

Validation and Optimisation of Player Motion Models in Football

Moritz Renkin¹[0000-0002-1404-857X], Jonas Bischofberger²[0000-0002-0878-0641],
Erich Schikuta¹[0000-0002-4126-4243], and Arnold Baca²[0000-0002-1704-0290]

¹ Research Group Workflow Systems and Technology, Faculty of Computer Science,
University of Vienna, Austria

`a11807211@unet.univie.ac.at`

² Department of Biomechanics, Kinesiology and Computer Science in Sport, Institute
of Sports Science, University of Vienna, Austria

`jonas.bischofberger@univie.ac.at`

Abstract. Modelling the trajectorial motion of humans along the ground is a foundational task in the quantitative analysis of sports like association football. Most existing models of football player motion have not been validated yet with respect to actual data. One of the reasons for this lack is that performing such a validation is not straightforward, because models of player motion are usually phrased in a way that emphasises possibly reachable positions rather than expected positions. Since positional data of football players typically contains outliers, this data may misrepresent the range of actually reachable positions.

This paper proposes a validation routine for trajectorial motion models that measures and optimises the ability of a motion model to accurately predict all possibly reachable positions by favoring the smallest predicted area of reachable positions that encompasses all observed reached positions up to a manually defined threshold. We demonstrate validation and optimisation on four different motion models, assuming (a) motion with constant speed, (b) motion with constant acceleration, (c) motion with constant acceleration with a speed limit, and (d) motion along two segments with constant speed. Our results show that assuming motion with constant speed or constant acceleration without a limit on the achievable speed is particularly inappropriate for an accurate distinction between reachable and unreachable locations. Motion along two segments of constant speed provides by far the highest accuracy among the tested models and serves as an efficient and accurate approximation of real-world player motion.

Keywords: Football · Positional data · Motion models · Performance analysis · Model validation · Complex systems

1 Introduction

Recently, professional association football has seen a surge in the availability of positional data of the players and the ball, typically collected by GPS, radar

or camera systems [10]. The growing availability of such data has opened up an exciting new avenue for performance analysis [8]. Positional data enables the development of sophisticated performance indicators that accurately measure the technical, tactical and athletic performance of teams and players beyond the possibilities of qualitative observation and simple event-based statistics like ball possession and passing accuracy. High-quality measures of performance are invaluable for effective performance analysis, training, opposition scouting, and player recruitment.

The modelling of human motion is a foundational component of many performance metrics based on positional data. For example, algorithms that compute space control [12] or simulate passes [13] implicitly or explicitly make assumptions about human kinematics. These kinematic assumptions have never been verified so far, which calls the validity of these assumptions and the resulting models into question.

In sports with many degrees of freedom like football, predicting player motion is generally a very hard task because player positioning and motion are the result of an intractable, individual decision-making process. Luckily, many applications do not require predictions of the actual position and kinematic state of players but merely of their possibly reachable positions. This requirement essentially shifts the purpose of a motion model from predicting expected positions towards estimating the most remote reachable positions. Estimating such extreme values from real-world data can be difficult, because extreme values are typically rare and particularly likely to include a component of measurement error. Since the distribution of measurement error within player trajectories can vary significantly between data sets due to the use of different measurement systems and post-processing methods, a validation procedure of player motion models needs to flexibly account for various, possibly unknown distributions of measurement error.

The contributions of this paper are twofold: First, we formally propose a validation routine for the quality of player motion models which measures their ability to predict all reachable positions depending on the player's current position and kinematic state. The procedure favors those models that predict the smallest reachable areas which still contain a certain, manually defined proportion of actually observed positions. Second, we use this routine to evaluate and optimise the parameters of four models of motion, assuming (a) motion with constant speed, (b) motion with constant acceleration, (c) motion with constant acceleration with a speed limit, and (d) motion along two segments of constant speed. This evaluation sheds light on the predictive quality of these models and suggests sensible parameter values for them.

The rest of this paper is structured as follows: Section 2 provides some background on motion models in football and their validation. Section 3 formally presents our validation routine. Section 4 describes our exemplary model validation and optimisation based on a real data set and discusses its results. Section 5 summarises the contributions of this paper and points out possible directions of further research.

2 Motion models in football: state of the art

Assumptions about the trajectorial motion of players are inherent to many performance indicators within the analysis of sports games. One example is the commonly used concept of space control which assigns control or influence over different areas on the pitch to players. It is used, for example, as a context variable to rate football actions [4, 11] and for time series analyses [5]. Controlled space is often defined as the area that a player is able to reach before any other player, given a specific model of motion for each player. Commonly used for this purpose are motion models assuming constant and equal speed, which results in a Voronoi partition of the pitch [3], or accelerated movement, possibly including a friction term [6] or velocity-dependent acceleration [14] to limit the achievable speed. Spearman et al. [13] assume accelerated player motion with a limit on both acceleration and velocity in the context of modeling ground passes.

Motion models have also been estimated directly from positional data by fitting a probability distribution over appropriately normalized future positions [1, 2]. However, such empirical models can be computationally expensive, prone to outliers and their current versions lend themselves less naturally to extreme value estimation than theoretically derived models. Attempts to validate trajectorial player motion models are rare. Notably, Caetano et al. [2] performed a validation of their space control model, and thus indirectly also the underlying motion model, by checking how many future positions of players fall within their associated controlled area for a number of time horizons.

3 Player motion model and validation procedure

3.1 Objectives & Requirements

The essential requirement for a player motion model with regards to our validation routine is that it defines a non-zero, finite area that corresponds to the set of positions that the player is able to reach according to the model.

The validation procedure can be represented as a function rating a suitable player motion model on how well it fits some real positional data. Usually, positional data in soccer consists of individual frames, with the position of each player defined for each frame. The frames are normally separated by a constant amount of time (seconds per frame). In order to abstract our validation procedure from the underlying positional data, we introduce the concept of a *trail*. A trail represents a slice of a player’s trajectory over some duration Δt . Formally, a trail is defined as the quadruple:

$$(\vec{x}_0, \vec{v}_0, \vec{x}_t, \Delta t) \tag{1}$$

- \vec{x}_0 : (2D) position of a given player at some arbitrary time t_0
- \vec{v}_0 : (2D) velocity of the player at time t_0
- \vec{x}_t : (2D) position of the player at time $t = t_0 + \Delta t$
- Δt : time horizon (predefined)

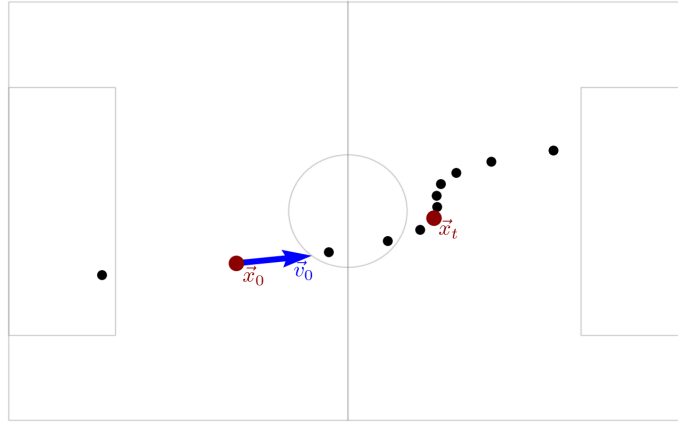


Fig. 1. A trail visualized. The black dots represent the positions of a player for a number of consecutive frames. In this case, the time horizon Δt is four times the duration of a frame. The player’s current velocity \vec{v}_0 can be calculated using numerical differentiation if it is not present in the data.

Figure 1 visualizes how a trail can be extracted from the positional data of a player.

The validation function takes a motion model m and a set of trails T as parameters and returns a numerical validation score. Formally, the function has the signature: $(m, T) \rightarrow \mathbb{R}$.

Since every reached position is trivially contained in a large enough area, the validation function should take not only correctness but also precision of the model into account. The correctness of a motion model measures its ability to make true predictions, i.e. to predict reachable areas that contain the true target position \vec{x}_t . Precision refers to how well narrowed-down the predicted areas of a model are. There is a trade-off relationship between correctness and precision.

One non-functional requirement worth mentioning is performance. Since a player motion model usually has to be invoked and evaluated separately for every trail, the computational complexity is generally $\theta(n)$ with n being the number of trails, unless some algorithmic optimisation or approximation is applied. Given that a high number of trails is desirable in order to get a representative sample of player displacements including a sufficient number of extrema, the validation of a trail should be as efficient as possible.

3.2 Conceptual approach

Measuring Correctness. For the conceptual understanding of the correctness measure, it helps to consider only a single trail (1) of a player. Using \vec{x}_0 , \vec{v}_0 and Δt , a motion model makes a prediction for the reachable area. The reachable area predicted by a motion model m is defined by the set of reachable positions R_m . If \vec{x}_t is contained in R_m , the model m has made a correct prediction. Following

this logic, a motion model achieves the highest possible correctness if and only if for every trail, the model predicts a reachable area in which \vec{x}_t is contained. The ratio between the number of correct predictions $n_{correct}$ and the number of total predictions n_{total} of a model m for a sample of trails T will be called *hit_ratio*.

$$hit_ratio(m, T) = \frac{n_{correct}}{n_{total}} = \frac{n_{correct}}{length(T)} \quad (2)$$

The total number of predictions is always equal to the length of the sample of trails as the motion model makes exactly one prediction for each trail. We can use the *hit_ratio* of a model as an indicator for its correctness. A high *hit_ratio* corresponds to a high correctness and vice-versa.

Measuring Precision. In the context of this paper, the precision of a motion model represents how much it narrows down the reachable area of a player. Smaller reachable areas imply a higher precision of the model and are generally preferable, given an equal *hit_ratio*.

To determine the precision of a model across multiple evaluated trails, we use the inverse of the mean surface area of all correctly predicted reachable areas. Incorrect predictions, where the target position \vec{x}_t is not contained in the predicted reachable area are excluded from this average, since the precision of a model would otherwise increase inappropriately for very narrow, incorrect predictions. The precision of model m across a sample of trails T is given by:

$$precision(m, T) = \frac{1}{\frac{1}{n_{correct}} \cdot \sum areas_{correct}} = \frac{n_{correct}}{\sum areas_{correct}} \quad (3)$$

where $\sum areas_{correct}$ is the sum of all correctly predicted reachable areas.

Defining an overall Validation Score. Since we aim for a single numerical value as a score for player motion models, correctness and precision have to be balanced in some way. Due to the fact that some measurement-related extreme outliers can usually be expected in positional data from football games, a model with a *hit_ratio* of 100% might not necessarily be desirable. Therefore, we introduce a minimum level of correctness *hit_ratio_{min}*, which represents a minimal required ratio between correct and total predictions of a model. We propose that if a motion model m satisfies the condition

$$hit_ratio(m, T) \geq hit_ratio_{min}$$

for a trail sample T , the exact *hit_ratio(m, T)* should be indifferent for the overall validation score of m . This way, extreme outliers in the positional data caused by measurement-related errors have no influence on the validation score, as long as *hit_ratio_{min}* is chosen adequately.

Consequently, for a motion model m that exceeds hit_ratio_{min} , the validation score is only determined by the precision of the model (3). We define the *score* of a motion model m with the sample of trails T as:

$$score(m, T) = \begin{cases} 0 & \text{if } hit_ratio(m, T) < hit_ratio_{min} \\ precision(m, T) & \text{else} \end{cases} \quad (4)$$

The *score* measures how well a motion models fits a sample of positional data. hit_ratio_{min} can be considered a free parameter of this validation procedure. Although the primary rationale behind introducing this parameter is to prevent outliers from influencing the validation score, it can also be tuned to some extent for adjusting the behaviour of the validation procedure. Using a relatively high hit_ratio_{min} favors models with a high correctness, whereas a lower one favors models with a high precision. It is important to note that due to this trade-off relationship, the "right" balance between correctness and precision depends on the specific use-cases of the model. Thus, the exact value for hit_ratio_{min} has to be chosen based on both the expected quality of the positional data in terms of outliers and the desired properties of the model with regard to correctness vs. precision.

In any case, hit_ratio_{min} has to be chosen such that

$$hit_ratio_{min} \leq 1 - \frac{n_{outlier}}{length(T)} \quad (5)$$

where $n_{outlier}$ is the (expected) number of outliers in the sample T . If (5) were violated, the validation score would be influenced by the outliers in the sample, which invalidates the primary purpose of using hit_ratio_{min} as a threshold for the validation score. The identification of outliers or estimation of their prevalence within a sample is naturally non-trivial, since the appropriate statistical method for achieving an acceptable estimate depends on the specific properties of the sample. However, this paper will not go into further detail here.

3.3 Implementation

For the actual computation of the validation score of a given motion model with reasonable efficiency, we need to overcome a few challenges. This section outlines how these challenges can be managed when implementing the validation procedure.

Implementing a motion model interface. First of all, a player motion model m predicts a set of reachable positions R_m based on the input parameters. This set R_m , however, is hardly useful for implementing our validation procedure because it is an infinite set. In practice, R_m typically corresponds to a bounded, simply-connected shape, so R_m is also defined through its boundary B_m . We

can use a simple polygon to approximate B_m and thus R_m . With an increasing number of polygon vertices, this approximation can become arbitrarily accurate.

Using a polygon to represent the reachable area predicted by a model does not only allow a straightforward computation of its validation score but is also practical for defining a common interface for motion models. Therefore, we implement the abstract concept of a motion model as an interface with the parameters $(\vec{x}_0, \vec{v}_0, \Delta t)$ and an array of vertices representing a polygon as return type.

Implementing the validation function. Translating the conceptual approach for calculating the validation score is fairly straightforward. Both the *hit_ratio* (2) and the *precision* (3) have to be calculated.

It has to be determined for each trail in T whether \vec{x}_t is inside the polygon returned by the motion model for the parameters $(\vec{x}_0, \vec{v}_0, \Delta t)$ of the current trail. For all trails where this is the case, it is also necessary to calculate the surface area of that polygon in order to later compute the precision (3).

However, looping over the entire sample of trails is not always necessary. Following the definition of the validation score (4), the score is always equal to 0 if the condition $hit_ratio(m, T) < hit_ratio_{min}$ is true. We can reformulate the hit ratio (2) as:

$$hit_ratio(m, T) = \frac{n_{correct}}{length(T)} = \frac{length(T) - n_{incorrect}}{length(T)}$$

with $n_{incorrect}$ being the number of incorrect predictions made by m for T . We can put this reformulated definition of $hit_ratio(m, T)$ into our condition from the definition of the validation score (4):

$$\frac{length(T) - n_{incorrect}}{length(T)} < hit_ratio_{min}$$

Solved for $n_{incorrect}$:

$$n_{incorrect} > (1 - hit_ratio_{min}) \cdot length(T) \quad (6)$$

Therefore, $score(m, T)$ is equal to 0 if and only if this condition (6) is satisfied. As $length(T)$ and hit_ratio_{min} are known beforehand, the execution of the validation procedure can be stopped prematurely if the number of incorrect predictions exceeds the threshold $(1 - hit_ratio_{min}) \cdot length(T)$. This small optimisation to the original procedure drastically accelerates the validation of motion models with a low correctness, i.e. a low *hit_ratio*.

Algorithm 1 describes the implementation for the computation of the validation score including the mentioned optimisation.

4 Experiment & Evaluation of results

To illustrate validation and parameter optimisation using the procedure defined in section 3, we evaluate four different models of motion, assuming (a) motion

inputs : motion model m , sample of trails T , free parameter hit_ratio_{min}
output: The validation score as defined in (4)

```

 $n_{incorrect} \leftarrow 0$ ;
Let  $A$  be an empty array;
Let  $length(T)$  be the number of trails in  $T$ ;
foreach  $trail \in T$  do
  if  $n_{incorrect} > (1 - hit\_ratio_{min})length(T)$  then
    | return 0
  end
  Let  $(\vec{x}_0, \vec{v}_0, \vec{x}_t, \Delta t)$  represent the current  $trail$ ;
  Let  $P$  be the reachable area polygon predicted by  $m$  based on  $(\vec{x}_0, \vec{v}_0, \Delta t)$ ;
  if  $\vec{x}_t$  is contained in  $P$  then
    | Let  $a$  be the surface area of  $P$ ;
    | Append  $a$  to  $A$ ;
  else
    |  $n_{incorrect} \leftarrow n_{incorrect} + 1$ ;
  end
end
 $n_{correct} \leftarrow length(T) - n_{incorrect}$ ;
return  $n_{correct}/sum(A)$ ;

```

Algorithm 1: Validation routine, optimized

with constant speed, (b) motion with constant acceleration, (c) motion with constant acceleration until a speed limit is reached, and (d) motion along two segments with constant speed.

4.1 Data set

For the evaluation, we use the public sample data set provided by Metrica Sports which consists of three anonymised games of football [9]. The positional data has been collected using a video-based system and is provided at a frequency of 25 Hz. Since the data contains no velocity, we compute a player’s velocity \vec{v}_i for each frame i as $\vec{v}_i = \frac{\vec{x}_{i+1} - \vec{x}_{i-1}}{2 \cdot 0.04s}$.

For performing our validation routine we convert the positional data into a list of trails. As outlined in (1), each trail is defined as a quadruple $(\vec{x}_0, \vec{v}_0, \vec{x}_t, \Delta t)$. For this experiment, we use a constant time horizon of $\Delta t = 1s$ for all trails. After visual inspection of the data, the minimal required hit ratio is set to $hit_ratio_{min} = 99.975\%$. We evaluate the models on a random sample of $5 \cdot 10^5$ trails across all three games and all participating players.

4.2 Preparation of motion models

We phrase the motion models such that they define a set of reachable positions R . This set has to form a non-zero, finite area. To approximate the reachable area as a polygon, R also has to be simply connected and have a computationally

approximable boundary B . The sets R and B depend on the player's initial position \vec{x}_0 and velocity \vec{v}_0 , the selected time horizon Δt , and the parameters of the respective model. For brevity of notation, we set $t_0 = 0$ such that $\Delta t = t$. The reachable areas defined by the models (a) - (d) are exemplarily visualized in Figure 2.

(a) Constant speed. Given motion with some constant speed $v \in [0, v_{max}]$ in direction ϕ from a starting location \vec{x}_0 , the set of reachable target locations R after time t forms a disk with the center \vec{x}_0 and radius $v_{max}t$.

$$R = \{\vec{x} \mid \exists \phi \in [0, 2\pi], \exists v \in [0, v_{max}], \vec{x} = \vec{x}_0 + v \begin{bmatrix} \cos \phi \\ \sin \phi \end{bmatrix} t\}$$

$$B = \{\vec{x} \mid \exists \phi \in [0, 2\pi], \vec{x} = \vec{x}_0 + v_{max} \begin{bmatrix} \cos \phi \\ \sin \phi \end{bmatrix} t\} \quad (7)$$

(b) Constant acceleration. Given motion with some constant acceleration $a \in [0, a_{max}]$ in direction ϕ from a starting location \vec{x}_0 with starting velocity \vec{v}_0 , the set of reachable positions R after time t forms a disk with the center $\vec{x}_0 + \vec{v}_0 t$ and radius $\frac{1}{2}a_{max}t^2$.

$$R = \{\vec{x} \mid \exists \phi \in [0, 2\pi], \exists a \in [0, a_{max}], \vec{x} = \vec{x}_0 + \frac{1}{2}a \begin{bmatrix} \cos \phi \\ \sin \phi \end{bmatrix} t^2 + \vec{v}_0 t\}$$

$$B = \{\vec{x} \mid \exists \phi \in [0, 2\pi], \vec{x} = \vec{x}_0 + \frac{1}{2}a_{max} \begin{bmatrix} \cos \phi \\ \sin \phi \end{bmatrix} t^2 + \vec{v}_0 t\} \quad (8)$$

(c) Constant acceleration with speed limit. We want to restrict model (b) such that the player's speed never exceeds a fixed maximum v_{max} . For that purpose, the simulated trajectory is divided into two segments: A first segment of motion with constant acceleration a until v_{max} is reached, and a second segment of motion with constant speed v_{max} . A non-negative solution for the time t_{acc} until v_{max} is reached is guaranteed to exist if the player's initial speed $|\vec{v}_0|$ does not exceed v_{max} . For that reason, \vec{v}_0 is clipped to $\vec{v}_0^* = \vec{v}_0 \frac{\min(|\vec{v}_0|, v_{max})}{|\vec{v}_0|}$. With $t_{acc}^* = \min(t_{acc}, t)$, the reachable area R corresponds to the following simply connected shape, which can be non-circular as t_{acc}^* depends on ϕ .

$$R = \{\vec{x} \mid \exists \phi \in [0, 2\pi], \exists a \in [0, a_{max}], \vec{x} = \vec{x}_0 + \vec{v}_0^* t + at_{acc}^* (t - \frac{1}{2}t_{acc}^*) \begin{bmatrix} \cos \phi \\ \sin \phi \end{bmatrix}\}$$

$$B = \{\vec{x} \mid \exists \phi \in [0, 2\pi], \vec{x} = \vec{x}_0 + \vec{v}_0^* t + a_{max}t_{acc}^* (t - \frac{1}{2}t_{acc}^*) \begin{bmatrix} \cos \phi \\ \sin \phi \end{bmatrix}\} \quad (9)$$

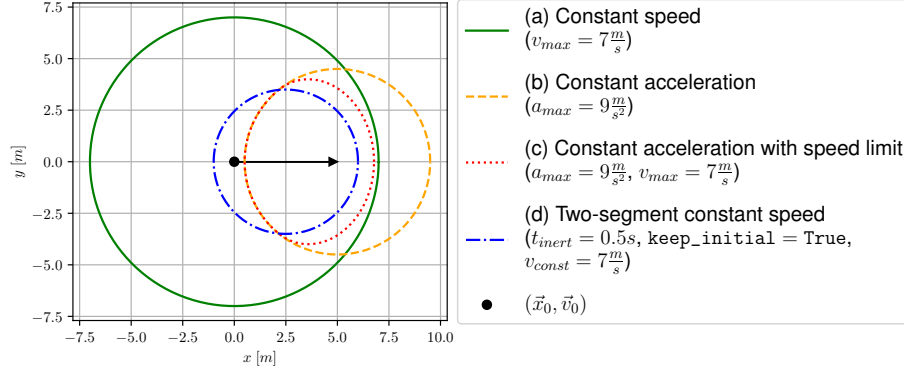


Fig. 2. Exemplary boundaries B of the reachable area defined by different motion models when the player starts at $\vec{x}_0 = \vec{0}$ with velocity $\vec{v}_0 = \begin{bmatrix} 5 \frac{m}{s} \\ 0 \end{bmatrix}$. The time horizon is $\Delta t = 1s$.

(d) Two-segment constant speed. We propose the following approximate model to respect the current kinematic state of a player: The player's motion is divided into two segments of constant-speed motion. First, with some speed v_{inert} for a predetermined amount of time t_{inert} in the direction of \vec{v}_0 and, subsequently, with speed v_{final} in some arbitrary direction ϕ . Using $t_{inert}^* = \min(t_{inert}, t)$, the set of reachable positions R forms a disk with the radius $v_{final}(t - t_{inert}^*)$ if $t_{inert} > t$ and is otherwise reduced to the point $\vec{x}_0 + v_{inert} \frac{\vec{v}_0}{|\vec{v}_0|} t$.

$$R = \{ \vec{x} \mid \exists \phi \in [0, 2\pi], \exists v \in [0, v_{final}], \vec{x} = \vec{x}_0 + v_{inert} \frac{\vec{v}_0}{|\vec{v}_0|} t_{inert}^* + v_{final} \begin{bmatrix} \cos \phi \\ \sin \phi \end{bmatrix} (t - t_{inert}^*) \}$$

$$B = \{ \vec{x} \mid \exists \phi \in [0, 2\pi], \vec{x} = \vec{x}_0 + v_{inert} \frac{\vec{v}_0}{|\vec{v}_0|} t_{inert}^* + v_{final} \begin{bmatrix} \cos \phi \\ \sin \phi \end{bmatrix} (t - t_{inert}^*) \} \quad (10)$$

The values of v_{inert} and v_{final} can in principle be set in various ways. In our parameterisation, we determine the value of v_{inert} according to a boolean parameter `keep_initial`, such that the speed of the player is either preserved or set to a fixed value v_{const} .

$$v_{inert} = \begin{cases} |\vec{v}_0| & \text{if } \text{keep_initial} \\ v_{const} & \text{else} \end{cases}$$

The value of v_{final} is either computed based on two parameters a_{max} and v_{max} which corresponds to pretending that the player has accelerated during the first segment, or set to the fixed value v_{const} .

$$v_{final} = \begin{cases} \min(v_{inert} + a_{max} t_{inert}, v_{max}) & \text{if } a_{max} \text{ and } v_{max} \text{ are set} \\ v_{const} & \text{else} \end{cases}$$

Computation of reachable area It follows from equations (7) - (10) that for each presented model, the reachable area can be approximated as a polygon by computing the vertices of that polygon as points from the boundary B along a discrete set of angles ϕ . For our evaluation, we choose to evaluate 200 evenly spaced angles.

4.3 Optimisation of model parameters

Based on the validation procedure outlined in section 3, we aim to find the motion model m out of the set of considered models M which maximises the *score* (4) for a given sample of trails T . We define the optimal model as:

$$\arg \max_{m \in M} (\text{score}(m, T)) \quad (11)$$

However, for each of the four player motion models defined in section 4.2, there are free parameters which specify its behaviours. For this reason, in order to determine the best model, the optimal combination of parameters has to be found for each model. We thus extend (11) by introducing the tuple P_m that represents the values for the free parameters of a model $m(P_m)$:

$$\arg \max_{m \in M} (\max_{P_m} (\text{score}(m(P_m), T))) \quad (12)$$

The parameters of the models (a) - (d) are, as outlined in section 4.2:

- (a) Constant speed: $P_{(a)} = (v_{max})$
- (b) Constant acceleration: $P_{(b)} = (a_{max})$
- (c) Constant acceleration with speed limit: $P_{(c)} = (a_{max}, v_{max})$
- (d) Two-segment constant speed: $P_{(d)} = (t_{inert}, \text{keep_initial}, v_{const}, a_{max}, v_{max})$

The main challenge here is to determine the optimal P_m for a model with regard to its *score* (4). For this experiment, the best parameter configuration P_m is determined using Bayesian optimisation. However, Bayesian optimization generally does not work for discrete parameter values such as the boolean variable `keep_initial` in model (d). There are proposals for enabling discrete variables, most notably by Luong et al. [7], but this works only if all values in P_m are discrete, which does not apply to our models. Luckily, our models have very few discrete parameters, so we perform one Bayesian optimisation for each combination of discrete parameter values and use the best score across those results.

4.4 Evaluation of results

The performance of the optimised models (a) - (d) and their parameter values are shown in Figure 3.

The constant-speed model (a) unsurprisingly shows a weaker performance ($\text{score}^{-1} = 218m^2$) than the more sophisticated models (c) and (d), since it does not factor in the initial kinematic state of the player. If the player moves at

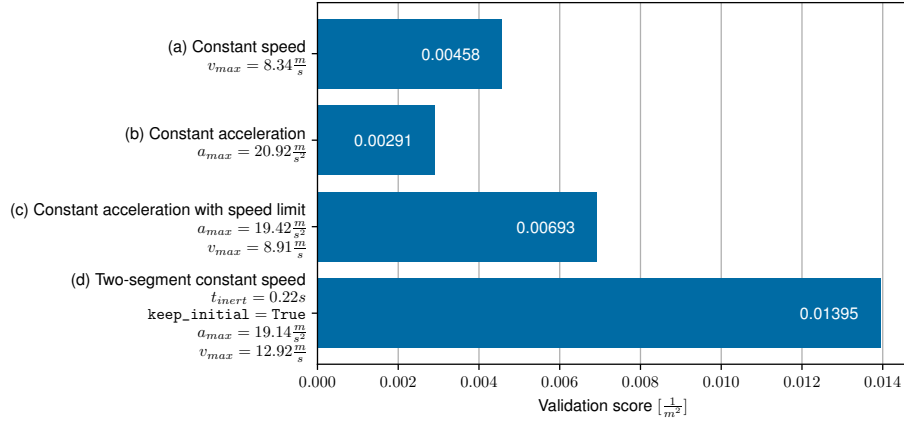


Fig. 3. Comparison of the performance of the four models (a) - (d) with their optimised parameter values.

a high speed, the model unrealistically assumes that the player can instantly run at maximal speed in the other direction. Thus, for high speeds, the model overestimates the amount of reachable space in directions which sufficiently deviate from the direction which the player is initially moving towards.

The naive constant acceleration model (b) ($score^{-1} = 344m^2$) performs even worse than model (a), likely because it makes the unrealistic assumption that the possible magnitude of acceleration is independent of the magnitude and direction of a player’s current velocity. This implies in particular that for high speeds, the amount of reachable space in the direction that a player is moving towards will be heavily overestimated since the model assumes that the player’s speed can increase unboundedly.

The model assuming constant acceleration with a speed limit (c) ($score^{-1} = 144m^2$) outperforms models (a) and (b). However, the optimised value of the maximally possible acceleration of a player a_{max} is physically unrealistic. A value of $a_{max} = 19.42 \frac{m}{s^2}$ assumes that a player can accelerate from zero to the top speed $v_{max} = 8.91 \frac{m}{s} (= 32.08 \frac{km}{h})$ within about half a second, which is implausibly fast. Therefore, the model still overestimates the reachable area. One approach to improve the constant acceleration model could be to view the maximally possible acceleration as being dependent on the direction and magnitude of the current velocity.

The two-segment constant speed model (d) ($score^{-1} = 71.7m^2$) is able to account for all reachable positions by predicting only about half the area of model (c). It successfully narrows down the area that a player can reach within one second to a circle with an average radius of 4.8 meters which is highly accurate. Model (d) not only achieves the best score in our evaluation, but is also mathematically simpler than model (c). For that reason, it is also computationally more efficient across the various tasks that motion models are used for, like

the computation of reachable areas or the shortest time to arrive at a specific location. A drawback of model (d) is that in its presented form, it only makes meaningful predictions for time horizons $\Delta t > t_{inert}$ (where the optimised value is $t_{inert} = 0.22s$). Below this duration, the model is not applicable unless it is appropriately extended.

In summary, our newly presented model (d) is both highly accurate and computationally efficient. The models (a) - (c) have obvious weaknesses and are not adequate to accurately identify reachable locations.

5 Conclusion

We presented a novel approach to the validation and optimisation of models of trajectorial player motion in football and similar sports. We also presented an empirical comparison of the accuracy of various such models. In line with our expectations, more sophisticated and accurate assumptions made by a motion model generally tend to be reflected in a better predictive performance. Yet, by far the best-performing model is our proposed approximate model which assumes motion along two segments with constant speed. Using this model allows researchers to compute complex performance indicators more efficiently and accurately over large data sets. In contrast, player motion should not be assumed to take place with constant speed or constant acceleration with unlimited speed. These assumptions are inappropriate to accurately distinguish reachable from unreachable locations.

The validation and optimisation approach described in this paper can be applied to data with arbitrary distributions of measurement error. However, this is also a disadvantage, since the threshold for the amount of outliers that are attributed to measurement error has to be determined manually. This threshold also has to be set for each distinguished population, depending on the frequency of extrema and the distribution of measurement error in the population. If for example one wants to evaluate motion models for each player individually, it would be misleading to assign the same threshold to each player because players who sprint regularly during the game produce more positional extrema (and thus outliers) than goalkeepers, for example, who are rarely forced to run with maximal effort. A solution to the problem of having to specify a threshold is to perform validation and optimisation with different thresholds and analyse how this choice affects the result. Also, the approach presented here can be extended to automatically determine an optimal threshold for known error distributions.

In the future, we plan to search for motion models that further exceed the presented ones in accuracy and computational efficiency. A key towards this goal is to estimate motion models from positional data. Many problems addressed in this paper are mirrored in empirical model fitting, for example the need to exclude outliers and the lack of generalisability across populations [1]. In the context of validation, empirical models can serve as a highly informative benchmark to reveal how well theoretical models are able to approximate actual human motion.

References

1. Brefeld, U., Lasek, J., Mair, S.: Probabilistic movement models and zones of control. *Machine Learning* **108**(1), 127–147 (2019). <https://doi.org/10.1007/s10994-018-5725-1>
2. Caetano, F.G., Barbon Junior, S., Torres, R., Cunha, S., Ruffino, P., Martins, L., Moura, F.: Football player dominant region determined by a novel model based on instantaneous kinematics variables. *Scientific Reports* **11**(1), 1–10 (2021). <https://doi.org/10.1038/s41598-021-97537-4>
3. Efthimiou, C.J.: The voronoi diagram in soccer: a theoretical study to measure dominance space. *arXiv preprint arXiv:2107.05714* (2021)
4. Fernandez, J., Bornn, L.: Wide open spaces: A statistical technique for measuring space creation in professional soccer. In: *12th MIT Sloan Sports Analytics Conference* (2018)
5. Fonseca, S., Milho, J., Travassos, B., Araújo, D.: Spatial dynamics of team sports exposed by voronoi diagrams. *Human movement science* **31**(6), 1652–1659 (2012). <https://doi.org/10.1016/j.humov.2012.04.006>
6. Fujimura, A., Sugihara, K.: Geometric analysis and quantitative evaluation of sport teamwork. *Systems and Computers in Japan* **36**, 49–58 (06 2005). <https://doi.org/10.1002/scj.20254>
7. Luong, P., Gupta, S., Nguyen, D., Rana, S., Venkatesh, S.: Bayesian optimization with discrete variables. In: Liu, J., Bailey, J. (eds.) *AI 2019: Advances in Artificial Intelligence*. pp. 473–484. *Lecture Notes in Computer Science*, Springer International Publishing (2019). https://doi.org/10.1007/978-3-030-35288-2_38
8. Memmert, D., Raabe, D.: *Data analytics in football: Positional data collection, modelling and analysis*. Routledge (2018). <https://doi.org/10.4324/9781351210164>
9. *Metrica sports sample data*. <https://github.com/metrica-sports/sample-data/commit/e706dd506b360d69d9d123d5b8026e7294b13996>. GitHub repository (2021)
10. Pino-Ortega, J., Oliva-Lozano, J., Gantois, P., Nakamura, F., Rico-González, M.: Comparison of the validity and reliability of local positioning systems against other tracking technologies in team sport: A systematic review. *Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology* pp. 1–10 (2021). <https://doi.org/10.1177/1754337120988236>
11. Rein, R., Raabe, D., Memmert, D.: “which pass is better?” novel approaches to assess passing effectiveness in elite soccer. *Human movement science* **55**, 172–181 (2017). <https://doi.org/10.1016/j.humov.2017.07.010>
12. Rico-González, M., Pino-Ortega, J., Nakamura, F., Moura, F., Los Arcos, A.: Identification, computational examination, critical assessment and future considerations of distance variables to assess collective tactical behaviour in team invasion sports by positional data: A systematic review. *International Journal of Environmental Research and Public Health* **17**(6), 1952 (2020). <https://doi.org/10.3390/ijerph17061952>
13. Spearman, W., Basye, A., Dick, G., Hotovy, R., Pop, P.: Physics-based modeling of pass probabilities in soccer. In: *11th MIT Sloan Sports Analytics Conference* (2017)
14. Taki, T., Hasegawa, J.: Visualization of dominant region in team games and its application to teamwork analysis. In: *Proceedings computer graphics international 2000*. pp. 227–235. IEEE (2000)