First Passage Value

Cenk Oguz Saglam and Katie Byl

April 18, 2021

Abstract

For many stochastic dynamic systems, the Mean First Passage Time (MFPT) is a useful concept, which gives expected time before a state of interest. This work is an extension of MFPT in several ways. (1) We show that for some systems the system-wide MFPT, calculated using the second largest eigenvalue only, captures most of the fundamental dynamics, even for quite complex, high-dimensional systems. (2) We generalize MFPT to Mean First Passage Value (MFPV), which gives a more general value of interest, e.g., energy expenditure, distance, or time. (3) We provide bounds on First Passage Value (FPV) for a given confidence level. At the heart of this work, we emphasize that for our goals, many hybrid systems can be approximated as Markov Decision Processes. So, many systems can be controlled effectively using this framework. However, our framework is particularly useful for metastable systems. Such systems exhibit interesting long-living behaviors from which they are guaranteed to inevitably escape (e.g., eventually arriving at a distinct failure or success state). Our goal is then either minimizing or maximizing the value until escape, depending on the application.

1 Introduction

First Passage Time, aka First Hitting Time, gives survival duration by answering "how long it will take on average before a specific event (or set of events) occurs". In discrete time models, one usually calculates the expected number of discrete time steps of survival, which corresponds to Mean First Passage Time (MFPT). While different initial conditions (states) generally result in different MFPTs, for some systems a scalar called system-wide MFPT is an accurate estimate across a large set of states. Although this paper also contains some useful ideas and extensions for systems for which this does not hold, some parts will build on the mentioned property. In this paper, we will abuse the notation and MFPT will refer to system-wide MFPT. Values for each state will be contained in the MFPT vector.

Of particular interest to the authors, stability of *metastable* systems can potentially be well represented by (system-wide) MFPT. These systems can be natural or human-made. They exist in a precarious state of stability, appearing to be locally stable for long periods of time until an external disturbance perturbs the system into a region of state space with a qualitatively different local behavior. Since these systems are guaranteed to exit these locally well-behaved regions with probability one given enough time, they cannot be classified as "stable", but it is also misleading to categorize them simply as "unstable". A toy example of metastable systems is a ball in a hollow on terrain where a sufficient disturbance would cause the ball to roll into another local minimum, as depicted in Figure 1. Physicists have explored this phenomenon in detail and have developed a number of tools for quantifying this behavior [1, 2, 3, 4]. Metastable processes have been observed in many other branches of science and engineering including familiar systems such as crystalline structures [5], flip-flops [6], and neuroscience [7].

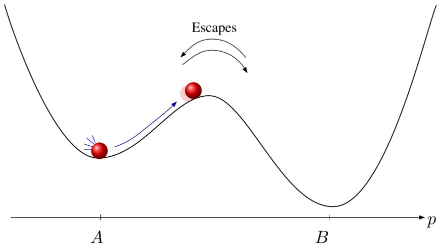

Figure 1: Cartoon with multiple locally-stable equilibria under deterministic conditions. With sufficient noise in the model, the particle is guaranteed to transition from one local minimum to another (back and forth), although transitions may still be quite rare. From region A's perspective, "escape" corresponds to moving to region B. Figure inspired by [8] and [9].

More recently, the tools for quantifying metastable systems has been applied to walking robots to predict how a robot will perform over variable terrain for a given control policy [8, 10]. For such analyses, the walking robot, the environment, the system noise, and the control actions can be modeled together as a Markov chain. Assuming that the initial state of the robot lies within a so-called "metastable region" of state space, the eigenvalues of the state transition matrix of the Markov chain, specifically the largest eigenvalue not associated with the (absorbing) failed system state, can be used to predict the number of steps the robot can take before failing. While this approach is described in some detail in [8], we have found when building upon the basic concepts that many important aspects of the analytic approach are not immediately obvious nor (as yet) well-documented. In this paper, we attempt to clarify the overall approach and the utility of the results, using simple toy examples.

To show the applicability of our methods to various problems we consider hybrid systems, which may exhibit either or both continuous and discontinuous dynamics. The requirement is the ability to model the full dynamics as a Markov chain. This can be done in many applications where a meaningful "step" definition can be proposed. As we justify in subsection 3.1, we will model the escape of interest (e.g., moving from region A to B in Fig. 1, or falling of a walking robot) as an absorbing *halt* state.

In previous work, we have used the Mean First Passage Time (MFPT) to characterize the average number of Markov chain steps until reaching an absorbing failure state. In this paper, we present a more generalized concept – First Passage Value (FPV) – and discuss both the mean and *variability* of a value of interest for a metastable system. Calculating different values corresponds to adopting different rewards. This was suggested in [10] and applied in [11].

The rest of this paper is organized as follows. In Section 2, we provide some motivating examples, which we will use to illustrate for the rest of the paper. In Section 3, we use the eigenvalues of the state transition matrix to calculate MFPT and discuss the relevance of employing a system-wide MFPT. Sections 4 and 5 introduces the FPV metric, which builds on MFPT. Finally, we conclude with control applications in Sec. 6.

2 Some Motivating Examples

In this section we provide five motivating examples. The first three of them are discrete-time with no control action and can be easily represented in the form of a Markov chain. We will later discuss how to deal with more general systems in Section 6. In all these examples, we are initially interested in discrete time steps before hitting a state of interest. We will later investigate how to calculate metrics other than discrete time.

2.1 Coin Toss

Consider tossing an unfair coin, for which the probability of having heads is 0.01. Say we are interested in the number of flips before two heads in a row. Then, we have three possible states: (1) Heads-heads, (2) Tails in the last flip (including 'not-flipped yet'), (3) Tails-Heads (including 'flipped once to get heads'). Figure 2 shows the corresponding Markov Chain.

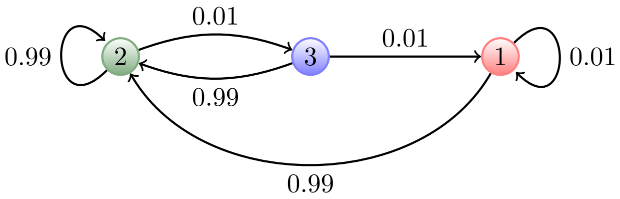

Figure 2: Markov Chain representing unfair coin toss to get two heads in a row.

2.2 Epidemics

Consider the discrete susceptible-infected-susceptible (SIS) model as in [12]. We are interested in number of discrete time steps to everyone being healthy. As a toy example, we will consider a network of only 2 people. Then, there are four states possible as listed in Table 1.

Table 1: Explanation of each state

	State 1	State 2	State 3	State 4
Patient 1	Susceptible	Susceptible	Infected	Infected
Patient 2	Susceptible	Infected	Susceptible	Infected

Let 'the probability of recovery when a node is infected' be $\delta = 0.01$, and 'the probability to be infected when the other node is infected' be $\beta = 0.8$.

2.3 Europe Tour

Consider a person traveling between some of the largest cities in Europe shown in Figure 3. After spending a day that person either stays in the same city, or moves to one of the connected cities. The probability of action is directly proportional to the population of the next city. For example when in London, this person stays there with probability 0.5922, moves to Berlin with probability 0.2507, or moves to Paris with probability 0.1571. Staying in London has the highest probability because it is more populated than Paris and Berlin. In this example, we will investigate number of days before reaching a specific city, say Istanbul.

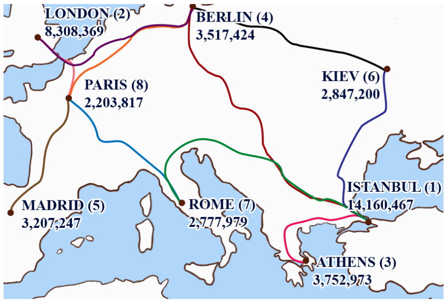

Figure 3: Populations of Europe's most populated 8 cities excluding Russia. Table 2 provides information about the road map drawn in this figure. Cities are ordered by their population as indicated in parenthesis, e.g., Athens (State 3) is the third most crowded city in this map.

2.4 Legged Locomotion

Arguably just like humans, underactuated two-legged robots are destined to fall down when the terrain is rough enough. We would like to estimate the number of steps before falling. The more steps taken on average, the more stable the robot is considered to be. Please see [13] for an application of the tools presented in this paper, where stability is dramatically increased by optimizing for number of steps.

Also, [14] provides a comparison between Lyapunov-based approach versus using the tools of this paper on a hybrid system, namely, the Rimless Wheel.

2.5 Driving a Car

Again arguably, given enough time (millions of years if necessary), any driver will be involved in an accident. The same is and will be true for autonomous cars including the Google car. We might be interested in, for example, the number of intersections before an accident, or the average number of miles driven between accidents. This example is provided just to motivate following sections, but we have not yet modeled it.

3 Absorbing Markov Chains

We will refer to a “state of interest” as a “halt state”, e.g., two heads in a row, everybody being healthy, being in Istanbul, falling of a robot, or accident for a car. Without loss of generality, we define this halt state to be *State 1* (x_1). Any halt state will be absorbing.

3.1 Why Absorbing?

Note that the epidemics model in the previous section is already absorbing because when both patients are susceptible (State 1), the state will not change (i.e., noone can become infected).

Let’s consider the coin toss example. State 1 is not absorbing, i.e., two heads in a row does not imply next flip to be heads. However, if we are interested in the number of flips before reaching State 1, then we should modify the Markov Chain by modeling State 1 to be absorbing as in Figure 4. This modification corresponds to the game ending when two heads are in a row. The dynamics until State 1 do not change, but the modification allows us calculate the number of flips before State 1.

Similarly, for the Europe tour example, if we are interested in the number of days before going to Istanbul, we should model the state of being in Istanbul as absorbing.

For the following, we focus on the toy example shown in Figure 4, which will be used to illustrate our analysis.

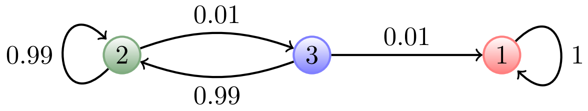

Figure 4: Figure 2 modified by modeling State 1 to be absorbing.

3.2 The Analysis

The state distribution vector at step n is denoted by $p[n]$ and defined by

$$p_i[n] := Pr(x[n] = x_i). \tag{1}$$

So $p_i[n]$ corresponds to the probability of being at state x_i at step n. Since probability cannot be negative, $p[n]$ is a non-negative vector, and because the system has to be at a state at any step, $p[n]$ sums to 1. The state transition matrix, aka the Markov matrix or the stochastic matrix, has the following structure by definition:

$$T_{s\{ij\}} := Pr(x[n+1] = x_j \mid x[n] = x_i). \tag{2}$$

So, the element of T_s on the ith row and jth column gives the probability of transitioning from state x_i to state x_j. To illustrate, the Markov chain of Figure 4 is represented with

$$T_s = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.99 & 0.01 \\ 0.01 & 0.99 & 0 \end{bmatrix}. \tag{3}$$

Similar to non-negativity of $p[n]$, we have $T_{s\{ij\}} \geq 0$. And because any state will transition (possibly to the halt state or the starting state itself) after each step, each row sums to one. For the rest of the paper, we assume the number of states is $\ell > 1$. So, the state transition matrix is ℓ by ℓ. The state transition matrix gives the next state distribution, given the current one:

$$p[n+1] = T_s' \, p[n] = (T_s')^{n+1} p[0], \tag{4}$$

where the prime ($'$) symbol denotes the transpose operation.

Let λ be an eigenvalue of T_s. Then, there exists a non-zero vector v such that

$$T_s v = \lambda v. \tag{5}$$

As shown in [15], we next show that every eigenvalue λ of T_s satisfies $|\lambda| \leq 1$. Let k be such that $|v_j| \leq |v_k|$ for all $1 \leq j \leq \ell$. Equating the k-th components in equation (5) gives

$$\sum_j T_{s\{kj\}} v_j = \lambda v_k. \tag{6}$$

We then have

$$\begin{aligned} |\lambda v_k| = |\lambda|\ |v_k| &= \left|\sum_j T_{s\{kj\}} v_j\right| \\ &\leq \sum_j T_{s\{kj\}} |v_j| \leq \sum_j T_{s\{kj\}} |v_k| = |v_k|, \end{aligned} \tag{7}$$

where we used $T_{s\{kj\}} \geq 0$ and $\sum_j T_{s\{kj\}} = 1$. $|\lambda||v_k| \leq |v_k|$ implies $|\lambda| \leq 1$.

For the rest of the paper, we will prefer working with the transpose of T_s to make the following sections easier to follow. Since T_s is square, T_s' has the same eigenvalues as T_s. Due to the nature of transpose operation and the structure of T_s, each column of T_s' sums to one.

Remember that x_1 is an absorbing state, which represents the end of game, no matter how the system escaped (e.g., the robot slipped or hit the wall). Then, T_s' has the following form:

$$T_s' = \begin{bmatrix} 1_{1\times 1} & T_1 \\ 0 & \hat{T} \end{bmatrix}_{\ell \times \ell}. \tag{8}$$

Note that $\lambda = 1$ and $v = [1\ 0\ ...\ 0]'$ satisfies the equation

$$T_s' v = \lambda v. \tag{9}$$

To distinguish (possibly non-distinct) eigenvalues, we will note them by λ_j, where $1 \leq j \leq \ell$. Without loss of generality, we will let $\lambda_1 = 1$ and the associated basis vector be $v_1 = [1\ 0\ ...\ 0]'$.

Existence of Jordan normal form for any square matrix is fundamental to Linear Algebra. Consider a Jordan normal form of $\hat{T}$ given by

$$\hat{T} = \hat{V}\hat{J}\hat{V}^{-1}, \tag{10}$$

Then, as we will verify, a Jordan normal form of T_s' is given by

$$T_s' = VJV^{-1}, \tag{11}$$

$$\text{where } J = \begin{bmatrix} 1 & 0 \\ 0 & \hat{J} \end{bmatrix}, \tag{12}$$

$$\text{and } V = \begin{bmatrix} 1 & -[1\ ...\ 1]\hat{V} \\ 0 & \hat{V} \end{bmatrix}. \tag{13}$$

Note that the sum of each column of V equals zero, except the first one. Furthermore, these columns form a basis in $\mathbb{R}^\ell$. Equation (11) can be verified as follows. The inverse of V is

$$V^{-1} = \begin{bmatrix} 1 & [1\ ...\ 1] \\ 0 & \hat{V}^{-1} \end{bmatrix}. \tag{14}$$

Then, the right hand side of (11) can be calculated as

$$\begin{bmatrix} 1 & [1 \;...\; 1] - [1 \;...\; 1]\hat{V}\hat{J}\hat{V}^{-1} \\ 0 & \hat{V}\hat{J}\hat{V}^{-1} \end{bmatrix} = \begin{bmatrix} 1 & [1 \;...\; 1](I - \hat{T}) \\ 0 & \hat{T} \end{bmatrix}. \tag{15}$$

Equation (11) is thus verified because $T_1 + [1 \;...\; 1]\hat{T} = [1 \;...\; 1]$ (columns of T_s' sum to one).

The spectrum of $\hat{T}$, denoted by $\sigma(\hat{T})$, is the set of distinct eigenvalues of $\hat{T}$. The spectral radius of $\hat{T}$ is given by

$$\rho(\hat{T}) = \max_{\lambda \in \sigma(\hat{T})} |\lambda|. \tag{16}$$

Let the spectral radius be $r = \rho(\hat{T})$. Remembering that $\hat{T}$ is a non-negative square matrix, we present the following two facts, which are proven in [16].

1. $r \in \sigma(\hat{T})$ (i.e., r is an eigenvalue of $\hat{T}$).
2. $\hat{T}z = rz$ for some $z \in \mathcal{N} = \{v|\ v \geq 0 \text{ with } v \neq 0\}$

We will let $\lambda_2 := r$ and v_2 will refer to the associated column in V. Then, $v_2 = [- \|z\|_1 \;\; z']'$. Now, let us consider the following state distribution

$$\phi := \begin{bmatrix} 0 & \dfrac{z'}{\|z\|_1} \end{bmatrix}' = v_1 + \frac{1}{\|z\|_1} v_2. \tag{17}$$

We will call ϕ the *metastable distribution.* Note that it is a valid initial state distribution since it sums to one and each element is non-negative. Back to our toy example, we have

$$\lambda_2 \approx 1 - 9.9020 \times 10^{-5} \text{ and } \phi \approx \begin{bmatrix} 0 \\ 0.9901 \\ 0.0099 \end{bmatrix} \tag{18}$$

This metastable distribution represents a simple probability distribution: The state is x_2 with probability ≈0.9901 and it is x_3 with probability ≈0.0099. The first element of ϕ, i.e., the probability of being at x_1, is zero, by definition (of not yet having escaped). It is actually interesting that the second element in ϕ is higher than the probability of having tails in a flip. Yet, this is true as can be verified by exactly solving for ϕ and λ_2 using the method we will show shortly.

Taking a step when ϕ is the initial condition, we obtain

$$T_s'\phi = T_s' \left(v_1 + \frac{1}{\|z\|_1} v_2 \right) = v_1 + \frac{\lambda_2}{\|z\|_1} v_2. \tag{19}$$

Naturally, the resulting distribution is also non-negative and sums to one. In addition, the first element is

$$1 + \frac{\lambda_2}{\|z\|_1}(- \|z\|_1) = 1 - \lambda_2. \tag{20}$$

This means the system escaped with probability $1 - \lambda_2$. Furthermore, given the system did not escape, we also have ϕ as the final probability distribution for the states. This can be seen by zeroing the first element of $T'_s\phi$ and scaling to sum to one. Then we have the following result: When ϕ is the initial state distribution, the probability of escaping is $1 - \lambda_2$, the probability of staying in the same distribution (ϕ) is λ_2. In our toy example, these can be seen by calculating

$$T'_s\phi \approx \begin{bmatrix} 0.0001 \\ 0.9900 \\ 0.0099 \end{bmatrix} \text{ and } \frac{T'_s\phi - (1-\lambda_2)\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}}{\left\| T'_s\phi - (1-\lambda_2)\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \right\|_1} = \phi. \tag{21}$$

Note that $\lambda_2 \approx 1 - 9.9020 \times 10^{-5}$ is close to one. This will be the case (actually $1 - \lambda_2$ will be much smaller in many cases) when we are interested in absorbing Markov chains with "rare escapes".

For this toy example, we can write ϕ and λ_2 in exact form as mentioned. This is done simply by solving the right hand side of (21) and noting $\|\phi\|_1 = 1$. In practice, a numerical eigenvalue solution gives fast and accurate results, even for quite complex (e.g., 411,000 states in [8]) models.

We then calculate the average number of steps before escape, given the initial condition was ϕ, i.e. $p[0] = \phi$. This corresponds to the term Mean First Passage Time (MFPT) in [8]. The higher MFPT is, the more stable a system is said to be. There are two cases depending on the probability of taking a step: If $\lambda_2 = 1$, then the probability of escape is zero. In this case the system will take infinitely many steps without escaping to the halt state. The other case ($\lambda_2 < 1$) is relatively more complicated and interesting as explained next. Note that from this point on we'll focus on the case $\lambda_2 < 1$.

3.3 (System-wide) First Passage Time (FPT)

Given the probability of taking a step without escaping is $\lambda_2 < 1$, the probability of taking n steps only, equivalently escaping at the nth step is simply

$$Pr(x[n] = x_1,\ x[n-1] \neq x_1) = \lambda_2^{n-1}(1 - \lambda_2). \tag{22}$$

Realize that as $n \to \infty$, the right hand side goes to zero, i.e., the system will eventually escape. Note that we also count the step which ended up escaping as a step. This can be verified considering escaping at the first step (taking 1 step only). When $n = 1$ is substituted, we get $1 - \lambda_2$ as expected. Then, the expected (mean) number of steps can be calculated as

$$\begin{aligned} MFPT &= E[FPT] \\ &= \sum_{n=1}^{\infty} n\ Pr(x[n] = x_1,\ x[n-1] \neq x_1) \\ &= \sum_{n=1}^{\infty} n\lambda_2^{n-1}(1-\lambda_2) = \frac{1}{1-\lambda_2}, \end{aligned} \tag{23}$$

where we used the fact that $\lambda_2 < 1$. As a result, MFPT can then be calculated using

$$M = \begin{cases} \infty & \lambda_2 = 1, \\ \frac{1}{1-\lambda_2} & \lambda_2 < 1. \end{cases} \tag{24}$$

The MFPT of our toy example is

$$\frac{1}{1-\lambda_2} \approx 1.0099 \times 10^4. \tag{25}$$

So, in the toy example if we start with the initial state distribution ϕ given in (18), then the system will take approximately 10,099 steps on average.

The standard deviation of FPT can be calculated by

$$\begin{aligned} E[FPT^2] &= \sum_{n=1}^{\infty} n^2\ Pr(x[n] = x_1,\ x[n-1] \neq x_1) \\ &= \sum_{n=1}^{\infty} n^2\lambda_2^{n-1}(1-\lambda_2) = \frac{1+\lambda_2}{(1-\lambda_2)^2} \end{aligned}$$

$$\implies \sqrt{E[FPT^2] - (E[FPT])^2} = M\sqrt{\lambda_2}. \tag{26}$$

This corresponds to $M\sqrt{\lambda_2} \approx 10,098.5$. For our toy example, as for any metastable system, λ_2 being close to one results in a standard deviation close to the mean!

We are also interested in obtaining the MFPT vector, m, which gives the MFPT for each state.

$$m_i := \begin{cases} 0 & i = 1, \\ 1 + \sum_j T_{s\{ij\}} m_j & otherwise. \end{cases} \tag{27}$$

The equation above says it will take zero steps to go to the halt state if the system escaped already. Otherwise, the number of steps until halt is 1 less after a step is taken. For the toy example, this means $m_1 = 0$, $m_2 = 1 + 0m_1 + 0.99m_2 + 0.01m_3$, and $m_3 = 1 + 0.01m_1 + 0.99m_2 + 0m_3$. These equations can be solved to get

$$m = \begin{bmatrix} 0 \\ 1.01 \times 10^4 \\ 10^4 \end{bmatrix}. \tag{28}$$

This means if we start at state x_2, we expect to take 1.01×10^4 steps before escape.

By using (8), it is straightforward to obtain

$$m = \begin{bmatrix} 0 \\ (I - \hat{T}')^{-1}1 \end{bmatrix}. \tag{29}$$

To be able to calculate (29), we need $(I - \hat{T}')$ to be invertible. This is equivalent to having $\lambda_2 < 1$, which is the hidden assumption we made while defining the MFPT vector. The system-wide MFPT calculated in (23) can be also obtained by

$$M = m'\phi = \frac{1}{\|z\|_1}[1 \ldots 1](I - \hat{T})^{-1}z. \tag{30}$$

This makes sense because each state has its own MFPT, and MFPT of the metastable distribution is just a convex combination of each state's MFPT weighted according to ϕ. We show the equivalence next.

$$\begin{aligned} \hat{M} &= \frac{1}{\|z\|_1}[1 \ldots 1](I - \hat{T})^{-1}z \\ &= \frac{1}{\|z\|_1}[1 \ldots 1](I - \hat{T})^{-1}(I - \hat{T} + \hat{T})z \\ &= \frac{1}{\|z\|_1}[1 \ldots 1](I + (I - \hat{T})^{-1}\hat{T})z \\ &= \frac{1}{\|z\|_1}[1 \ldots 1]z + \frac{1}{\|z\|_1}[1 \ldots 1](I - \hat{T})^{-1}\hat{T}z \\ &= 1 + \lambda_2\frac{1}{\|z\|_1}[1 \ldots 1](I - \hat{T})^{-1}z \\ &= 1 + \lambda_2\hat{M} \quad \Longrightarrow \hat{M} = 1/(1 - \lambda_2) = M \end{aligned} \tag{31}$$

Note that M is upper bounded by the largest element in m. In fact, any initial state distribution, $p[0]$, will have an MFPT that is a convex combination of the m_i values.

3.4 Why (system-wide) MFPT?

We would like to answer why we should be using MFPT of the metastable distribution. First of all, it is a lower bound for average steps taken from at least one of the states, because it is a convex combination of m_is. So, there are state(s) at least as stable. Secondly, it is a good measure of overall stability. Often systems quickly converge to their metastable distributions, where MFPT becomes the true value. Thirdly, system-wide MFPT also has advantages over calculating the MFPT vector. In case T_s is very large, estimating the second largest eigenvalue is relatively very easy, whereas finding the inverse to calculate MFPT vector costs more time. Also, a scalar representing the stability is much easier to understand than a possibly huge vector.

Let's assume (T_s) has distinct eigenvalues. Please see [17] for other cases. Let the initial distribution be

$$p[0] = c_1 v_1 + c_2 v_2 + ...c_\ell v_\ell. \tag{32}$$

Note that $c_1 = 1$ to have $\|p[0]\|_1 = 1$. Then,

$$p[n] = (T_s')^n p[0] = v_1 + c_2 \lambda_2^n v_2 + ... + c_\ell \lambda_\ell^n v_\ell. \tag{33}$$

In the light of this section we see that the metastable distribution is also given by

$$\phi_i = \lim_{n\to\infty} Pr(X[n] = x_i \;|X[n] \neq x_1]), \tag{34}$$

when the limit exists. (It does for distinct eigenvalues)

3.5 How quickly is the Initial Distribution Forgotten?

First, let's explain what we mean by being "forgotten". We say the initial distribution (condition) is forgotten if either the distribution is the metastable distribution ($p = \phi$) or the game ended (halt state) ($p = [1\ 0\ ...\ 0]'$). So, the question can be paraphrased as "how quickly do we converge towards the metastable distribution, given the system is not absorbed yet?". For systems with distinct eigenvalues, we propose using

$$\text{memory constant} = \frac{1-\lambda_2}{1-|\lambda_3|}. \tag{35}$$

We are motivated by the fact that $\sum_{n=0}^{\infty} \lambda_i^n = 1/(1-\lambda_i)$ for $|\lambda_i| < 1$. Although this holds for complex λ_i values too, since $|\lambda_i^n| = |\lambda_i|^n$, we use $1/(1-|\lambda_i|)$ to quantify how many steps it takes before vanishing. The higher λ_3 is, the slower the initial condition is forgotten, i.e., the more steps are required to forget the same amount of initial condition information ($1/(1-|\lambda_3|)$ is higher). We look at λ_3, because it gives the worst case scenario, i.e., it gives a conservative value. We divide ($1/(1-|\lambda_3|)$) by $M = 1/(1-\lambda_2)$ to get a relative memory constant, which is upper bounded by 1.

It is also worth noting that to get a memory constant of 10^{-6}, we need $\lambda_2 = 1 - 10^{-6}(1-|\lambda_3|) > 1 - 10^{-6}$, or equivalently MFPT $M > 10^6$. Thus, small memory constants require metastability.

If the memory constant is very close to one, then we have another mode almost as stable (as the one associated with λ_2). In such cases it may be useful to use the next $|\lambda_i|$ instead of $|\lambda_3|$, until we have a memory much smaller than one.

When the eigenvalues are not necessarily distinct, we may have $\lambda_2 = |\lambda_3|$. This case is very similar to the one just explained.

3.6 Epidemics

For the epidemics model outlined in Section 2.2, the stochastic matrix is given by

$$\begin{aligned} T_s &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ (1-\beta)\delta & (1-\delta)(1-\beta) & \beta\delta & \beta(1-\delta) \\ (1-\beta)\delta & \beta\delta & (1-\delta)(1-\beta) & \beta(1-\delta) \\ \delta\delta & \delta(1-\delta) & \delta(1-\delta) & (1-\delta)(1-\delta) \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.002 & 0.198 & 0.008 & 0.792 \\ 0.002 & 0.008 & 0.198 & 0.792 \\ 0.001 & 0.0099 & 0.0099 & 0.9801 \end{bmatrix}. \end{aligned} \tag{36}$$

λ_2 is such that MFPT is $M = 6.8383 \times 10^3$. In addition, $\lambda_3 = 0.19$, and the memory constant is 1.8×10^{-4}. Small memory constant results in states having a MFPT close to either zero or M.

$$m = \begin{bmatrix} 0 \\ 6822.7 \\ 6822.7 \\ 6838.7 \end{bmatrix} \qquad \phi = \begin{bmatrix} 0 \\ 0.0122 \\ 0.0122 \\ 0.9757 \end{bmatrix} \tag{37}$$

3.7 Europe Tour

We now look at the example provided in Section 2.3. We have $\lambda_2 = 0.866$ and $\lambda_3 = 0.6355$. As a result MFPT is $M = 7.4643$ and the memory constant is 0.3676. Due to the high memory constant, we see a high variation between the MFPT of each state.

$$m = \begin{bmatrix} 0 \\ 8.4084 \\ 1.265 \\ 4.5381 \\ 10.6749 \\ 2.064 \\ 2.2767 \\ 8.2196 \end{bmatrix} \qquad \phi = \begin{bmatrix} 0 \\ 0.4620 \\ 0 \\ 0.2006 \\ 0.1028 \\ 0.0253 \\ 0.0338 \\ 0.1754 \end{bmatrix} \tag{38}$$

As an interesting fact, note that there is a zero probability of being in Athens in the metastable distribution, because you can only go to Athens from Istanbul, which is the absorbing state.

3.8 Modified Europe Tour

Since metastable systems with a small memory constant are of interest to the authors, we modify the Europe tour example by hypothetically assuming the population of Paris is 1 billion instead. Then, we have a MFPT of $M = 10,823$ and the memory constant drops to 1.1688×10^{-4}. This results in system-wide

MFPT being more "valid", because most of the states either have a MFPT close to either zero or M.

$$m = \begin{bmatrix} 0 \\ 10,825 \\ 1 \\ 10,652 \\ 10,825 \\ 2,121 \\ 10,674 \\ 10,824 \end{bmatrix} \qquad \phi = \begin{bmatrix} 0 \\ 0.0081 \\ 0 \\ 0.0034 \\ 0.0031 \\ 0 \\ 0.0027 \\ 0.9826 \end{bmatrix} \tag{39}$$

This time in addition to Athens, Kiev also no longer appears in the metastable distribution. This is because within several steps we will typically move directly from Kiev to either Istanbul (the absorbing state) or Berlin. The latter almost implies going to Paris in the following step due to high population there. Kiev has a MFPT that is not very small or close to the system-wide MFPT, because $m_1 = 0$, $m_4 \approx M$, and

$$m_6 = T_{s\{61\}} m_1 + T_{s\{64\}} m_4 + T_{s\{66\}} m_6 \tag{40}$$

implies

$$m_6 \approx \frac{T_{s\{64\}}}{1 - T_{s\{66\}}} = 2,153. \tag{41}$$

4 Measuring Metrics Other Than Steps

Calculating the discrete time to a state of interest has potential to be very useful, but in some applications we might be interested in metrics other than discrete time. To illustrate, we will build on the modified Europe Tour of Section 3.8. Instead of how many decisions it took before reaching Istanbul, we might be interested in travel-time or distance to Istanbul. For this matter, we propose the term Mean First Passage Value (MFPV), where "value" depends on the task, e.g., Mean First Passage Distance.

4.1 Mean First Passage Value (MFPV)

First, note that Mean First Passage Value (MFPV) is a generalization of MFPT. Equivalently, MFPT is a special case of MFPV, where value is discrete time steps.

Let us start by redefining m from (27) as the MFPV vector, which gives the MFPV for each state, as

$$m_i := \begin{cases} 0 & i = 1 \\ \sum\limits_j T_{s\{ij\}} T_{v\{ij\}} + \sum\limits_j T_{s\{ij\}} m_j & otherwise, \end{cases} \tag{42}$$

where $T_{v\{ij\}}$ is the value (reward) of transitioning from state x_i to x_j. For example, travel from Rome to Paris takes 12 hours 46 minutes. Then, we can calculate vector m as

$$m = \begin{bmatrix} 0 \\ (I - \hat{T}')^{-1} \begin{bmatrix} \sum_j T_{s\{2j\}} T_{v\{2j\}} \\ \cdot \\ \cdot \\ \cdot \\ \sum_j T_{s\{\ell j\}} T_{v\{\ell j\}} \end{bmatrix} \end{bmatrix}. \tag{43}$$

And the (system-wide) MFPV is simply

$$M = m'\phi. \tag{44}$$

4.2 Modified Europe Tour

For the modified Europe tour example and the information provided in Table 2, Mean First Passage Distance is 325.68 thousand km. This is achieved by setting, for example, $T_{v\{24\}} = 1,098$ km, which corresponds to the distance between London and Berlin.

Table 2: Explanation of each state

	Time (h:m)	Distance (km)
London (2) - Paris (8)	5:06	454
London (2) - Berlin (4)	10:25	1,098
Madrid (5) - Paris (8)	11:10	1,270
Rome (7) - Paris (8)	12:46	1,419
Berlin (4) - Paris (8)	9:18	1,055
Berlin (4) - Kiev (6)	14:41	1,329
Berlin (4) - Istanbul (1)	21:39	2,210
Kiev (6) - Istanbul (1)	19:00	1,459
Rome (7) - Istanbul (1)	22:46	2,262
Athens (3) - Istanbul (1)	11:13	1,095

To calculate the travel-time only (excluding the days spent in a city), we set $T_{v\{ii\}} = 0$. Then, it takes 128 days on average before we are in Istanbul. The

MFPV vector for this case is

$$m = \begin{bmatrix} 0 \\ 128.4756 \\ 0.4674 \\ 126.6098 \\ 128.7334 \\ 25.9478 \\ 127.0149 \\ 128.2681 \end{bmatrix} \text{ days.} \tag{45}$$

On the other hand, if we also include the days spent in a city (remember that a decision is made after spending a day in the city), we need to set $T_{v\{ii\}} = 1$ day to obtain MFPV of 29 years, which is much higher than 128 days. One reason of this is because once in Paris, there is a 0.9825 probability to stay in Paris.

4.3 The Value

Often, one wishes to include multiple objectives in a single value function, for example for walking robots penalizing failure events (e.g., falling down) while also rewarding fast speed and low energy use. This can be achieved since the value (i.e., cost function) does not need to have a physical correspondence, number of steps minus 10^{-3} times energy consumption is a valid value definition. Please see [11] for further details and usage of this example.

5 Confidence Levels on Value

In some applications, instead of the mean FPV, we may want to have a conservative FPV bound for a particular "confidence level", *pr*. That is, one would observe a value above LFPT with probability *pr*, and one would observe a value below UFPT with probability *pr*.

5.1 First Passage Time (FPT)

The probability of taking more than LFPT steps is λ_2^{LFPT}. Then, the lower bound on number of steps taken with probability *pr* can be calculated by

$$\text{LFPT}(pr) = log_{\lambda_2} pr. \tag{46}$$

Note that LFPT(1)=0, so we can only guarantee taking a single step, which leads to the halt state. The probability of taking less than UFPT steps is $1-\lambda_2^{\text{UFPT}-1}$. Then, the upper bound on number of steps taken with probability *pr* is

$$\text{UFPT}(pr) = log_{\lambda_2}(1 - pr) + 1. \tag{47}$$

Note that $\lim_{pr\to 1} \text{UFPT}(pr) = \infty$, so it may take infinitely many steps before converging to the halt state (theoretically). We can then limit the FPT for a given probability

$$\text{LFPT} \leq \text{FPT} \leq \text{UFPT}. \tag{48}$$

To have LFPT<UFPT we require $pr > \lambda_2/(1+\lambda_2)$, which is approximately 0.5 for $\lambda_2 \approx 1$. To illustrate the advantage of looking to FPT, let's consider the modified Europe tour example. The probability of the journey taking more than $M = 10,823$ days is only 36.79 %. On the other hand, we have

$$\begin{gathered} 1,140 \leq \text{FPT}(0.9) \leq 24,921 \\ 108 \leq \text{FPT}(0.99) \leq 49,842. \end{gathered} \tag{49}$$

These numbers are not coincidence. As $\lambda_2 \to 1$, the probability of taking M steps is

$$\lim_{\lambda_2\to 1} \lambda_2^{1/(1-\lambda_2)} = \frac{1}{e} \approx 0.3679. \tag{50}$$

In fact, 0.3679 is an upper limit for the probability of taking $1/(1-\lambda_2)$ steps for any λ_2. In addition, when λ_2 and pr are close to one, we have

$$\text{LFPT}(pr) = \frac{ln(pr)}{ln(\lambda_2)} \approx \frac{1-pr}{1-\lambda_2} = (1-pr)M, \tag{51}$$

$$\text{UFPT}(pr) = \frac{ln(1-pr)}{ln(\lambda_2)} + 1 \approx -ln(1-pr)M, \tag{52}$$

where M denotes MFPT.

Note that LFPT is of interest for walking systems, to give a conservative bound on steps to failure, while UFPT would be helpful in modeling epidemics, to get a conservative time when everyone will be healthy (i.e., recurrence to an "all-healthy" system state).

5.2 First Passage Value

Using MFPT and MFPV we can calculate value per step. We then, simply use

$$\text{FPV} = \frac{\text{MFPV}}{\text{MFPT}} \,\text{FPT}. \tag{53}$$

When the value is the distance in the modified Europe tour example, we obtain

$$\begin{gathered} 3.4 \times 10^4 \leq \text{FPV}(0.9) \leq 7.5 \times 10^5 \\ 3.2 \times 10^3 \leq \text{FPV}(0.99) \leq 1.5 \times 10^6. \end{gathered} \tag{54}$$

So, with probability 0.99, the journey will take more than 3.2 thousand or less than 1.5 million kilometers.

6 Control Applications

In order to control a system, one needs a goal, e.g., for the Europe tour example, trying to go to Istanbul as quickly as possible. It is then useful to quantify the performance towards achieving that goal, e.g., number of days before reaching Istanbul would be a meaningful metric to minimize. We believe FPV tool would give useful metrics for many applications. To illustrate, the more steps a walking robot takes before falling, the more stable it is said to be. Once we quantify the performance, we can optimize the control.

For the examples we studied in the previous sections, there was no control. We just illustrated how to calculate FPV for a given Markov Chain. There are two ways control action comes into play: Low-level and high-level. Let's explain by illustrating with the Europe tour. Say the probability for city change is directly proportional to population to the power k, where $1 \leq k \leq 3$. What we previously studied was a special case of this setting with $k = 1$. Any choice of k will give a Markov Chain, for which we calculate FPV as shown. Choosing k is the low-level control problem. Moreover, if we are capable of choosing an integer k every time a city is to be chosen, then we have three low-level controllers available. Deciding which one to use for a given city is the high-level control problem.

Remember that the first three examples in Section 2 were all discrete-time. So, they were special cases of hybrid systems, which may have either or both discontinuities and continuous phases. For systems in which an exact (discrete) Markov chain does not naturally exist, we can create one, using the meshing process described below.

6.1 Hybrid Model

Let x, γ, and ζ be the internal state, the randomness system experiences, and the control action respectively. To illustrate, for a walking robot, x is the robot's state, γ is random variable representing factors such as terrain variation or system noise, and ζ is the control action which may be a function of x and γ. We define vector $y := [x; \gamma; \zeta]$ to represent them all. Then our general hybrid model is represented as

$$\begin{aligned} \dot{y} &= f(y) \qquad & y \in C \\ y^+ &= g(y) \qquad & y \in D. \end{aligned} \tag{55}$$

C and D are known as flow and jump sets [18]. Note that this setting is compatible with less general cases like continuous and discrete systems with/without a control action or randomness.

6.2 Meshing for Markov Decision Process (MDP) Model

The first step is choosing a Poincaré-like section, noted by S, which does not necessarily decrease the dimension of the state. However, if the system has not yet escaped (from the region of interest), it needs to keep passing through this section. For example, the hybrid dynamics of walking systems are punctuated by discrete impacts when a foot comes into contact with the ground. These impacts provide a natural discretization of the robot motion.

After defining this section, we abuse the notation and refer to $x \in S$ simply by x. Then, the next state (intersecting S) is a function of the current state $x[n]$, the randomness experienced $\gamma[n]$, and the controller action during that step $\zeta[n]$ i.e.,

$$x[n+1] = h(x[n], \gamma[n], \zeta[n]). \tag{56}$$

To obtain a (discrete) Markov Decision Process (MDP) model, we need to have finite sets for control action, randomness, and state. The first one is rather easy, we simply design finitely many low-level controllers. The second (randomness), is straightforward to handle when number of noise sources is low, e.g., when randomness is in the slope ahead for a walking robot, randomness set is just one dimensional, which is very easy to discretize by meshing. State space can be discretized similarly when the state is low dimensional. However, if the state is high dimensional, discretization is not as intuitive. In case the intrinsic dimension is low, we can still mesh by cleverly exploring the reachable state space, as explained in [13].

Once we have finite control, randomness and state sets, we simply simulate for each possible $h(x, \gamma, \zeta)$ to obtain a MDP similar to Figure 5, where $\gamma[n] \in \{\gamma_1, \gamma_2\}$ and $\zeta[n] \in \{\zeta_1, \zeta_2\}$, i.e., there are two available actions (low-level control) and two possible randomness. Also there are just 3 states, that is $x[n] \in \{x_1, x_2, x_3\}$.

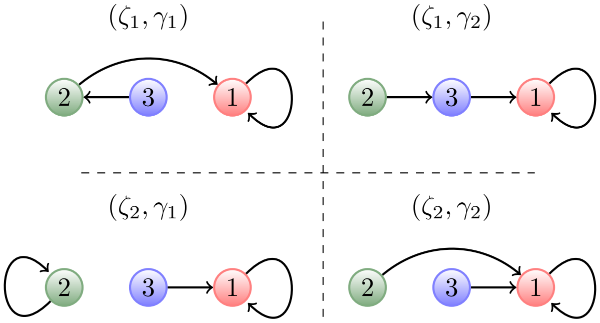

Figure 5: Representation of a Markov Decision Process with two available actions and two possible randomness

6.3 Policy for ζ

Next, we describe the methodology for deriving the Markov chain model from a MDP. A policy, π, is what determines which (low-level) control action to take at each step. It is the high-level control. Optimal and robust policies can be obtained using dynamic programming tools [10, 19]. Let's say, we decide to use policy

$$\pi(x[n],\gamma[n]) = \begin{cases} \zeta_2 \text{ if } x[n]=x_2 \text{ and } \gamma[n]=\gamma_1 \\ \zeta_1 \text{ if } x[n]=x_2 \text{ and } \gamma[n]=\gamma_2 \\ \zeta_1 \text{ if } x[n]=x_3 \text{ and } \gamma[n]=\gamma_1 \\ \zeta_2 \text{ if } x[n]=x_3 \text{ and } \gamma[n]=\gamma_2. \end{cases} \tag{57}$$

Notice that we don't determine a control action for state x_1, since nothing can be done differently at halt state. When we use $\zeta[n]=\pi(x[n],\gamma[n])$, (56) becomes

$$x[n+1] = h(x[n],\gamma[n],\pi(x[n],\gamma[n])), \tag{58}$$

which is a function of the state and randomness only. The result is illustrated in Figure 6.

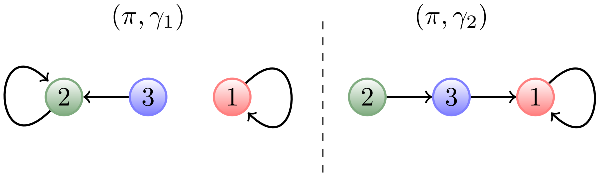

Figure 6: MDP of Figure 5 after applying policy π defined in (57)

6.4 Distribution for γ

The last step before obtaining an absorbing Markov chain is to assume a distribution for randomness. For Figure 6, say $P(\gamma_2)=0.01$, i.e., with probability 0.01, $\gamma[n]$ will be γ_2. We then obtain Figure 4, which we already studied. More complicated systems will end up being surprisingly similar to Figure 4.

7 Conclusion

In this paper, we studied First Passage Time, which is a survival metric. After providing some motivational examples, we presented system-wide Mean First Passage Time (MFPT), which is calculated using the second largest eigenvalue of the stochastic transition matrix. We showed that for metastable systems, system-wide MFPT is an accurate indicator across a large set of states, including those frequently visited. We then introduced Mean First Passage Value (MFPV), which gives a more general value of interest, e.g., energy expenditure, distance, or time. We then provided bounds on First Passage Value (FPV) for a given confidence level. The last section of this paper was perhaps the most important one, because it showed how these tools explained can be used to low-level and high-level control a hybrid systems.

Acknowledgments

This work was supported by the Institute for Collaborative Biotechnologies through grant W911NF-09-0001 from the U.S. Army Research Office. The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

We would also like to thank Beril Pisgin for drawing Figure 3.

References

[1] P. Talkner, P. Hanggi, E. Freidkin, and D. Trautmann, “Discrete dynamics and metastability: Mean first passage times and escape rates,” *Journal of Statistical Physics*, vol. 48, no. 1-2, pp. 231–254, Jul. 1987. [Online]. Available: http://link.springer.com/article/10.1007/BF01010408

[2] P. Hanggi, P. Talkner, and M. Borkovec, “Reaction-rate theory: fifty years after kramers,” *Reviews of Modern Physics*, vol. 62, no. 2, pp. 251–341, Apr. 1990. [Online]. Available: http://link.aps.org/doi/10.1103/RevModPhys.62.251

[3] R. Muller, P. Talkner, and P. Reimann, “Rates and mean first passage times,” *Physica A: Statistical Mechanics and its Applications*, vol. 247, no. 14, pp. 338–356, Dec. 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378437197003907

[4] N. Kampen, *Stochastic Processes in Physics and Chemistry.* Elsevier, Aug. 2011.

[5] A. E. Larsen and D. G. Grier, “Like-charge attractions in metastable colloidal crystallites,” *Nature*, vol. 385, no. 6613, pp. 230–233, Jan. 1997. [Online]. Available: http://www.nature.com/nature/journal/v385/n6613/abs/385230a0.html

[6] H. J. Veendrick, “The behaviour of flip-flops used as synchronizers and prediction of their failure rate,” *IEEE Journal of Solid-State Circuits*, vol. 15, no. 2, pp. 169–176, Apr. 1980.

[7] A. A. Fingelkurts and A. A. Fingelkurts, “Making complexity simpler: multivariability and metastability in the brain,” *The International Journal of Neuroscience*, vol. 114, no. 7, pp. 843–862, Jul. 2004.

[8] K. Byl and R. Tedrake, “Metastable walking machines,” *The International Journal of Robotics Research*, vol. 28, no. 8, pp. 1040–1064, Aug. 2009. [Online]. Available: http://ijr.sagepub.com/cgi/doi/10.1177/0278364909340446

[9] M. Benallegue and J.-P. Laumond, “Metastability for high-dimensional walking systems on stochastically rough terrain.” in *Robotics: Science and Systems.* Citeseer, 2013.

[10] C. O. Saglam and K. Byl, “Robust policies via meshing for metastable rough terrain walking,” in *Proceedings of Robotics: Science and Systems*, Berkeley, USA, 2014.

[11] ——, “Quantifying the trade-offs between stability versus energy use for underactuated biped walking,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[12] H. J. Ahn and B. Hassibi, “Global dynamics of epidemic spread over complex networks,” in *2013 IEEE 52nd Annual Conference on Decision and Control (CDC)*, Dec. 2013, pp. 4579–4585.

[13] C. O. Saglam and K. Byl, “Robust policies via meshing for metastable walking,” in *International Journal Robotics Research (IJRR)*, 2015, invited and submitted.

[14] C. O. Saglam, A. Teel, and K. Byl, “Lyapunov-based versus Poincare map analysis of the rimless wheel,” in *IEEE Conference on Decision and Control (CDC)*, Dec. 2014, accepted for publication.

[15] K. Matthews, “Number theory web,” 1995. [Online]. Available: http://www.numbertheory.org/courses/MP274/markov.pdf

[16] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra.* SIAM, Jun. 2000.

[17] C. O. Saglam and K. Byl, “Metastable markov chains,” in *IEEE Conference on Decision and Control (CDC)*, Dec. 2014, accepted for publication.

[18] R. Goebel, R. G. Sanfelice, and A. R. Teel, *Hybrid Dynamical Systems: Modeling, Stability, and Robustness.* Princeton University Press, 2012.

[19] R. Bellman, “A markovian decision process,” *Indiana University Mathematics Journal*, vol. 6, no. 4, pp. 679–684, 1957. [Online]. Available: http://www.iumj.indiana.edu/IUMJ/FULLTEXT/1957/6/56038