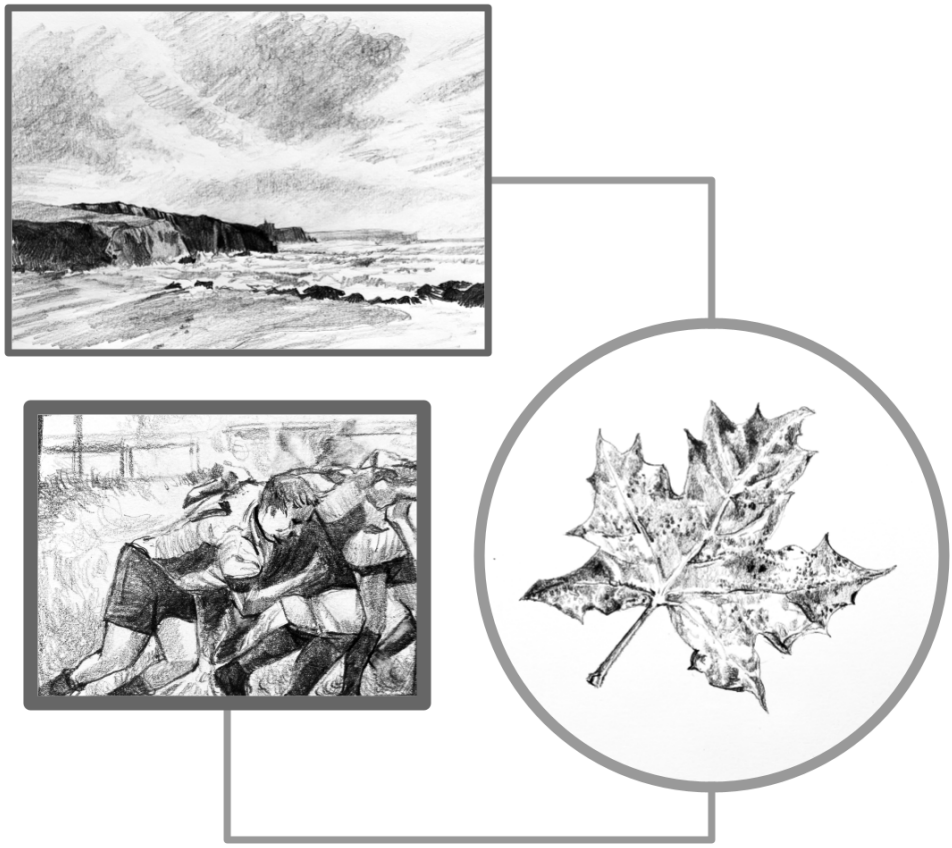

11

Managing a rugby match

Concept. Building a toy model simulation of a rugby match whose outcome can be manipulated through correctly-timed player substitutions and game management decisions. The dexetera state manipulation framework we have built around the stochadex can meet these requirements, and a dashboard can be created for user interaction. All this combines together to make a simple dashboard game, which we call: 'trywizard'. For the mathematically-inclined, this chapter will motivate the construction of a specific modelling framework for rugby match simulation. For the programmers, the public Git repository for the code described in this chapter can be found here: https://github.com/umbralcalc/trywizard.

11.1 Designing the event simulation engine

Since the basic state manipulation framework and simulation engine will run using dexetera, the mathematical novelties in this project are all in the design of the rugby match model itself. And, as ever, we're not especially keen on spending a lot of time doing detailed data analysis to come up with the most realistic values for the parameters that are dreamed up here. Even though this would also be interesting.[1]

Let's begin by specifying an appropriate state space to live in when simulating a rugby match. It is important at this level that events are defined in quite broadly applicable terms, as it will define the state space available to our stochastic sampler and hence the simulated game will never be allowed to exist outside of it. It seems reasonable to characterise a rugby union match by the following set of states: Penalty, Free Kick (the punitive states); Penalty Goal, Drop Goal, Try (the scoring states); Kick Phase, Run Phase, Knock-on, Scrum, Lineout, Maul and Ruck (the general play states). Using this set of states, in Fig. 11.1 we have summarised our approach to match state transitions into a single event graph. In order to capture the fully detailed range of events that are possible in a real-world match, we've needed to be a little imaginative in how we define the

[1] One could do this data analysis, for instance, by scraping player-level performance data from one of the excellent websites that collect live commentary data such as rugbypass.com or espn.co.uk/rugby.

kinds of state transitions which occur. In other words; it's fair to say that our simplified model here represents just a subset of states that a real rugby match could exist in.

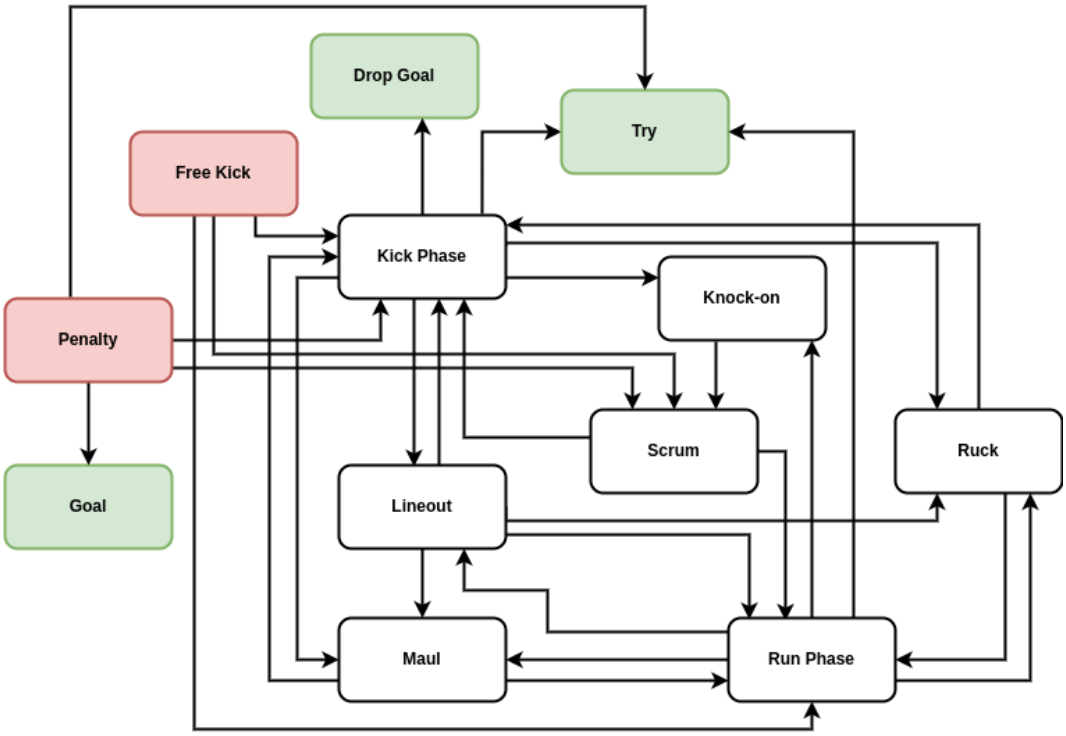

Figure 11.1: Simplified event graph of a rugby union match.

In addition to occupying some state in the event graph, the state of a rugby match must also include a binary 'possession' element which encodes which team has the ball at any moment. We should also include the 2-dimensional pitch location of the ball as an element of the match state in order to get a better sense of how likely some state transitions are, e.g., when playing on the edge of the pitch near the touchline it's clearly more likely that a Run Phase is going to result in a Lineout than if the state is currently in the centre of the pitch. To add even more detail; in the next section we will introduce states for each player.

Since a rugby match exists in continuous time, it is natural to choose a continuous-time event-based simulation model for our game engine. As we have discussed in previous chapters already, this means we will be characterising transition probabilities of the event graph in Fig. 11.1 by ratios of event rates in time. Recalling our notation in previous chapters, if we consider the current state vector of the match X_t, we can start by assigning each transition $\mathsf{a} \to \mathsf{b}$ on the event graph an associated expected rate of occurance $\lambda_{\mathsf{a}\to\mathsf{b}}$ which is defined in units of continuous time, e.g., seconds. In addition to the transitions displayed on the graph, we can add a 'possession change transition'; where the possession of the ball in play moves to the opposing team. This transition may occur while the match is also in any of the white-coloured states on the graph (apart from Knock-on which determines a possession change immediately through a Scrum) and let's assign this

a state, parameter and timestep-dependent expected rate of occurance $\lambda_{\text{pos}}(X_{\mathsf{t}}, z, \mathsf{t})$.

Based on our dicussion above, an appropriate encoding for the overall game state at timestep index t could be a state vector X_{t} whose elements are

$$X^0_{\mathsf{t}} = \begin{cases} 0 & \text{Match State} = \textsf{Penalty} \\ 1 & \text{Match State} = \textsf{Free Kick} \\ \dots & \end{cases} \tag{11.1}$$

$$X^1_{\mathsf{t}} = \begin{cases} 0 & \text{Possession} = \textsf{Home Team} \\ 1 & \text{Possession} = \textsf{Away Team} \end{cases} . \tag{11.2}$$

But how does this overall game state connect to the event rates? The probabilistic answer is quite straightforward. If the probability of the match state being $X^0_{\mathsf{t}} = \mathsf{a}$ at timestep t is written as $P^0_{\mathsf{t}}(\mathsf{a})$, then the probability of $X^0_{\mathsf{t}+1} = \mathsf{b}$ in the following timestep is

$$P^0_{\mathsf{t}+1}(\mathsf{b}) = \frac{\frac{1}{\tau} P^0_{\mathsf{t}}(\mathsf{b}) + \sum_{\forall \mathsf{a} \neq \mathsf{b}} \lambda_{\mathsf{a}\to\mathsf{b}} \mathcal{T}_{\mathsf{a}\to\mathsf{b}}(X_{\mathsf{t}}, z, \mathsf{t}) P^0_{\mathsf{t}}(\mathsf{a})}{\left[\frac{1}{\tau} + \sum_{\forall \mathsf{a} \neq \mathsf{b}} \lambda_{\mathsf{a}\to\mathsf{b}} \mathcal{T}_{\mathsf{a}\to\mathsf{b}}(X_{\mathsf{t}}, z, \mathsf{t})\right]} , \tag{11.3}$$

where $\forall \mathsf{a} \neq \mathsf{b}$ in the summation indicates that all the available transitions from a to b, where $\mathsf{a} \neq \mathsf{b}$, should be summed over and $\mathcal{T}_{\mathsf{a}\to\mathsf{b}}(X_{\mathsf{t}}, z, \mathsf{t})$ is a time, parameter and state-dependent transition probability that is determined by the playing tactics of each team as well as the general likelihoods of gameplay which are expected from a real rugby match. Note that in the expression above, we have also defined τ as a timescale short enough such that no transition is likely to occur during that interval. An equivalent to Eq. (11.3) should also apply to the possession change transition rate, i.e., the probability that the Home Team has possession P^1_{t} at time t evolves according to

$$P^1_{\mathsf{t}+1} = \frac{\frac{1}{\tau} P^1_{\mathsf{t}} + \lambda_{\text{pos}}(X_{\mathsf{t}}, z, \mathsf{t})(1 - P^1_{\mathsf{t}})}{\left[\frac{1}{\tau} + \lambda_{\text{pos}}(X_{\mathsf{t}}, z, \mathsf{t})\right]} . \tag{11.4}$$

Before we move on to other details, it's quite important to recognise that because our process is defined in continuous time, the possession change rate may well vary continuously (this will be especially true when we talk about, e.g., player fatigue). Hence, Eq. (11.3) is only an *approximation* of the true underlying dynamics that we are trying to simulate — and this approximation will only be accurate if τ is small. The reader may recall that we discussed this same issue from the point of view of simulating time-inhomogeneous Poisson processes with the rejection method when we were building the stochadex in an earlier chapter.

While these match state transitions and possession changes are taking place, we also need to come up with a model for how the ball location L_{t} changes during the course of a game, and as a function of the current game state. Note that, because the ball location is a part of the overall game state, it will be included as information contained within some elements of X_{t} as well. To make this explicit, we can simply set $X^2_{\mathsf{t}} = L^{\text{lon}}_{\mathsf{t}}$ and $X^3_{\mathsf{t}} = L^{\text{lat}}_{\mathsf{t}}$ — where $L^{\text{lon}}_{\mathsf{t}}$ denotes the longitudinal component (lengthwise along the pitch) and $L^{\text{lat}}_{\mathsf{t}}$ denotes the lateral component (widthwise across the pitch). If we associate every state on the event graph with a single change in spatial location of the ball on the pitch, we then need to construct a process which makes 'jumps' in 2-dimensional space each time a state transition occurs. To keep things simple and intuitive, we will say that movements of the ball are only allowed to occur during either a Run Phase or a Kick Phase.

In the case of a Run Phase, let's choose the longitudinal component of the ball location $L_{\mathsf{t}}^{\text{lon}}$ to be updated by the difference between samples drawn from two exponential distributions (one associated to each team). Hence, the probability density $P_{\mathsf{t}}(\ell)$ of $L_{\mathsf{t}}^{\text{lon}} = \ell$ at timestep t evolves according to

$$P_{\mathsf{t}+1}(\ell) = \int_0^\infty \mathrm{d}\ell' \, \mathsf{ExponentialPDF}(\ell + \ell'; a_{\text{run}}) \mathsf{ExponentialPDF}(\ell'; d_{\text{run}}) \,, \tag{11.5}$$

where a_{run} and d_{run} are the exponential distribution scale parameters for an attacking and defending player, respectively, and we have chosen positive values of ℓ to be aligned with the forward direction for the attacking team. We shall elaborate on where a_{run} and d_{run} come from when we discuss associating events for player abilities in due course. If we now consider lateral component of the ball location $L_{\mathsf{t}}^{\text{lat}}$ during a Run Phase; it makes sense that this wouldn't be affected much by either team within the scope of detail in this first version of our model. Hence, the probability density $P_{\mathsf{t}}(w)$ of $L_{\mathsf{t}}^{\text{lat}} = w$ at timestep t can just be updated like so

$$P_{\mathsf{t}+1}(w) = \mathsf{NormalPDF}(w; L_{\mathsf{t}}^{\text{lat}}, \sigma_{\text{run}}^2) \,, \tag{11.6}$$

where σ_{run} is the typical jump in lateral motion (the standard deviation parameter of the normal distribution).

Turning our attention to the Kick Phase; the longitudinal component is only realistically controlled by the attacking team. There's also an angular component to this motion which is associated to the direction that the kicker is orientating their kick. Therefore, the probability density of $P_{\mathsf{t}}(\ell)$ with $L_{\mathsf{t}}^{\text{lon}} = \ell \cos\theta$ and $L_{\mathsf{t}}^{\text{lat}} = \ell \sin\theta$ at timestep t will evolve according to

$$P_{\mathsf{t}+1}(\ell) = \mathsf{ExponentialPDF}(\ell; a_{\text{kick}}) \,, \tag{11.7}$$

where a_{kick} this time is the exponential scale parameter for the attacking team player who is kicking and θ is the angle they have chosen to kick at (where straight ahead is defined as $\theta = 0$). Similarly, the lateral component for a Kick Phase is only going to be controlled by the accuracy of the kicker with the angle and longitudinal distance already applied. Therefore, the corresponding update to the probability density $P_{\mathsf{t}}(w)$ where $L_{\mathsf{t}}^{\text{lon}} = w \sin\theta$ and $L_{\mathsf{t}}^{\text{lat}} = w \cos\theta$ at timestep t will look like

$$P_{\mathsf{t}+1}(w) = \mathsf{NormalPDF}(w; L_{\mathsf{t}}^{\text{lat}} + \ell \sin\theta, \sigma_{\text{kick}}^2) \,, \tag{11.8}$$

with a standard deviation σ_{kick}, and its interpretation now being the accuracy of the kicker.

11.2 Associating events to player states and abilities

In the last section we introduced a continuous-time event-based simulation model for a rugby union match. In this section we are going to add more detail into this model by inventing how to associate specific player states and abilities to the event rates of the simulation. Before continuing, we want to reiterate that this model is entirely made up and, while we hope it illustrates some interesting mathematical modelling ideas in the context of rugby, there's no particular reason why a statistical inference with a reliable dataset should prefer our model to others which may exist.

In Fig. 11.2 we began by separating playing positions on the rugby field into their usual descriptions and then associating each player type with a short list of simplified attributes. Our model is then to associate a player with an 'possession attacking' and 'possession defending' ability which

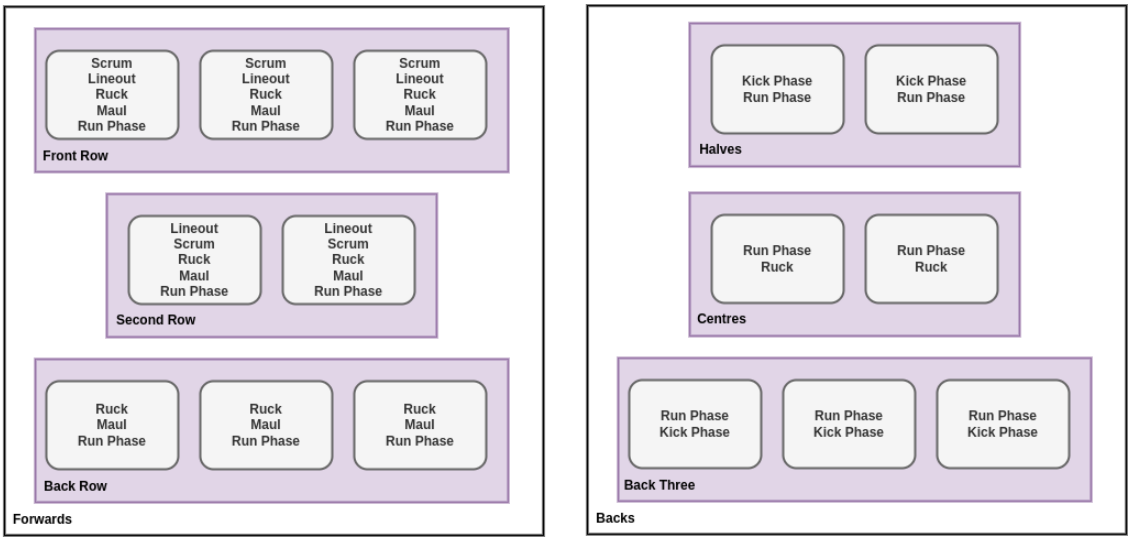

Figure 11.2: Associated playing abilities for each position type.

corresponds to each of their attributes. For example, a Front Row Forward will have 8 abilities associated to them: 2 for each of their Scrum, Lineout, Ruck and Maul attributes.

Let's now say that z contains all of these parameters for all of the players on both sides, and also whether or not each player is actively playing or on the bench. With this information, and the knowledge of which team is in possession from $X_{\mathtt{t}}^1$, it should be simple to create a vector-valued function $a_{\text{pos}}(X_{\mathtt{t}}, z)$ which returns all of the posession attacking attributes that are associated to match state $X_{\mathtt{t}}^0$ and an analogous one $d_{\text{pos}}(X_{\mathtt{t}}, z)$ for the possession defending attributes. The dependencies of these functions on the ball possession state $X_{\mathtt{t}}^1$ comes from the fact that when, e.g., the Home Team has possession of the ball it will be their possession attacking attributes that are returned by $a_{\text{pos}}(X_{\mathtt{t}}, z)$ and the Away Team's possession defending attributes that are returned by $d_{\text{pos}}(X_{\mathtt{t}}, z)$.

In order to model the effect of player fatigue over the course of a match, we can add some vectors of player fatigue values f and times at which each player started playing t_{start} into the collection of parameters that are contained within z. These new parameters can then be used to define a formula for the decline of each attribute over the course of a match. Let's assign these declining values to be part of the overall match state like so

$$X_{\mathtt{t}}^j = a_{\text{pos}}^j(X_{\mathtt{t}}, z)e^{-f^j[t(\mathtt{t})-t_{\text{start}}^j]} \tag{11.9}$$

$$X_{\mathtt{t}}^k = d_{\text{pos}}^k(X_{\mathtt{t}}, z)e^{-f^k[t(\mathtt{t})-t_{\text{start}}^k]}\,, \tag{11.10}$$

where the j and k each index the range of indices in the overall match state which correspond to the attacking and defending team player states, respectively.

So how does each player affect the events of a match? In our model, we would argue that players should be able to directly influence the possession change rate $\lambda_{\text{pos}}(X_{\mathtt{t}}, z, \mathtt{t})$ through a balance of

attacking and defensive attributes in the following relation

$$\lambda_{\text{pos}}(X_{\mathsf{t}}, z, \mathsf{t}) = \frac{\lambda^*_{\text{pos}} \sum_{\forall j} X^j_{\mathsf{t}}}{\sum_{\forall j} X^j_{\mathsf{t}} + \sum_{\forall k} X^k_{\mathsf{t}}}\,, \tag{11.11}$$

where λ^*_{pos} is the maximum rate that is physically possible and the $\forall j$ and $\forall k$ in the summations indicate summing over all attacking and defending player attributes, respectively. In addition to this possession change influence, players who have Run Phase and Kick Phase attributes may affect the gain in distance (and accuracy in the case of Kick Phase) that each state translates to on the pitch.

Let's first describe how we intend the Run Phase to work. Every time the match state transitions into a Run Phase, an individual player on the attacking side is chosen at random (uniformally across the team[2]) to be the the nominal 'attacker'. At the same time, an individual player on the defending side is chosen at random (again, uniformally across the team) to be the the nominal 'defender'. Once these players have been chosen (and hence the a_{run} and d_{run} parameters have been determined), the longitudinal motion update we described in Eq. (11.5) can then be applied. Note that the a_{run} and d_{run} parameters should also receive a fatigue decrement depending on the time that each player has remained on the pitch, much like the exponential factors we applied in Eqs. (11.9) and (11.10).

Finally, we turn or attention to the mechanics of a Kick Phase. For tactical kicking in the middle of play, this operates in a similar way as a Run Phase; one of the players on the attacking side (the side in possession of the ball) is chosen at random (uniformally) and their a_{kick} and σ_{kick} attributes are used in Eqs. (11.7) and (11.8) to update the position of the ball. In the specific case of Goal kicking from a Penalty, the designated place kicker on each side also uses their same Kick Phase attributes to determine the success/failure of the kick at the posts.

11.3 Deciding on managerial actions

So how does managing a rugby match map to taking actions with our formalism? We have to figure out what sorts of managerial actions $\mathcal{A}_{\mathsf{t}}$ are parametric in nature (affecting z) and if any directly act on the state of the system itself (affecting X_{t}). Let's jump straight to the answer to the second question — we can't really think of any situation where the overall match state X_{t} itself is directly changed by a managerial action, so we will be using only parametric actions in this chapter.

Our model structure would suggest that the only way in which a manager can influence the state of a match is through modifying the parameters which are used by the $a_{\text{pos}}(X_{\mathsf{t}}, z)$, $d_{\text{pos}}(X_{\mathsf{t}}, z)$ or $\mathcal{T}_{\mathsf{a}\to\mathsf{b}}(X_{\mathsf{t}}, z, \mathsf{t})$ functions. In the case of the possession attacking and defending attributes $a_{\text{pos}}(X_{\mathsf{t}}, z)$ and $d_{\text{pos}}(X_{\mathsf{t}}, z)$, a parametric action that the manager can perform would be to modify which players are actively playing through substitutions. To indicate that this underlying data may change, we can promote z to its actionable, time-dependent version Z_{t} such that the functions now become $a_{\text{pos}}(X_{\mathsf{t}}, Z_{\mathsf{t}})$ and $d_{\text{pos}}(X_{\mathsf{t}}, Z_{\mathsf{t}})$. In order to map substitutions/initial squad selections to the vector $\mathcal{A}_{\mathsf{t}}$, we can define the first set of 15 indices $\mathcal{A}^i_{\mathsf{t}}$ (where $i = 0, \ldots, 14$) as the player IDs chosen to be on the pitch for either the Home Team or the Away Team, depending on which side the manager is in charge of. If Z_{t} contains all of these IDs and the positions that they are allowed to play (as well as whether they are currently playing or not), then when the manager wishes to make a substitution, all that is needed is a change to $\mathcal{A}^i_{\mathsf{t}}$ and the parametric action function $G_{\mathsf{t}+1}(Z_{\mathsf{t}}, \mathcal{A}_{\mathsf{t}}, \mathsf{t})$ can be defined to make a change to the Z_{t} in response to this.

[2] This uniform sampling could be refined later to associate sampling probabilities with game state and player roles.

But what about $\mathcal{T}_{\mathsf{a}\to\mathsf{b}}(X_\mathsf{t}, Z_\mathsf{t}, \mathsf{t})$? What kinds of managerial actions can change the state transition probabilities through parameters? Given that these transition probabilities mostly arise from the tactics of each team, if the tactics of either team were changed throughout the match due to managerial decisions, these actions could be mapped to $\mathcal{T}_{\mathsf{a}\to\mathsf{b}}(X_\mathsf{t}, Z_\mathsf{t}, \mathsf{t})$. In order for these actions to have a clear influence on the game, however, we need to specify how team tactics get translated into transition probabilities. To keep things as simple as possible, we're going to specify only two tactical 'axes' on which a manager has to decide a position during each moment of the match.

The first, and perhaps more obvious, tactical decision axis to dynamically manipulate is the ratio between Kick Phase and Run Phase that the team chooses when it has possession of the ball, depending on what part of the pitch they are playing in. The second axis maps how aggressively a team pursues scoring tries over any other points (even when they may be on offer from a Penalty Goal). This latter axis also only really matters for the team in possession of the ball, so we aren't planning to map out any defensive tactics in our model for now. Since both of these actions can be mapped to a single axis each, these ratios r (each defined between $1 \geq r \geq 0$) can populate the last two indices of the actions vector: $\mathcal{A}_\mathsf{t}^{15}$ and $\mathcal{A}_\mathsf{t}^{16}$. When either of them has been changed, this can be mapped to Z_t using the parametric action function $G_{\mathsf{t}+1}(Z_\mathsf{t}, \mathcal{A}_\mathsf{t}, \mathsf{t})$ and then Z, in turn, can change the corresponding ratios between transition probabilities $\mathcal{T}_{\mathsf{a}\to\mathsf{b}}(X_\mathsf{t}, Z_\mathsf{t}, \mathsf{t})$ when the attacking team is making these in-play decisions.

Before moving on to the next section, there's a quick point to make about how managerial actions can affect ball locations on the pitch. In addition to the changes that we have discussed above, let's recall that the parameters a_run, d_run, a_kick and σ_kick in Eqs. (11.5), (11.7) and (11.8) can be indirectly determined by the manager to some extent through player selection/substitutions. So, while tactical managerial actions can change the ratios of state transitions themselves (at least while in possession of the ball), the players which are chosen in the first place (or by substitution) can have quite a significant influence on the outcome of a match.

11.4 Writing the game itself

- Show which stochadex/dexetera methods were called and how they were used to simulate the game.
- Give a summary of how the dashboard backend works (diagram would help) and how this connects up to the streamlit frontend via protobuf messages.

Bibliography