Combining Foot Placement Prediction with Obstacle Detection to Detect Tripping

Urban Bobek[1], **Elmar Rueckert**[3], **Marko Jamšek**[1,2], **Saša Barišić**[4], **Jan Babič**[1]

[1]*Laboratory for Neuromechanics and Biorobotics, Department of Automation, Biocybernetics and Robotics*
[2]*Jozef Stefan International Postgraduate School, Jamova cesta 39, 1000 Ljubljana, Slovenia*
[3]*Institute for Robotics and Cognitive Systems, University of Luebeck, Ratzeburger Allee 160, 23562 Luebeck, Germany*
[4]*Bjelovar University of Applied Sciences,Trg Eugena Kvaternika 4, 43000 Bjelovar, Croatia*
E-pošta: bobek.urban@gmail.com

Abstract

Tripping is a major cause of fall related injuries, especialy among the elderly population. Some reaserch has been done on the mechanics of tripping and strategies to gain balance afterwards. But what if you could detect a potential trip in advance and possibly prevent it? We propose a system that involves detecting obstacles infront of the user and a method to predict whether they will hit it.

1 Introduction

Falls and fall related injuries are very common among people, especially in the elderly population [1]. This is becoming an ever increasing problem, because the amount of elderly people is getting larger with every year [2]. The mechanism of tripping have been explored in the past [3, 4], but here we are interested in preventing tripping from even occuring. For this we would need a way to predict the future placement of the subjects feet and a prediction system of obstacles.

Some research has already been done on predicting foot placement. In [5] they predicted foot placement in the mediolater direction based on the position and velocity of the subjects center of mass. Similarly, [6] predicted recovery foot placement when pertubating a subject in the sagital plane. A method, where the next contact location of the subjects foot gets predicted needed to be made. The prediction method we implemented takes advantage of the properties of probabilistic motion primitives (ProMPs) [7], that allow operations from probability theory. ProMPs have been used in many different robotic applications [8, 9] and also postural studies [10] to encode and predict movements. We combined this method with an obstacle detection system mounted on the subject. In this article we present a system that can both detect potential obstacles infront of the wearer and also predict if there is a chance of tripping.

2 Equipment and setup

The entire system consist of two major parts, the foot placement prediction and the obstacle detection. The latter was achieved with the combination of a depth camera (RealSense Depth Camera D435, Intel, Santa Clara, USA) and 3D motion capture system (3D Investigator, NDI, Waterloo, Canada). For the foot placement prediction, an Xsens motion capture suite (MTw Awinda, Xsens, Enschede, Netherlands) was used to capture subjects walking gait. Using this data we could then online predict where the subject is going to step using methods described in Section 3. A block diagram of the whole setup is depicted in Figure 1). The main computer receives the foot placement predictions from the Xsens computer, the data from the RealSense camera and the 3D motion capture system. It then combines this data to calculate if the subject is in danger of tripping.

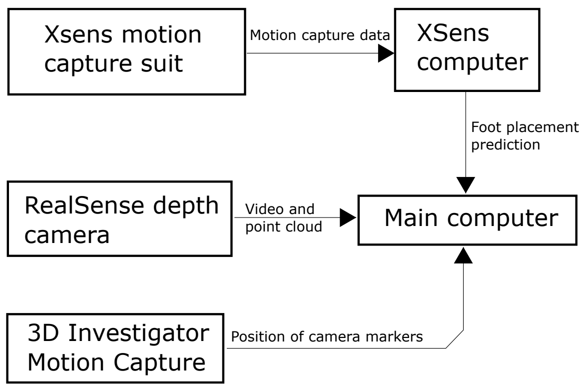

Figure 1: Block diagram of the setup used in the experiment.

2.1 Motion Capture

In the setup we used two diffent motion capture (MC) systems. The first one is the Awinda human motion capture system (MTw Awinda, Xsens, Enschede, Netherlands) that uses IMUs placed on certain parts of the body. Using a complex kinematic model it then returns the whole body kinematics of the wearer. Using the proprietary software we could stream the data at 60Hz to a local Simulink (Mathworks, Nat- ick, MA, United States) scheme. For this specific experiment, we were only interested in the position and velocity of the left and right ankle. The Simulink scheme then calculated the most probable foot placement of the subject and sent it foward to the main computer.

The other MC system used was the 3D Investigator Motion Capture system (3D Investigator, NDI, Waterloo, Canada) that can track special markers in 3D space with an accuracy of 0.4 mm. We used the RealSense depth camera to detect obstacles infront of the subject. Point cloud data captured by the camera is relative to its own coordinate frame, so we needed a way to know the cameras position and orientation in the global coordinate frame. The camera was attached to the subjects front and on it we placed 3 markers. Using the position of these markers we could then transform the depth camera data to the global frame. Both the depth camera and the motion tracking system streamed their data to the main computer, where all of the calculation necesarry was done.

2.2 Depth Camera for Obstacle Detection

The RealSense depth camera was used for object detection. We used both the RGB video and the point cloud output of the camera to help represent the results on the main computer. The camera was slanted at an angle (approx. 40 °), so that the floor infront of the subject was visible.

3 Methods

In this section we will describe the general theory of how the foot placement was predicted using probabilistic motion primivitives (ProMP) and the approach we used to detect obstacles.

3.1 Probabilstic Model of Trajectories

Before any predictions could be made, we first needed a model of how the subjects gait looks like. Several repetitions of the subjects gait had to be recorded in order to obtain this model. We had the subject wear the Xsens MC suite and walk on a treadmill at a constant velocity (0.8 m/s). After 20 steps we could than process the data and learn the model.

3.1.1 Data aqusition

The human gait is a periodic movement that consist of two main phases, the stance and the swing phase [11]. We focused on the latter, because only in this one the leg moves in space. This means that we had to seperate the recorded gait cycle into this two phases. Because the foot only moves during the swing phase we observed when the velocity of the foot in the sagital plane became positive. This moment indicated the end of the stance and the begining of the swing phase. The exact threshold was set empiricaly so that the small variations in velocity and the noise did not have an effect. During the learning of the model all three dimension and the time were recorded while the subject walked on a treadmill. After we processed the recorded data to extract the trajectories of the swing phases of all the gait cycles. Using these trajectories we could then train the probabilistic model.

3.1.2 Encoding recorded trajectories

To keep the amount of parameters needed to represent trajectories as low as possible, ProMPs uses a basis function representation approach. To better understand the formulation, let us take a look at a simple example where we describe a point in time a_t using this method. Let $\phi_t \in \mathbf{R}^{1\times J}$ denote a basis function vector containing values of *J* basis functions at time *t*. Variable $w \in \mathbf{R}^{J\times 1}$ represents a *J*-dimensional feature vector that encodes weights for each of the *J* basis functions. With w and ϕ_t defined, a point at time *t* can be approximated as

$$a_t = \phi_t\, w = \begin{bmatrix}\phi_{1,t} & \cdots & \phi_{J,t}\end{bmatrix}\begin{bmatrix}w_1 & \cdots & w_J\end{bmatrix}^T .$$

This concept can be applied to multi-dimensional states by using block diagonal matrices. Let‘s assume that our variable a_t now has *D* dimensions $a_t = \begin{bmatrix}a_{1,t} & \cdots & a_{D,t}\end{bmatrix}^T$. In this case the basis function vector becomes a block diagonal matrix $\Phi_t \in \mathbf{R}^{D\times JD}$ and the weight vector w becomes a concatenation of the weight vectors of each dimension $w \in \mathbf{R}^{JD\times 1}$. Variable a_t is now approximated as

$$a_t = \Phi_t\, w = \begin{bmatrix}\phi_t & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & \phi_t\end{bmatrix}\begin{bmatrix}w_1 & \cdots & w_i & \cdots & w_D\end{bmatrix}^T ,$$

where

$$\phi_t = \begin{bmatrix}\phi_{1,t} & \phi_{2,t} & \cdots & \phi_{J,t}\end{bmatrix}$$

and

$$w_i = \begin{bmatrix}w_{1,i} & w_{2,i} & \cdots & w_{J,i}\end{bmatrix}^T .$$

Using the same idea we can approximate a sequence of *T* states denoted by $\tau = y_{1:T}$ where

$$\tau = \Phi_{1:T} w \qquad (1)$$

with

$$\Phi_{1:T} = \begin{bmatrix}\Phi_1 & \cdots & \Phi_t & \cdots & \Phi_T\end{bmatrix}^T \in \mathbf{R}^{TD\times JD},$$

where the vector w and the matrix Φ_t are the same as before. In Figure 2), you can see 1 dimension of a trajectory and its approximation using the described method. In our application we used Gaussian basis functions which are often used for point to point movements.

To approximate the trajectories in the previously described manner, the weights for each trajectory need to be calculated. For the *i*-th trajectory τ_i the corresponding weight vector w_i can be estimated using a simple least squares estimate. In our application the ordinary least square (OLS) method was used

$$w_i = \left(\Phi_{1:T}^T \Phi_{1:T} + \lambda I\right)^{-1} \Phi_{1:t}^T \tau_i, \qquad (2)$$

where λ represents a regularization parameter used to avoid numerical singularities. Its value should be small, in our case we used $\lambda = 10^{-6}$.

3.1.3 Creating the probabilistic model

When the weight vectors of all trajectories are calculated, we assume their values to be normally distributed, *i.e.*, $p(w) = \mathcal{N}(w|\mu_w, \Sigma_w)$. The mean μ_w and the covariance matrix Σ_w can be estimated with sample mean and sample covariance of the w_i vectors.

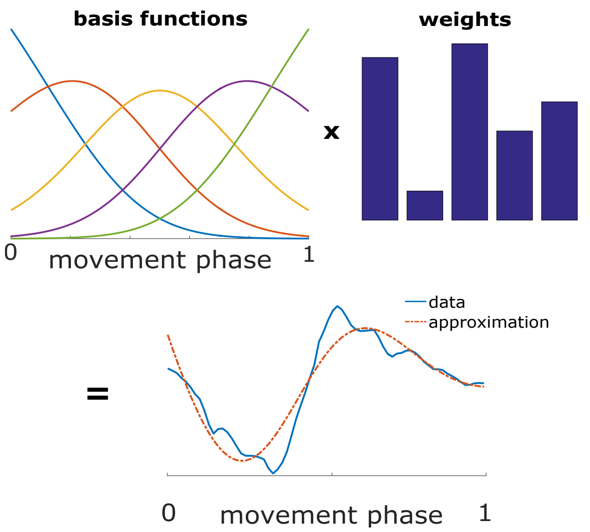

Figure 2: Approximating a sequence of states using 5 basis functions and their coresponding weights.

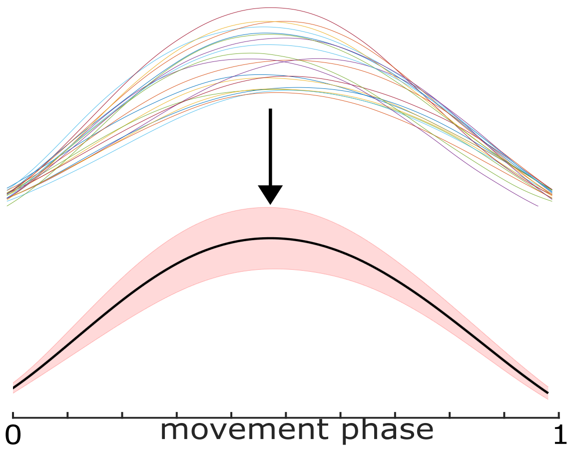

Figure 3: Trajectory distribution model calculated from several different trajectories. This model represents only one degree of freedom.

With the function approximation (1) and the weight vectors w_i defined, we can define a probabilistic model for trajectories as

$$p(\tau|w) = \prod_{t=1}^{T} \mathcal{N}(y_t|\Phi_t w, \Sigma_y) = \mathcal{N}(y_{1:t}|\Phi_{1:T} w, \Sigma_y).$$

This model describes the probability of observing trajectory τ, given the weight vector w that is given as a linear basis function $y_{1:t} = \Phi_{1:T} w + \epsilon_{y,1:T}$. The parameter Σ_y represents independent and identically distributed (*i.i.d.*) Gaussian noise in the trajectories $y_t = \Phi_t w + \epsilon_y$, where $\epsilon_y \sim \mathcal{N}(\epsilon_y|0, \Sigma_y)$.

On the top pane of Figure 3) you can see 1 dimension of several recordings of the leg swing trajectory. The bottom pane shows the model that we calculated from these recordings. The thick line represents the mean of the model and the shaded area the 1-σ standard deviation. The movement phase denotes the normalized time.

3.2 Computing Predictions from Observations

Statistical theory tells us that we can model predictions as computing the conditional probability. First we need to define the probability distribution over the trajectories τ, which can be computed by marginalizing out the weight vector w. In the case of Gaussian distribution the marginal can be computed in closed form as

$$p(\tau) = \int p(\tau|w)p(w)dw$$

$$= \int \mathcal{N}(y_{1:T}|\Phi_{1:T} w, \Sigma_y)\mathcal{N}(w|\mu_w, \Sigma_w)dw$$

$$= \mathcal{N}(y_{1:t}|\Phi_{1:T} w, \Phi_{1:T}\Sigma_w\Phi_{1:T}^T + \Sigma_y). \quad (3)$$

What we get is a multivariate Gaussian distribution, the conditional probability of which we can compute in closed form.

When we receive a previously unseen point a^*, we can predict the most likely path of the foot (parametrized through $\overline{\mu^*}$ and $\overline{\Sigma^*}$) by conditioning the observed state over the weight vectors. Say that we observed a sequence of states y_{t1} to y_{tM} at m=1, 2,..., M-different time points. We declare ν as a concatenation of the observed states y_{tm} and Φ_ν as the concatenation of the basis function matrices for the observed time points.

With the observed trajectories encoded as previously described, we can obtain a conditioned distribution $p(w_\nu|\nu)$ over the weight vectors w as

$$p(w_\nu|\nu) \propto \mathcal{N}(\nu|\Phi_\nu w_\nu, \Sigma_0)p(w)$$

$$:= \mathcal{N}(w_\nu|\mu_{w|\nu}, \Sigma_{w|\nu}).$$

We can compute the mean $\mu_{w|\nu}$ and the covariance matrix $\Sigma_{w|\nu}$ as

$$\mu_{w|\nu} = \mu_w + \Sigma_w \Phi_\nu^T L(\nu - \Phi_\nu \mu_w)$$

and

$$\Sigma_{w|\nu} = \Sigma_w - \Sigma_w \Phi_\nu^T L \Phi_\nu \Sigma_w$$

where

$$L = \left(\Sigma_0 + \Phi_\nu \Sigma_w \Phi_\nu^T\right)^{-1}.$$

With the feature mean $\mu_{w|\nu}$ and covariance matrix $\Sigma_{w|\nu}$ obtained, we can now use this conditional distribution to calculate the distribution over the trajectories $p(\overline{\tau})$ using (3)

$$p(\overline{\tau}) = \mathcal{N}(\tilde{y}_{1:T}|\Phi_{1:T}\mu_{w|\nu}, \Phi_{1:T}\Sigma_{w|\nu}\Phi_{1:T}^T + \Sigma_y),$$

where the predicted sequence of states $\tilde{y}_{1:T}$ is represented by the product $\Phi_{1:T}\mu_{w|\nu}$. In Figure 4) a prediction is calculated through conditioning with one and two observed states.

3.3 Obstacle Detection

For detecting obstacles the Intel RealSense D435 depth camera was used. Alongside RGB video it returns a point

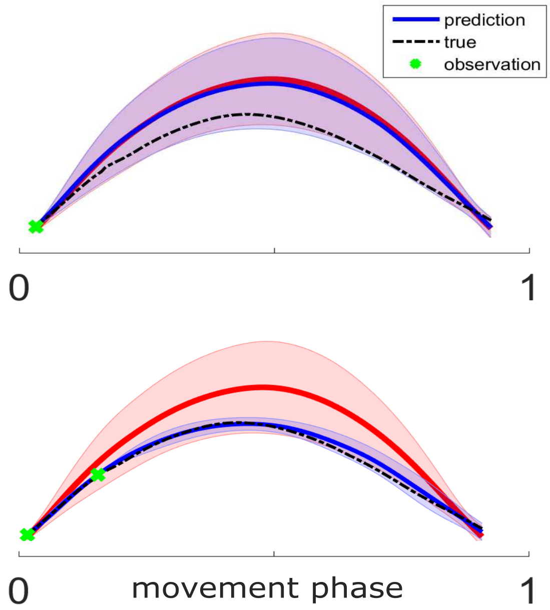

Figure 4: A demonstration of predicting a path through conditioning on observed states.
On the top pane only one point was observed, so the predicted path is very similar to the model mean. The dark blue area represents a $1-\sigma$ deviation from the predicted path. On the bottom pane, two points were observed so the quality of the prediction increases.

cloud - a matrix of distances from the camera to the objects infront. To simplify the object detection we splited the visible 3D space of the camera in small cubes, i.e. voxels. We then used the point cloud to fill in the corresponding voxels. Using the data from the three markers above the camera, this output gets transformed to the world coordinate system. This approach also gives us a simpler and computationaly less demanding visualization of the world. Because the location of the treadmill was known, we knew which voxels represent it. This way, when voxels above the treadmill were observed, we knew there was an obstacle in the way.

This computation is all done on the main computer to which also the Xsens computer streams its foot placement prediction alongside the current position of the subjects feet. This data is already in the world coordinate frame. With each iteration we checked if the predicted foot placement coincides with any of the obstacle voxels and if they did, we identified this as a potential collision.

4 Discussion

The presented system was able to predict the foot placement of both left and right foot of a subject walking on a treadmill. When an obstacle was presented, the system accurately detected it and sent a warning if the predicted foot placement was coinciding with it. The main computer returned a visual feedback where the output of the camera, the current foot positions and the predicted foot placements were visible. A system like this could eventualy be implemented in a assistive exoskeleton, that would help the user avoid tripping by extending or shortening their step. The prediction process does not only return the final position of the foot, but also the whole path. This is why it could be used by itself for example in a gait analysis study, where certain events could be triggered in advance based on the calculated prediction.

The downside of the system is that because the prediction model is based on the previous recordings of the subjects gait, the system will only work accurately for this specific movement. Changing the subject or even the velocity of the treadmill will result in much worse predictions. Implementing a phase estimation like in [9] could help with the changes in velocity.

5 Aknowledgements

This work was supported by the European Union's Horizon 2020 through the SPEXOR project (contract no. 687-662); AnDy project (contract nr. 731540); and by the Slovenian Research Agency (research core funding no.P2-0076).

References

[1] W. R. Berg, H. M. Alessio, E. M. Mills, and T. O. N. G. Chen, "Circumstances and consequences of falls in independent community- dwelling older adults," pp. 261–268, 1997.

[2] U. N. P. F. U. Internationa, , and HelpAge, *Ageing in the Twenty-First Century: A Celebration and A Challenge, 2012.*

[3] A. M. Schillings, J. Duysens, and S. Maartenskliniek, "Muscular Responses and Movement Strategies During Stumbling Over Obstacles," pp. 2093–2102, 2000.

[4] B. M. H. V. Wezel and J. Duysens, "Mechanically induced stumbling during human treadmill walking," vol. 67, 1996.

[5] M. Arvin, M. J. Hoozemans, M. Pijnappels, J. Duysens, S. M. Verschueren, and J. H. van Dieën, "Where to step? Contributions of stance leg muscle spindle afference to planning of mediolateral foot placement for balance control in young and old adults," 2018.

[6] C. F. Lei Zhang, "Predicting foot placement for balance through a simple model with swing leg dynamics, 2018."

[7] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Probabilistic Movement Primitives," *Advances in Neural Information Processing Systems 26*, pp. 2616–2624, 2013. [Online]. Available: http://papers.nips.cc/paper/5177-probabilistic-movement-primitives.pdf

[8] ——, "Using probabilistic movement primitives in robotics," pp. 529–551, 2018.

[9] G. Maeda, M. Ewerton, G. Neumann, R. Lioutikov, and J. Peters, "Phase estimation for fast action recognition and trajectory generation in human–robot collaboration," pp. 1579–1594, 2017.

[10] E. Rueckert, J. Camernik, J. Peters, and J. Babic, "Probabilistic Movement Models Show that Postural Control Precedes and Predicts Volitional Motor Control," *Scientific Reports*, vol. 6, no. February, pp. 1–12, 2016.

[11] A. Alvarez-alvarez, "Linguistic Description of the Human Gait Quality," vol. 26, no. 1, pp. 13–23, 2013.