Conceptual Road Map

Model Structure

The model consists of a nested structure:

1. Outer loop: Climate policy maker (social planner)
2. Inner loop: Two-Period General Equilibrium(GE) Model

Markov Decision Process Formalization

The system evolves as a Markov Decision Process (MDP) where:

$$S_t = f(a_t | S_{t-1})$$

Where:

- S_t is the state vector at time t containing:
    - Economic variables from GE model (Y_t, C_t, etc.)
    - Cumulative emissions $E_t = \sum_{i=0}^{t} \eta_i Y_i$
    - Current damage function parameters θ_t
- a_t is the policy action vector: $[\tau_t, \tau_{t+1}]$
- $f(\cdot)$ is the transition function (Two-Period GE model)

Policy Maker's Problem

Objective Function

The policy maker maximizes expected social welfare:

$$\max_{a_t} \mathbb{E} \left[\sum_{t=0}^{\infty} \beta^t \left(u(S_t) - D(E_t, \theta_t) Y_t \right) \right]$$

Where:

- β is the social discount factor
- $u(S_t)$ is the social utility function

- $D(E_t, \theta_t)$ is the damage function
- Y_t is aggregate output
- E_t is cumulative emissions
- θ_t is the vector of damage function parameters

Damage Function Learning

The damage function uncertainty evolves as:

$$\theta_t = g(E_t, \theta_{t-1}, \epsilon_t)$$

Where:

- $g(\cdot)$ represents the learning process
- ϵ_t is new information received in period t
- Uncertainty band narrows as $t \to \infty$
- Distribution shape depends on E_t

Policy Action Space

$a_t \in A$ where A is the discrete action space:

$$A = \{\tau_t \pm \{5, 10\} \text{ basis points}\}$$

Agent Expectations

Agents in the GE model form expectations of future carbon taxes:

$$\mathbb{E}_t[\tau_{t+1}] = h(a_t, \Omega_t)$$

Where:

- Ω_t is the information set at time t containing:
  - Policy maker's announced rates $[\tau_t, \tau_{t+1}]$
  - Policy maker's objective function
  - Current state S_t
- $h(\cdot)$ is the expectation formation function

Time Structure

1. Period t begins with state S_t

2. Policy maker observes S_t and chooses $a_t = [\tau_t, \tau_{t+1}]$
3. Agents observe a_t and form expectations
4. GE model solves for equilibrium
5. Damages realized, new information ϵ_t received
6. System transitions to S_{t+1}

Implementation Plan

1. State Management

```
mutable struct StateNode
    time::Int
    economic_state::Dict{String, Float64}  # Y_t, C_t, etc.
    emissions::Float64                     # E_t
    damage_params::Vector{Float64}         # θ_t
    policy_actions::Vector{Float64}        # [τ_t, τ_{t+1}]
    parent::Union{Nothing, StateNode}
    children::Vector{StateNode}
    debug_log::Vector{String}
end
```

2. Transition Function Integration

1. **GE Model as Inner Loop**
   - Use existing GE model as $f(a_t|S_{t-1})$
   - Extend equilibrium computation to track emissions
   - Add damage calculations to economic outcomes
2. **State Transitions**

```
function state_transition(current_state, policy_action)
    # 1. GE Model Equilibrium
    GE_equilibrium = compute_equilibrium(policy_action)

    # 2. Emissions Accumulation
    E_t = current_state.emissions + η_t * GE_equilibrium["Y_t"]

    # 3. Damage Learning
    θ_t = damage_function_learning(current_state.damage_params, E_t)

    return new_state
end
```

3. Monte Carlo Policy Search

1. **Graph Building**
    - Start from initial state S_0
    - For each time period:
        - Generate policy action combinations
        - Compute state transitions
        - Track feasible paths
2. **Path Evaluation**

```
function evaluate_path(path)
    # Calculate expected social welfare:
    # Σ βᵗ(u(Sₜ) - D(Eₜ,θₜ)Yₜ)
    utility = sum(
        β^t * (social_utility(state) - damage_function(state))
        for (t, state) in enumerate(path)
    )
    return utility
end
```

4. Learning Implementation

1. **Damage Function Learning**

```
function damage_function_learning(θ_prev, E_t)
    # Update damage parameter distribution
    # Narrow uncertainty based on emissions
    # Return updated parameters
end
```

2. **Agent Expectations**

```
function form_expectations(policy_announcement, state)
    # Form expectations about future tax rates
    # Based on policy maker's history and current state
end
```

5. Simulation Framework

1. **Single Period**
    i. Start with state S_t
    ii. Policy maker chooses $[\tau_t, \tau_{t+1}]$
    iii. Agents form expectations
    iv. GE model computes equilibrium
    v. Update state (emissions, damages, learning)
    vi. Transition to S_{t+1}
2. **Monte Carlo Analysis**
    - Run multiple simulations
    - Collect feasible paths
    - Analyze policy effectiveness
    - Identify robust strategies

6. Analysis Tools

1. **Path Analysis**
    - Policy frequency analysis
    - Transition path statistics
    - Damage learning visualization
2. **Economic Outcomes**
    - Emissions trajectories
    - Output and welfare metrics
    - Uncertainty reduction tracking