Structural Integration Management Decision Making Modeling using RL

ISM Colloquium

Wanchang Zhang

Supervisor: Prof. Dr. Tom Lahmer

Institut für Strukturmechanik (ISM)
an der Bauhaus Universität Weimar

5. Dezember 2025

1. What is a Decision Making Process? an optimization problem

2. How to model the decision making process?
   - Markov Decision Process
     - Basic Terminology
     - Value functions definition
     - Bellman Equations for Value functions
     - On policy learning
     - Off-policy learning
   - Partially Observable Markov Decision Process

3. How to model Structural Integration Management as POMDP

Basic Idea: an optimization problem

Decision Making

Decision Making is about choice in everyone's life. We tend to make decisions like where to live, how we react etc. based on our experience instinction and constraints. Finally the decision would be made to maximize our benefits, no matter physically or mentally.

Seqeuntial Decision Making

Sequential Decision Making focuses on the decision process in the whole time-domain. It is about not only one choices but a sequence of choices. The decision made at every time step would dynamically influence the environment and the following state. Thus the decison would be made to maximize the accumulated total satisfaction of the whole time span.

How to model the decision making process?

The basic formulation of a decision making modelling

The decision making process is a dynamic interaction between decision makers (agents) and the environment. The basic elements would be the state space, action space, transition model, reward model. For the case when we can not observe the true state, but partially get some observations of the true state, we would add the observation space and observation model.

Typical Applications of Decision Making Modeling

- alpha-go chess playing: finite discrete state(discrete observation), discrete action, discrete time step
- Autonomous driving: continuous state like the velocity, location, continuous action space like the steering wheel angles, continuous time step
- Strucutral Integration Management

How to model the decision making process?

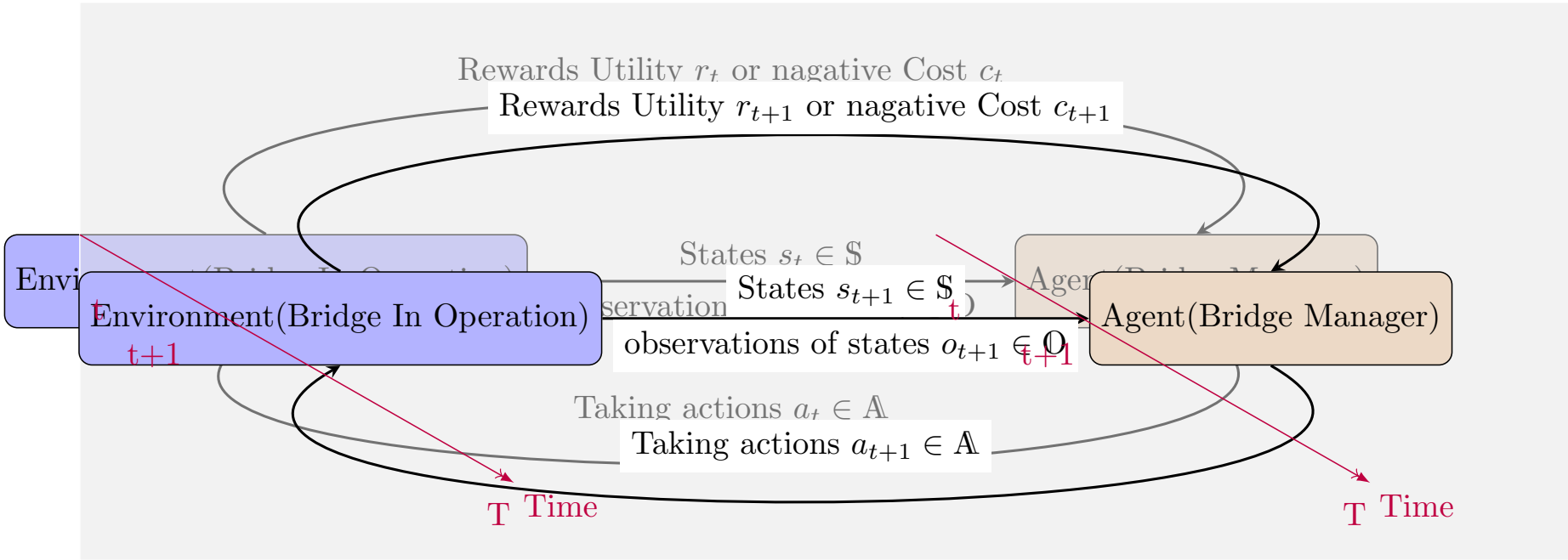


Abbildung 1: Sequential Decision Making Modeling in Structural Integration Management

Basic terminologies in Sequential Decision Making Analysis

In summary, a basic discrete decision making process with discrete time steps (aka. Markov Decision Process) can be generally defined by 4-tuple $(\mathbb{S}, \mathbb{A}, \mathrm{T}, \mathrm{R})$

- a finite set of states $\mathbb{S}$;
- a finite set of actions $\mathbb{A}$; $A_{i \geq t} \in \mathbb{A}$
- a policy $\pi(a|s)$: the state-dependent sequence of actions
  - Deterministic policy $\pi(a|s) : \mathbb{S} \to \mathbb{A}$
  - or Stochastic policy $\pi(a|s) : \mathbb{S} \times \mathbb{A} \to \mathbb{R} \in [0, 1]$

  Depending on how we choose the a_t we can have: Online-policy $A_{i \geq t} \sim \pi$ and Offline-policy: $A_{i > t} \sim \pi$, $a_t \sim \mu$
- a state transition model T:
  - It could be a deterministic transition function: e.g. $s_{t+1} = f(s_t, a_t)$;
  - or a transition probability: $P(s_{t+1}|s_t, a_t, \cdots, s_1, a_1)$ ($P(s_{t+1}|s_t, a_t)$ under the Markov Assumption)
- a reward function , R (negative cost function), could be formulated as:
  - A general reward function $R(s_t, a_t, s_{t+1})$
  - Or an immediate reward function $R(s_t, a_t)$

Value function definitions I

Accumulated Discounted Reward for one episode

$$U_t^\pi = U^\pi(s_t, a_t, \cdots, s_T, a_T) = \sum_{i=t}^{T} \gamma^{i-t} R(s_i, a_i, s_{i+1}), \tag{1}$$

where $\gamma \in [0, 1]$ is the discount factor weighting more on the current reward than the future reward.$\gamma = 0$: only the current reward matters, $\gamma = 1$: rewards in all steps equally matter.

Value of state-action pair following a policy π: $Q^\pi(s_t, a_t)$

$$Q^\pi(s_t, a_t) = \mathbb{E}_{s_{i>t} \sim \rho, A_{i>t} \sim \pi}[U_t^\pi | s_t, a_t] \tag{2}$$

Value of state following a policy π: $V^\pi(s_t)$

$$V^\pi(s_t) = \mathbb{E}_{a_t \sim \mu}[Q^\pi(s_t, a_t)] = \mathbb{E}_{s_{i>t} \sim \rho, A_{i>t} \sim \pi}[U_t^\pi | s_t] \tag{3}$$

Value function definitions II

Optimal Value of state-action pair: $Q(s_t, a_t)$

$$Q(s_t, a_t) = \max_{\pi} \mathbb{E}_{S_{i>t} \sim \rho} \left[\sum_{i=t}^{T} \gamma^{i-t} R(s_i, a_i, s_{i+1}) \middle| s_t, a_t \right] \tag{4}$$

Optimal Value of state: $V(s_t)$

$$V(s_t) = \max_{\pi} \mathbb{E}_{S_{i>t} \sim \rho} \left[\sum_{i=t}^{T} \gamma^{i-t} R(s_i, a_i, s_{i+1}) \middle| s_t \right] \tag{5}$$

Bellman Equations for general reward $R(s_t, a_t, s_{t+1})$ I

Value of state-action pair following a policy π: $Q^\pi(s_t, a_t)$

$$Q^\pi(s_t, a_t) = \mathbb{E}_{S_{t+1} \sim \rho}\left[R(s_t, a_t, s_{t+1}) + \gamma \mathbb{E}_{A_{t+1} \sim \pi} Q^\pi(S_{t+1}, A_{t+1})\right] \tag{6a}$$

$$= \sum_{s_{t+1} \in \mathbb{S}} p(s_{t+1}|s_t, a_t) \left[R(s_t, a_t, s_{t+1}) + \gamma \sum_{a_{t+1} \in \mathbb{A}} \pi(a_{t+1}|s_{t+1}) Q^\pi(s_{t+1}, a_{t+1})\right] \tag{6b}$$

Value of state following a policy π: $V^\pi(s_t)$

$$V^\pi(s_t) = \mathbb{E}_{A_t \sim \mu, S_{t+1} \sim \rho}\left[R(s_t, A_t, S_{t+1}) + \gamma V^\pi(S_{t+1})\right] \tag{7a}$$

$$= \sum_{a_t \in \mathbb{A}} \mu(a_t|s_t) \sum_{s_{t+1} \in \mathbb{S}} p(s_{t+1}|s_t, a_t)\left[R(s_t, a_t, s_{t+1}) + \gamma V^\pi(s_{t+1})\right] \tag{7b}$$

Bellman Equations for general reward $R(s_t, a_t, s_{t+1})$ II

Optimal Value of state-action pair: $Q(s_t, a_t)$

$$Q(s_t, a_t) = \mathbb{E}_{S_{t+1} \sim \rho} \left[R(s_t, a_t, S_{t+1}) + \gamma \max_{a_{t+1} \in \mathbb{A}} Q(S_{t+1}, a_{t+1}) \right] \tag{8a}$$

$$= \sum_{s_{t+1} \in \mathbb{S}} p(s_{t+1}|s_t, a_t) \left[R(s_t, a_t, s_{t+1}) + \gamma \max_{a_{t+1} \in \mathbb{A}} Q(s_{t+1}, a_{t+1}) \right] \tag{8b}$$

s

Optimal Value of state: $V(s_t)$

$$V(s_t) = \max_{a_t \in \mathbb{A}} \mathbb{E}_{S_{t+1} \sim \rho} \left[R(s_t, a_t, S_{t+1}) + \gamma V(S_{t+1}) \right] \tag{9a}$$

$$= \max_{a_t \in \mathbb{A}} \sum_{s_{t+1} \in \mathbb{S}} p(s_{t+1}|s_t, a_t) \left[R(s_t, a_t, s_{t+1}) + \gamma V(s_{t+1}) \right] \tag{9b}$$

Bellman Equations for intermediate reward $R(s_t, a_t)$ I

Value of state-action pair following a policy π: $Q^\pi(s_t, a_t)$

$$Q^\pi(s_t, a_t) = R(s_t, a_t) + \gamma \mathbb{E}_{S_{t+1} \sim \rho, A_{t+1} \sim \pi} \left[Q^\pi(S_{t+1}, A_{t+1}) \right] \tag{10a}$$

$$= R(s_t, a_t) + \gamma \sum_{s_{t+1} \in \mathbb{S}} \sum_{a_{t+1} \in \mathbb{A}} p(s_{t+1}|s_t, a_{t+1}) \pi(a_{t+1}|s_{t+1}) Q^\pi(s_{t+1}, a_{t+1}) \tag{10b}$$

Value of state following a policy π: $V^\pi(s_t)$

$$V^\pi(s_t) = \mathbb{E}_{A_t \sim \mu} \left[R(s_t, A_t) \right] + \gamma \mathbb{E}_{A_t \sim \mu, S_{t+1} \sim \rho} \left[V^\pi(S_{t+1}) \right] \tag{11a}$$

$$= \sum_{a_t \in \mathbb{A}} \mu(a_t|s_t) R(s_t, a_t) + \gamma \sum_{a_t \in \mathbb{A}} \mu(a_t|s_t) \sum_{s_{t+1} \in \mathbb{S}} p(s_{t+1}|s_t, a_t) V^\pi(s_{t+1}) \tag{11b}$$

Bellman Equations for intermediate reward $R(s_t, a_t)$ II

Optimal Value of state-action pair: $Q(s_t, a_t)$

$$Q(s_t, a_t) = R(s_t, a_t) + \gamma \mathbb{E}_{S_{t+1} \sim \rho} \left[\max_{a_{t+1} \in \mathbb{A}} Q(S_{t+1}, a_{t+1}) \right] \tag{12a}$$

$$= R(s_t, a_t) + \gamma \sum_{s_{t+1} \in \mathbb{S}} p(s_{t+1}|s_t, a_t) \max_{a_{t+1} \in \mathbb{A}} Q(s_{t+1}, a_{t+1}) \tag{12b}$$

Optimal Value of state: $V(s_t)$

$$V(s_t) = \max_{a_t \in \mathbb{A}} \{ R(s_t, a_t) + \gamma \mathbb{E}_{S_{t+1} \sim \rho} [V(S_{t+1})] \} \tag{13a}$$

$$= \max_{a_t \in \mathbb{A}} \left\{ R(s_t, a_t) + \gamma \sum_{s_{t+1} \in \mathbb{S}} p(s_{t+1}|s_t, a_t) V(s_{t+1}) \right\} \tag{13b}$$

From the Bellman Equation to Temporal Difference Learning

The Bellman equations can be seen as a recursive update rule
tell us the true value of a state or the true value of state-action pair would be, given the dynamics of the environment and the policy

On-policy Temporal Difference Learning I

Use $\hat{q}^{\pi}(s_t, a_t)$ to approximate State-action Value function:
$Q^{\pi}(s_t, a_t) = R(s_t, a_t) +$
$\gamma \sum_{s_{t+1} \in \mathbb{S}} \sum_{a_{t+1} \in \mathbb{A}} p(s_{t+1}|s_t, a_{t+1}) \pi(a_{t+1}|s_{t+1}) Q^{\pi}(s_{t+1}, a_{t+1})$

- we are in state s_t and we take action a_t
- we observe next state s_{t+1} and reward r_{t+1}
- we take the next action a_{t+1} from s_{t+1} following the policy (e.g. ϵ-greedy)

The algorithm SARSA (State-Action-Reward-State-Action) uses the experience tuple $(S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1})$ to form its TD target:
SARSA TD Target is

$$\hat{y}_t = R_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) \tag{14}$$

SARSA TD error is

$$\delta_t = \hat{y}_t - Q(s_t, a_t) \tag{15}$$

SARSA Update Rule:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \delta_t \tag{16}$$

On-policy Temporal Difference Learning II

Value of state following a policy π:
$V^{\pi}(s_t) = \sum_{a_t \in \mathbb{A}} \mu(a_t|s_t) R(s_t, a_t) +$
$\gamma \sum_{a_t \in \mathbb{A}} \mu(a_t|s_t) \sum_{s_{t+1} \in \mathbb{S}} p(s_{t+1}|s_t, a_t) V^{\pi}(s_{t+1})$

- we are in state s_t
- we take an action a_t (according to some policy e.g. ϵ-greedy)
- we observe the next state s_{t+1} and receive a reward r_{t+1}
- TD target $\hat{y}_t = r_{t+1} + \gamma V(s_{t+1})$. The target is a biased estimate of the true value of s_t, but it is often better than your current estimate $V(s_t)$
- TD error $\delta_t = \hat{y}_t - V(s_t)$
- TD (0) Update

$$V(s_t) \leftarrow V(s_t) + \alpha \delta_T \tag{17}$$

where α is the learning rate.

Off-policy Temporal Difference Learning I

Off-policy Temporal Difference Learning II

Optimal Value of state-action pair: StateActionRewardStateAction

$Q(s_t, a_t) = R(s_t, a_t) + \gamma \sum_{s_{t+1} \in \mathbb{S}} p(s_{t+1}|s_t, a_t) \max_{a_{t+1} \in \mathbb{A}} Q(s_{t+1}, a_{t+1})$

- we are in state s_t and we take action a_t
- we observe next state s_{t+1} and reward r_{t+1}
- we take the next action a_{t+1} from s_{t+1} following the policy π(e.g. ϵ-greedy)

The algorithm SARSA (State-Action-Reward-State-Action) uses the experience tuple $(S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1})$ to form its TD target:
SARSA TD Target is

$$\hat{y}_t = R_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) \tag{18}$$

SARSA TD error is

$$\delta_t = \hat{y}_t - Q(s_t, a_t) \tag{19}$$

SARSA Update Rule:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\delta \tag{20}$$

How to use the value functions to find the best policy?

Tabelle 1: Example of a table estimation for the optimal state-action value function $Q(S, A)$

$Q(S, A)$	A_1 do nothing	A_2 inspection	A_3 repair	A_4 replace
S_0 perfect state	300	50	-40	-300
S_1 minor damage	200	100	70	-100
S_2 major damage	-7	150	240	100
S_3 total failure	-500	-300	-300	200

If we already know such a value table, our acitons could be chosen guided by this table. If we are in thse major damage states, the repair action is most valuable choices since this optimal state-action value function is the largest.

Ways to solve MDP

Temporal Difference Learning

Derived from the Bellman Equation to recursively update the value approximation based on the new collected experience every time step

Dynamic Programming

Dynamic Programming requires a perfect model of environment, i.e. we need to know exactly the transition probability $T(s_{t+1}|s_t, a_t)$ and the reward model $R(s_t, a_t)$

Reinforcement Learning

RL could be used for model-free cases when we can not exactly describe the transition model and the reward model. The agent earn from the interaction with the environment via trial and errors. This requires a very good simulator.

Basic terminologies in POMDP

A discrete-time POMDP models the relationship between an agent and its environment. Formally, a POMDP is a $6-$tuple $(\mathbb{S}, \mathbb{O}, \mathbb{A}, \mathrm{T}, \mathrm{R}, \mathrm{O})$

- $\mathbb{S}$: a finite set of states
- $\mathbb{A}$: a finite set of actions
- $\mathbb{O}$: the finite set of observations
- T: a set of conditional transition probabilities between states
- R: $\mathbb{S} \times \mathbb{A} \to \mathbb{R}$: the reward function
- O: the set of conditional observation probabilities

At each time period, the environment is in some state $s \in \mathrm{S}$, the agent takes an action $a \in \mathbb{A}$, which causes the environment to transition to states s' with probability $T(s'|s, a)$. At the same time, the agent receives an observation $o' \in \mathbb{O}$ which depends on the new state of the environment, s' and the just taken action a with probability $O(o'|s', a)$ or sometimes $O(o'|s')$ depending on the sensor model. Finally the agent receives a reward r equal to $R(s, a)$ or $R(s, a, s')$ depending on the reward model. The goal is for the agent to choose actions at each time step that maximize its expected future discounted reward $E[\sum_{i=t}^{T} \gamma^t R_t]$

How to model Structural Integration Management as MDP I

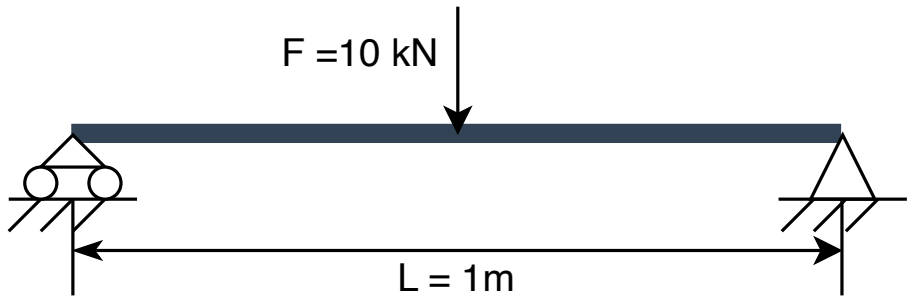


Tabelle 2: Configuration of Steel Truss Bridge Structure

Geometry Property	length L Cross Section A	$1m$ $1.0m^2$
Machanical Property	Youngs modulus E	$E(t=0) = 210e9Pa$
Material Property	Density ρ	$7800kg/m^3$
Loading	Mid point load F	10kN
Boundary Condition	$u_L = 0,\ u_R = 0$	

How to model Structural Integration Management as MDP I

To model as a POMDP model, we will think through the following components of the model

- States space $\mathbb{S}$: is a continuous spaces $\mathbb{R}^N$ containing the Youngs modulus for all elements $\mathrm{E} = [E_1, E_2, \cdots, E_{n_{elements}}]$. In other words, we will assume that the only changing parameters over time is the mechanical property Youngs modulus. The mass (usually depends on density and geometry) is constant if we assume that the density and geometry do not change over time.
- Action space $\mathbb{A}$: is a discrete space contains three elements a_0, a_1, a_2
  The state-dependent sequence of actions is defined as policy π. There could be two types of policies, the deterministic policy $\pi(a|s) : \mathbb{S} \to \mathbb{A}$ (mapping from the state space to the action space) and the stochastic policy $\pi(a|s) : \mathbb{S} \times \mathbb{A} \to \mathbb{R} \in [0, 1]$(mapping from the state space to the probability of actions PDF or PMF).
- Observation space $\mathbb{O}$: is a continous observation space containing the observation of the mid-point displacement observations or damage sensitive features.

How to model Structural Integration Management as MDP II

- Observation model $O(o_{t+\Delta t}|s_{t+\Delta t}, a_t)$:
  - Static analysis: Observation from the static analysis is the midpoint displacement. It is continuous observation. The displacement vector is calculated via

$$u_{t+\Delta t} = StaticSolver(K(E(t+\Delta t)), F_{t+\Delta t}). \tag{22}$$

    We could generate the synthetic observations by adding the noise

$$o_{t+\Delta t} = f(u_{t+\Delta t} + \mathrm{N}(0, \sigma^2)). \tag{23}$$

  - Dynamic analysis: We could generate the synthetic observation from dynamic analysis is the acceleration time series data.

$$acc_{t+\Delta t} = dynamicSolver(K(E(t+\Delta t)), F_{t+\Delta t}). \tag{24}$$

    We could use the Vibration-based SHM method to extract the damage sensitive feature from the acceleration time serires data.

$$DSF_{t+\Delta t} = DamageSensitiveFeatureExtraction(acc_{0:t+\Delta t}) \tag{25}$$

How to model Structural Integration Management as MDP III

- Reward Model r(s, a):
  Accumulated Discount Reward for the whole episode T is defined as the weighted sum of reward at each time step:

$$R = R(s_0, a_0, \cdots, s_T.a_T) = \sum_{i=0}^{T} \gamma^{i-t} R(s_i, a_i) \tag{26}$$

  where the discount factor $\gamma \in [0, 1]$. It weights more on the current reward than the future reward. When $\gamma = 0$: only the current reward matters; when $\gamma = 1$: rewards in all steps equally matter.
  The total reward is also composed of three parts:
  $R = R_{insp} + R_{disp} + R_{repair} + R_{replace} + R_{failure}$

How to model Structural Integration Management as MDP IV

Tabelle 3: Cost Definition in the beam monitoring process

Displacement Cost	$R_{disp} = -\sum_{i=1}^{T} k_i u_i^2$ or simpler $-\beta \lvert u_i \rvert$	$k_i = 10$ for $i = 0, \cdots, T$ $\beta = 50$
Repair Cost	$R_{repair} = -c_{repair} \cdot n_{repair}$	$c_{repair} = 500$ n_{repair} is the total repair times
Replace Cost	$R_{replace} = -c_{replace} \cdot n_{replace}$	$c_{replace} = 2000$ $n_{replace}$ is the total replace times
Inspection Cost	$R_{insp} = -c_{insp} \cdot n_{inspection}$	$c_{insp} = 200$ n_{insp} is the total inspection times
Failure Cost	$R_{failure} = -c_{failure}$	$c_{failure} = 10^4$ equivalent injury cost Failure terminates the process

How to model Structural Integration Management as MDP V

- Transition Model T: $s_{t+\Delta t} \leftarrow \mathrm{T}(s_t, a_t)$. The transition depends on the actions:
  when a_0: do nothing $\rightarrow$ natural deterioration.
  We could define a deterioration level as $D(t) = E(t=0) - E(t)$ to indicate the deterioration extent from the beginning to the time t.
  - gradual deterioration (aging process): The gradual deterioration process can be modelled as a simple rate function **ellingwood2005risk**:

$$D(t) = E(t=0) - E(t) = At^B e^{w(t)} \tag{27}$$

    where A is the random variable modelling the deterioration rate, B is the random variable modelling the nonlinearity effect in terms of a lower law in time and $w(t)$ models the gaussian stochastic process noise. Realization plot of an aging process is shown in Figure 2

How to model Structural Integration Management as MDP VI

The changes of the deterioration can be approximated by the derivative $\frac{dD}{dt}\Delta t$ if we treate $w(t)$ as a constant number w_k

$$\Delta D(t) = E(t) - E(t + \Delta t) \approx ABt^{B-1}e^{w_k}\Delta t \tag{28}$$

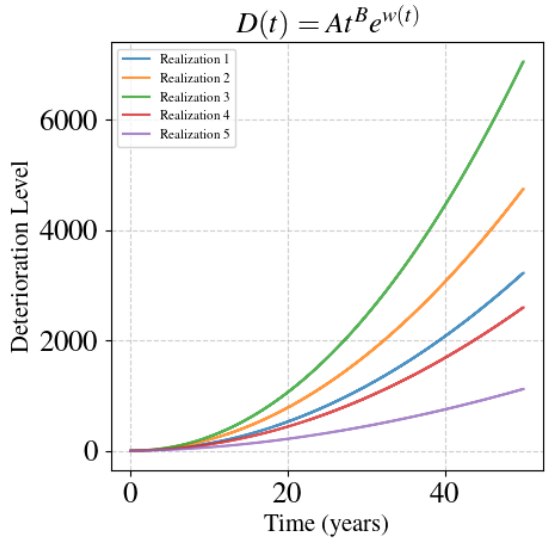


Abbildung 2: Gradual deterioration realization modeled by simple rate fu

How to model Structural Integration Management as MDP VII

- sudden deterioration: The sudden deterioration can be modelled as a homogeneous compound Poisson Process(CPP) **van2009survey**; **sanchez2016reliability**

$$D(t) = E(t=0) - E(t) = \sum_{i=1}^{N(t)} D_i \sim CPP(t; \lambda, F_D(d)) \tag{29}$$

where the number of jumps in the time interval t
$N(\Delta t) \sim PoissonProcess(\lambda)$; the amplitude of each jump $D_i \sim F_D(d)$ are independent and identically distributed random variables following a given distribution $F_D(d)$ e.g. a Gamma distribution. Realization plot of a CPP process is shown in Figure 3
The change of the deterioration during the time interval Δt is calculated as:

How to model Structural Integration Management as MDP VIII

$$\Delta D(t) = E(t) - E(t + \Delta t) = \sum_{i=1}^{N(\Delta t)} D_i \sim CPP(\Delta t; \lambda, F_D(d)) \quad (30)$$

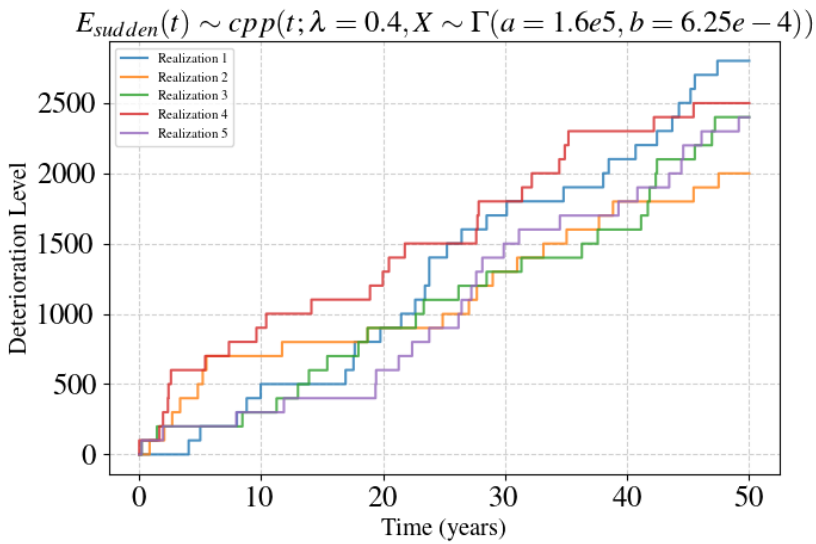
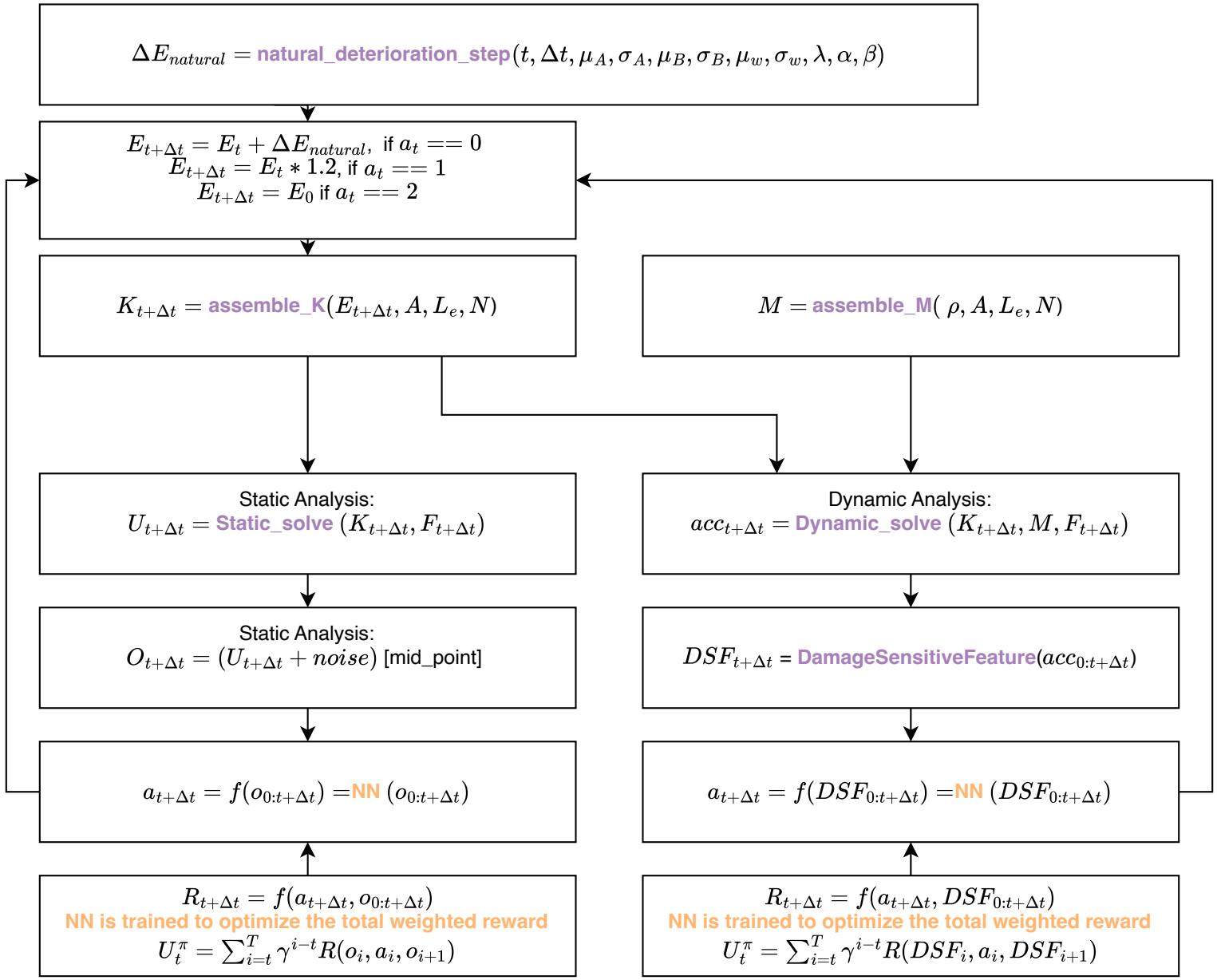


Abbildung 3: sudden deterioration realization modelled by CPP proce

How to model Structural Integration Management as MDP IX

- a_1: minor repair $\rightarrow E(t+\Delta t) = E(t) + (E(t=0) - E(t)) \cdot \alpha_{repair}$
- a_2: full replacement $\rightarrow E(t+\Delta t) = E(t=0)$

Implementation Idea to model SIM as POMDP



References I