

# Using Quadric-Representing Neurons (QRENS) for Real-Time Learning of an Implicit Body Model

Manfred Hild, Matthias Kubisch, and Sebastian Höfer  
Neurorobotics Research Laboratory  
Humboldt-Universität zu Berlin  
Unter den Linden 6  
10099 Berlin, Germany  
{hild|kubisch|hoefer}@informatik.hu-berlin.de

**Abstract**—Controlling the body requires self-explorative behavior as well as the ability to build a model of the body. This model does not need to explicitly encode body shape, joint positions, and so on, but can just as well be built up implicitly. We introduce Quadric-Representing Neurons (QRENS) and show why they are very well-suited to model the plurality of body morphologies. QRENS can be either learned in a batch manner or on-the-fly in real-time. They possess the important property to extrapolate behavioral manifolds from a reduced and localized sensorimotor data set. QRENS can be used to elegantly control a robot’s body in a straightforward way. We comment on how QRENS have the potential to allow for modular and hierarchical learning strategies.

## I. INTRODUCTION

Learning how to control the body within a given environment is a fundamental perceptual task. It requires self-explorative behavior and at the same time the ability to build a model of the body – be it a human being, animal, or robot. In biological systems, information of body posture is available in real-time by afferent proprioceptive sensory signals, but there is no such sensory signal which is directly informative about the body’s size and shape. Although the need for a stored body model has been recognized for quite some time, only recently techniques have been introduced to systematically isolate and measure this model for one limb: in [1] the authors produced maps of the mental representation of people’s hand size and shape.

Clearly, body models do not need to explicitly encode body shape, joint positions, lengths and mass distributions of the limbs. This would only be required if a robot is to be driven using inverse kinematics, like with industrial robots, where this is still the preferred method. A large amount of literature is dealing with optimal control of smooth motion trajectories, whilst circumnavigating singularities of the inverse model; for an overview see [2]. Body models can just as well be built up implicitly. One example how information about the full body size of a segmented artificial organism can emerge within each body segment using a local homeostatic learning rule can be found in [3].

In the paper at hand we introduce Quadric-Representing Neurons (QRENS) and show why they are very well-suited to model the plurality of body morphologies. The remaining sections are organized as follows. We first define quadrics,

describe what they are able to represent geometrically, and introduce QRENS by using quadrics as kernel functions. Then, we describe the different body morphologies under investigation and give examples on how the body morphologies are mapped onto the QRENS. We discuss learning with small data sets, show how a robot can be controlled by using the QRENS, and finally comment on the QRENS’ potential to allow for modular and hierarchical learning strategies.

## II. QUADRICS AND QRENS

A quadric  $Q$  is a surface defined by an algebraic equation of degree two. Formally, we have

$$Q = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b} \mathbf{x} + c = 0\}, \quad (1)$$

where  $\mathbf{A} \in \mathbb{R}^{n \times n}$  symmetric,  $\mathbf{b} \in \mathbb{R}^n$ , and  $c \in \mathbb{R}$ . The vector notation is neat and advantageous if we need the eigenvalues and eigenvectors of  $\mathbf{A}$ . This is the case if we want to normalize the quadric, i.e., successively get rid of the mixed terms of order two (off-diagonal elements of  $\mathbf{A}$  are all zero) and, if possible, also the linear terms ( $\mathbf{b} = 0$ ) and the constant ( $c = 0$ ). Using the normalized standard form, quadrics can be categorized, and for  $n = 3$  also be visualized. Some examples are shown in figure 1. A parabolic cylinder, a hyperboloid of one sheet, and two parallel planes are respectively defined by

$$\begin{aligned} x_1^2 - x_2 &= 0, \\ x_1^2 + x_2^2 - x_3^2 - 1 &= 0, \\ x_1^2 - 1 &= 0. \end{aligned}$$

Quadrics play an important role in algebraic geometry. We can link quadrics to robot morphologies which are situated in an environment by identifying the variables with sensor values. This will briefly be addressed in the following section.



Figure 1. Three examples of different quadrics in  $\mathbb{R}^3$ , namely a parabolic cylinder, a hyperboloid of one sheet, and two parallel planes (from left to right).

### A. Why Second Order is Sufficient

For a large range of robots quadrics are able to describe invariants, i.e., when choosing the appropriate coefficients, the quadric stays zero for a specific subset of the robot's configurations. In other words, quadrics exist, which are invariant under some type of the robot's behavior. This is inter alia the case for robot arms with a series of revolute joints [4].

Why is it sufficient to use a second order polynomial to get a behavior-invariant constant expression? This depends on the type of sensor values used. Let us assume we have a robot arm with two revolute joints in series and arbitrary angular ranges and segment lengths of the arm. We are interested in all configurations where the robot touches a plane. If we use angular sensor values, then we have the trivial case, that the sum of the angles remains constant. If we use cartesian coordinates (e.g., if the sensor values are derived from an image of the scene), then we have

$$x_1 = r \cos \varphi, \quad x_2 = r \sin \varphi, \quad (2)$$

for a single joint. Obviously, we get a constant expression by squaring and summing. If we use a completely different type of sensor value, namely acceleration forces of sensors mounted on the robot arm, then again the squared sum of all values will be constant and represent the static gravitational force. This is of course only exactly true for moderately slow motions of angular joints, but approximately still holds otherwise. All in all, there is good reason why quadrics are sufficiently accurate to describe behavioral relationships between a robot's body and the environment.

### B. Using Quadrics as Kernel Functions

The main idea of a QREN is to use quadrics as kernel functions and let the QREN indicate the presence of a specific behavioral mode of a robot within a given environmental context.

Since we are not interested in normalizing and categorizing the quadric but want to learn the quadric in real-time, it is advantageous to switch from matrix notation to the following notation:

$$Q = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{w}^T \mathbf{f}_n(\mathbf{x}) = 0\}, \quad (3)$$

where  $\mathbf{f}_n : \mathbb{R}^n \rightarrow \mathbb{R}^m$  is the expansion of  $\mathbf{x}$  including all quadratic terms and the constant 1,  $\mathbf{w} \in \mathbb{R}^m$  is a weight vector, and  $m = \frac{n(n+3)}{2} + 1$ .

We are now ready to define the output of a QREN as follows:

$$q_{\mathbf{w}}(\mathbf{x}) = e^{-(\mathbf{w}^T \mathbf{f}_n(\mathbf{x}))^2}. \quad (4)$$

Clearly,  $q_{\mathbf{w}}(\mathbf{x})$  acts as an indicator neuron, as we have

$$q_{\mathbf{w}}(\mathbf{x}) = 1 \iff \mathbf{x} \in Q. \quad (5)$$

The more  $\mathbf{x}$  is distant from  $Q$ , the more  $q_{\mathbf{w}}$  is close to zero. Please note, that  $\mathbf{w}$  is only defined uniquely up to a multiplicative factor. We therefore have to normalize  $\mathbf{w}$  in some way, if we want to compare the output signals of different QRENS to find out which quadric  $\mathbf{x}$  is closer to. A straightforward approach would be to demand  $\|\mathbf{x}\| = 1$ , but

we already succeeded with the even easier trick to always have the quadric's constant term equal one (we get more explicit below).

In other fields, there exists quite some literature on how to fit the parameters of a quadric to a given data set, mostly for data stemming from stereo images or laser range measurements, e.g., see [5], [6], or, very recently, [7]. But also more general approaches have already been addressed [8]. There exist comparative surveys on the quality of different methods, as in [9] and [10], as well as reports on stability [11]. Of specific interest for our QRENS is the comparative survey of neural learning rules in [12], most standard methods of which can be applied here. Using a competitive neural network with a simple delta rule will already work well when learning several QRENS simultaneously. In this paper, we will not go into detail concerning the selection of an appropriate learning rule, but focus on the peculiarities of robot learning, namely, that sensory data from real behaviors will mostly only cover a small part of a full quadric. Therefore, we are interested in the QRENS' extrapolating abilities.

For what follows we use  $n = 3$ , so the parameter vector to be learned is of dimension  $m = 10$ . We have

$$\begin{aligned} \mathbf{x} &= (x_1 \ x_2 \ x_3)^T, \\ \mathbf{f}_3(\mathbf{x}) &= (x_1^2 \ x_2^2 \ x_3^2 \ x_1 x_2 \ x_2 x_3 \ x_1 x_3 \ x_1 \ x_2 \ x_3 \ 1)^T, \\ \mathbf{w} &= (w_1 \ w_2 \ \dots \ w_{10})^T, \end{aligned}$$

where we demand  $w_{10} = 1$  during the least squares fitting of  $\mathbf{w}$ . Coefficients shown in the section on experiments will always be in the order  $w_1, w_2, \dots, w_9$ . Since  $w_{10} = 1$  by construction, this coefficient will be omitted.

## III. DIFFERENT BODY MORPHOLOGIES

For the investigation of the QRENS' properties, the two robots SEMNI and Myon have been used, which significantly differ in size, mass, actuation, and overall morphology. They will briefly be described within the following subsections.

Despite their different properties, they are equipped with the same data interface, so data acquisition and experimental setting could be identical for both hardware platforms. Thus, it can be excluded that resulting body models are distorted systematically due to platform-dependent data quality (e.g. noise, resolution, sampling rate). Sensorimotor loops have always been guaranteed to run tightly at a rate of 100 Hz.

### A. The Self-Exploring Robot SEMNI

The acronym SEMNI stands for *Self-Exploring Multi-Neural Individual* – and that is exactly the purpose the robot has been built for. It possesses only two degrees of freedom: one revolute joint at the hip, and another one at the knee, as can be seen in figure 2.

Proprioceptive sensors continuously measure the current and temperature of each actuator, the angular positions of both joints and the acceleration forces within the robot's mid-sagittal plane, relative to the center of the printed circuit board in the head. Thus, there are two motor values and eight sensor values per 10 ms time slot.

The actuators can be controlled in various ways. For the experiments reported here, we choose a constant velocity paradigm to always keep the robot in motion at a moderate speed. This way, we can omit the robot’s velocity vector and restrict ourselves to the analysis of the manifold

$$M_r = \{(\varphi_h, \varphi_k, \varphi_b) \in \mathbb{R}^3 \mid \chi_r(\varphi_h, \varphi_k, \varphi_b) = 1\}, \quad (6)$$

where  $\varphi_h, \varphi_k$  and  $\varphi_b$  are the angular positions of the hip joint, knee joint and the robot’s body, the latter being measured relative to the horizontal ground surface. Due to the moderately slow motion, the horizontal and vertical acceleration forces  $F_h$  and  $F_v$  are dominated by the earth’s gravitational force, so the body position can be calculated as  $\varphi_b = \arctan(F_v/F_h)$ .

The characteristic function  $\chi_r$  equals one, iff its argument represents a body posture which the robot can reach and hold on its own. Thus, the manifold  $M_r$  is defined by all reachable body positions on a flat ground, without using dynamic motions and excluding transient motions when the robot falls over.

Obviously,  $M_r$  completely depends on the robot’s body shape, moving abilities, and environment. It can thus be regarded as an implicit body model of the robot, situated in a fixed environment. Whenever the robot is at rest, its body posture and position relative to the ground correspond to a point  $\mathbf{p} \in M_r$ . So, at low velocities, we stay within  $M_r$ , whereas dynamic motions temporarily leave  $M_r$ , be they induced by the robot’s motors or by tumbling accidentally.

Having noted this, we are now able to inspect  $M_r$  for SEMNI on a flat ground. Please refer to figure 3 for the following explanation. The manifold has been cut into four parts. We start with the bottom left image, which corresponds to the situations where the robot is lying on its front side (i.e., 90 degrees counter-clockwise from the position shown in figure 2). The  $xy$ -position corresponds to the posture of the leg, with the horizontal position representing the joint angle of the hip, and the vertical position that of the knee.



Figure 2. Left: The 30 cm tall robot SEMNI standing in an upright position. Proprioceptive sensors are located within the actuators (hip, knee) and on the printed circuit board in the head. The robot is facing to the left, with the leg standing on the back side. Right: Head, torso, and right arm of the modular humanoid research robot *Myon*. The black solid line indicates the area which can be reached by the arm. The hand is not attached in this experiment.

The border of the missing corner (top right white part of the image) represents all leg postures where the foot touches the back part of the head. For each  $xy$ -position (i.e., leg posture) the color encodes the body’s angle relative to the ground. The darker the blue, the more the head is near the ground, whereas the darker the red, the more the head is in the air. The pale greyish regions indicate leg postures where the robot’s body is lying on the ground horizontally.

As can be seen, the robot is tilted a bit in one or the other direction, depending on where the barycenter of the leg is, relative to the hip joint. Now, the robot is starting to sit up, if the leg is following a specific trajectory, which is shown in the top left image. Coming from the bottom left and continuing to the top right (in the image), the robot is quickly raising its head and then falling over – either onto its back or back to the front again. This can be seen in the two bottom images by the isolated blue dots which correspond to impacts of the robot’s body. Obviously, the manifold has borders which correspond either to the joints’ end positions, or to unstable body positions. In the former case we are just stuck, whereas in the latter case, we fall off the manifold and back onto it to another place.

The two images on the right side are analogous to the left ones, but describe the robot starting from lying on its back. The darkest blue positions correspond to the robot performing a headstand. Due to the leg length and slow motion, the robot is not able to do a backward somersault on the ground. When speeding up the motion, this is indeed possible.

Summing up, the manifold  $M_r$ , which describes SEMNI’s capabilities during moderately slow motions, is already complex enough to be of interest for building up an implicit body model. As we will see later, only few QRENS are needed to accurately capture  $M_r$ .

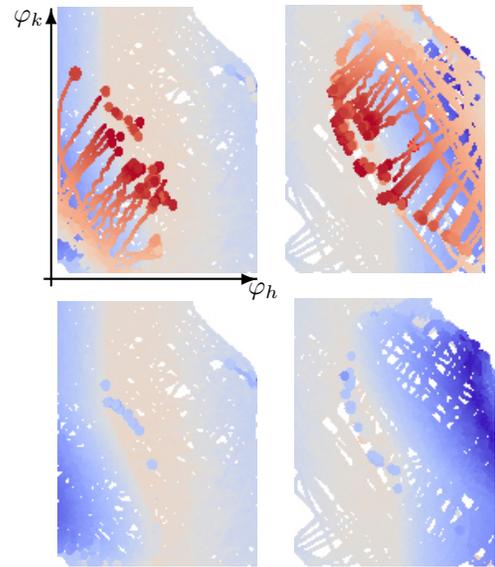


Figure 3. A sampled version of the manifold  $M_r$  of the robot SEMNI, situated on a flat ground. It indicates the body posture ( $xy$ -coordinates) and body position relative to the ground (color-coded). See text for an explanation.

## B. The Humanoid Robot Myon

Since the shape and configuration of SEMNI is rather uncommon, we choose a more standard scenario for additional tests of the QRENS. The humanoid robot *Myon* is a modular research robot, the body parts of which can be detached and reattached during run-time, since they all possess their own processing power and energy supply. Figure 2 shows Myon composed of three body parts, namely the torso, the head, and the right arm. This is the experimental setting we used to record sensory data when the arm was moving, while the end of the arm was touching the table.

## IV. EXPERIMENTS AND RESULTS

In order to put the QRENS to test, we first recorded sensory data of the robot SEMNI while it was touching the ground with both feet. This is a somehow artificial situation, since only part of the poses are stable and we had to hold the robot still in place to get the rest of the data. The result can be seen in figure 4. This is obviously an elliptic hyperboloid of the kind shown in middle of figure 1.

We then attached a QREN to the behavioral primitive *standing-on-the-ground-with-both-feet* by fitting the corresponding weights using a least squares approach. The weights are shown by the crosses in figure 5. To check the quality of the fit, we calculated the QREN's kernel function for the original raw sensory data, sorted the results in descending order, and plotted the result in figure 6. As can be seen, the QREN quite nicely fits the original data – the error remains small, even without having filtered the raw sensor values beforehand.

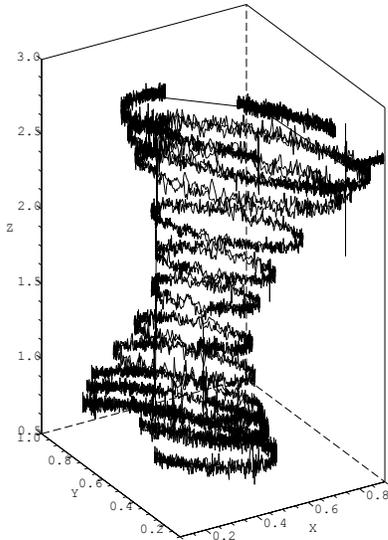


Figure 4. Sensory data of the robot SEMNI which has been recorded while the robot was touching the ground with both feet, like shown in figure 2. The  $xy$ -plane encodes the angular position of the joints in the same way as in figure 3. The  $z$ -axis shows the body's angle relative to the ground, the standing position being at  $z = \pi/2$ .

## A. Using Massively Reduced Data Sets

The weight vector contains nine free parameters, so only nine out of the over 30.000 data values are sufficient to fully specify the QREN. In order to find out how much the weight vector varies depending on the data sample, we randomly picked nine data values and calculated the weights. This has been repeated 50 times and plotted together, as shown in figure 7. Obviously, there is not much variance, since the sampled values are far apart by chance. The subset-dependent weight vector variance is of more behavioral relevance, if the subset is not spread over the whole raw values, but localized. This is equivalent to sampling the stream of sensor values over a short time period, where the pose of the robot does not change too much. We used the lower corner of the hyperbolic ellipsoid, as the robot passes through this part during exploration (see figure 3). The used subset is shown in figure 8 and the result can be found in figure 5 by comparing the crosses (full data set) with the solid dots (localized subset of the data). The differences are almost unnoticeable, which illustrates the QRENS' excellent extrapolation characteristics. This in turn allows for feeding the very first quadric estimates back to the control of exploration – the QRENS can very soon help to steer the direction of exploration.

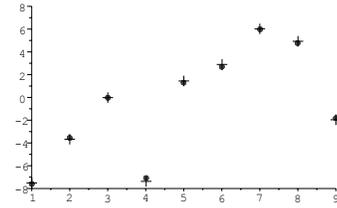


Figure 5. Crosses: weight vector of the elliptic hyperboloid which optimally fits the robot's sensory data. Solid dots: weight vector found by the least squares approach when using only a small subset of the data.

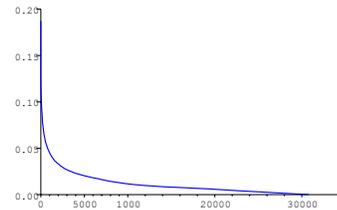


Figure 6. Values of the QREN's kernel function for all original sensor values, sorted in descending order. The error remains small, even without having filtered the data.

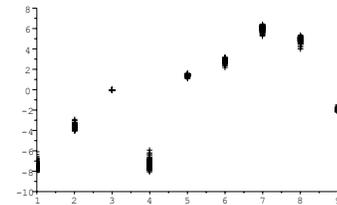


Figure 7. The weight vector does not change very much when only nine random samples from the whole data set are drawn for parameter calculation. The process has been repeated 50 times.

### B. Fitting the Quadric of a Moving Arm

Next, we used the angular data which we recorded using the arm of the humanoid robot Myon, as shown in figure 2. The arm was moving in all directions, while the end of the arm was touching the table all the time.

The raw sensory data is shown in figure 9 and reveals part of an ellipsoid. The corresponding weight vector can be found in figure 10. Although the raw data forms less than one eights of a full ellipsoid, again, the quadric which has been fitted by the QREN is as close to the sensory data as in the former case of the hyperboloid. Due to the limited space, we do not explicitly show the model and the quality of fit here. Interestingly, we also got an ellipsoid when used the acceleration sensors instead of the angular ones.

### C. Quadric-Based Movements

Since QRENS represent quadrics that encode an implicit body model of a robot given in a specific environment, they are of behavioral relevance. As it turns out, the weight vector of a QREN can be used in a straightforward manner to purposefully control a robot's motion.

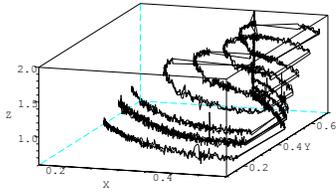


Figure 8. Localized subset of the full sensory data which has been used to test the QRENS' extrapolation abilities.

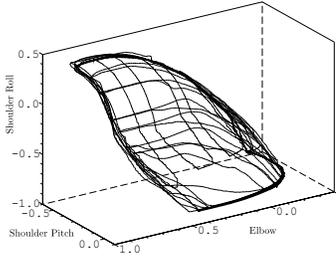


Figure 9. Raw angular sensor data of the three joints of Myon's right arm, as shown in figure 2. Obviously, the data can be modeled by an ellipsoid.

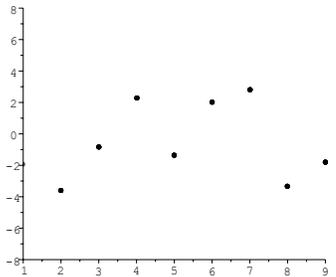


Figure 10. Weight vector of the ellipsoid which best fits the sensor data of Myon's moving arm.

The following example will illustrate this. Say we want the robot SEMNI enable to sit-up from arbitrary starting postures. There are many target positions for the robot to stand upright and they all lie on the hyperboloid shown in figure 4. To be more precise, they satisfy the two conditions:

$$\begin{aligned} \mathbf{w}^T \mathbf{f}_3 \left( (\varphi_h \ \varphi_k \ \varphi_b)^T \right) &= 0, \\ \varphi_b &= \pi/2. \end{aligned} \quad (7)$$

Using the well-known gradient descent, we control the voltage of the hip actuator ( $U_h$ ) and of the knee actuator ( $U_k$ ) as follows:

$$U_h = -\mu \mathbf{w}^T \mathbf{f}_3(\hat{\mathbf{x}}) \frac{d\mathbf{w}^T \mathbf{f}_3}{d\varphi_h}(\hat{\mathbf{x}}), \quad (9)$$

$$U_k = -\mu \mathbf{w}^T \mathbf{f}_3(\hat{\mathbf{x}}) \frac{d\mathbf{w}^T \mathbf{f}_3}{d\varphi_k}(\hat{\mathbf{x}}), \quad (10)$$

where  $\mu$  is a fixed motor constant and

$$\hat{\mathbf{x}} = (\varphi_h \ \varphi_k \ \frac{\pi}{2})^T \quad (11)$$

is the current posture, but with  $\varphi_b$  clamped to the desired target value. Of course this approach can also be used to define arbitrary targets, e.g., stretching the leg ( $\varphi_k = 0$ ).

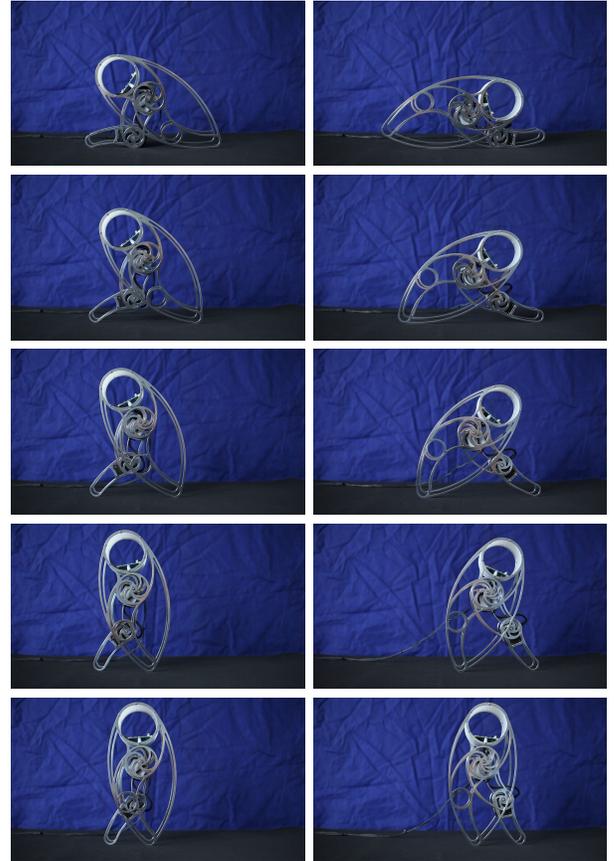


Figure 11. Series of snapshots (top to down) illustrating two different sit-up motions of the robot SEMNI. Left column: Starting with the robot's front side facing the ground. Right column: Starting with the robot's front side looking away from the ground.

For the experimental setup with the robot Myon this would include lifting the arm to arbitrary heights above the table. Of course sensory signals from the vision system need to be recruited during the learning process in first place, but after that the robot is able to reach a target position blindly.

Figure 11 shows the results after implementation of the aforementioned motion control loop. The robot successfully sits up from different starting postures and, as a consequence, also stops in an upright position with different leg postures. The motion trajectories are highly efficient in the sense that as less movements as possible are being executed.

#### D. Speeding Up the Learning Process

As already mentioned above, a standard competitive neural network, where only the winning QREN is updated by the delta rule, will learn the manifold of a robot, like the one shown in figure 3. Using the sensory information in a smart way, one can considerably speed up the learning process. Figure 12 shows the impacts of SEMNI's body. They are derived from the acceleration data. Since we used a moderately slow motion for the exploration of the robot's behavioral manifold, those impacts can only be due to jumps off and back onto the manifold (also see the isolated dots in figure 3). This information can be used to introduce new QRENS in early stages of the learning process.

Another strategy uses the current sensors of each actuator: Whenever a stall current situation is detected, a specific QREN is learned. This QREN will soon anticipate self-harm of the robot and most likely be representing a quadric with two parallel planes, as shown in figure 1, right. Finally, referring to the QREN's excellent extrapolation characteristics, and using the quadric-based motion control we introduced, it is a good idea to explore the manifold QREN by QREN. This reduces the amount of impacts which could potentially harm the robot, and, at the same time, improves the accuracy of the QREN which is currently active.

### V. CONCLUSION AND OUTLOOK

We have formally introduced QRENS and given experimental evidence that they are able to implicitly model different

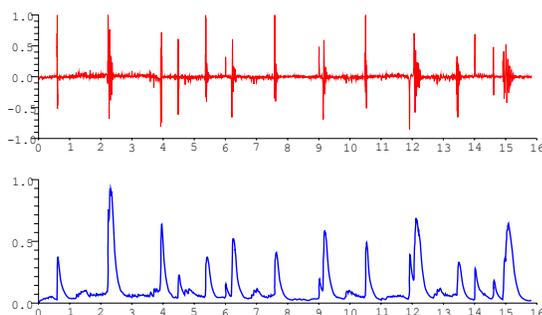


Figure 12. Sensory information of the type shown here can be used to speed up the learning process. Top: Squared sum of the three perpendicular acceleration forces minus the squared static value of the earth's gravitation. Bottom: Smoothed and normalized absolute value of the above curve.

body morphologies and motion capabilities. QRENS can be learned using standard methods, but we also outlined how sensory information can be used to speed up the learning process. For a single QREN first results concerning the dependence on the data set have been presented. They show the QREN's robustness and extrapolating capabilities.

There are two very promising directions, we would finally like to comment on. First, when modeling behavioral manifold by multiple quadrics the latter will intersect. So-called QSICs (Quadric Surfaces Intersection Curves), see [13] for a classification, show up. Those QSICs are behaviorally of special interest, because a robot who will spend most of its time on QSICs will have better options of quickly doing one or the other.

Second, the scalar output of a quadric can serve as a virtual sensor. We intend to use this approach for the modular humanoid robot Myon. If a QREN is able to encode the behavioral condition, that a leg or arm stands upright (like shown with the robot SEMNI), the scalar output of another QREN can encode the height of the limb relative to the ground (like with the arm of Myon). This way, QRENS inside the torso can make use of this virtual sensor information to model *crawling*, *walking*, and the like.

### REFERENCES

- [1] M. R. Longo and P. Haggard, "An implicit body representation underlying human position sense," *Proceedings of the National Academy of Sciences of the United States of America*, 2010.
- [2] J. Lenarcic and P. Wenger, Eds., *Advances in Robot Kinematics: Analysis and Design*. Springer, Netherlands, 2008.
- [3] M. Hild and F. Pasemann, "Self-adjusting ring modules (sarms) for flexible gait pattern generation," in *IJCAI*, 2007, pp. 848–852.
- [4] J. M. Selig, *Geometric Fundamentals of Robotics*. Springer, New York, 2005.
- [5] S. Lee and H. Hahn, "An optimal sensing strategy for recognition and localization of 3D natural quadric objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 10, October 1991.
- [6] G. Cross and A. Zisserman, "Quadric reconstruction from dual-space geometry," in *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*. Washington, DC, USA: IEEE Computer Society, 1998, p. 25.
- [7] M. R. Rahayem and J. A. P. Kjellander, "Quadric segmentation and fitting of data captured by a laser profile scanner mounted on an industrial robot," *The International Journal of Advanced Manufacturing Technology*, 2010.
- [8] G. J. Agin, "Fitting ellipses and general second-order curves," Robotics Institute, Carnegie-Mellon University, Tech. Rep. CMU-RI-TR-81-5, 1981.
- [9] M. Dai and T. S. Newman, "Hyperbolic and parabolic quadric surface fitting algorithms – comparison between the least squares approach and the parameter optimization approach," Computer Science Department, University of Alabama, Huntsville, Tech. Rep. TR-UAH-CS-1998-02, 1998.
- [10] S. Petitjean, "A survey of methods for recovering quadrics in triangle meshes," *ACM Computing Surveys*, vol. 34, no. 2, pp. 211–262, June 2002.
- [11] R. Halir and J. Flusser, "Numerically stable direct least squares fitting of ellipses," 1998. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.1.7559>
- [12] S. Haykin, *Neural Networks and Learning Machines*. Pearson Prentice Hall, New Jersey, 2009.
- [13] C. Tu, W. Wang, B. Mourrain, and J. Wang, "Using signature sequences to classify intersection curves of two quadrics," *Computer Aided Geometric Design*, vol. 26, pp. 317–335, 2009.