A. Data Acquisition

Knee kinematic data was utilized for CMAC training. This data was captured through motion capture techniques, using 12 cameras Qualisys Oqus MRI, passive markers and software package Qualisys QTM 3.2. Data were converted to the appropriate format supported by Octave 3.8.1 language (MATLAB format). Data were captured from a subject in Human Performance Laboratory at Faculty UnB Ceilândia. A healthy male subject was selected. He repeated a walk of approximately 5 seconds for 5 times. This process generated spatial variables, regarding the position of the markers. The markers were distributed along 34 positions at inferior limbs. As only knee flexion and extension were necessary for this work, only the markers of tibias, knees and trochanters were utilized. It was possible with these data, to calculate knees angles, angular velocities and angular accelerations. The initial and final data of each sample had to be eliminated, because they constitute the comfortable gait cycle beginning and end.

The data acquisition was approved by the UnB Health Faculty Ethics Committee, protocol N11911/12.

B. Kinematic Calculations

Kinematic calculations were made using the Octave programming language which is compatible with MATLAB. This choice was made, because Octave is open source and because the QTM export data to MATLAB format.

Angles were obtained using the equation (1) from [10].

$$\theta = acos\left(\frac{u.v}{\| \; u \| \; \| \; v \|}\right) \tag{1}$$

The *u* variable is equivalent to the vector at the trochanter point; *v* is the vector at the tibia point. The knee point must be the origin of *u and v*, so these must be translated to the new origin [11].

The angular velocities were obtained using adjacent points of collected data following equation (2):

$$\omega = \frac{\theta 2 - \theta 1}{t} \tag{2}$$

The variable t is the time between adjacent calculated angles $\theta 1$ and $\theta 2$. For this paper t is equal *1*/315 seconds.

The source code to extract markers position and to generate knee angles, angles velocities and angles accelerations, is available in http://github.com/rob-nn/gait_data_loader.

C. CMAC setting

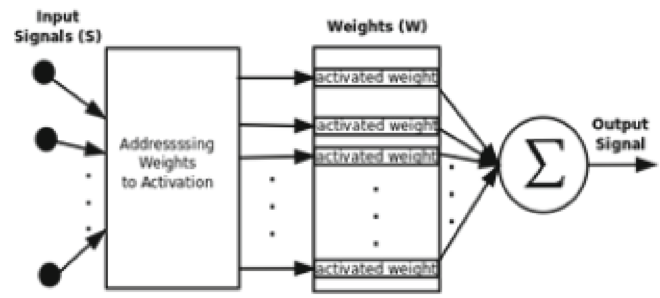

Fig. 2: Simulation Process

The CMAC model proposed by [3] is depicted in Figure 2. This type of ANN receives a signal vector *S,* and then uses this vector to calculate addresses. These addresses correspond to positions in weight vector *W*. Only activated weights will be summed to compose the CMAC output signal.

The input signals for a CMAC must have their enter space known. For example, for a vector *S,* with signals *s1*, *s2* and *s3*, the signal *s1* could have a value between 0 and 1, the signal *s2* a value between -500 and 100 and *s3* a value between -0.1 and 0.1. Furthermore, each signal must be converted to an exact quantity of discrete values. For example, a signal *sx*, which is between 0 and 2, has five discrete values, such that their possible values are 0, 0.5, 1, 1.5 and 2. If for example, it has the value 1.6, then *sx* takes the value 2, because there is not the value 1.6 in the list of discrete values. The quantity of discrete values by signal defines the signal resolution. More discrete values more resolution. Less discrete values less resolution. See Figure 3. The calculation of addresses has been implemented using the same algorithm proposed by [3].

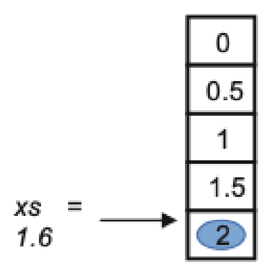

Fig. 3: Input value discretization for a signal *sx*

An essential parameter in CMAC is the number of activations. As the name says, it defines the number of weights which will compose the output. How much higher this pa-