Integrating multi-sensory input in the body model — a RNN approach to connect visual features and motor control

Malte Schilling

Abstract— **An internal model of the own body can be assumed to be a central and early representation as such a model is already required in simple behavioural tasks. More and more evidence is showing that such grounded internal models are applied in higher level tasks. Internal models appear to be recruited in service for cognitive function. Understanding what another person is doing seems to rely on the ability to step into the shoes of the other person and map the observed action onto ones own action system. This rules out dedicated and highly specialized models, but presupposes a flexible internal model which can be applied in different context and fulfilling different functions. Here, we are going to present a recurrent neural network approach of an internal body model. The model can be used in the context of movement control, e.g. in reaching tasks, but can also be employed as a predictor, e.g. for planning ahead. The introduced extension allows to integrate visual features into the kinematic model. Simulation results show how in this way the model can be to utilised in perception.**

I. Introduction

The way in which we perceive our surroundings is strongly predetermined through our expectations and memories [1]. Our internal representations guide how and what we perceive. We exploit past experiences in order to be better prepared as our memories allow us to anticipate what will come next or what might become important in a specific context. In this sense, what comes to mind is not a large set of unstructured signals from different sensory modalities, but a coherent impression of a few rich concepts which are multimodal in their nature [2]. It has often been assumed that conceptualisations were formed in order to allow for higher level functions as planning ahead. But today there is more and more evidence accumulating that internal representations are a form of by-product and their primary purpose was to subserve action and perception in linking related information [3]. Internal models reflect the structure of co-occuring sensory and motor signals.

Nowadays, there are many psychological, neuropsychological and neurophysiological data supporting the tight coupling between the system responsible for controlling an action and the ones responsible for perceiving, imagining and understanding such an action (see e.g., the common coding principle [4], [5], imaging studies, e.g., [6], mirror neurons [7]). It is assumed that already existing functions in the brain are recruited in different tasks, e.g., in planning ahead already existing internal models are reused in a new context in an internal simulation [8]. In perception, internal models seem to be recruited in order to understand the meaning of an action of another person. The perception is driving the body model and the following experience of being moved is making us understand what is going on as we step into the shoes of the other.

Recruitment of internal models presupposes that the internal models are quite flexible. This is in contrast to many models of internal representation which concentrate on one single function. In this paper, we want to present a body model which can serve different function. The model is implemented as a recurrent neural network. First, the model can be used in motor control. This provides an evolutionary account how such a model might have evolved in the first place and in this way grounds the internal representation. A model of the own body can be assumed as one of the first models acquired [9]. Already in simple tasks as targeted movements such a model is required and must have co-evolved in parallel and in service for this action [10]. The MMC model presented here can be used to control reaching movements and solve inverse kinematic tasks. In addition, the model also can act as a predictor or forward model (kinematic or dynamic) which is necessary when performing fast movements, but might also be exploited in planning ahead. In planning ahead the model can be used to predict consequences and instead of performing an action it can be tested in advance in internal simulation [8] (for application of our model for prediction see [11]). In this paper we are extending our model to integrate visual information, i.e., do sensor fusion.

How does an internal model of the body subserve perception? Loula and colleagues performed a series of experiments showing that humans are using their own body model in seeing others doing a movement [12]. Test subjects were watching short point-light display movies. These movies do not show a person performing a movement. Only markers which are attached to the joints of the actor can be seen. The set of markers on the person is in addition masked by a number of markers that are randomly oscillating around a random position. Even though the visual given information is not sufficient to understand what is shown, subjects can easily and quickly recognise what is going on in such a clip. This must rely on additional information which is integrated with the visual stimuli. A body model provides an explanation. It can be used as a set of constraints how movements of body parts relate. While perceiving a movement the brain is constantly trying to detect structure in the movements. It tries to connect the dots and assumes that there might be a

Malte Schilling is with the International Computer Science Institute, Berkeley (1947 Center Street, Suite 600, Berkeley, California 94704, email: mschilli@icsi.berkeley.edu) and Sony Computer Science Laboratory (6 rue Amyot, 75005 Paris, France).

This work was supported by a DAAD grant to Malte Schilling and has been carried out with partial support from the ALEAR project, funded by the EU Cognitive Systems program.

body in there. The seen markers are mapped to parts of the body model. The ones moving randomly around are treated as noise, but the ones placed on a joint of a person are moving in the same structured way as predicted by the body model. The internal model can pick up the visually given movement. When going on watching the movement, the body model and visual markers are moving together leading to the impression that one can see a body moving. There is more and more evidence accumulating that internal models are recruited in such a way in perception. Interestingly, subjects are not only able to easily distinguish different movements, but they are also able to recognise who is shown. And even though one only rarely sees oneself walking or moving, when watching the point-light movies the subjects are best in identifying themselves compared to clips showing people they knew well and have often seen making the shown movements. These and other findings [13] indicate that the underlying information can not be only visual. Instead, the underlying structure appears to be grounded in movement control and that it is the same internal body model used for the control of ones own movement which is also recruited in perception. This is also supported by findings from neuroscience and neurobiology showing that single neurons or parts of the brain—which were assumed for a long time to be subserving motor control—are also engaged in perception of movements (Mirror Neurons found in monkeys [7] and the Mirror Neuron System in humans [6].).

In the following, we are presenting an approach for an internal body model allowing for all the requested functions, a Mean of Multiple Computation network. The model will be introduced in the context of motor control and we will show briefly how it can control reaching movements and solve the inverse kinematic problem. The main contribution of the paper follows, showing in principle how visual features can be integrated into the model. Simulation results are presented demonstrating how the body model can be used in perception in a similar task as explained above, i.e., how the model can be driven by visual information.

II. Mean of Multiple Computation Principle

A Mean of Multiple Computation (MMC) network is a recurrent neural network [14]. The connections of the net encode constraints which are derived from a set of equations and which are usually not learnt. The activation of the network is constrained by the defining equations and the network acts as an autoassociator. Such a network can be used to describe the kinematics of a manipulator like a human or robotic arm. In this case, the model can be applied for solving inverse, forward or mixed kinematic function due to its pattern completion capabilities. The constraints of the network establish the attractor characteristics of the network. A task can be given to the network as a partial activation state which is enforced onto some of the variables of the net. The problem is solved by the network through filling in complementary information. At the same time the overall state is kept in agreement with the encoded kinematic relations which drive the relaxation behaviour of the network. MMC networks have shown to be able to solve inverse, forward and mixed kinematic problems and always stabilise to a valid geometric state (i.e., a state which can be adopted by the arm)—even if there is no solution [15].

The MMC principle will be introduced using a descriptive example: An arm consisting of three segments and three joints (see fig. 1). At first, this arm shall be restricted to movements in a plane and will be later-on extended to work in three dimensions. The arm is redundant as it works in two dimensions while featuring three control variables, one for each joint (for the three dimensional case the arm has nine degrees of freedom). Usually, redundancy is a problem for approaches to inverse kinematics as in a redundant system there are many solutions possible and the system has to choose one of them [16]. In contrast, the MMC approach exploits redundancy to find a solution in an iterative fashion. As a representation of the arm, we choose a Cartesian space as this makes the introduction of the MMC principle straightforward. Nonetheless, the principle can be applied to other representations as well and has been successfully applied for axis-angle representations as are usually employed in robotic descriptions [17].

The arm consists in general of three segments represented as vectors L_1, L_2 and L_3. For the two dimensional case, each vector consists of two components, a x- and a y-component. For each of the components a single network is used, but the structural relations are the same for both of the networks and therefore, we will only discuss the network used for the x-component (written as x^{L_0},...). In addition to the three segment vectors, we define an end-effector vector R and two diagonal vectors D_1 and D_2 (fig. 1).

The kinematics of the network can be described through local relationships. In this way each variable is related to the others. One can think of these local relations as closed chains of vectors, i.e., three vectors which form a triangle. The first

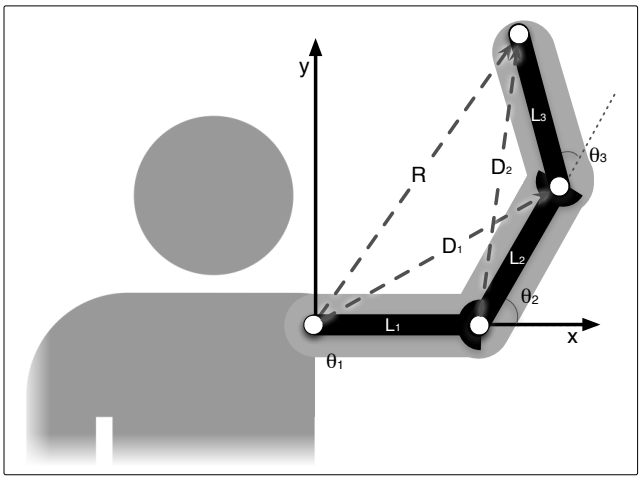

Fig. 1. Graphic representation of a planar (2D) arm consisting of three segments, upper arm (L_1), lower arm (L_2) and hand (L_3). Vector R points to the position of the end effector (tip of the hand). D_1 and D_2 represent additional diagonal vectors.

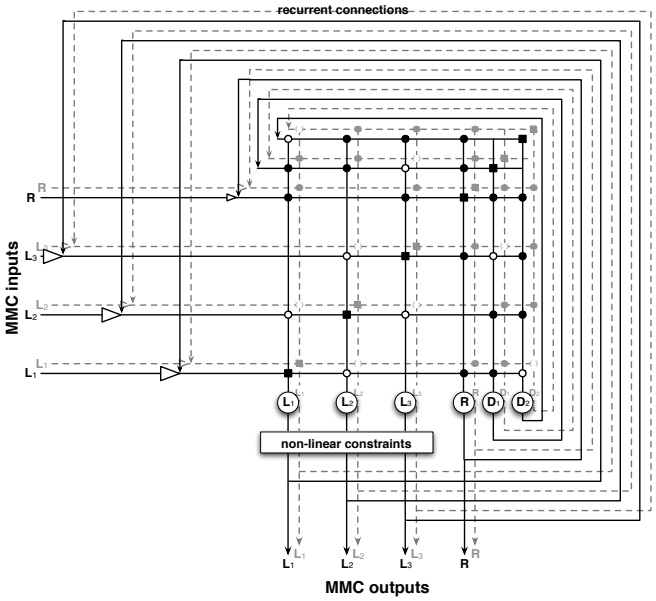

Fig. 2. A MMC network: The net consists of two identical linear networks, one for the x-components (black lines) and the other for the y-components (grey dashed lines) of the vectors. The units represent the components of the six vectors L_1, L_2, L_3, D_1, D_2 and R of the planar arm (see Fig. 1 for graphic illustration). If an input is given, the corresponding recurrent channel is suppressed (symbolised by the open arrow heads). Positive weights are indicated by filled circles and negative weighs are indicated by open circles. The recurrent connections are shown as black squares (in the diagonal). All other weights are zero. For details see text.

segment can be described in this way by two equations (see fig. 1):

$$\begin{aligned} x^{L_1} + x^{D_2} - x^R &= 0 \\ x^{L_1} + x^{L_2} - x^{D_1} &= 0 \end{aligned} \tag{1}$$

We got multiple equations containing each variable—for the arm example there are overall four different equations and each variable is present in two of them (in general, one obtains $\binom{n}{3}$ equations). For each variable all equations containing this variable are solved. Again, for the first segment this leads to the following two equations:

$$\begin{aligned} x^{L_1} &= x^R - x^{D_2} \\ x^{L_1} &= x^{D_1} - x^{L_2} \end{aligned} \tag{2}$$

There are now Multiple ways of Computing each variable. Each variable is time dependent and shall be computed in an iterative fashion. Calculating a new value of a variable for the next time step shall incorporate all of the above computations. To integrate the—possibly different—solutions we simply calculate the Mean value of all the solutions and also include the current value of the variable. Including the current value inhibits fast changes and therefore prevents oscillations (the recurrent connection introduces low pass properties and the weight of the connection can be related to the time constant [18]). A new value for the first segment can be calculated therefore as:

$$\begin{aligned} x^{L_1}(t+1) =& \frac{1}{d}(x^R(t) - x^{D_2}(t)) + \frac{1}{d}(x^{D_1}(t) - x^{L_2}(t)) \\ &+ \frac{d-2}{d} x^{L_1}(t) \end{aligned} \tag{3}$$

For each variable one gets a similar equation which consists of the integrated equations and the current value. This set of equations describes the relations between all the variables and how to calculate one variable from the other values. It also can be interpreted directly as a weight matrix defining a neural network (see fig. 2, equation 3 corresponds to the first column of the network weights: R and D_1 are of positive weight while D_2, L_2 have a negative sign).

To explain the behaviour of the network, let us consider some examples. As the network is setup using forward kinematic relations, it is obvious that the network can calculate the position of the end-effector when the single segment vectors are given and solve the forward kinematic problem. In the inverse kinematic case the end-effector position R is given to the network and the network has to find a set of complying segment orientations. While in a stable state all of the multiple equations would provide the same result for a variable, the introduction of a new end-effector vector changes this and all equations containing R are affected. These equations lead now to different results and the introduced disturbance is spread through these equations onto the other variables. Over time the network converges to a harmonic state again, that is a state in which the encoded kinematic relations are valid once again and the new end-effector position has been accounted for [15]. This can be seen in fig. 3. The initial position is shown in light grey and for every second iteration step the new arm configuration as provided by the network is shown through dotted lines. As can be seen in fig. 3 a), the network approaches the end position over a small number of iteration steps. In b) a case is shown in which the target can not be reached as it is outside of the workspace. But the network comes up with a coherent arm configuration which is minimizing the distance of the end-effector to the target. The model solves the task as good as possible.

The presented model above requires for a vector representation an additional processing step after each iteration. As all variables can be changed independently the length of the vectors is not guaranteed to stay the same. But the vectors representing rigid segments should always be of the same size. Therefore it is necessary to normalise these vectors after each iteration. Using a joint angle representation—as common in robotics—for control of the movements of the segments makes this unneccessary [17].

To summarise, MMC networks can easily be derived to describe the kinematics of any manipulator. For complex structures the structure can be divided onto different levels of a hierarchy of connected MMC networks [19]. A MMC net is not restricted to one type of problems, but can solve forward, inverse or any mixed kinematic problem. States of the network always correspond to geometric valid configurations.

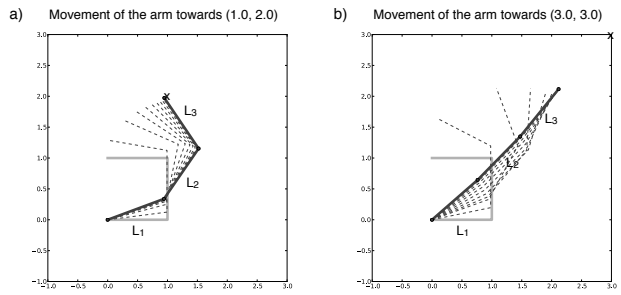

Fig. 3. Solution of the inverse kinematic problem (damping $d = 5$). The manipulator arm should point to a given position, marked by a cross, starting from an initial configuration (dotted lines). In a) a situation is shown in which one of the infinite numbered possible solutions is chosen. In b) the target is situated outside the workspace of the arm, therefore there is no solution possible. Nevertheless, the network solves the task "as good as possible".

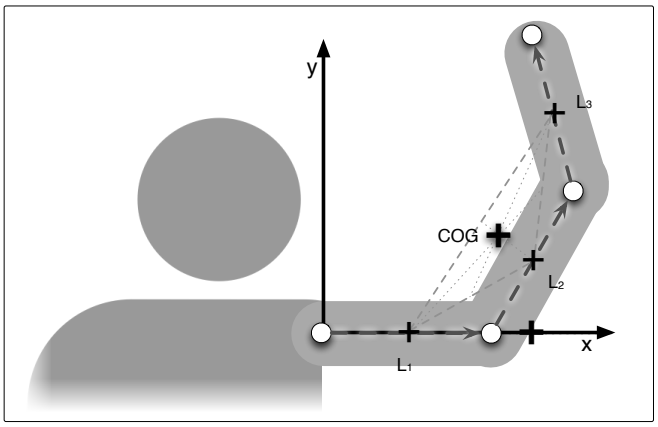

Fig. 4. Determination of the visual center of gravity. + signals the single segment COG. The overall visual COG lies at the centroid of the triangle spanned by the three segment COGs and is marked by a bold + sign.

III. Integration of visual features

The MMC model has been introduced in the context of reaching movements, solving the inverse kinematic problem. Besides this, we extended MMC networks to include dynamic influences and to use such a model as a predictor of movement dynamics [20]. A predictive model may also allow for higher-level cognitive functions. Planning ahead can be realised using such a model of the body in internal simulation [8] (for details of applying the MMC model for planning of movements of a hexapod robot see [11]).

As mentioned in the introduction, internal models are also involved in perception. The internal representations govern how we perceive our environment and we constantly try to map incoming perceptions onto our existing embodied representations [21], [22]. First, to make sense of them. That is to recognise and categorise what we perceive. And second, to be able to draw conclusions. That is to predict what might happen next or to remember what might be also important in a context. The example in the introduction demonstrates how our visual perception of movements is guided by our own movement expertise as encoded in a body model. It appears that we use our internal body model constantly during perception of others moving around to understand them (See research on Mirror Neurons, that are neurons in areas which were for a long time assumed to be solely responsible for motor control, but have also been found active in perception of movements. These neurons appear to encode goal directed action, independent of who is carrying them out [7], [23].).

How can our model serve visual perception? The task of the model would be to map an observed body to the body model and to bring the model in line with the observed body. Therefore, the model has to incorporate visual features. This approach is comparable to the application of Kalman filters which also exploit known underlying structural relations to predict future states (see e.g. [22], [24]). The difference is that in our model the prediction relies explicitly on the given body model regardless of what values have to be predicted (proprioceptive information or visual information). In the following, we will show how visual descriptors can be introduced into the model. We will use image moments which describe a seen body, in our case an arm. Image moments [25] are features that describe shape properties of the foreground object, like visual center of gravity (COG) or orientation. They capture statistical regularities of the pixels. An advantage of image moments is that they are easy and inexpensive to compute and are at the same time descriptive. In general, image moments are calculated using binary pixel-based images ($I(x, y)$ intensity function, object pixels are equal to 1) as visual input and applying the formula

$$M_{pq} = \sum_x \sum_y x^p y^q I(x, y) \tag{4}$$

The order of the visual moment is defined as the sum of p and q. These two factors are weighting the summation over all pixels. The zeroth order moment represents the covered area of the foreground object. From the first order image moments one can derive the visual COG $(\overline{x}, \overline{y})$ of the object

$$\overline{x} = \frac{M_{10}}{M_{00}}, \overline{y} = \frac{M_{01}}{M_{00}} \tag{5}$$

While higher order moments provide information as orientation and shape, we will only use these lower order image moments to show how visual descriptors can be incorporated into a MMC network in principle. Already this information is sufficient for the arm model to follow an observed arm.

Image moments can be calculated from an input image. But to influence the network behaviour they have to be related to the kinematic variables of the network. It is possible to derive the image moments for any given configuration of the arm (see fig. 4, derivation of the centroid). First, the overall image moments can be decomposed into moments representing the single segments

$$M_{pq}^{ges} = M_{pq}^{L_1} + M_{pq}^{L_2} + M_{pq}^{L_3} \tag{6}$$

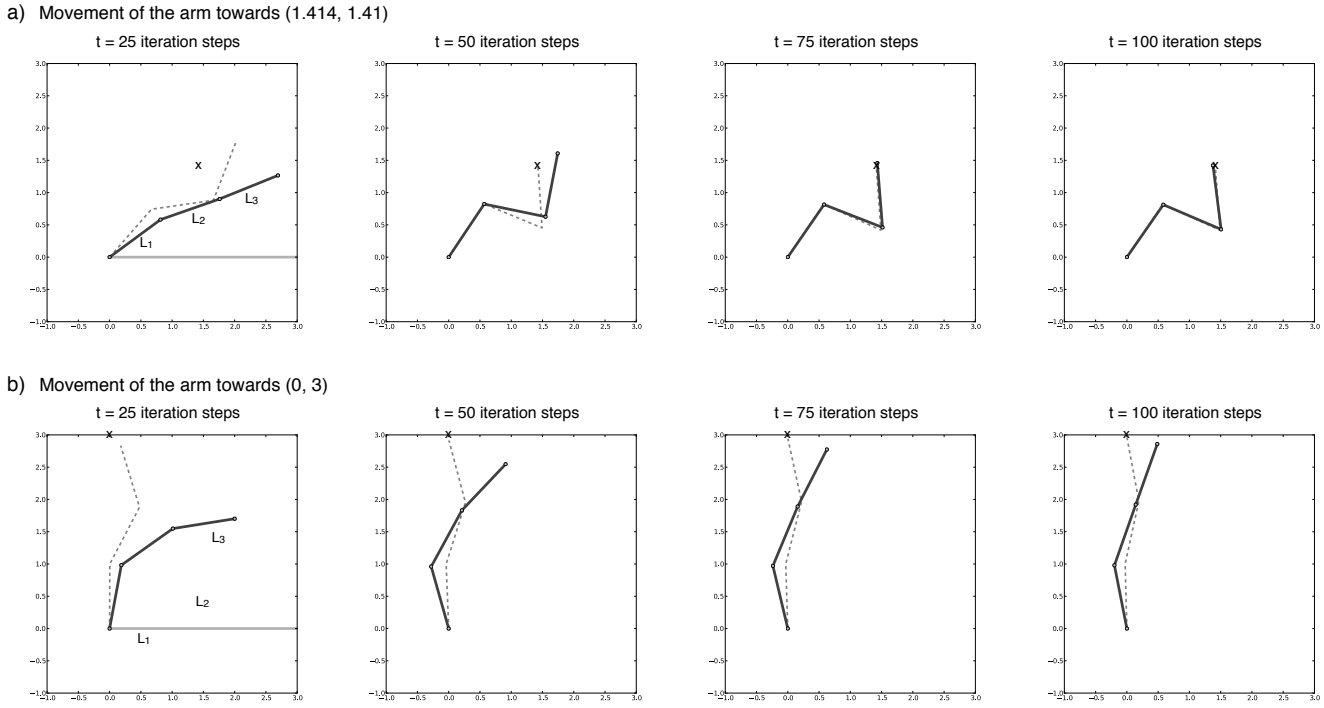

Fig. 5. Two examples of the perceived arm movement. Course of time is going from left to right, shown are snapshots of iteration 25, 50, 75 and 100. In the first figure at the left the initial configuration is shown in light gray. The moving arm is shown as a dashed line and the current state of the MMC model used for perception is represented as the dark grey line. In a) a near target is shown and in b) a target is shown for the fully stretched arm.

The centroid is defined as (shown only for the x-component)

$$\begin{aligned}\overline{x}^{ges} &= \frac{M_{10}^{ges}}{M_{00}^{ges}} = \frac{M_{10}^{L_1} + M_{10}^{L_2} + M_{10}^{L_3}}{3M_{00}^{L}} \\ &= \frac{1}{3}(\overline{x}^{L_1} + \overline{x}^{L_2} + \overline{x}^{L_3}) \end{aligned} \tag{7}$$

The centroid of each segment is given as the midpoint between starting and end point of the segment vector:

$$\begin{aligned}\overline{x}^{L_1} &= \frac{1}{2}(0 + x^{L_1}) \\ \overline{x}^{L_2} &= \frac{1}{2}(x^{L_1} + (x^{L_1} + x^{L_2})) \\ \overline{x}^{L_3} &= \frac{1}{2}((x^{L_1} + x^{L_2}) + (x^{L_1} + x^{L_2} + x^{L_3})) \end{aligned} \tag{8}$$

When employing these equations (8) into the centroid equation (7), we get

$$\begin{aligned}\overline{x}^{ges} &= \tfrac{1}{6} \quad ((0 + x^{L_1}) \\ &\qquad +(x^{L_1} + (x^{L_1} + x^{L_2})) \\ &\qquad +((x^{L_1} + x^{L_2}) + (x^{L_1} + x^{L_2} + x^{L_3}))) \\ &= \tfrac{5}{6} \quad x^{L_1} + \tfrac{1}{2}x^{L_2} + \tfrac{1}{6}x^{L_3} \end{aligned} \tag{9}$$

This equation provides us with a visual COG vector for the whole arm when the single segment vectors are given. The centroid vector can now be introduced into the network. In addition, for each of the contained kinematic variables, we can derrive an equation relating this variable to the centroid. Solving equation (9) for the segment variables leads to

$$\begin{aligned}x^{L_1} &= \frac{6}{5}\overline{x}^{ges} - \frac{3}{5}x^{L_2} - \frac{1}{5}x^{L_3} \\ x^{L_2} &= 2\overline{x}^{ges} - \frac{5}{3}x^{L_1} - \frac{1}{3}x^{L_3} \\ x^{L_3} &= 6\overline{x}^{ges} - 5x^{L_1} - 3x^{L_2} \end{aligned} \tag{10}$$

These equations are integrated following the MMC principle into the overall network. New segment vectors are now computed using three equations and the current variable value. In the case of L_1 the kinematic equations (2) and equation (10) are combined[1] (compare to equation 3):

$$\begin{aligned}x^{L_1}(t+1) = &\frac{1}{d}(x^R(t) - x^{D_2}(t)) \\ &+ \frac{1}{d}(x^{D_1}(t) - x^{L_2}(t)) \\ &+ \frac{k_1}{d}(\frac{6}{5}\overline{x}^{ges} - \frac{3}{5}x^{L_2} - \frac{1}{5}x^{L_3}) \\ &+ \frac{d - (k_1 + 2)}{d}x^{L_1}(t) \end{aligned} \tag{11}$$

In addition to the centroid, we want to also use depth information. Therefore, we have to to extend our MMC network to three dimensions which is straightforward. Until now, two identical networks were used. One for the x- and

[1] k_i is a factor weighting differentially the influence of the centroid equations. As can be seen from the equations in (10) the centroid is affecting the different segments not equally. Therefore, the influence of this equation has to be weighted. We used in our computation $k_1 = 2$, $k_2 = 1$, $k_3 = 0.5$.

one for the y-component. To extend the network to three dimension one has just to add a third network representing the z-component. The kinematic equations are the same as given in equation (3). As the z-component is orthogonal to the picture plane the centroid position does not contain any depth information. But the visual information does contain depth information. The visual area is changing depending on where the arm is pointing. Again, we make simplifying assumptions. First, the depth of the segments is negligible, that means when a segment is pointing towards the viewer the area is approaching zero. Second, we are assuming a parallel visual projection of the whole scene, therefore objects does not appear smaller when they are farer away. We define the visual area of a segment when seen from the side as one ($M_{00}^{L_p} = 1$). For all cases in between the area equals the projection of the segment onto the x-y-plane (the view plane). This can be calculated as

$$M_{00}^{L_i} = \sqrt{(x^{L_i})^2 + (y^{L_i})^2} \tag{12}$$

Taken together for all the segments, the summed visual area is

$$M_{00}^{ges} = M_{00}^{L_1} + M_{00}^{L_2} + M_{00}^{L_3} \tag{13}$$

We can now introduce the visual area for each segment as an additional variable and calculate it from the segment vectors (as the summation of the partial visual areas calculated for one segment). But in addition, when a visual area is provided we can estimate the depth information for the whole arm and integrate this depth information. First of all, we have to share the visual area information onto the segments, e.g. gor the first segment we estimate

$$M_{00}^{L_1} = M_{00}^{ges} - (M_{00}^{L_2} + M_{00}^{L_3}) \tag{14}$$

As the projection on the x-y-plane, the z-component of the segment vector and the segment vector (assumed of unit length) form an orthogonal triangle, it holds $(x^{L_i})^2 + (y^{L_i})^2 = 1 - (z^{L_i})^2$. Therefore, we can substitute $(x^{L_i})^2 + (y^{L_i})^2$ in equation (12) and can calculate the matching z-component from the estimated visual area for that segment

$$\begin{aligned} M_{00}^{L_i} &= \sqrt{1 - (z^{L_i})^2} \\ z^{L_i} &= \sqrt{1 - (M_{00}^{L_i})^2} \end{aligned} \tag{15}$$

The estimate for the z-component can now be integrated in the MMC network through application of the MMC principle (it is only incorporated in the network of the z-component). In the following, the two extensions will be evaluated in simulations.

IV. Results

The network can be used in simulations for perceiving movements of an arm. On the one hand, there is a MMC network controlling the movements of an arm. The task for the arm is to perform reaching movements. Input to this network are target points. From its initial configuration the arm is then reaching towards the targets. On the other hand, there is a second MMC network acting as an observer. This network shall follow the performed movement. The visual features of the moving arm are input to this network. The image moments then drive the network. We then compare the movement with the perceived movement over time.

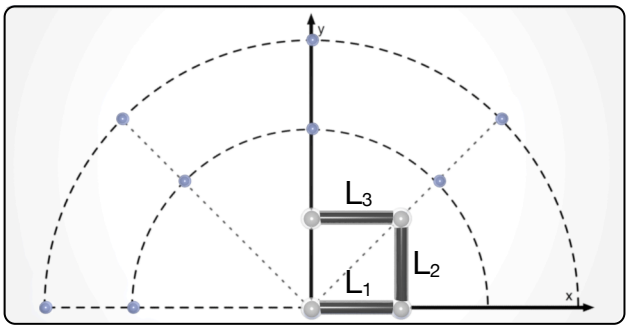

Fig. 6. Arrangements of the target points. Shown is the robot arm in the initial configuration. The targets (white crosses) are arranged around the base of the manipulator on two circles and in intervals of $45°$.

A. Perceiving movements in a plane

In the first series of simulations, the movements are restricted to movements of the arm in a plane (no depth information is used). Input to the MMC network controlling the arm are target points. Two examples for movements are shown in fig. 5. The arm driven by the image moment properties nicely follows the moving arm. What immediately becomes apparent is that the movement is slower than in the original approach. This is not surprising as now there are more equations involved while for the perceiving network only one equation is used for steering the network. The effect could be counteracted through reducing the damping value—but for better comparison we choose the same damping value as in the initial simulations $d = 5$ in all simulations.

To analyse the networks performance quantitatively, we set-up a set of 10 targets. The targets are arranged on two half circles (fig. 6). The vectors pointing to the five targets on each circle enclose $45°$ respectively. The two distances to the two sets of targets are two segment lengths and three segment lenghts which would mean the arm is fully stretched. As can be seen in fig. 5, for the near targets the arm has to fold. The joints connecting the segments are not restricted and there are two distinct different ways of folding the arm. For this series we therefore decided not to start from a fully stretched arm configuration, as in this case both ways of folding in one joint are equally likely. As a visual centroid is not distinguishing the shape, there is no information provided to the observer network in which way to fold. As a consequence the perceived arm and the moved arm could end up in very different configuration just because the MMC model of the perceiving network chose the other possibility of folding the arm. This could been circumvented incorporating additional visual features like higher order image moments. As in this paper we only want to present the general approach, we decided to use a pragmatic and

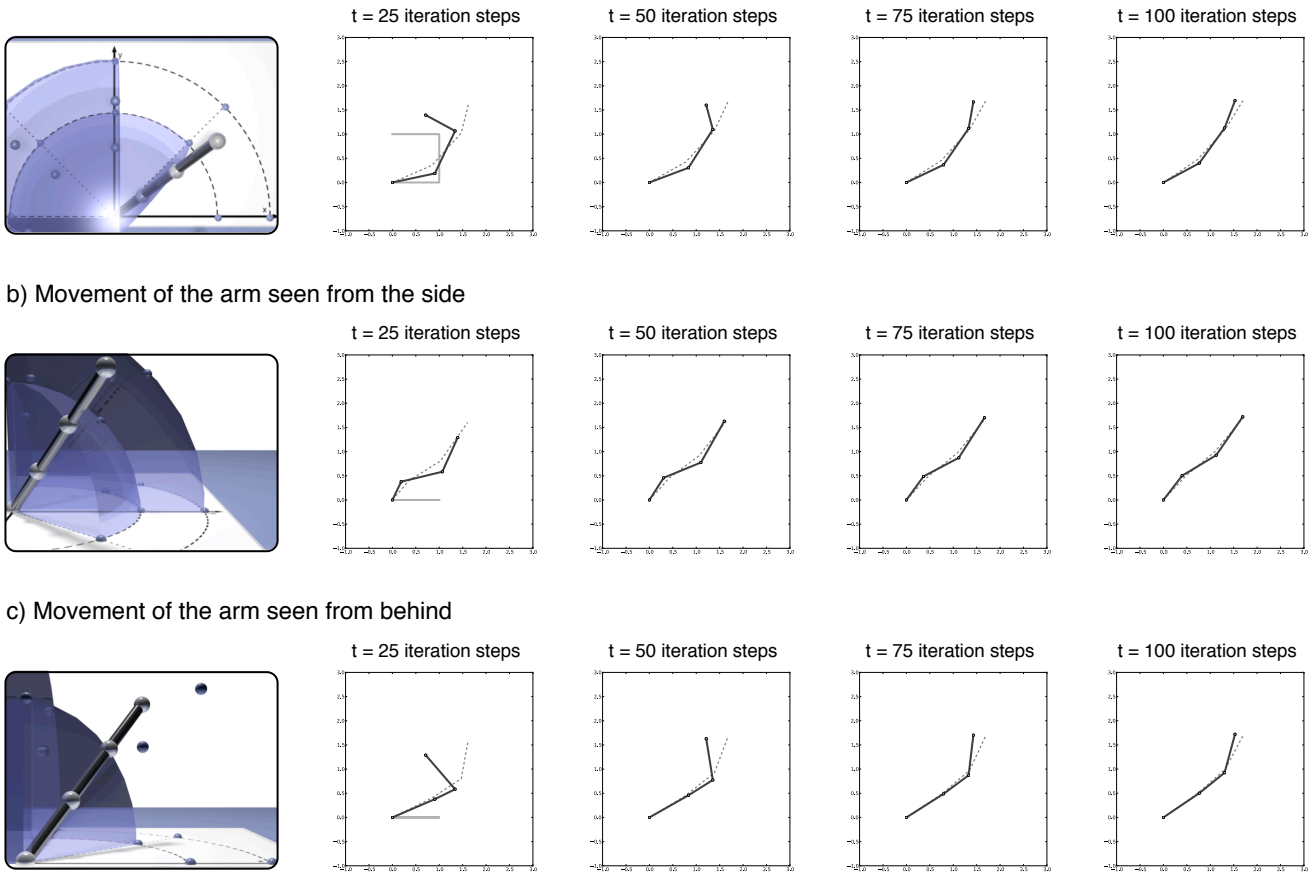

Fig. 7. Example movement in three dimensions, shown from different perspectives (a) view from above, b) side view, c) view from behind). Consider the initial configuration of the arm as shown in fig. 6 (in b) this means that the second segment lies in front of the viewer; in c) the third segment lies behind the other two segments). Course of time is going from left to right, shown are snapshots of iteration 25, 50, 75 and 100. At the left a rendering is visualising the perspective. In the first snapshot at the left the initial configuration is shown in light gray. The moving arm is shown as a dashed line and the current state of the MMC model used for perception is represented as the dark grey line.

simple solution, i.e., to use an initial configuration in which the arm is not fully stretched but already folded. In this case the MMC network does not have to choose how to fold as this is already determined (the arm in fig. 6 shows the initial configuration). The controlled arm reached out during a period of 200 iteration steps towards one target point after the other. Again, the observing network adopted in all cases a similar configuration. First, we compared the distance between the end-points of the movement control network and the network visually perceiving the movement. The mean difference between the two end-points was 0.178 units (a unit equals one segment length, standard deviation of ± 0.142 units). A better measurement for comparing the configurations of the networks states is to look at the differences of the single segment orientations. The difference angle for the segment orientations of the perceived arm and the moving arm were computed for each segment. The mean difference was $-0.49°$ (standard deviation $\pm 12.29°$). Mostly the last segment was responsible for the high variation which is not surprising as the orientations of the first two segments are weighted higher. The mean difference in segment orientations for the last segment was $-1.41°$ (standard deviation $\pm 18.90°$; for the first segment: $-2.73°$, st.d. $\pm 4.33°$; second segment: $2.71°$, st.d. $\pm 9.63°$).

B. Perceiving three dimensional movements

In a second series of movements, we also used depth information. Again, we used a set of targets in three dimensional space. As before, the targets are arranged on two spheres (radii of two and three segment lengths). In the x-y-plane the same targets are used. Additional targets have been introduced. On the one hand, two targets directly above the base of the manipulator. One target on the inner sphere and one on the outer sphere. In addition, another target has been placed on each sphere in the middle between a target lying in the x-y-plane and the target on top of the sphere (in this way the vectors towards these targets and the x-y-plane enclose a $45°$ angle, see fig. 7 on the left). Information about the covered visual area from the moving arm was given to the observer network. Fig. 7 shows one example of a perceived movement. As can be seen, this information is sufficient

to drive the network also in three dimensions. Again, we analysed this quantitatively. For the twenty two different targets we got a mean difference between controlled arm and perceived arm of 0.135 units (st.d. ± 0.113 units). The orientation of the single segments differed by $-0.63°$(st.d. $\pm 10.47°$)

V. Conclusion

In this article, we presented a body model which can fulfil different function and can be flexibly applied in different contexts. Using the MMC principle, we first used the network for motor control of a redundant arm (The model has already been used for more complex structures [19].). The model can also be used for prediction and allows for planning ahead [11]. We extended the model and incorporated visual information. Additional equations are integrated into the model and due to its autoassociator capabilities the presented body model can be now used to mediate between visual spaces and segment orientations. This has been demonstrated in simulations. For application in a real world scenario on a robot more sophisticated descriptors might be needed and preprocessing becomes necessary. When during perception another person or robot should be mapped onto the body model, it is required to, first, find the agent in the picture which means it has to be segmented from the background. Then the features have to be extracted from the picture and one must compensate changes in scale and orientation. As an example, normalised and centralised moments [26] of higher order could be used in the future as descriptors which are invariant against rotation and scale changes. At the same time this would improve the results as these image features include shape information. In the simple scenario as presented here, perceived and controlled arm align in general, even though we only used visual moments of first order and therefore no form describing features at all. As the model is realised as a neural network approach, we want to integrate additional and more descriptive visual features in the future through learning parts of the neural network.

References

[1] D. L. Schacter, D. R. Addis, and R. Buckner, "Remembering the past to imagine the future: the prospective brain," *Nature Reviews Neuroscience*, vol. 8, no. 7, pp. 657–661, 2007.

[2] A. M. Glenberg, "What memory is for," *Behavioral and Brain Sciences*, vol. 20, no. 1, pp. 1–55, March 1997.

[3] P. F. Verschure and P. Althaus, "The study of learning and problem solving using artificial devices: Synthetic epistemology," *Bildung und Erziehung*, vol. 52, no. 3, pp. 317–333, 1999.

[4] W. Prinz, "Perception and action planning," *The European Journal of Cognitive Psychology*, vol. 9, no. 2, pp. 129–154, 1997.

[5] M. Jeannerod, "To act or not to act: Perspectives on the representation of actions," *Quarterly Journal of Experimental Psychology*, vol. 52A, pp. 1–29, 1999.

[6] G. Buccino, F. Binkofski, G. R. Fink, L. Fadiga, L. Fogassi, V. Gallese, R. J. Seitz, K. Zilles, G. Rizzolatti, and H. J. Freund, "Action observation activates premotor and parietal areas in a somatotopic manner: an fMRI study." *European Journal of Neuroscience*, vol. 13, no. 2, pp. 400–404, January 2001.

[7] G. Rizzolatti, "The mirror neuron system and its function in humans," *Anatomy and Embryology*, vol. 210, no. 5–6, pp. 419–421, 2005.

[8] G. Hesslow, "Conscious thought as simulation of behaviour and perception," *Trends in Cognitive Sciences*, vol. 6, no. 6, pp. 242–247, 2002.

[9] H. Cruse, "Feeling our body - the basis of cognition?" *Evolution and Cognition*, vol. 5, no. 2, pp. 162–173, 1999.

[10] L. Steels, "Intelligence with representation," *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, vol. 361, no. 1811, pp. 2381–2395, 2003.

[11] M. Schilling and H. Cruse, "The evolution of cognition – from first order to second order embodiment," in *Modeling Communication with Robots and Virtual Humans*, I. Wachsmuth and G. Knoblich, Eds. Berlin: Springer, 2008, pp. 77–108.

[12] F. Loula, S. Prasad, K. Harber, and M. Shiffrar, "Recognizing people from their movement," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 31, no. 1, pp. 210–220, 2005.

[13] M. Shiffrar, "Movement and event perception," in *The Blackwell Handbook of Perception*, B. Goldstein, Ed. Blackwell Publishers, Oxford, 2001, pp. 237–272.

[14] H. Cruse and U. Steinkühler, "Solution of the direct and inverse kinematic problems by a common algorithm based on the mean of multiple computations," *Biological Cybernetics*, vol. 69, pp. 345–351, 1993.

[15] U. Steinkühler and H. Cruse, "A holistic model for an internal representation to control the movement of a manipulator with redundant degrees of freedom," *Biological Cybernetics*, vol. 79, no. 6, pp. 457–466, 1998.

[16] N. A. Bernstein, *The Co-ordination and regulation of movements*. Pergamon Press Ltd., Oxford, 1967.

[17] M. Schilling and H. Cruse, "Universally manipulable body models – dual quaternion representations in layered and dynamic MMCs," Submitted, 2009.

[18] V. Makarov, Y. Song, M. Velarde, D. Hübner, and H. Cruse, "Elements for a general memory structure: properties of recurrent neural networks used to form situation models," *Biological Cybernetics.*, vol. 98, no. 5, pp. 371–395, 2008.

[19] M. Schilling and H. Cruse, "Hierarchical MMC Networks as a manipulable body model," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2007), Orlando, FL*, 2007, pp. 2141–2146.

[20] M. Schilling, "Dynamic equations in MMC networks: Construction of a dynamic body model," in *Proc. of The 12th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR)*, 2009.

[21] J. Decety and J. Grezes, "Neural mechanisms subserving the perception of human actions," *Trends in Cognitive Sciences*, vol. 3, no. 5, pp. 172–178, May 1999.

[22] R. Grush, "The emulation theory of representation: Motor control, imagery, and perception," *Behavioral and Brain Sciences*, vol. 27, pp. 377–442, 2004.

[23] V. Gallese, C. Keysers, and G. Rizzolatti, "A unifying view of the basis of social cognition," *Trends in Cognitive Sciences*, vol. 8, no. 9, pp. 396–403, 2004.

[24] D. Wolpert, "Probabilistic models in human sensorimotor control." *Human Movement Science*, vol. 26, pp. 511–524, 2007.

[25] R. Mukundan and K. Ramakrishnan, *Moment Functions in Image Analysis: Theory and Applications*. London, UK: World Scientific, 1998.

[26] M.-K. Hu, "Visual pattern recognition by moment invariants," *IEEE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962. [Online]. Available: http://dx.doi.org/10.1109/TIT.1962.1057692