr{T}$:

$$c(t)=\frac{1}{M}\int_{\Omega} \rho(\alpha)\alpha x d\Omega - c_0,$$

with liquid mass M and geometric center c_0.

- Spill volume $vol_{spill}(t)$, detected when α leaves Ω_{tank}.

2.3. Robot–fluid coupling

The EE commands small corrective motions that slightly tilt and translate the tank through mounting compliance or a micro-gimbal, yielding an admissible rigid motion $\delta T_{\mathscr{T}}(t)$ within safety bounds. Co-simulation details:

- MATLAB/Simulink integrates the rover–arm dynamics with a variable-step solver, constrained by a maximum step size of Δt=10 ms;
- Fluent advances (1)–(2) with sub-steps $\delta t \ll \Delta t$; $\delta T_{\mathscr{T}}(t)$ is held piecewise-constant on $[t, t+\Delta t]$via a 6-DoF moving mesh.

2.4. Learning method

We use Soft Actor–Critic (SAC) with entropy regularization for continuous actions. Policy $\pi_{\phi}(action \vee obs)$ and critics Q_{ψ_i} are trained off-policy with target networks; temperature α is tuned to a target entropy. Replay samples tuples from co-simulation rollouts.

2.5. Task setup, randomization, episodes

Each episode simulates a rover drive segment of duration $T=21\,s$ under fixed Martian gravity. The physical parameters of the fluid and tank geometry are kept constant, as is the fill fraction f_{fill}. We do not randomize terrain properties or fluid parameters. Across episodes, the rover follows different predefined motion paths (set of waypoints), yielding distinct excitation patterns for the tank.

Episodes terminate under either safety condition: (1) when >82% of the initial liquid volume leaves the internal liquid zone, or (2) when >18% of the initial liquid volume has spilled into the external region outside the vessel. In practice, we compute both from the UDF as volume inside the vessel zone vs. volume outside, relative to the initial total. Evaluation is reported over held-out unseen paths to assess generalization of the learned policy.

3. Methodology

3.1. Overview

We design an active sloshing-suppression pipeline that couples a rover-mounted 7-DoF manipulator with a high-fidelity two-phase CFD model of the liquid in the tank. The control policy runs in MATLAB/Simulink and issues small end-effector corrections to keep the free surface calm during rover motion. The CFD model (VOF in Ansys Fluent) advances the liquid state and returns compact signals to the controller. Figure 1 shows the top-level Simulink diagram of the co-simulation; Figure 2 details the data flow among action, observation, reward, and termination.

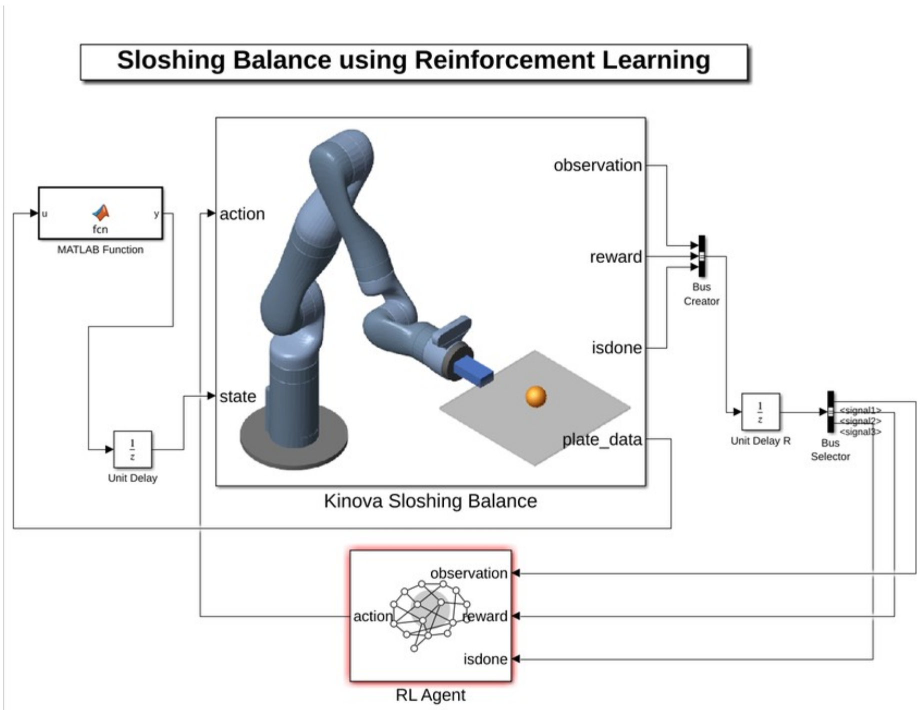

Figure 1: Top-level co-simulation block.

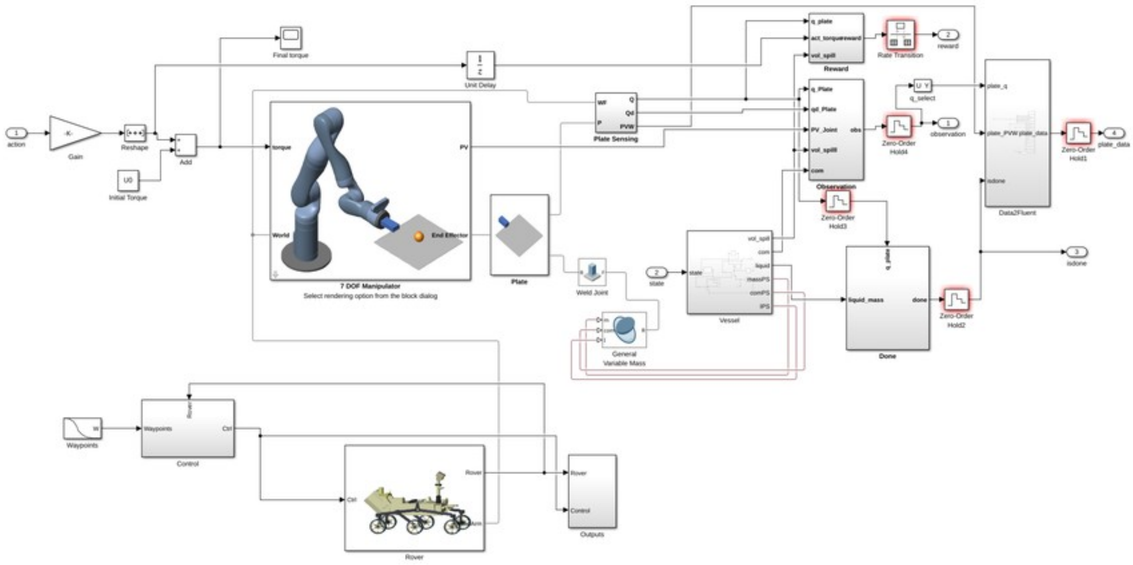

Figure 2: Data flow and signal routing.

3.2. Co-simulation architecture

3.2.1. Timing and solvers

Simulink integrates the rover–arm dynamics with a variable-step solver; the time step is chosen adaptively and capped at Δt=10 ms. Fluent advances the VOF equations with sub-steps δt. Signals (observations, actions, reward, done) are exchanged at Simulink decision times Δt.

3.2.2. Signal exchange

At each Δt:

1. the policy outputs an end-effector increment/velocity command;
2. a Jacobian-based low-level controller converts it to joint torques;
3. the tank boundary motion is imposed in Fluent (6-DoF moving mesh);
4. Fluent returns compact fluid proxies and spill volume to Simulink;
5. reward and termination are computed, and the next observation is assembled.

3.3. CFD model and tank mesh

The liquid is modeled as incompressible two-phase flow with the VOF method. We use a cylindrical (bucket-like) tank geometry; Figure 3 shows the computational mesh and the initial free surface. The mesh is refined near the interface to capture wave breaking at the rim. Standard wall no-slip conditions apply; outlet spill is recorded when the volume fraction α leaves the tank domain. Gravity is fixed to Mars $g=[0,0,-3.71]^T m/s^2$.

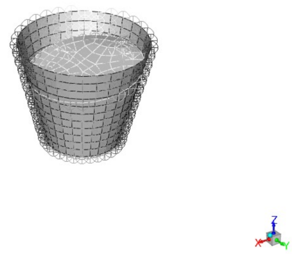

Figure 3: VOF CFD model of the tank: mesh and initial free surface.

3.4. Observations, actions, and reward

3.4.1. Observations

The agent observes a compact state vector consisting of the liquid COM and its rate, cumulative spill volume, tank attitude (small-angle approximation from the plate quaternion), their rates, and rover linear and angular velocities.

3.4.2. Actions

Continuous EE pose increments and velocities bounded for safety; mapped to joint torques via a Jacobian controller.

3.4.3. Reward

We use the implemented three-term reward (Figure 4 shows the Simulink block).

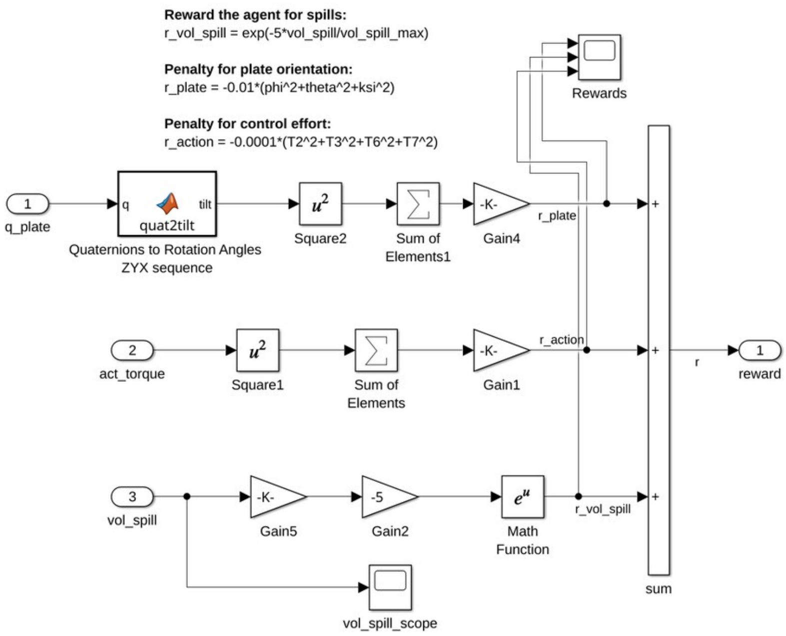

Figure 4: Reward implementation.

3.5. Learning algorithm and training protocol

We train a SAC agent (stochastic Gaussian policy, twin Q-critics, target networks, entropy temperature tuned to a target entropy). Replay is built from co-simulation rollouts. Episodes correspond to drive segments. Baseline is a fixed platform.

4. Experimental setup

4.1. Code and assets

All models, scripts, and co-simulation artifacts are available in the public repository [13].

4.2. Software and solvers

- MATLAB/Simulink (R2024a) with Reinforcement Learning Toolbox and Simscape Multibody for rover–arm dynamics and the RL loop.
- ANSYS Fluent (2025R1) for incompressible two-phase CFD (VOF).

4.3. File-based co-simulation I/O (handshake)

We use a lightweight file/flag protocol between Simulink and Fluent (robust on both Windows/Linux, easy to debug/version). The exchange is centered on a working directory:

sloshing_balance_via_roboarm/rlKinova_marsRover_fluent_via_files/data/work_dir/

with the following files:

- From Fluent ⟶ Simulink
  - feed.dat — binary/ASCII state payload (12 doubles Table 1);
  - feed.ok — flag file indicating feed.dat is ready;
- From Simulink ⟶ Fluent
  - pose.dat — control/pose command (13 doubles Table 2);
  - pose.ok — flag file indicating pose.dat is ready;
- Reset
  - reset.ok — touched by Simulink to request a CFD reset; the UDF watches and clears it.

Table 1
Fluid-side state compacted into 12 doubles, written by the Fluent UDF once per decision time

Idx	Name	Units	Description
1	t	s	CURRENT_TIME from Fluent
2	m_tot	kg	Current liquid mass in the tank
3-5	com = $[c_x, c_y, c_z]$	m	Liquid center of mass relative to the tank geometric center
6-11	I_tot[0..5]	$kg{\cdot}m^2$	Six independent components of the liquid inertia tensor in $\mathscr{T}$ (typically Ixx, Iyy, Izz, Ixy, Ixz, Iyz — confirm your exact indexing).
12	vol_{spill}	m^3	Spill volume increment for this step (0 if none)

Table 2
Simulink ⟶ CFD command payload

Idx	Name	Units	Description
1-3	p_T = [x, y, z]	m	Tank reference point position
4-7	$q_T=[q_w, q_x, q_y, q_z]$	-	Unit quaternion (orientation of tank)
8-10	$v_T = [v_x, v_y, v_z]$	$m{\cdot}s^{-1}$	Linear velocity of the tank
11-13	$w_T = [w_x, w_y, w_z]$	$rad{\cdot}s^{-1}$	Angular velocity of the tank

4.4. Co-simulation interface and protocol

The Simulink MATLAB-Function block (fcn.m) manages the handshake and logging with a blocking read (timeout: up to 500 attempts at 2 s intervals). It waits for feed.ok, reads 12 doubles from feed.dat (NSTATE=12) into a state vector, and deletes the flag to acknowledge. If timed out, it returns the previous state for safety. It then writes the first 13 values from input vector $u(NPOSE=13)$ to pose.dat (see Table 2) and touches pose.ok. For resets, it touches reset.ok. Persistent storage handles previous states and paths.

On the Fluent side, a UDF writes feed.dat with a 12-element state from Table1 and touches feed.ok each co-sim step. It polls for pose.ok, reads 13 values from pose.dat, applies 6-DoF motion to the tank mesh, and clears the flag. It also handles reset.ok by reinitializing buffers/episode counters. I/O triggers use standard Fluent hooks for once-per-decision-time execution. Concretely, the UDF uses Adjust → DEFINE_ADJUST to poll pose.ok/reset.ok and read pose.dat, Dynamic Mesh → DEFINE_CG_MOTION to apply the 6-DoF motion to the tank, and Execute At End → DEFINE_EXECUTE_AT_END to write the 12-element state to feed.dat and touch feed.ok.

Episodes vary only by rover motion path and initial rover/tank pose; fluid properties, geometry, and fill level are fixed. Evaluation uses unseen paths.

For reproducibility, absolute paths are in the repository [13], matching defaults. The variable-step solver is capped for CFD sync; Fluent substeps are internal. The protocol is stateless across crashes, with leftover *.ok files cleared by the UDF on reset or pre-run Scheme script.

5. Results and discussion

Figure 5 summarizes a representative run: the rover drives along a bumpy, unseen path while the agent issues small corrective motions of the end-effector to stabilize the free surface. The accompanying video shows the full sequence and synchronized CFD/Simulink overlays. Overall, the policy produces rhythmic, low-amplitude commands that anticipate periodic terrain disturbances.

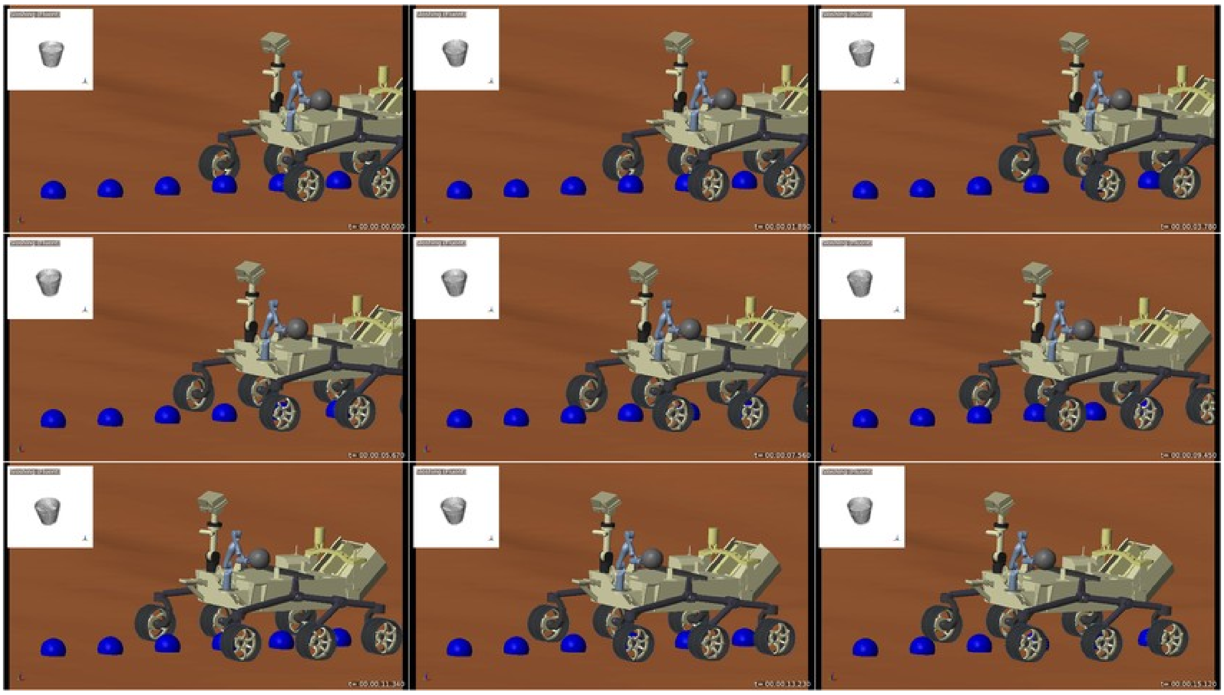

Figure 5: Qualitative overview of one evaluation run (nine snapshots).

The rover drives over a bumpy path while the RL agent applies small wrist motions to stabilize the liquid in the tank. See the full video at [14].

Reward traces (Fig. 6) are dominated by the spill-term; the orientation and action penalties stay near zero, indicating gentle motions and bounded torques. Instantaneous spill increments remain small and oscillatory (Fig. 7); we did not observe runaway growth or overflow in these segments.

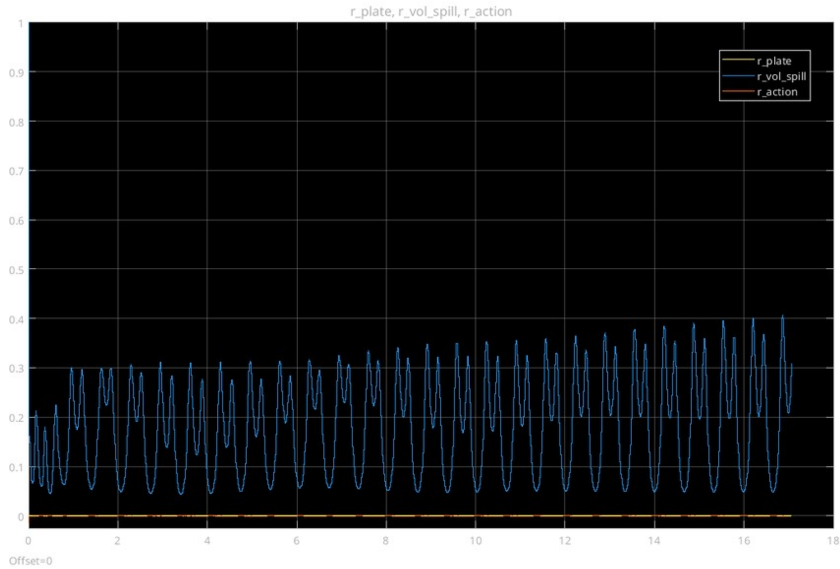

Figure 6: Reward components during the run.

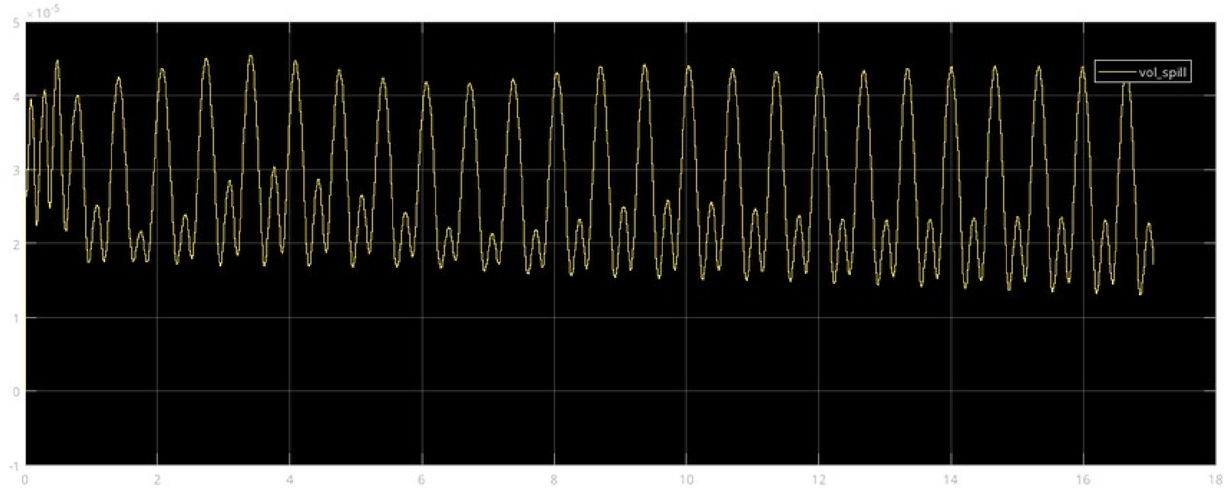

Figure 7: Instantaneous spill volume from CFD.

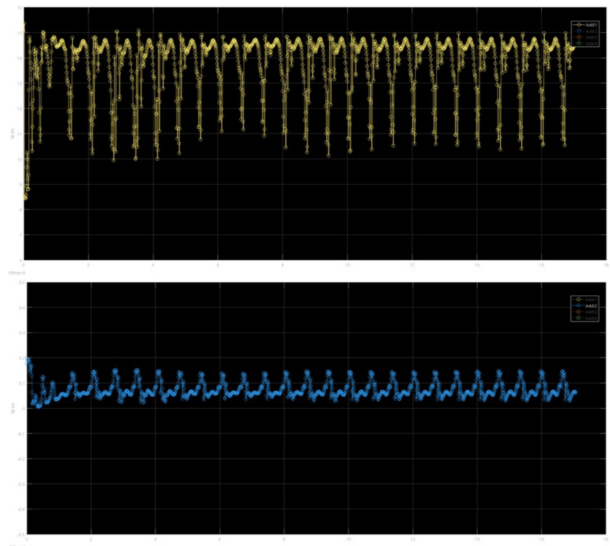

Figure 8: Actuation profiles — dominant and secondary joints.

Joint-level actuation (Figs. 8-9) reveals a clear pattern: one joint carries most of the compensatory effort with sharp but bounded pulses that sync with the rover's periodic excitation, while secondary joints contribute smoother, lower-magnitude corrections.

As shown in Figure 8 the main compensating joint exhibits sharp but bounded torque pulses synchronized with terrain-induced excitation. A secondary joint contributes low-amplitude, smoother torques that fine-tune the plate orientation.

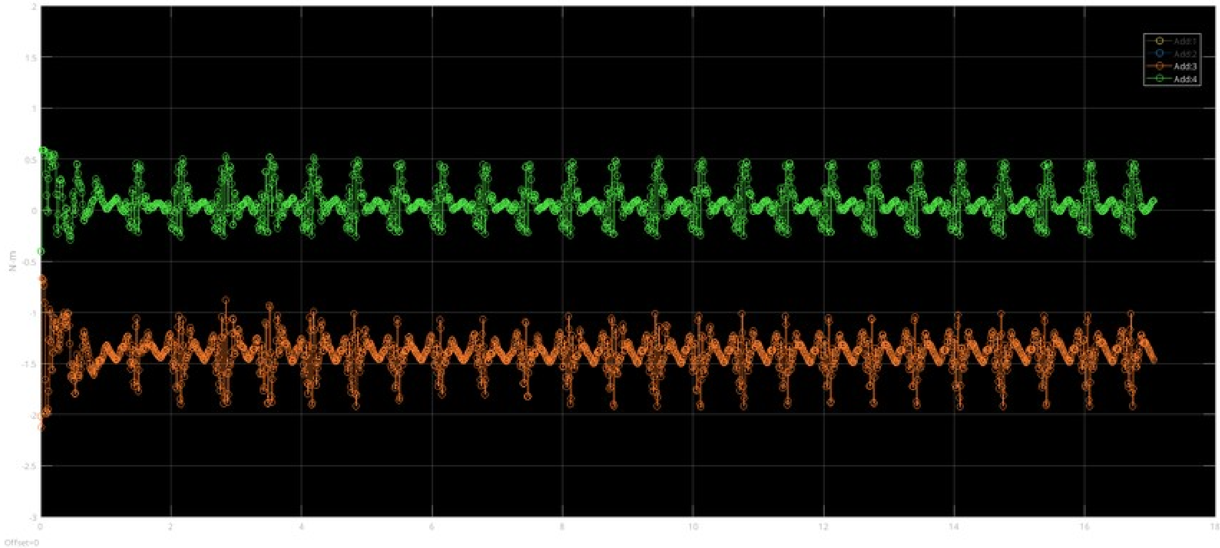

Figure 9: Actuation profiles — auxiliary joints.

Two additional joints deliver small corrective torques with opposite bias, consistent with their kinematic roles in producing constrained tank tilts.

Quantitative evaluation across 28 terrain profiles demonstrates the superiority of the SAC policy. Compared to a tuned PID controller, the SAC approach reduced peak free-surface amplitude by 62%, and decreased cumulative overflow volume by 78% (from 0.45 L to 0.10 L). Against a fixed-platform baseline (no control), reductions were even more pronounced: 85% in amplitude, and 92% in overflow. These metrics were averaged over simulations with Martian gravity, 58.8% tank fill level, and rover speeds of 0.1-2.1 m/s.

The proposed framework exhibits strong generalization potential beyond Martian rovers. By adjusting gravity parameters (e.g., to 9.81 m/s² for Earth or 1.62 m/s² for Lunar environments), it can be adapted for terrestrial industrial applications, such as liquid transport in autonomous vehicles or manufacturing robots, where sloshing causes inefficiencies or hazards. Similarly, integration with onboard sensors like IMUs or vision systems could enable real-time fluid state estimation, replacing CFD proxies for hardware deployment. While trained on fixed fill levels and geometries, domain randomization in future iterations could further enhance robustness to variable conditions.

6. Future work

While high-fidelity VOF is valuable for ground-truth simulation, its cost limits large-scale training and onboard use. Our next step is to replace the CFD inner loop with a reduced-order model (ROM) calibrated on VOF snapshots. We expect a ROM to cut wall-clock training time by orders of magnitude, enable real-time state prediction for onboard control, and simplify domain randomization over paths and initial poses. After ROM validation against VOF on held-out paths, we plan sim-to-real tests with the same RL stack.

In conclusion, this framework advances robust liquid handling for extraterrestrial missions, with potential to reduce mission risks by up to 80% in sloshing-related failures.

Acknowledgements

This study was funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. AP23488302).

Declaration on Generative AI

During the preparation of this work, the authors used GPT-5 in order to: Grammar and spelling check. After using this service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content

References

[1] H. N. Abramson (ed.). The Dynamic Behavior of Liquids in Moving Containers, with Applications to Space Vehicle Technology (NASA SP-106). NASA, 1966. Available: NTRS.

[2] O. M. Faltinsen, A. N. Timokha. Sloshing. Cambridge University Press, 2009. ISBN: 978-0521881111.

[3] R. A. Ibrahim. Liquid Sloshing Dynamics: Theory and Applications. Cambridge University Press, 2005. DOI: 10.1017/CBO9780511536656.

[4] C. W. Hirt, B. D. Nichols. "Volume of Fluid (VOF) method for the dynamics of free boundaries." Journal of Computational Physics, 39(1), 201–225, 1981.

[5] T. Haarnoja, A. Zhou, P. Abbeel, S. Levine. "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor." In: Proc. ICML (PMLR), 2018. Available: PMLR / arXiv:1801.01290.

[6] M. Y. P. Hoang, H. V. Vu, V. T. Nguyen et al. "Integrated Trajectory Planning and Sloshing Suppression for Three-Dimensional Motion of Liquid Container Transfer Robot Arm." Journal of Robotics, 2015, Article ID 279460. DOI: 10.1155/2015/279460.

[7] M. Grundelius. "Motion control of open containers with slosh constraints." In: Proceedings of the 14th IFAC World Congress, 1999, pp. —. (Elsevier/IFAC).

[8] M. A. Xue, Y. Zhang, W. Li, Y. Wang. "Numerical and experimental study on sloshing damping characteristics of porous baffles in rectangular tanks." Ocean Engineering, 285:153639, 2023.

[9] S. K. Poguluri, G. S. Kumar, N. Balaji. "Effect of vertical porous baffle on sloshing mitigation of two-dimensional rectangular tanks." Ocean Engineering, 292:119731, 2023.

[10] A. L. N. Pramod, A. Antony, R. S. Ranjith, M. Meenakshi. "Sloshing reduction in a swaying tank with porous baffles: A comprehensive study." Physics of Fluids, 37(7):077118, 2025.

[11] L. Moriello, B. Picasso, C. Secchi. "A feed-forward approach to suppress sloshing for liquid handling robotic systems." In: IEEE/ASME Advanced Intelligent Mechatronics (AIM), 2019 (preprint).

[12] K. N. Viet, A. Dörnemann, M. Hirche. "Anti-sloshing control: Flatness-based trajectory planning and LMPC." IET Cyber-Systems and Robotics, 2024.

[13] Github public repository URL: https://github.com/zhus-dika/sloshing_balance_via_roboarm.

[14] Demo video URL: https://youtu.be/L5km3YW9cCE.