**Markov Decision Processes** Lecture Notes 02

# Dynamics

*Authors: Benjamin Van Roy* *September 24, 2025*

## 1 Trajectories

Given a finite-state MDP $(\mathcal{S}, \mathcal{A}, P)$, an initial state $S_0$, and a policy $\pi$, we can simulate a trajectory $S_0, A_0, S_1, A_1, S_2, A_2, \ldots$. Given a history $H_t = (S_0, A_0, \ldots, S_t)$, the next action and state are generated by:

1. sampling an action $A_t \sim \pi(\cdot|H_t)$;
2. sampling a next state $S_{t+1} \sim (P_{A_t,S_t,s} : s \in \mathcal{S})$.

Alternatively, if an MDP is specified by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{W}, f, \nu)$, the next action and state are generated by:

1. sampling an action $A_t \sim \pi(\cdot|H_t)$;
2. sampling a disturbance $W_{t+1} \sim \nu(\cdot|S_t, A_t)$;
3. sampling a next state $S_{t+1} = f(S_t, S_t, W_{t+1})$.

### 1.1 The Markov Property

Consider a stochastic process $X_0, X_1, X_2, \ldots$ that takes values in a set $\mathcal{X}$. This process is said to satisfy the *Markov property* if, for all $t$ and $\mathcal{Y} \subseteq \mathcal{X}$,

$$\mathbb{P}(X_{t+1} \in \mathcal{Y}|X_0, \ldots, X_t) = \mathbb{P}(X_{t+1} \in \mathcal{Y}|X_t). \tag{1}$$

Intuitively, the stochastic process is Markov if, given $X_t$, previous samples $X_0, \ldots, X_{t-1}$ would not further inform predictions about $X_{t+1}$.

Must a state sequence $S_0, S_1, S_2, \ldots$ generated by an MDP $(\mathcal{S}, \mathcal{A}, P)$ and a policy $\pi$ satisfy the Markov property? Not necessarily. This is because the choice of next action $A_t$ can depend on previous history $S_0, A_0, \ldots, S_{t-1}, A_{t-1}$ in addition to the current state $S_t$. For example, consider the routing environment described in Section 1.2.2 of Lecture Notes 01. Recall that, for this environment, $\mathcal{S} = \mathcal{A} = \mathcal{V}$, where $\mathcal{V}$ is the set of vertices in the graph. Consider a policy that, at vertex $S_t$, samples randomly among actions that do not return to the previous vertex. In other words $\pi(\cdot|H_t) \sim \text{unif}(a \in \mathcal{A} : a \neq S_{t-1})$. Under this policy, given $S_t$, knowing $S_{t-1}$ would improve our prediction of $S_{t+1}$. Hence, the state sequence does not satisfy the Markov property.

The state sequence of an MDP $(\mathcal{S}, \mathcal{A}, P)$ *does* satisfy the Markov property if actions are selected by a stationary policy $\pi$. If the state and action spaces are countable, this can be verified via the tower property:

$$\begin{aligned}
\mathbb{P}(S_{t+1} = s|S_0, \ldots, S_t) &= \sum_{a \in \mathcal{A}} \mathbb{P}(A_t = a|S_0, \ldots, S_t)\mathbb{P}(S_{t+1} = s|S_0, \ldots, S_t, A_t = a) \\
&= \sum_{a \in \mathcal{A}} \pi(a|S_t) P_{aS_t s} \\
&= \sum_{a \in \mathcal{A}} \mathbb{P}(A_t = a|S_t)\mathbb{P}(S_{t+1} = s|S_t, A_t = a) \\
&= \mathbb{P}(S_{t+1} = s|S_t).
\end{aligned}$$

This argument can be extended to uncountable state and action spaces.

Under a stationary policy $\pi$, the sequence of state-action pairs also satisfies the Markov property. For countable state and action spaces, this can be verified via Bayes' rule. In particular, letting $X_t = (S_t, A_t)$ and $x = (s, a)$,

$$\begin{aligned}\mathbb{P}(X_{t+1} = x|X_0, \ldots, X_t) =& \mathbb{P}(S_{t+1} = s, A_{t+1} = a|S_0, A_0, \ldots, S_t, A_t)\\ =& \mathbb{P}(A_{t+1} = a|S_0, A_0, \ldots, S_t, A_t, S_{t+1} = s)\mathbb{P}(S_{t+1} = s|S_0, A_0, \ldots, S_t, A_t)\\ =& \pi(a|s)P_{aS_t s}\\ =& \mathbb{P}(A_{t+1} = a|S_{t+1} = s, S_t, A_t)\mathbb{P}(S_{t+1} = s|S_t, A_t)\\ =& \mathbb{P}(S_{t+1} = s, A_{t+1} = a|S_t, A_t)\\ =& \mathbb{P}(X_{t+1} = x|X_t).\end{aligned}$$

Again, the argument can be extended to uncountable state and action spaces.

## 1.2 Transition Matrices and Kernels

For an MDP $(\mathcal{S}, \mathcal{A}, P)$ with countable state and action spaces, the sequence of states under a stationary policy follows a Markov chain with transition matrix $P_\pi$. Components are given by

$$P_{\pi ss'} = \sum_{a \in \mathcal{A}} \pi(a|s) P_{ass'}. \tag{2}$$

For an MDP $(\mathcal{S}, \mathcal{A}, f, \nu)$ with an uncountably infinite state space, individual transition probabilities do not suffice to characterize dynamics. For example, if $\mathcal{S} = \mathbb{R}$ and, conditioned on $S_t$ and $A_t$, the next state distribution is Gaussian then the probability assigned to each next state is zero. In order to characterize dynamics we can instead use a transition kernel defined by

$$P_a(B|s) = \nu(\{w \in \mathcal{W} : f(s, a, w) \in B\}|s, a). \tag{3}$$

For each action $a \in \mathcal{A}$ and set $B \subseteq \mathcal{S}$ of possible next states, this is the probability that $S_{t+1}$ will be an element of $B$. Technically, for each $s \in \mathcal{S}$ and $a \in \mathcal{A}$, $P_a(\cdot|s)$ is a probability measure. The sequence of states under a stationary policy $\pi$ is then a Markov process with transition kernel defined by

$$P_\pi(B|s) = \int_{a \in \mathcal{A}} \pi(da|s) P_a(B|s). \tag{4}$$

## 1.3 Examples

We provide examples of policies and state trajectories for a few MDPs.

### 1.3.1 Queueing

Recall the queueing system described in the Lecture 01 Notes. At each time, an additional customer arrives with probability $q$. The station can operate in a fast or slow mode of service. If there are customers in the queue, the fast mode removes one with probability $p_{\text{fast}}$ and the slow mode, $p_{\text{slow}}$.

We modeled the queue length dynamics in terms of an MDP $(\mathcal{S}, \mathcal{A}, P)$ with $\mathcal{S} = \{0, 1, 2, \ldots\}$ and $\mathcal{A} = \{\text{fast}, \text{slow}\}$. Each state $S_t$ is the current queue length, and the action $A_t$ indicates whether the fast or slow service mode is applied. Transition probabilities are as provided in Table 1.

Consider a simple policy that uses the slow mode if and only if the queue length is below a threshold. This is a deterministic policy defined by

$$\pi(s) = \begin{cases} \text{slow} & \text{if } s < \theta \\ \text{fast} & \text{otherwise,} \end{cases}$$

| $P_{ass'}$ | $s' = s-1$ | $s' = s$ | $s' = s+1$ |
|---|---|---|---|
| $s = 0$ | 0 | $1-q$ | $q$ |
| $s > 0$ | $(1-q)p_a$ | $qp_a + (1-q)(1-p_a)$ | $q(1-p_a)$ |

**Table 1**: Queue transition probabilities. The rows provide formulas that apply when $s = 0$ or $s > 0$, respectively.

where $\theta$ is the threshold. Under this policy, state transition matrix is defined by

$$P_{\pi,0,s'} = \begin{cases} 1-q & \text{if } s' = 0 \\ q(1-p_{\pi(s)}) & \text{if } s' = 1, \end{cases}$$

and, for $s > 0$,

$$P_{\pi ss'} = \begin{cases} (1-q)p_{\pi(s)} & \text{if } s' = s-1 \\ qp_{\pi(s)} + (1-q)(1-p_{\pi(s)}) & \text{if } s' = s \\ q(1-p_{\pi(s)}) & \text{if } s' = s+1. \end{cases}$$

Figure 6 plots state trajectories simulated with thresholds of $\theta = 0$ and $\theta = 10$. Queue lengths are volatile due to random arrivals and service times. As one would expect, the queue lengths tend to be much shorter with $\theta = 0$ because in that case we always use the fast mode of service.

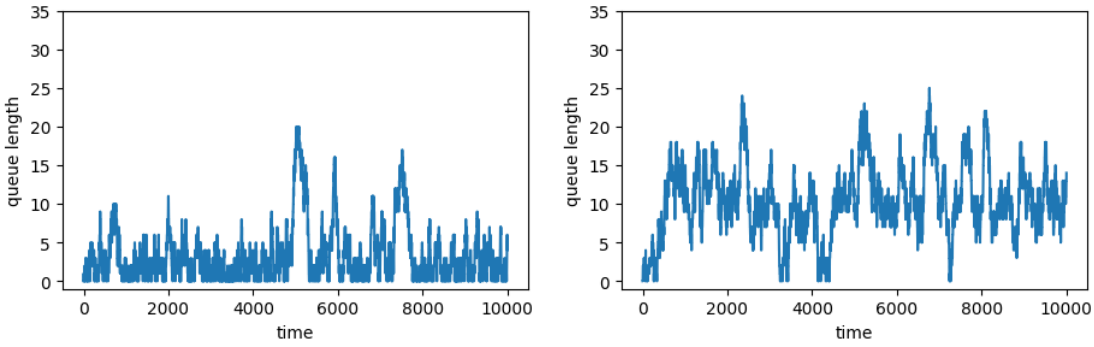


**Figure 1**: Queue length dynamics with policy thresholds of 0 (left) and 10 (right).

### 1.3.2 Pursuit

Consider a stochastic linear system that models pursuit of a target in one dimension, as introduced in Lecture Notes 01. Those notes provided two mathematical formulations, and we will use the second one here. That formulation involved two state variables. The first represented the difference $x_{t,1}$ between the location of the pursuer and that of the target. The second represented the velocity $x_{t,2}$ of the pursuer. The dynamics of these quantities were modeled by an MDP $(\mathcal{X}, \mathcal{U}, \mathcal{W}, f, \nu)$ with $\mathcal{X} = \mathbb{R}^2$, $\mathcal{U} = \mathbb{R}$, $\mathcal{W} = \mathbb{R}^2$, $f(x, u, w) = Ax + Bu + w$, where

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 1/m \end{bmatrix} \qquad \Sigma = \begin{bmatrix} \sigma^2 & 0 \\ 0 & 0 \end{bmatrix}.$$

Here, $m$ denotes the mass of the pursuer and each action $U_t$ denotes force applied to accelerate.

Consider a policy $\pi(s) = -0.3mx_{t,1} - 0.5mx_{t,2}$. This policy tends to accelerate the pursuer by an amount that decreases in both $x_{t,1}$ and $x_{t,2}$. This is intuitively desirable, since the pursuer needs to displace the

distance $x_{t,1}$ to reach the target and should tend to decelate if the current velocity $x_{t,2}$ is too large. Figure 2 plots simulated trajectories with $\sigma^2 = 0$ and $\sigma^2 = 1$. In the first case, where the target remains still, the pursuer approaches the target, and thus distance and velocity vanish. In the second case, the target moves randomly and thus we observe continual pursuit.

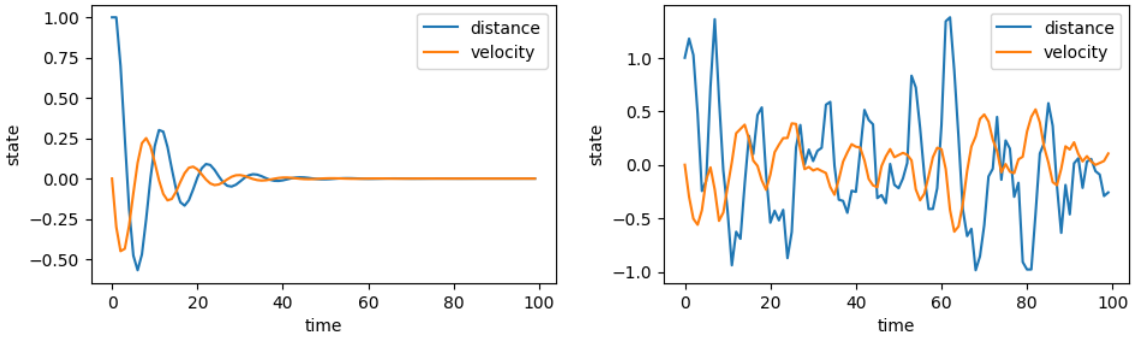


**Figure 2**: Pursuit dynamics without (left) and with (right) random noise.

### 1.3.3 Pair Writing

Consider a human who uses a chatbot as a companion in pairs writing. The human and chatbot take turns writing sentences. The chatbot's decision process can be modeled by an MDP $(\mathcal{S}, \mathcal{A}, \mathcal{W}, f, \nu)$. The state $S_t$ encodes the sentences written thus far. The action $A_t$ is the next sentence written by the chatbot. The disturbance $W_{t+1}$ is the next sentence written by the human. Finally, $S_{t+1} = f(S_t, A_t, W_{t+1})$ is the concatenation of previous text $S_t$, the chatbot's sentence $A_t$, and the human's sentence $W_{t+1}$. Note that $\mathcal{S}$, $\mathcal{A}$, and $\mathcal{W}$ are countable sets, each consisting of finite-length text strings. Figure 3 provides a simulated trajectory as well as python code that produced it. A language model was used to generate chatbot responses. Thus, this language model defines the policy $\pi$. Rather than requiring a human to write the 'human' sentences, for the purpose of this simulation, we used the language model in place of the human. Hence, in this simulation, the language model also defines the disturbance distribution $\nu$.

## 2 Probabilities and Expectations

Given an MDP $(\mathcal{S}, \mathcal{A}, P)$, a policy $\pi$, and an initial state $S_0$, we will often study events that are determined by the realized infinite trajectory $H_\infty = (S_0, A_0, S_1, A_1, \ldots)$. Such an event identifies a set $E \subseteq (\mathcal{S} \times \mathcal{A})^\infty$ of trajectories, where $\mathbb{Z}_+$ is the set of nonnegative integers. The event occurs if and only if $H_\infty \in E$.

The event probability $\mathbb{P}(H_\infty \in E)$ is the fraction of simulated trajectories that would lie in $E$. If it is possible to determine whether $H_\infty \in E$ by observing a finite-length partial trajectory $H_T$, then we can estimate $\mathbb{P}(H_\infty \in E)$ via Monte Carlo simulation.

For example, suppose we would consider the queueing system a failure if the queue length exceeds 20 at some point within the first $T = 1000$ timesteps. Let $E$ be the set of trajectories that constitute failures. Then, there is a set $E_T$ such that $H_\infty \in E$ if and only if $H_T \in E_T$. Hence, the probability of failure is $\mathbb{P}(H_T \in E_T)$. This probability can be estimated by simulating many trajectories of length $T$ and taking the estimate to be the fraction that lie in $E_T$.

Figure 6 plots failure probabilities estimated using 1000 simulated trajectories. Estimates are generated for policy thresholds $\theta$ varying from 0 to 20. This plot can inform design of a policy. For example, to trade off between cost of service and failure probability, one might choose the largest threshold that keeps the failure probability below some value, say, 10%.

```python
import google.generativeai as genai
genai.configure(api_key=GOOGLE_API_KEY)

def generate_next_sentence(context, model):
  response = model.generate_content(context + "  Provide a sentence that could follow the previous text.  "
                                    + "Do not provide any text beyond that one sentence.")
  return response.text

model = genai.GenerativeModel('models/gemini-1.5-flash')
context = "human: Once upon a time, there was a PhD student taking a class on Markov decision processes.\n"
print(context)
for n in range(5):
  for writer in ['chatbot', 'human']:
    next_sentence = generate_next_sentence(context, model)
    print(writer + ': ' + next_sentence)
    context = context + "  " + next_sentence
```

```
human: Once upon a time, there was a PhD student taking a class on Markov decision processes.

chatbot: He found the subject surprisingly engaging, despite the complex mathematics involved.

human: He especially enjoyed the challenge of designing optimal policies for intricate, multi-stage decision problems.

chatbot: His professor, noticing his keen interest, suggested he consider a research topic in reinforcement learning.

human: Intrigued by the potential to apply these concepts to real-world problems, he readily agreed.

chatbot: His journey into the world of artificial intelligence had begun.

human: Little did he know, this decision would lead him down a path of groundbreaking discoveries and unexpected collaborations.

chatbot: The future held not only academic success, but also the thrilling possibility of changing the world.

human: His first project involved training a robot to navigate a maze using Q-learning.

chatbot: The seemingly simple task proved to be a surprisingly fertile ground for innovation.

human: He discovered a novel approach to reward shaping that significantly improved the robot's learning speed and efficiency.
```

**Figure 3**: Pair writing with a chatbot.

Expectations can be estimated in a similar way. For example, $\mathbb{E}[g(H_T)]$ can be estimated by simulating many trajectories of length $T$, applying the function $g$ to each, and averaging the resulting values.

Figure 6 plots the expected queue length $\mathbb{E}[S_T]$ as a function of $T$, with policy thresholds 0 and 10. In each case, the initial expected length is $\mathbb{E}[S_0] = 0 = S_0$ and the expected length converges. The limit is larger when the policy threshold is 10. This is intuitive because, when the policy threshold is 0, the fast service mode is always applied.

# 3 State Probabilities

In this section, as with much of this course, we will focus on finite MDPs – those with finite state and action spaces. This is because much can be learned from the finite case, and dealing with infinite MDPs requires more advanced mathematical analysis that can encumber that learning. But we will continue to study examples with infinite spaces when infinite MDPs allow for more intuitive models. And we will occasionally discuss how results established for finite MDPS extend to infinite ones.

Fix a finite MDP $(\mathcal{S}, \mathcal{A}, P)$ and a stationary policy $\pi$. This MDP and policy induce a Markov process $(\mathcal{S}, P_\pi)$. Fix an initial state $S_0$, and let $\mu_t(s) = \mathbb{P}(S_t = s)$. This is the probability that the state is $s$ at time $t$ and can be estimated by simulating trajectories of length $t$ and calculating the fraction the end in state $s$. We denote the set of probability mass functions over a finite or countable state space $\mathcal{S}$ by $\Delta_\mathcal{S} = \{\mu \in [0,1]^\mathcal{S} : \sum_{s\in\mathcal{S}} \mu(s) = 1\}$. Hence, $\mu_t \in \Delta_\mathcal{S}$.

## 3.1 The Forward Equation

State probabilities satisfy the *forward equation*

$$\mu_{t+1}(s') = \sum_{s\in\mathcal{S}} \mu_t(s) P_{\pi ss'} \qquad \forall s' \in \mathcal{S}. \tag{5}$$

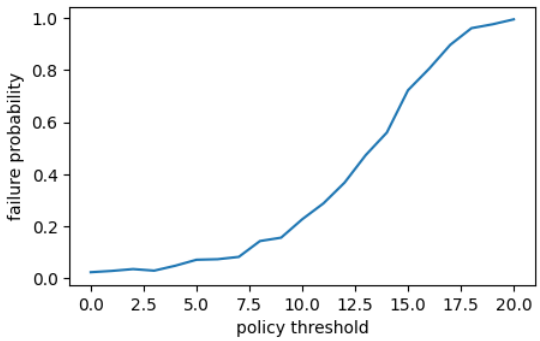


**Figure 4**: Failure probability as a function of the policy threshold.

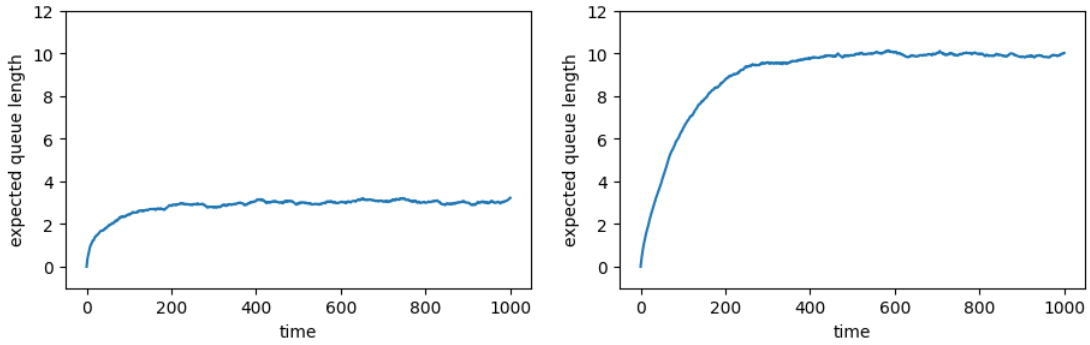


**Figure 5**: Expected queue length with policy thresholds of 0 (left) and 10 (right).

This is intuitive: the probability of being in state $s'$ at time $t+1$ is the sum over $s$ of the probability of being in state $s$ at time $t$ times the probability of transitioning from $s$ to $s'$.

Figure 6 plots $\mu_t$ for various values of $t$, for the special case of our queueing system, with policy thresholds of 0 and 10, initialized with $S_0 = 0$ and thus $\mu_0(s) = \mathbf{1}_0(s)$. The probability distributions become more diffuse over time, converging to a steady state distributions. The steady state distribution assigns higher probabilities to shorter queues when the threshold is 0 rather than 10. This makes sense because when the threshold is 0, the system always applies the fast service mode. With a threshold of 10, there is a kink in steady state distribution due to the switching behavior from slow to fast service.

## 3.2 The Balance Equation

Consider a Markov process $(\mathcal{S}, P_\pi)$ induced by an MDP $(\mathcal{S}, \mathcal{A}, P)$ with finite state and actions paces and stationary policy $\pi$. Given an initial state $S_0$, let $\mu_\infty = \lim_{t\to\infty} \mu_t$ denote the vector of limiting probabilities, if that exists. Limiting probabilities, if they exist, satisfy the *balance equation*

$$\mu(s') = \sum_{s\in\mathcal{S}} \mu(s) P_{\pi ss'} \qquad \forall s' \in \mathcal{S}, \tag{6}$$

which is obtained by taking limits of the left and right hand sides of the forward equation. We refer to any state distribution $\mu$ that solves this equation as a *steady state distribution*. Such a solution is also commonly referred to as an *invariant distribution*.

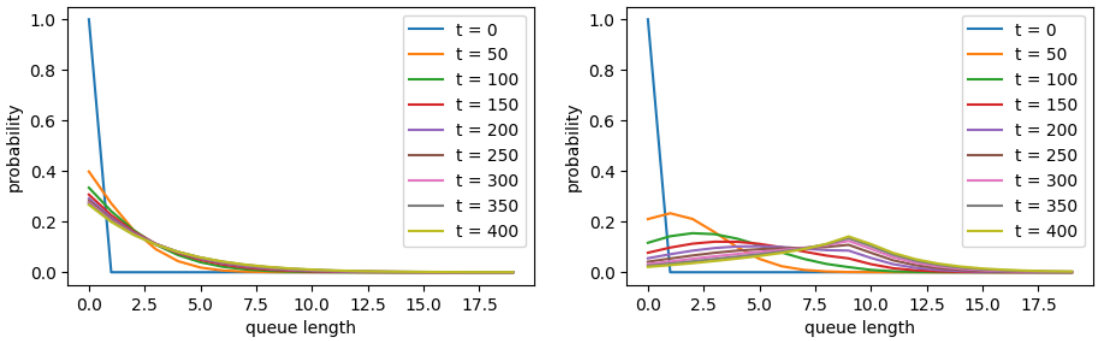


**Figure 6**: Queue length probability distributions with policy thresholds of 0 (left) and 10 (right).

It is natural to wonder whether a steady state distribution exists and, if so, whether it is unique. In particular, whether there exists a unique solution to

$$\mu(s') = \sum_{s \in \mathcal{S}} \mu(s) P_{\pi ss'} \qquad \forall s' \in \mathcal{S}, \tag{7}$$

within $\Delta_{\mathcal{S}}$.

### 3.2.1 Existence

Recall that the limiting distribution $\mu_\infty$, if one exits, constistutes a steady state distribution. However, a limiting distribution need not exist when dynamics exhibit periodicity. To understand why, consider the Markov chain of Figure 7. If $S_0 = 1$ then each $S_t = 1$ when $t$ is even and $S_t = 2$ when $t$ is odd. As such $\mu_t(1)$ is 1 if $t$ is even and 0 if $t$ is odd. Thus, the limit $\mu_\infty$ does not exist.

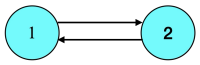


**Figure 7**: A periodic Markov chain.

What *is* guaranteed to exist is the Cesàro limit:

$$\overline{\mu} = \lim_{t \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mu_t. \tag{8}$$

And this solves the balance equation, since

$$\begin{aligned}
\overline{\mu}(s) &= \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mu_t(s) \\
&= \lim_{T\to\infty} \frac{1}{T-1} \sum_{t=1}^{T-1} \mu_t(s) \\
&= \lim_{T\to\infty} \frac{1}{T-1} \sum_{t=1}^{T-1} \sum_{s'\in\mathcal{S}} \mu_{t-1}(s') P_{\pi s's} \\
&= \sum_{s'\in\mathcal{S}} P_{\pi s's} \lim_{T\to\infty} \frac{1}{T-1} \sum_{t=1}^{T-1} \mu_{t-1}(s') \\
&= \sum_{s'\in\mathcal{S}} P_{\pi s's} \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mu_{t-1}(s') \\
&= \sum_{s'\in\mathcal{S}} \overline{\mu}(s') P_{\pi s's}.
\end{aligned}$$

Hence, there always exists a steady state distribution.

### 3.2.2 Uniqueness

The steady state distribution is not necessarily unique. To see why, consider the Markov chain in Figure 8. The Cesàro limit defined by Equation (8) depends on the initial state $S_0$. In particular, it is easy to verify that

$$\overline{\mu} = \begin{cases} [0 \quad 1/2 \quad 1/2]^\top & \text{if } S_0 = 1, \\ [0 \quad 1 \quad 0]^\top & \text{if } S_0 = 2, \\ [0 \quad 0 \quad 1]^\top & \text{if } S_0 = 3. \end{cases}$$

As we established earlier, each Cesàro limit constitutes a steady state distribution. As such, there are at least three distinct solutions. In fact, there are many more, because any convex combination of these three solutions are also steady state distributions.

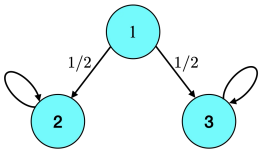


**Figure 8**: A Markov chain with multiple recurrent classes.

A *recurrent class* is a set $\tilde{S}$ of states such that if $S_t \in \tilde{S}$ then every state in $\tilde{S}$ will be visited infinitely often and no state outside $\tilde{S}$ will be visited after time $t$. In the example of Figure 8, there are two recurrent classes: $\{2\}$ and $\{3\}$. This example highlights the fact that, when there are multiple recurrent classes, there are multiple steady state distributions. Further, if a Markov chain has a single recurrent class $\tilde{S}$ then there is a unique steady state distribution.

# 4 Infinite MDPs

While a steady state distribution exists for any finite state Markov chains, there are infinite state Markov chains with no steady state distribution. For example, when dynamics are unstable, there is no steady state

distribution. Figure 9 illustrates such instability arising in models we formulated earlier for a queueing system and for pursuit dynamics. The queueing system becomes unstable, with queue length growing unbounded over time, if we set the policy threshold to infinity, meaning that we only ever use the slow service mode. With this slow mode, on average, the number of customers arriving per unit time exceeds the number served. Pursuit dynamics become unstable if the pursuer overaccelerates. To produce the instability illustrated in the plot, we increased acceleration by between four and five times.

A sufficient condition for existence of a steady state distribution is that there is a positive recurrent state. A *positive recurrent state* is a state that will be visited infinitely often if visited at all and for which the expected return time is finite.

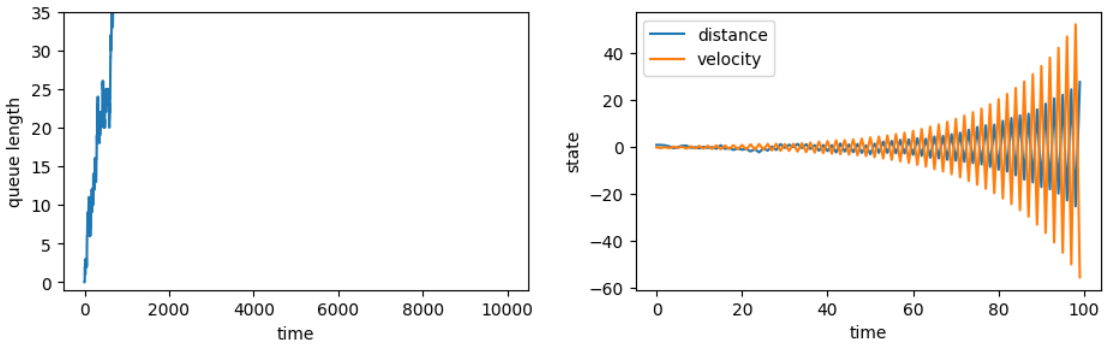


**Figure 9**: Instability arising in queueing and pursuit.