Mosquito population Feedback Control with Deep Reinforcement Learning

CANUM: 30 Mai 2024

Kala Agbo bidi

Joint works with Luis Almeida, Jean-Michel Coron,
Amaury Hayat, Nathan Lichtlé

Sorbonne Université, Laboratoire Jacques-Louis Lions, LJLL, INRIA équipe CAGE

Vector-born Diseases

Vector-born Diseases

- Vector Borne Diseases involve at least 3 components :

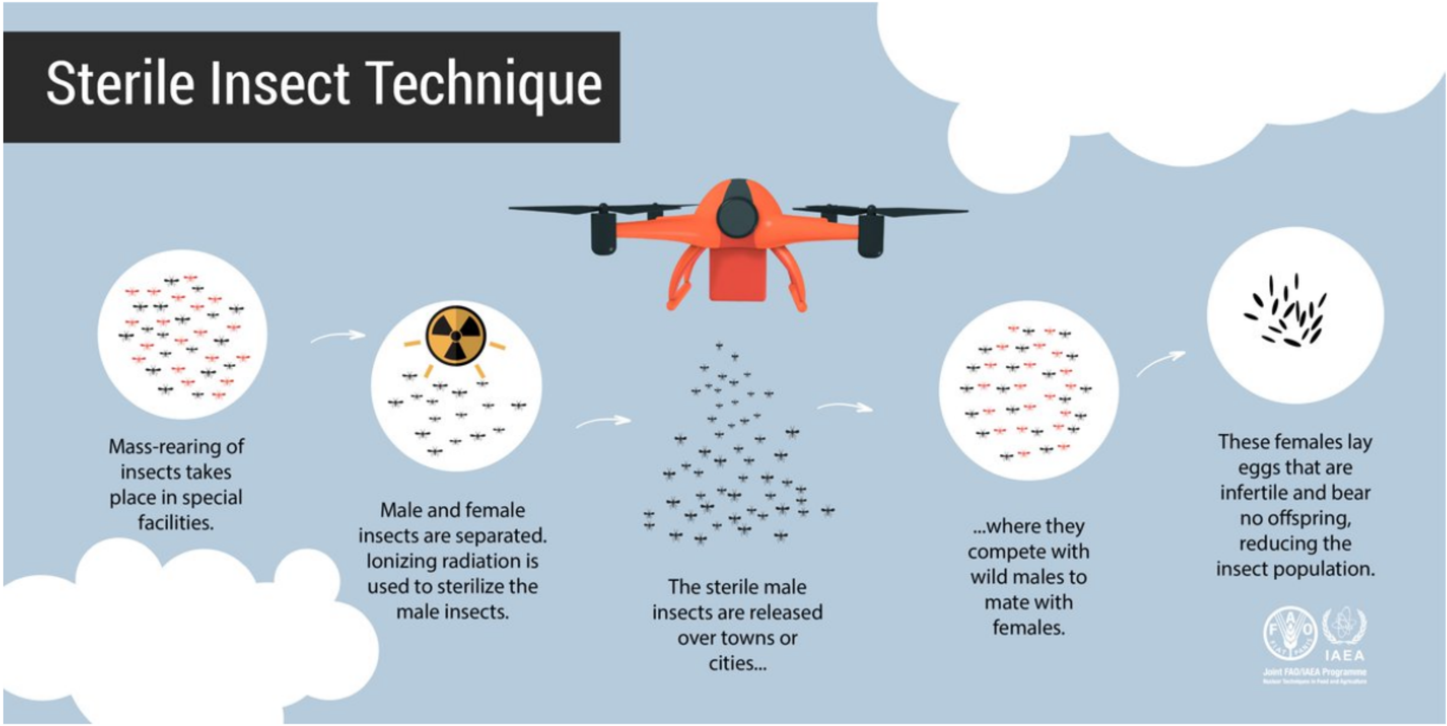
Vector-born Diseases

- Vector Borne Diseases involve at least 3 components :
- Vectors: usually bloodsucking arthropods like mosquitoes (malaria, dengue, chikungunya, yellow fever, west Nile fever), ticks (Lyme disease), fleas (Black Death)....
- Hosts (Humans)
- Pathogens: plasmodium (malaria), virus (dengue, chikungunya, Zika), bacteria (Yersinia pestis for the Black Death)

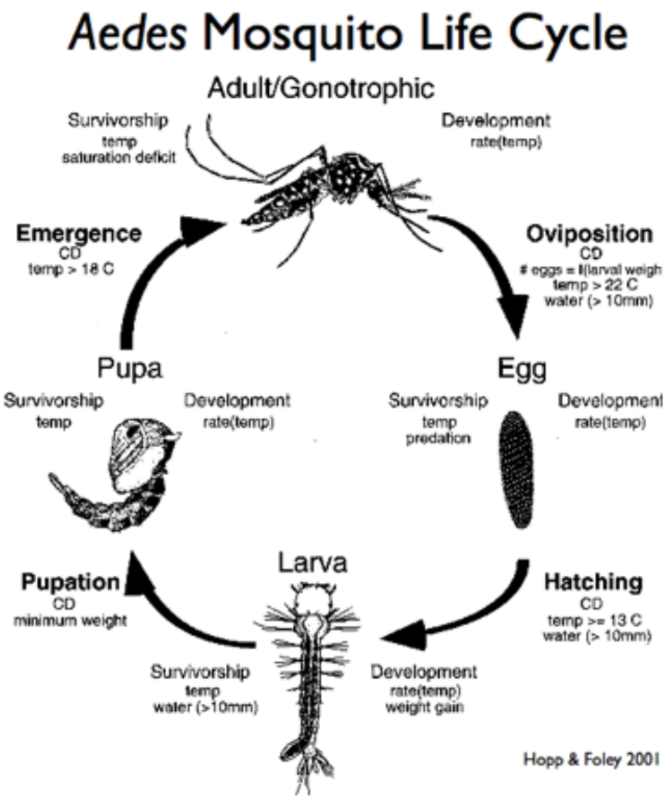
Vector-born Diseases

- Vector Borne Diseases involve at least 3 components :
- Vectors: usually bloodsucking arthropods like mosquitoes (malaria, dengue, chikungunya, yellow fever, west Nile fever), ticks (Lyme disease), fleas (Black Death)....
- Hosts (Humans)
- Pathogens: plasmodium (malaria), virus (dengue, chikungunya, Zika), bacteria (Yersinia pestis for the Black Death)
- Other species can be involved, for instance as reservoir (like birds for the west Nile fever or rodents for the Black Death)

Targeting sexual reproduction: SIT

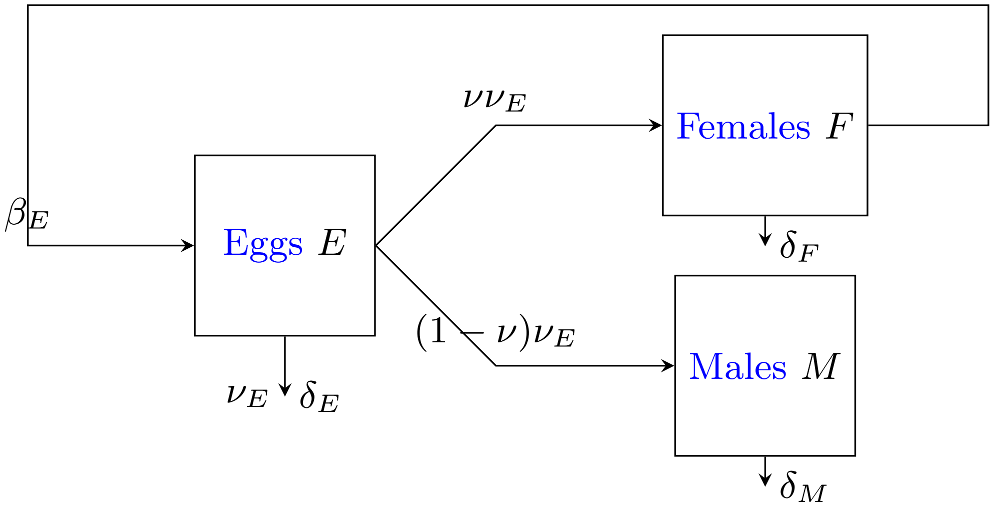


Mosquito life cycle



- Aquatic phase: egg (few days to several months) larvae (3 days to several weeks) pupa (1-3 days)
- Adult phase: (1 month)

Mosquito life cycle



- β_E birth rate;
- ν_E transition rate;
- $\delta_E, \delta_M, \delta_F$ death rates.

Mosquito life cycle

$$\dot{E} = \beta_E F \left(1 - \frac{E}{K}\right) - \big(\nu_E + \delta_E\big) E, \tag{1}$$

$$\dot{M} = (1-\nu)\nu_E E - \delta_M M, \tag{2}$$

$$\dot{F} = \nu \nu_E E - \delta_F F, \tag{3}$$

SIT mathematical model

Sterile Insect Technique (SIT): releases of sterilized male mosquitoes. The release function is denoted by u. Introduce a new compartment for sterilized males, denoted M_s. Probability for a female to meet a non-sterilized male is proportional to the proportion of them, with a preferential parameter, denoted γ_s

$$\dot{E} = \beta_E F\left(1 - \frac{E}{K}\right) - (\nu_E + \delta_E)E, \tag{4}$$

$$\dot{M} = (1-\nu)\nu_E E - \delta_M M, \tag{5}$$

$$\dot{F} = \nu\nu_E E \frac{M}{M + \gamma_s M_s} - \delta_F F, \tag{6}$$

$$\dot{M}_s = u - \delta_s M_s, \tag{7}$$

Backstepping feedback stabilization result

We defined for $\theta > 0$ and $\alpha > 0$

$$u((x^T, M_s)^T) := \max\left(0, G((x^T, M_s)^T)\right). \tag{8}$$

$$G((x^T, M_s)^T) := \frac{\gamma_s \psi E(\theta M + M_s)^2}{\alpha(M + \gamma_s M_s)(3\theta M + M_s)} + \frac{1}{\alpha}(\theta M - M_s) + \frac{((1-\nu)\nu_E \theta E - \theta\delta_M M)(\theta M + 3M_s)}{3\theta M + M_s} + \delta_s M_s, \tag{9}$$

$$\mathcal{R}(\theta) := \frac{\beta_E \nu \nu_E}{\delta_F(1 + \gamma_s\theta)(\nu_E + \delta_E)}. \tag{10}$$

Theorem

Assume that $R_\theta < 1$, then $\mathbf{0}$ is globally asymptotically stable in $\mathcal{D} = \mathbb{R}^4_+$ for the system (4)-(7) with the feedback law (8).

1

1

Agbo Bidi, Kala and Almeida, Luis and Coron, Jean-Michel (2023)
Global stabilization of sterile insect technique model by feedback laws
arXiv, 2307.00846

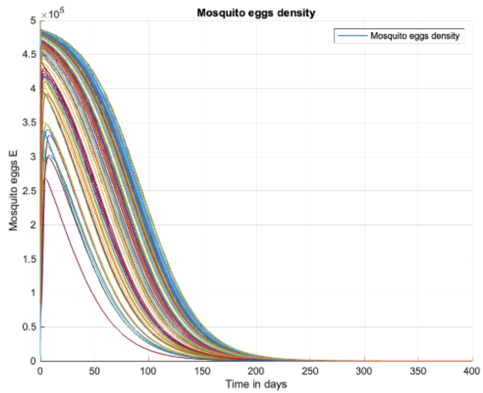
Numerical simulation

The parameters we use are given in the following table.

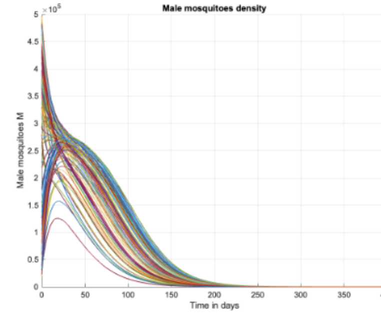
	Parameter name	Typical interval	Value	Unit
β_E	Effective fecundity	[7.46, 14.85]	8	Day^{-1}
ν_E	Hatching parameter	[0.005, 0.25]	0.25	Day^{-1}
δ_E	Aquatic phase death rate	[0.023, 0.046]	0.03	Day^{-1}
δ_F	Female death rate	[0.033, 0.046]	0.04	Day^{-1}
δ_M	Males death rate	[0.077, 0.139]	0.1	Day^{-1}
δ_s	Sterilized male death rate	-	0.12	Day^{-1}
ν	Probability of emergence	-	0.49	
K	Environmental capacity for eggs	-	22200	

- The condition $\mathcal{R}_\theta < 1$ is $\theta > 60,25$.
- We take $\theta = 290$ and $\alpha = 90$.

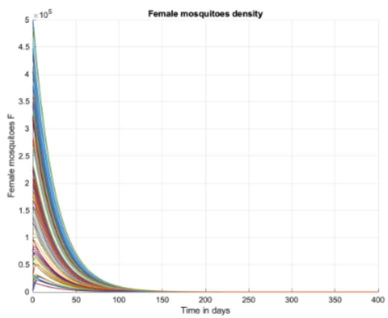
Numerical simulation



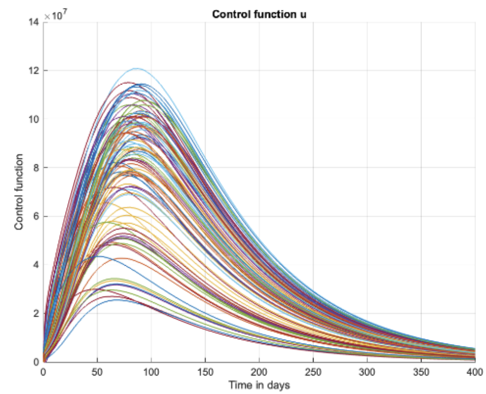
(a) Eggs density for robustness test.



(b) Males density for robustness test.



(c) Females density for robustness test



(d) Control function u for robustness test

Reinforcement Learning (RL) for Control design

[2]

Agbo bidi, Kala and Coron, Jean-Michel and Hayat, Amaury and Lichtlé, Nathan (2023)
Reinforcement Learning in Control Theory: A New Approach to Mathematical Problem Solving
The 3rd Workshop on Mathematical Reasoning and AI at NeurIPS'23

Reinforcement Learning (RL) for Control design

We denote by $F_s(t) = \frac{F(t)M_s(t)}{M(t)}$ the non-fecondated female density and

$$M_T(t) = M(t) + M_s(t) \longrightarrow \text{total males} \tag{11}$$

$$F_T(t) = F(t) + F_s(t) \longrightarrow \text{total females} \tag{12}$$

Mathematical open question: find $u(t) = f(M_T(t), F_T(t))$ where $f \in L^\infty(\mathbb{R}^2)$ such that $(0,0,0)$ is globally asymptotically stable and M_s is asymptotically small, i.e

$$\lim_{t\to+\infty} \|E(t), M(t), F(t)\| = 0, \quad \text{and} \quad \lim_{t\to+\infty} M_s(t) = \varepsilon \tag{13}$$

where ε can be chosen arbitrarily small (with f depend on ε) [2]

[2]

Agbo bidi, Kala and Coron, Jean-Michel and Hayat, Amaury and Lichtlé, Nathan (2023)
Reinforcement Learning in Control Theory: A New Approach to Mathematical Problem Solving
The 3rd Workshop on Mathematical Reasoning and AI at NeurIPS'23

RL for control design

$$\mathcal{U}^* := \frac{K\beta_E \nu(1-\nu)\nu_E^2\delta_s}{4(\delta_E+\nu_E)\delta_F\delta_M}\left(1-\frac{\delta_F(\nu_E+\delta_E)}{\beta_E\nu\nu_E}\right)^2 \tag{14}$$

Theorem (Almeida et al.2022)

If $u(t) = \mathcal{U} > \mathcal{U}^*$, *then* $(0, 0, 0, U^*/\delta_s)$ *is globally asymptotically stable for the system* (Σ).

RL for control design

$$\mathcal{U}^* := \frac{K\beta_E\nu(1-\nu)\nu_E^2\delta_s}{4(\delta_E+\nu_E)\delta_F\delta_M}\left(1-\frac{\delta_F(\nu_E+\delta_E)}{\beta_E\nu\nu_E}\right)^2 \tag{14}$$

Theorem (Almeida et al.2022)

If $u(t) = \mathcal{U} > \mathcal{U}^*$, then $(0,0,0,U^*/\delta_s)$ is globally asymptotically stable for the system (Σ).

- Applying constant control does not consider the population decrease after an intervention.

RL for control design

$$\mathcal{U}^* := \frac{K\beta_E\nu(1-\nu)\nu_E^2\delta_s}{4(\delta_E+\nu_E)\delta_F\delta_M}\left(1-\frac{\delta_F(\nu_E+\delta_E)}{\beta_E\nu\nu_E}\right)^2 \tag{14}$$

Theorem (Almeida et al.2022)

If $u(t) = \mathcal{U} > \mathcal{U}^*$, then $(0,0,0,U^*/\delta_s)$ is globally asymptotically stable for the system (Σ).

- Applying constant control does not consider the population decrease after an intervention.
- By deep learning the SIT model and its evolution, we want the control to adjust instead of remaining constant.

RL for control design

$$\mathcal{U}^* := \frac{K\beta_E \nu(1-\nu)\nu_E^2\delta_s}{4(\delta_E+\nu_E)\delta_F\delta_M}\left(1-\frac{\delta_F(\nu_E+\delta_E)}{\beta_E\nu\nu_E}\right)^2 \tag{14}$$

Theorem (Almeida et al.2022)

If $u(t) = \mathcal{U} > \mathcal{U}^*$, then $(0,0,0,U^*/\delta_s)$ is globally asymptotically stable for the system (Σ).

- Applying constant control does not consider the population decrease after an intervention.
- By deep learning the SIT model and its evolution, we want the control to adjust instead of remaining constant.
- We want to find the explicit formula of this control

1. **Discretize the equations** in a numerical scheme and use those dynamics to create a training environment by implementing the observations, actions, and rewards.
2. **Train an RL model** that learns to maximize the objective function we assign it through many simulations and obtain a numerical control feedback based on this numerical scheme.
3. **Recover an explicit mathematical control** from the numerical control feedback.
4. **Perform several tests** using different numerical schemes and discretizations to ensure that the explicit control is efficient.

Reinforcement Learning (RL) for Control design

We model the environment using the usual formalism of a partially-observable Markov decision process (POMDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, T, R, \gamma, \mu, \Omega, \mathcal{O})$ where

- $\mathcal{S} \subseteq \mathbb{R}^n$ is a set of states,
- $\mathcal{A} \subseteq \mathbb{R}^m$ a set of actions,
- $T : \mathcal{S} \times \mathcal{A} \to \Delta\mathcal{S}$ is the state transition function (ie. $T(s'|s,a)$ is the probability of transitioning to state s' given state s and action a),
- $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function,
- $\gamma \in [0,1)$ is the discount factor,
- $\mu \in \Delta(\mathcal{S})$ is the initial state distribution,
- $\Omega \subseteq \mathbb{R}^p$ is a set of observations of the hidden state,
- $\mathcal{O} : \mathcal{S} \to \Delta(\Omega)$ is the observation distribution.

Reinforcement Learning (RL) for Control design

The goal for the agent is to learn a policy $\pi_\theta : \Omega \to \Delta(\mathcal{A})$ (stochastic in our case) mapping observations to actions, where θ are the parameters of the policy (typically the weights of a neural network in the case of deep RL), which maximizes the expected discounted sum of rewards

$$J(\pi_\theta) = \mathrm{E}_{\tau \sim (\pi_\theta, \mathcal{M})} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right] \tag{15}$$

where the expectation is taken over all trajectories $\tau = (s_t, a_t, r_t)_{t \geq 0}$ generated by the current policy π_θ acting in the POMDP $\mathcal{M}$

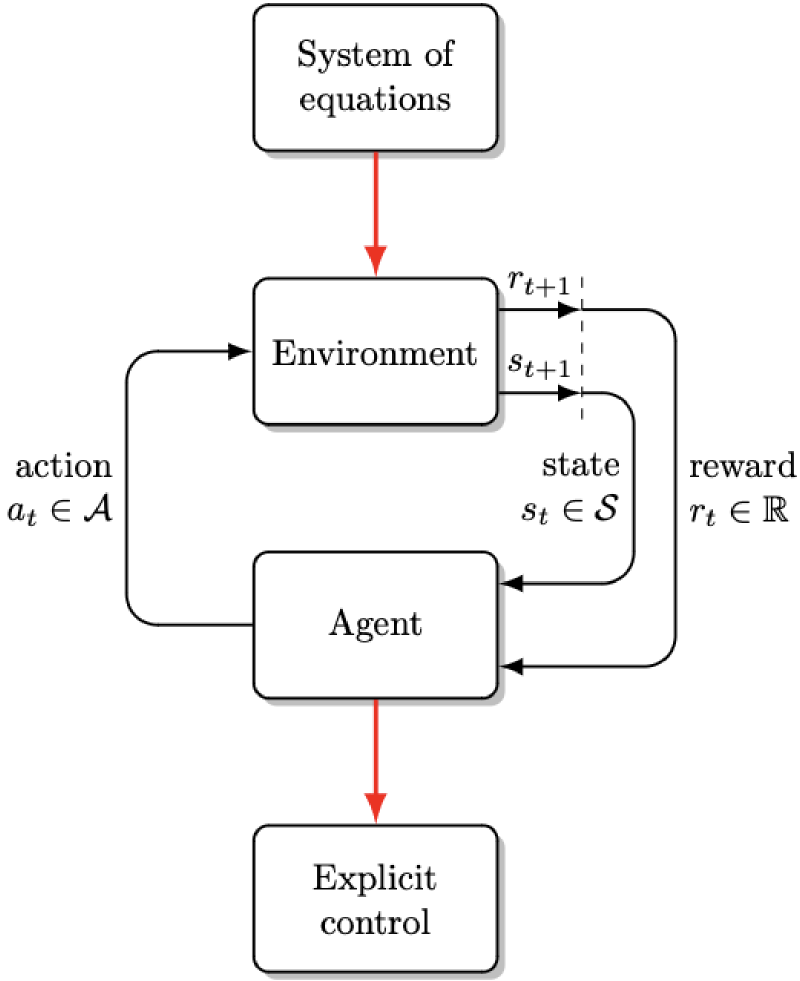


Diagram representing the procedure by which we simulate our model in an environment that is used to train an RL agent, whose policy we then convert into an explicit control.

Reinforcement learning framework

We learn a control $u(t) = \pi_\theta(s_t)$ that maximizes the expected return $\mathbb{E}\left[\sum_t \gamma^t r_t\right]$ with $\gamma \in (0,1)$.

Observations

We observe a combination of:

- Total males $M_T(t)$
- Total females $F_T(t)$

For better convergence, we provide states are at several scales then normalize them:

$$s_t = \left(\frac{\min(M_T(t),k)}{k}, \frac{\min(F_T(t),k)}{k}\right)_{k\in\mathcal{K}}$$

Reward function

We penalize the norm of the states:

$$r_t = c_1\|E(t), M(t), F(t)\|_2 + c_2(t)\|M_s(t)\|_2$$

Around the end of the horizon, we penalize $M_s(t)$ to encourage training convergence.

$$c_2(t) = \begin{cases} c_3 & \text{if } t < 0.9T, \\ c_3 + c_4 & \text{otherwise.} \end{cases}$$

$$J(u) = \int_0^T c_1\|E(t), M(t), F(t)\|_2 + c_2(t)\|M_s(t)\|_2 dt. \qquad (16)$$

Reinforcement Learning (RL) for Control design

Training algorithm

for iteration $= 1, 2, \ldots$ **do**
  for rollout $= 1, 2, \ldots, 12$ **do** *in parallel*
    Generate a random initial condition $(E, M, F, M_s)(0) \in [0, 50000]^4$
    for $D = 0, 1, \ldots, 1000$ days **do**
      Get control action $U(D) = \pi_\theta(s_t) \in [0, 500000]$
      Run 700 simulation steps (1 week) using $u(t) = U(D)$
      Compute reward r_t
    end for
  end for
  Optimize PPO loss[2] wrt. θ to maximize expected return using data collected during rollouts
  $\theta \leftarrow \theta_{\text{optimized}}$
end for

[2] Schulman, John, et al. "Proximal policy optimization algorithms", arXiv preprint arXiv:1707.06347 (2017).

Learned control

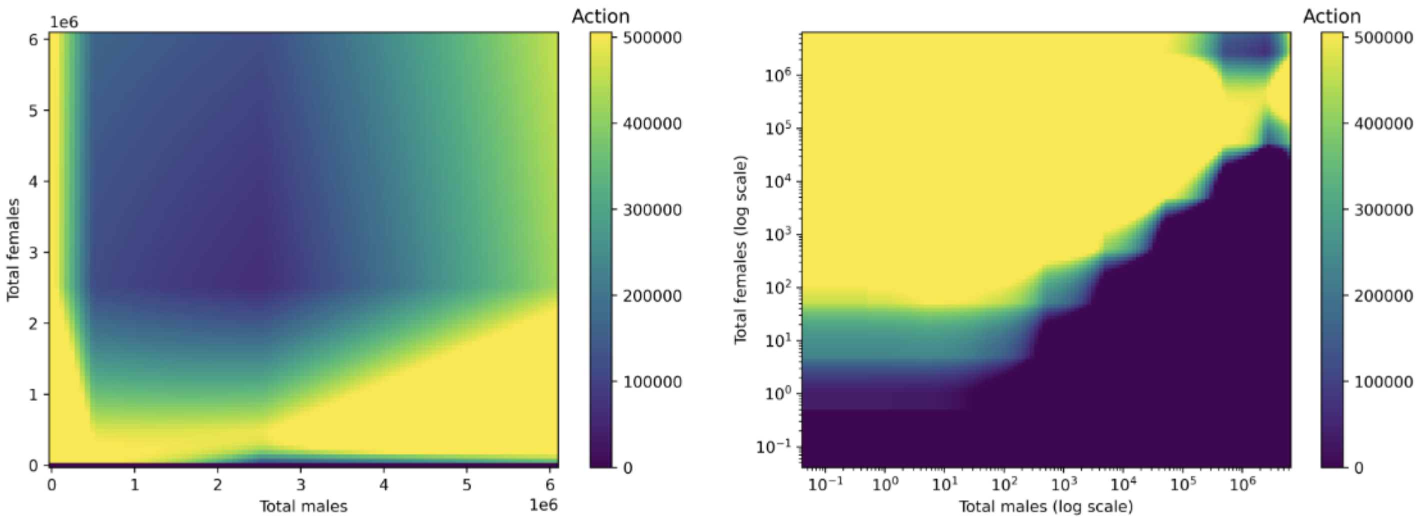


Figure: Heatmap of the model's action $u(M + M_S, F + F_s)$ as a function of total males and total females, in linear scale (left) and logarithmic scale (right).

Main result

We approximate this numerical control with the explicit mathematical control

$$u_{\text{reg}}(M+M_s,F+F_s)=\begin{cases}u_{\text{reg}}^{\text{left}}(M+M_s,F+F_s) & \text{if } M+M_s<M^*,\\ u_{\text{reg}}^{\text{right}}(M+M_s,F+F_s) & \text{otherwise,}\end{cases} \tag{17}$$

where u is defined on $(0,+\infty)^2$ and

$$u_{\text{reg}}^{\text{left}}=\begin{cases}u_{\min} & \text{if } I_1(F+F_s)>\alpha_2,\\ u_{\max}(\alpha_2-I_1) & \text{if } I_1\in(\alpha_1,\alpha_2],\\ u_{\max} & \text{otherwise,}\end{cases}\quad u_{\text{reg}}^{\text{right}}=\begin{cases}u_{\min} & \text{if } I_2>\alpha_2,\\ u_{\max}(\alpha_2-I_2) & \text{if } I_2\in(\alpha_1,\alpha_2],\\ u_{\max} & \text{otherwise.}\end{cases}$$

where $I_1(x)=\frac{\log(M^*)}{\log(x)}$ and $I_2(x,y)=\frac{\log(x)}{\log(y)}$, $M^*=200$, $\alpha_1=3$, $\alpha_2=4$, $u_{\max}=3\cdot 10^5$ is imposed by physical constraints and $u_{\min}$ can be chosen.

Numerical simulations

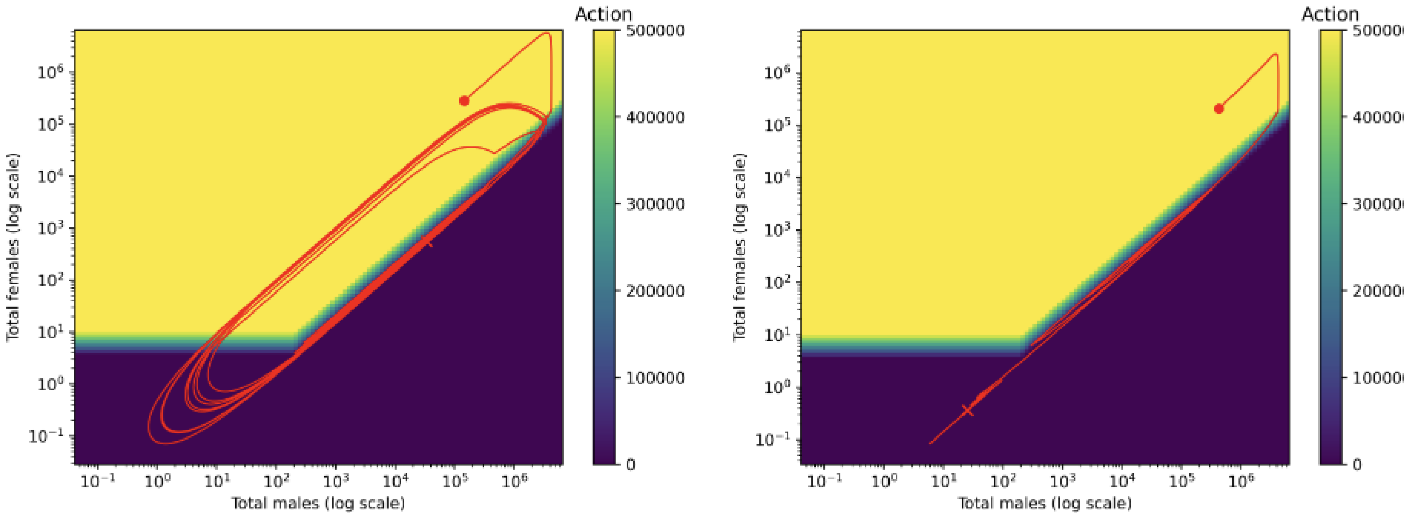


Figure: Heatmap of the regression model's action $u(M + M_s, F + F_s)$ as a function of total males and total females. A state-space trajectory is plotted in red, with the dot indicating initial state and the cross final state, for the heatmap only (left) and when a small noise $\mu \sim \mathcal{N}(0, 5)$ is added on top of the action (right)

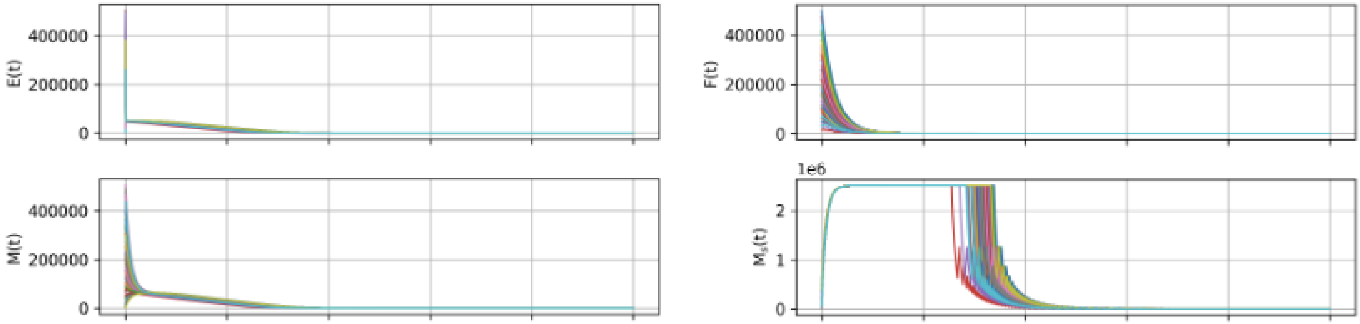
Numerical simulations

$$\textbf{Final control:} \quad U_{\text{reg}}(M_T, F_T) = \begin{cases} \varepsilon\delta_s & \text{if } \frac{\log(M_T)}{\log(F_T)} > 12, \\ u_{\max} & \text{otherwise,} \end{cases} \tag{18}$$

Numerical simulations

$$\textbf{Final control:} \quad U_{\text{reg}}(M_T, F_T) = \begin{cases} \varepsilon\delta_s & \text{if } \frac{\log(M_T)}{\log(F_T)} > 12, \\ u_{\max} & \text{otherwise,} \end{cases} \tag{18}$$

Using this control, $M_s(t)$ is large at first then quickly decreases as did other states:



Final control stats	200 days	400 days	600 days	800 days
average $\lvert E\rvert + \lvert M\rvert + \lvert F\rvert$	49×10^3	689	2.47	0.002
average $\lvert M_s\rvert$	2.5×10^6	129×10^3	2204	41.67

Numerical simulation

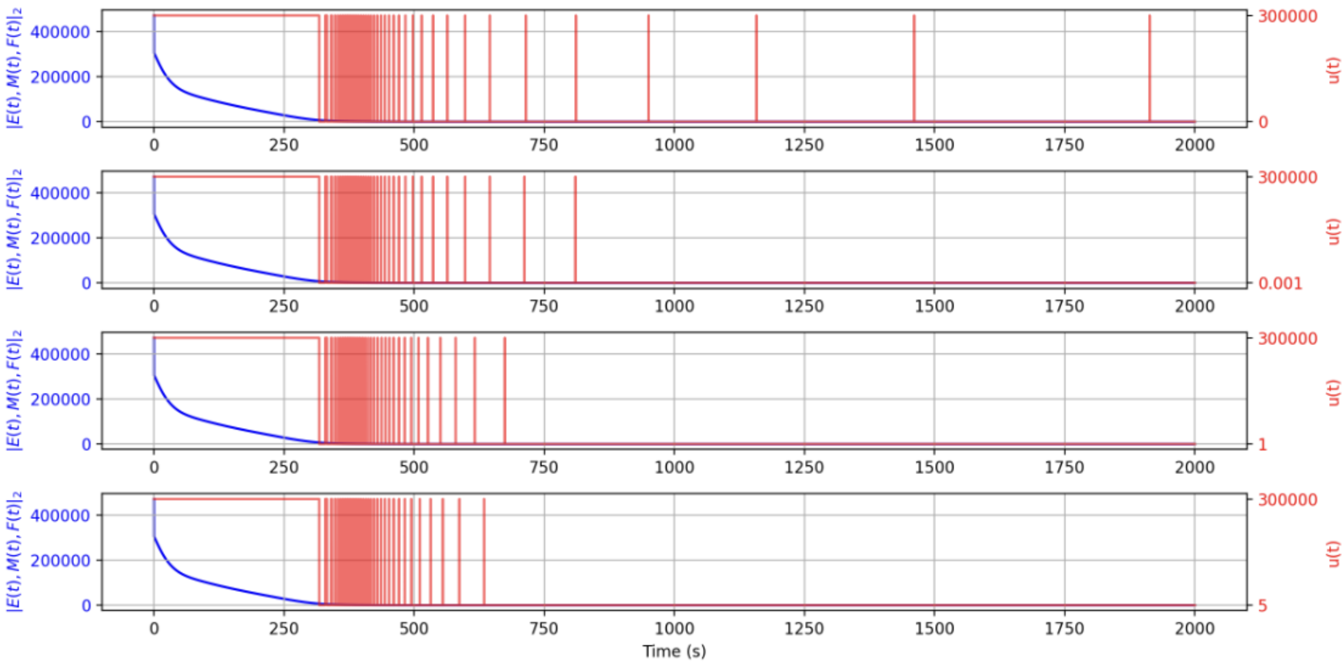


Figure: Norm of the states $\|E(t), M(t), F(t)\|_2$ (blue) and control v_{reg} (red) as a function of time for different values of u_{min} (0, 0.001, 1, and 5 respectively from top to bottom) and $u_{\text{max}} = 300000$, over 2000 days and with the same initial condition.

Conclusion & Perspective

Our results highlight the usefulness of machine learning and control theory in developing effective control strategies for complex biological systems. Further research in this field could lead to even more powerful techniques.

Conclusion & Perspective

Our results highlight the usefulness of machine learning and control theory in developing effective control strategies for complex biological systems. Further research in this field could lead to even more powerful techniques.

Thank you!