OCTAVIUS: Reinforcement Learning for Fusion Plasma Control

Overview

OCTAVIUS (Optimized Controller for Tokamaks with A Virtual Intelligence Ushering Simulations) is a MATLAB-based reinforcement learning framework for tokamak plasma control. The environment models coupled particle and energy balance equations and applies **Deep Deterministic Policy Gradient (DDPG)** to learn continuous control policies for auxiliary heating and fueling.

The primary goal of OCTAVIUS is to control burn and track plasma energy, density, and fuel mix under simulated tokamak conditions. OCTAVIUS was developed and validated at Lehigh University's Plasma Control Lab to support this work, specifically enabling comparison of reinforcement learning–based fusion control methods with traditional approaches

---

How to Run OCTAVIUS (Important)

Main Entry Point

To run the system, **execute the Run file**:

```
OCTAVIUS_Network.m
```

This script: - Initializes tokamak parameters - Builds the environment and agent - Loads or trains the RL controller - Starts the simulation loop

---

Customization Guide (Most Important Parameters)

OCTAVIUS is highly configurable. The sections below highlight the parameters that control **simulation behavior**, **tokamak physics**, **training**, and **visualization**.

Simulation Control Parameters

Located in the *Simulation Control* section of the *environment* file.

These parameters govern time resolution, episode length, and how targets evolve during training or evaluation.

Parameter	Description	Typical Effect
Ts	Simulation time step (s)	Smaller → higher fidelity, slower training

Parameter	Description	Typical Effect
TT	Total simulation time per episode (s)	Longer episodes → more learning per run
target_length	Steps spent tracking each target	Controls steady-state duration
target_transition_time	Steps used to transition between targets	1 = instant step, >1 = smooth ramp
S_I_inj	Wall impurity injection rate	Used for impurity-driven scenarios

Tips: - Set `target_transition_time = 1` for abrupt control challenges - Increase `target_length` for steady-state burn control studies

---

Tokamak Geometry & Magnetic Parameters

These parameters define the physical machine being simulated. The default configuration is **ITER-like** and is the only fully tested scenario.

Key Tokamak Parameters

Parameter	Meaning
V	Plasma volume (m^3)
a	Minor radius (m)
R	Major radius (m)
epsilon	Inverse aspect ratio (a/R)
kappa	Plasma elongation
B_T	Toroidal magnetic field (T)
I_P	Plasma current (MA)
H_H	Confinement enhancement factor

Example Configurations

Predefined parameter sets are provided for: - **ITER** (default, validated) - JET - DIII-D - SPARC - ARC

Only the ITER configuration has been rigorously validated. Other machines are included for exploratory use.

Environment Details

State Space (12 Dimensions)

The observation vector includes normalized plasma states and targets:

- Electron energy density (Ee)
- Ion energy density (Ei)
- Deuterium, Tritium, Impurity, Alpha densities
- Total density and tritium fraction
- Target values for density, fraction, and energies

Normalization improves numerical stability and training efficiency.

Action Space (6 Dimensions)

All actions are continuous and normalized to **[0, 1]**:

Action	Physical Meaning	Range
ECRH	Electron heating	0–20 MW
ICRH	Ion heating	0–20 MW
NBI1	Neutral beam 1	0–16.5 MW
NBI2	Neutral beam 2	0–16.5 MW
S_D	Deuterium fueling	0–10^{19} $m^{-3}s^{-1}$
S_DT	D-T fueling	0–10^{19} $m^{-3}s^{-1}$

Reward Function

OCTAVIUS uses a **multi-scale Gaussian reward** to balance coarse stabilization and fine tracking accuracy:

[R = (e{-e2/2_1^2} + e{-e2/2_2^2} + e{-e2/2_3^2})]

Where e represents normalized tracking error in: - Electron energy - Ion energy - Total density - Tritium fraction

Scales: - σ_1 = 0.25 (coarse control) - σ_2 = 0.05 (medium precision) - σ_3 = 0.005 (high precision)

This structure encourages both stability and precision.

Visualization (Live Plotting)

Live plots can be enabled or disabled directly in the *Network* file.

Enable Live Plotting

Uncomment the desired plot functions:

```
plotEe(env);
plotEi(env);
```

```
plotN(env);
plotGamma(env);
plotHeating(env);
plotFueling(env);
```

Disable Plotting

Simply comment out these lines.

- Disabling plots significantly speeds up training.

Training Control

Enable / Disable Training

```
dotraining = 0;
```

Value	Behavior
0	Train agent from scratch or continue training
1	Skip training and load saved agent

Using Pre-Trained Agents

```
USE_PRE_TRAINED_MODEL = true;
```

When enabled: - Loads AGENT.mat - Continues training from previous weights - Preserves experience buffer (optional reset supported)

Saved agents should be placed in:

/savedAgents/

Running Old or Archived Agents

To evaluate a previously trained controller without retraining:

1. Set dotraining = 1

2. Ensure the desired AGENT.mat is in savedAgents/

3. Run OCTAVIUS_Run.m

This allows direct comparison of policies across experiments.

Thesis & Research Context

This codebase was developed and used in the doctoral thesis:

A Reinforcement Learning Approach to Burn Control in ITER
Ian Silvestri Ward

The thesis contains:

- Full model derivations

- Training methodology

- Validation experiments

- Discussion of limitations and future work

References

- Bosch, H.-S., & Hale, G. M. (1992). *Improved formulas for fusion cross-sections and thermal reactivities*. Nuclear Fusion, 32(4), 611.
- ITER Physics Expert Group on Confinement and Transport. (1999). Nuclear Fusion, 39(12), 2175.
- Lillicrap, T. P., et al. (2015). *Continuous control with deep reinforcement learning*. arXiv:1509.02971.
- Graber, V., & Schuster, E. (2022). *Nonlinear burn control in ITER*. Nuclear Fusion, 62(2), 026016.
- Ward, I. S. *A Reinforcement Learning Approach to Burn Control in ITER*.

Contact

For questions, issues, or collaboration:

- **Email:** ianwardengineer@gmail.com
- **GitHub Issues:** https://github.com/isw223/OCTAVIUS_Fusion/issues