# Neural Control and System Identification Using a Similarity Approach

Steffen Brückner* and Stephan Rudolph

Institute for Statics and Dynamics of Aerospace Structures
Universität Stuttgart, Pfaffenwaldring 27, D-70569 Stuttgart, Germany

## ABSTRACT

For nonlinear and adaptive control of smart structures direct and indirect neural network control strategies have been suggested. In indirect neural network control the identified plant models are usually implemented as black-box neural networks using no a priori knowledge. Designing a neural network for system identification using dimensional analysis results in neural networks, where in contrary to black-box solutions no dimensionally inhomogeneous states can occur. Furthermore, the generalization and learning properties of neural networks designed using dimensional analysis are usually improved compared to conventional black-box networks. This work describes a technique of using neural networks for system identification and control, where the neural network has been constructed according to a dimensional analysis of the governing equations.

**Keywords:** Neural Control, Similarity Theory, Pi-Theorem, Dimensional Analysis

## 1. INTRODUCTION

Neural networks have been proposed in the literature[1-4] for nonlinear and time-variant control problems as arise frequently in smart structure control. By using neural networks for control, the need for an explicit plant model is avoided, but the black-box character of this artificial intelligence technique does not satisfy the need of the engineers to a posteriori understand the behaviour of the system as a function of the system parameters.

Dimensional analysis on the other hand is a powerful engineering tool, which reduces the dimensionality of the problem using only minimal a priori knowledge. Applying dimensional analysis to the system identification problem leads to smaller neural network sizes, due to the simpler structure of the dimensionless relation compared to the dimensional relation, and usually also to better learning properties. Neural networks designed using dimensional analysis have well-defined input and output layers, for this reason, the black-box character of the neural networks can be partially lifted and is restricted to the hidden layers.

At the University of Stuttgart, the Collaborative Research Project "Smart Structures (SFB 409)" investigates actor and sensor integration, new design methodologies for smart structures and new control strategies for smart structure applications. Inside this research project, dimensionally homogeneous neural networks are investigated as controllers for nonlinear and time-variant systems, such as for a space tether deployment system, and for the control of helicopter rotor blades with active noise reduction. In this context, the technique of dimensional analysis is discussed first. Then, a short introduction into neural networks follows. After this the principles of neural networks for control are shown and the design of a neural network for the identification of a space tether system is used as an example for the implemented indirect neural network control strategy.

## 2. DIMENSIONAL MODELLING

A major concept of physical relationships is the principle of dimensional homogeneity. Based on this principle, the systematic method of dimensional analysis allows to simplify dimensional homogeneous equations by transforming the physical variables into a smaller number of dimensionless groups describing the same physical relationship. The equations written in terms of these dimensionless groups possess the advantage of having less parameters and of being scale-invariant. This means that one solution of the dimensionless equation(s) is equally valid for an infinite number of completely similar but distinct physical states.

*Correspondence: Email: pigroup@isd.uni-stuttgart.de; WWW: http://www.isd.uni-stuttgart.de/,
Phone: +49 711 685 3799, Fax: +49 711 685 3706



In *Smart Structures and Materials 2000: Mathematics and Control in Smart Structures*,
Vasundara V. Varadan, Editor, Proceedings of SPIE Vol. 3984 (2000) • 0277-786X/00/$15.00

The method of dimensional analysis is briefly described here, a more detailed description and additional examples of the application of dimensional analysis in different domains can be found in Szirtes[5] or others.[6]

Suppose a physical variable $x_j$ is given as $x_j = \{value\}\,[dimension] = \{x_j\}\,[x_j]$. The $[dimension]$ is a monomial combination of the chosen base units (e.g. the 7 SI base units, or units in another coherent unit system)

$$[x_j] = \prod_{i=1}^{7} d_i^{e_{ij}} = kg^{e_{1j}}\, m^{e_{2j}}\, s^{e_{3j}}\, A^{e_{4j}}\, cd^{e_{5j}}\, mol^{e_{6j}}\, I^{e_{7j}} \tag{1}$$

The exponents $e_{ij}$ of the dimensions $d_i$ of the variables $x_j$ involved are written in the so-called dimensional matrix $E = e_{ij}$, where the $d_i$ are the base units of the chosen system of units

$$E \rightarrow \begin{array}{c|cccc} & x_1 & x_2 & x_3 & \cdots \\ \hline d_1 & e_{11} & e_{12} & e_{13} & \cdots \\ d_2 & e_{21} & e_{22} & e_{23} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{array} \tag{2}$$

For a variable containing a moment, which is measured in the SI units "Newton meter" $[N\ m] = \left[kg^1\, m^2\, s^{-2}\right]$, the corresponding column in the dimensional matrix would be $x_i = [1,\ 2,\ -2,\ 0,\ 0,\ 0,\ 0]^T$ if the first row corresponds to mass in $[kg]$, the second row to length in $[m]$, and the third row to time in $[s]$. For the construction of the dimensional matrix, only occupied rows (i.e. dimensions) remain in the matrix, so in mechanical problems usually only the three rows for mass, length, and time appear. Although the choice of the base dimensions is basically free as long as the set of dimensions is consistent, usually the seven SI base units are used for their consistency and for convenience.

In order to find the dimensionless groups, the dimensional matrix is partitioned according to Szirtes[5] in a matrix $A$ containing the rightmost *rank(E)* columns of the dimensional matrix $E$. The submatrices $A$ and $B$ are defined as shown in equation (3), $B$ being the remaining columns of $E$ not included in $A$.

$$E = [\ B \mid A\ ] \tag{3}$$

The submatrix $A$ is the largest rightmost, not singular determinant of the dimensional matrix. Hence $A$ is always a square matrix. Now creating the dimensional set leads to the following matrix

$$\begin{array}{c|cc} & x_j & \\ \hline d_i & B & A \\ \hline \pi_k & D & C \end{array} \tag{4}$$

When the submatrix $D$ is chosen to be a unity matrix, the submatrix $C$ can be calculated, such that

$$C = -\left(A^{-1} \cdot B\right)^T \tag{5}$$

The values in the $C$ and $D$ matrices are the exponents of the $n$ basis variables $x_i$ forming the $m = n - rank(A)$ dimensionless groups.

$$P = [\ D \mid C\ ] \tag{6}$$

$$\pi_k = \prod_{j=1}^{n} x_j^{P_{kj}} \qquad k = 1; \ldots; n - rank(A) \tag{7}$$

Using this scheme, a set of dimensionless similarity parameters $\pi_k$ can be obtained where the number $m$ of dimensionless groups equals the number $n$ of physical variables $x_j$ reduced by the rank of the sub-matrix $A$. This usually corresponds to the number of independent dimensions $d_i$ involved. The complexity of the functional relation to be identified is therefore substantially reduced.

As a simple example, dimensional analysis is applied to a second-order linear equation of motion as commonly used in structural mechanics

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = \hat{p}f(t) \tag{8}$$

The relevance list for this problem is given by the variables appearing in this equation

| Parameter | Symbol | Dimension |
|---|---|---|
| mass | m | $[kg]$ |
| damping | c | $[kg/s]$ |
| stiffness | k | $[kg/s^2]$ |
| max. ext. force | $\hat{p}$ | $[kg\ m/s^2]$ |
| ext. force | $f$ | $[-]$ |
| position | $x$ | $[m]$ |
| time | $t$ | $[s]$ |

The derivatives $\dot{x}$ and $\ddot{x}$ are taken into account by including the position $x$ and the time $t$ in the relevance list. The dimensional matrix $E$ is then given by

$$E \rightarrow \begin{array}{c|ccccccc} & f & c & \hat{p} & t & k & m & x \\ \hline kg & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ m & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ s & 0 & -1 & -2 & 1 & -2 & 0 & 0 \end{array} \tag{9}$$

This dimensional matrix $E$ is then partitioned into the largest rightmost determinant $A$ and the remaining matrix $B$. The rank of the dimensional matrix is $\text{rank}(E) = 3$, so the largest rightmost determinant $A$ is given by the first three columns from the right. $B$ is formed of the remaining four columns

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & -2 & 1 \end{bmatrix} \tag{10}$$

The four rows for the dimensionless variables $\pi_1, \ldots, \pi_4$ are be generated, defining the matrix $D$ as the identity matrix of size $4 \times 4$. The $C$-matrix $C = -(A^{-1} \cdot B)^T$ is then given by

$$C = \begin{bmatrix} 0 & 0 & 0 \\ -1/2 & -1/2 & 0 \\ -1 & 0 & -1 \\ 1/2 & -1/2 & 0 \end{bmatrix} \tag{11}$$

This leads to the dimensional set combining the submatrices $A$,$B$,$C$, and $D$ according to equation (8).

$$\begin{bmatrix} E \\ P \end{bmatrix} \rightarrow \begin{array}{c|ccccccc} & f & c & \hat{p} & t & k & m & x \\ \hline kg & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ m & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ s & 0 & -1 & -2 & 1 & -2 & 0 & 0 \\ \hline \pi_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \pi_2 & 0 & 1 & 0 & 0 & -1/2 & -1/2 & 0 \\ \pi_3 & 0 & 0 & 1 & 0 & -1 & 0 & -1 \\ \pi_4 & 0 & 0 & 0 & 1 & 1/2 & -1/2 & 0 \end{array} \tag{12}$$

The dimensionless groups can now be taken from the $\pi_i$ rows of this dimensional set

$$\pi_1 = f \qquad \pi_2 = \sqrt{\frac{c^2}{k \cdot m}} = 2D \qquad \pi_3 = \frac{\hat{p}}{k \cdot x} = \frac{1}{\xi} \qquad \pi_4 = \sqrt{\frac{k \cdot t^2}{m}} = \omega_0 \cdot t = \tau \tag{13}$$

The equation of motion equation (8) can then be written in terms of these dimensionless groups which leads to the formulation in equation (14), which is well known from structural mechanics

$$\xi_{,\tau\tau} + 2D\xi_{,\tau} + \xi = f(\tau/\omega_o) \qquad ; \quad (\bullet)_{,\tau} = \frac{\partial}{\partial \tau}(\bullet) \tag{14}$$

As can be easily seen, this formulation has less parameters than the dimensional formulation equation (8). It is therefore much easier to use this dimensionless formulation for system identification. Another major advantage of this dimensionless formulation in equation (14) is that the attenuation constant $D$ and the eigenfrequency $\omega_0$ of the system can be easily determined from simple experimental plots.

## 3. NEURAL NETWORKS

Neural networks are an artificial intelligence technique where the input-output relations of the networks are trained using example data rather than being programmed in a top-down manner, as is the case with traditional sequential computer programming. This learning capability of neural networks makes them of special interest in non-linear system identification and control.

Neural networks are built from neurons, which are the computational atoms, and which are interconnected with each other using weighted connections. The information is propagated from each neuron's output to a number of other neurons' inputs. The neurons are arranged in layers, and if there are only connections from neurons to neurons in the next layer, the network is called a feedforward neural network. In case where there are feedback connections, the network is called a recurrent neural network.

The neurons themselves consist of three parts. First the incoming information is summarized using an integration function. The result of the integration function is then activated using the activation function. The activation is then propagated through an output function. Usually sigmoidal functions, or transcendental functions, such as sin, cos, hyperbolic tangent tanh, or linear functions are used as activation functions.[7] The output function is usually assumed to be the unity function.

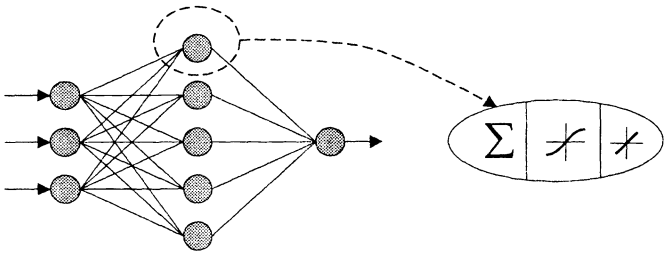



**Figure 1.** The structure of an artificial feedforward neural network (ANN)

Recurrent neural networks often suffer from instability and long training and recall times. These problems are not present in feed-forward neural networks. Considering these advantages of feed-forward neural networks only this type of neural networks is used in this work. The general structure of an feed-forward neural network with three input neurons, five hidden neurons in one hidden layer, and one output neuron is shown in figure 1.

The training algorithms for the iterative adjustment of the neural network weights according to the training data are based upon optimization methods. Backpropagation as an implementation of steepest descent is the best known of these training algorithms. In recent years several other algorithms like Levenberg-Marquardt and Threshold Accepting have gained broader attention, since they are more sophisticated and also available in standard toolboxes.[8]

Two different training paradigms are known, on- and offline learning. While in online learning the network parameters are adjusted after each training pattern, the weight adjustment in offline learning is done after all patterns have been presented to the network. Offline learning is known to be more stable and shows better convergence than online learning, whereas online learning has the advantage that not all patterns have to be presented to adjust the network. In the context of neural network system identification and control, offline learning is used for the first identification of the neural networks. During operation it is switched to online learning to ensure the adaptivity of the neural network controller to environmental changes and so forth.

## 4. NEURAL CONTROL

Two different neural network control schemes have been proposed in the literature, the direct and the indirect control scheme. Whereas the direct neural control uses no model of the plant, the indirect neural network control makes use of an identified (neural) model of the plant. The direct inverse control, as shown in figure 2(a) is known to suffer from stability problems. More often, indirect neural control is used. Indirect training of a plant inverse usually results in a particular plant inverse, even if the input-output behaviour of the plant is not one-one.[9] Other indirect

neural control methodologies, such as model reference adaptive control, internal model control, or predictive control are also commonly used, an overview of these neural network control methodologies is given in Hunt et al.[9]

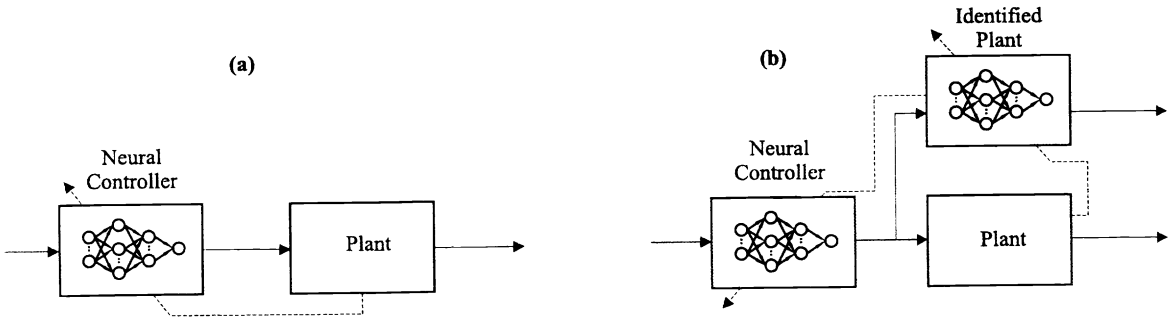



**Figure 2.** Neural network control schemes: (a) direct and (b) indirect neural control

In indirect neural control, the plant model has to emulate the dynamic behaviour of the plant. This can be accomplished using recurrent dynamic neural networks. This approach has several drawbacks, such as extremely slow learning and problems of instability. Using feedforward neural networks however, these problems can be avoided. Feedforward neural networks provide fast learning and recall, and well-known training algorithms, such as backpropagation and Levenberg-Marquardt, are available and known to be stable. The dynamic behaviour of the plant is modelled using external feedback lines and delayed values of the neural network input as shown in figure 3.

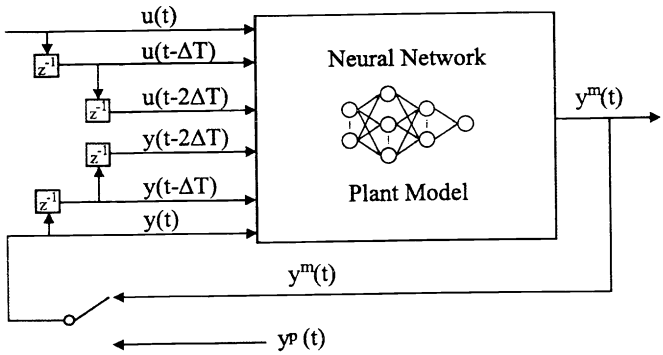



**Figure 3.** Neural network for system identification using external feedback, either actual plant output $y^p(t)$ or neural model output $y^m(t)$

These feedforward neural networks with external feedback can be written as nonlinear mathematical functions

$$y^m(t) = f\left[y^p(t-1),\ldots,y^p(t-n),u(t-1),\ldots,u(t-m)\right] \quad (15)$$

$$y^m(t) = f\left[y^m(t-1),\ldots,y^m(t-n),u(t-1),\ldots,u(t-m)\right] \quad (16)$$

where equation (15) uses the actual plant output as feedback and equation (16) uses the neural network model output as feedback. Equation (15) is referred to by Narendra[10] as series-parallel model and equation (16) as the parallel model. The feedback of the actual plant output has advantages in stability, whereas the feedback of the model output allows for a use of the model independently of the plant. This has also advantages if the plant output is extremely noisy, because the neural network model usually provides a smoother output that can be used for the controller design.

The neural plant model is trained using the squared error between the actual plant output and the model output as shown in figure 4. Training is stopped when the approximation of the plant is good enough (i.e. $y^m(t) \approx y^p(t)$) and the neural network can be used as a plant model for the training of the controller without the use of the actual system. In online use the actual plant output is used again to ensure better stability properties of the control system.

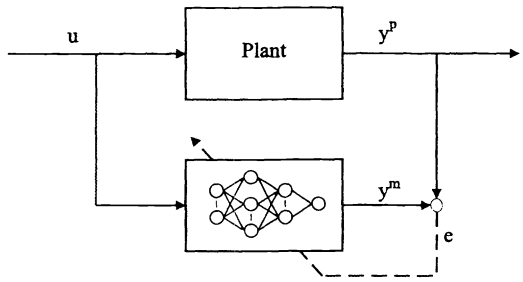



**Figure 4.** System identification using neural networks, training using error between plant and network output

After the successful identification of the plant, the controller is trained using the specialized inverse training scheme[9] as shown in figure 5. The control input $u$ is fed into the plant and the neural plant model. The error $e'$ between the commanded input $w$ and the plant output $y^p$ or the neural network output $y^m$ (in the case $y^m \approx y^p$) is then propagated back through the neural plant model using the first steps of the well-known backpropagation algorithm. The error found for the $u$-input neuron of the neural plant model corresponds directly to the error of the $u$-output neuron of the neural controller. The controller can therefore be trained using this error information with training algorithms, such as backpropagation, threshold accepting, or Levenberg-Marquardt.

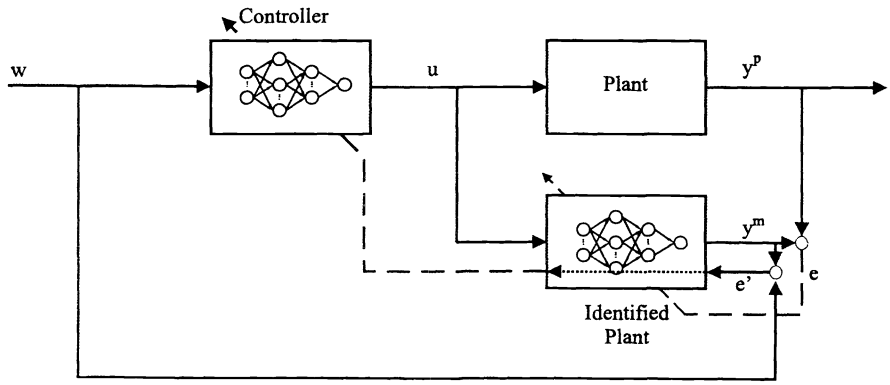



**Figure 5.** Training of the neural network controller using a neural network plant model

The neural network controller is usually structured according to the neural network plant model shown in figure 3 using external feedback of the control signal $u$ and delayed values of the commanded input $w$

$$u(t) = f\left[w(t), \ldots, w(t-n), u(t-1), \ldots, u(t-m)\right] \tag{17}$$

The above described scheme of the neural network controller training is used in the example of the space tether deployment system for re-entry capsules from the International Space Station (ISS) as shown in the following section.

## 5. EXAMPLE: THE SPACE TETHER SYSTEM

The tethered deorbit system for the International Space Station (ISS), as shown in figure 6, is conceived as an inexpensive deorbit system for small reentry capsules.[11] The reentry capsule is deployed from the space station until a tether length of approximately 30 km is reached. Then the tether is cut and the capsule is brought into a deorbit trajectory. This is accomplished by orbit mechanics. The ISS-capsule system orbits with the orbit velocity of the joint center of mass. When the tether is cut, the space station is too slow for its orbit and the capsule is too fast. Therefore, the space station's orbit is increased and the capsule's orbit is decreased. If the decrease of the capsule's orbit is large enough, then the capsule gets on a reentry trajectory, reaching inside the earth's atmosphere at perigee. The perigee reduction of the capsule's orbit can additionally be influenced e.g. by choosing between a static or dynamic tether release maneuver.

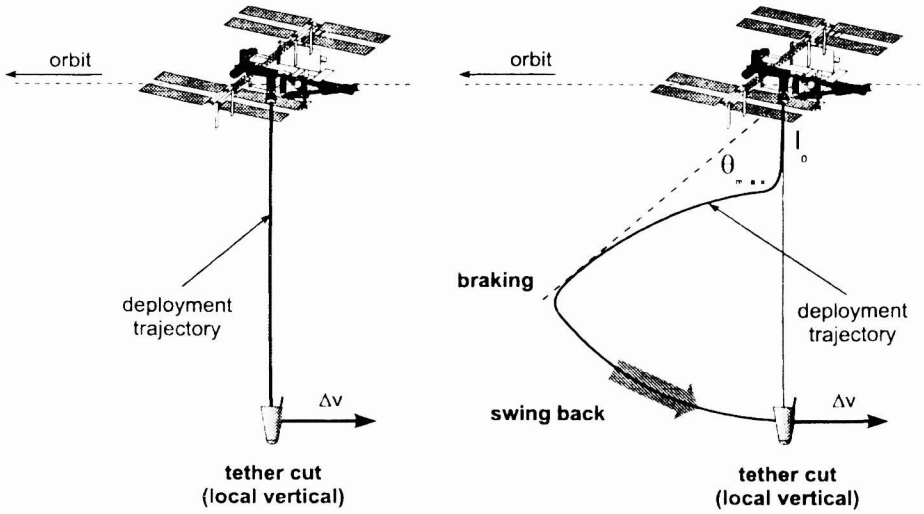



**Figure 6.** Tether-assisted deorbit of a small capsule from the International Space Station (ISS). Left: static maneuver, Right: dynamic maneuver

Using a static tether release, i.e. deploying the tether slowly and cutting the tether while in the equilibrium state, the perigee reduction is

$$r_0 - r_p \approx 7l \tag{18}$$

The dynamic release uses a defined deployment procedure and the tether is cut while swinging back through the local vertical. Compared to the static release, the perigee reduction can be substantially increased to

$$r_0 - r_p \approx 7l + 4\sqrt{3} \cdot l \cdot \sin\theta_{max} \tag{19}$$

For this reason, the dynamic release, shown in figure 6, is preferred. To achieve a dynamic release, the deployment procedure has to be controlled, by using a brake inside the spool mechanism. The quality of the control of the deployment procedure is crucial to ensure a given landing area of the capsule.[12] The reduced set of equations of motion for in-plane movement during the tether deployment starting with an initial tether length of $l_0 \approx 1\ km$ until the tether release with an final length of $l_{final} \approx 30\ km$ are given by[13]

$$\begin{aligned} \ddot{\theta} &= -3\Omega^2 \sin\theta\cos\theta - 2\frac{\dot{l}}{l}(\dot{\theta} - \Omega) \\ \ddot{l} &= -\frac{F_B}{m_1} + l(\dot{\theta}^2 - 2\dot{\theta}\Omega + 3\Omega^2\cos^2\theta) \end{aligned} \tag{20}$$

The inital values are

$$\theta(0) = \theta_0 \qquad \dot{\theta}(0) = \left(\frac{\partial\theta}{\partial t}\right)_0 \qquad l(0) = l_0 \qquad \dot{l}(0) = \left(\frac{\partial l}{\partial t}\right)_0 \tag{21}$$

The relevance list for this set of equations is given by

| Parameter | Symbol | Dimension |
|---|---|---|
| initial angle | $\theta_0$ | $[-]$ |
| in-plane angle | $\theta$ | $[-]$ |
| braking force | $F$ | $[kg\ m\ s^{-2}]$ |
| ISS angular velocity | $\Omega$ | $[s^{-1}]$ |
| mass of capsule | $m$ | $[kg]$ |
| tether length | $l$ | $[m]$ |
| initial tether length | $l_0$ | $[m]$ |
| time | $t$ | $[s]$ |

The parameters $\theta_0$ and $l_0$ have to be included in the relevance list since inital values are an important part of the differential equations.

Applying dimensional analysis to this tether deployment problem leads to the dimensional matrix $E$ and calculating the matrix $C = -(A^{-1} \cdot B)^T$, the dimensional set can be established

$$\begin{bmatrix} E \\ P \end{bmatrix} \rightarrow \begin{array}{c|cccccccc} & \theta & l & F & t & \theta_0 & m & l_0 & \Omega \\ \hline kg & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ m & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ s & 0 & 0 & -2 & -1 & 0 & 0 & 0 & -1 \\ \hline \pi_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \pi_2 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ \pi_3 & 0 & 0 & 1 & 0 & 0 & -1 & -1 & -2 \\ \pi_4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \pi_5 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array} \tag{22}$$

From this dimensional set the dimensionless groups $\pi_1 \ldots \pi_5$ can be found

$$\pi_1 = \theta \qquad \pi_2 = \frac{l}{l_0} \qquad \pi_3 = \frac{F}{m \cdot \Omega^2 \cdot l_0} \qquad \pi_4 = \Omega \cdot t = \tau \qquad \pi_5 = \theta_0 \tag{23}$$

The equations of motion (equation (20)) can then easily be transformed into the dimensionless notation where $\dot{\pi}$ denotes differentiation with respect to dimensionless time $\tau$ instead of dimensional time $t$

$$\begin{aligned} \ddot{\pi}_1 &= -3\sin\pi_1\cos\pi_1 - 2\frac{\dot{\pi}_2}{\pi_2}\left(\dot{\pi}_1 - 1\right) \\ \ddot{\pi}_2 &= -\pi_3 + \pi_2 \cdot \left(\dot{\pi}_1^2 - 2\dot{\pi}_1 + 3\cos^2\pi_1\right) \end{aligned} \tag{24}$$

with the initial values

$$\pi_1(0) = \pi_5 \qquad \dot{\pi}_1(0) = \left(\frac{\partial\pi_1}{\partial\tau}\right)_{\tau=0} \qquad \pi_2(0) = 1 \qquad \dot{\pi}_2(0) = \left(\frac{\partial\pi_2}{\partial\tau}\right)_{\tau=0} \tag{25}$$

Although the overall structure of the differential equations is not affected by dimensional analysis, the number of parameters involved has been reduced from 8 physical parameters to 5 dimensionless parameters.

The system equation (24) has been simulated using a given braking force profile. This optimized braking profile has been taken from Zimmermann,[14] where it has been found for the above equations of motion, using the cost function

$$J = \int_{\tau_0}^{\tau_1} \pi_3^2 \, d\tau \tag{26}$$

under the restriction to ensure stability

$$\frac{1}{\pi_2} \frac{\partial \pi_2}{\partial \tau} \leq 0.75 \qquad \pi_2, \dot{\pi}_2 > 0 \tag{27}$$

The orbit velocity $\Omega$ is given by $\Omega = \sqrt{\mu / r_0^3} = 0.0011 \frac{1}{s}$ (with $\mu = 3.986005 \cdot 10^{14} \frac{m^3}{s^2}$ and $r_0 = 6771000\, m$), the initial tether length is chosen to be $l_0 = 1000\, m$, and the mass of the reentry capsule is assumed to be $m = 170\, kg$. The 4 states of the system are drawn against dimensionless time $\tau = \Omega \cdot t$ and are shown in figure 7. The initial values of the simulation of the dimensionless differential equations are $\pi_1(0) = 0$, $\dot{\pi}_1(0) = 0$, $\pi_2(0) = 1$, and $\dot{\pi}_2(0) = 0$.

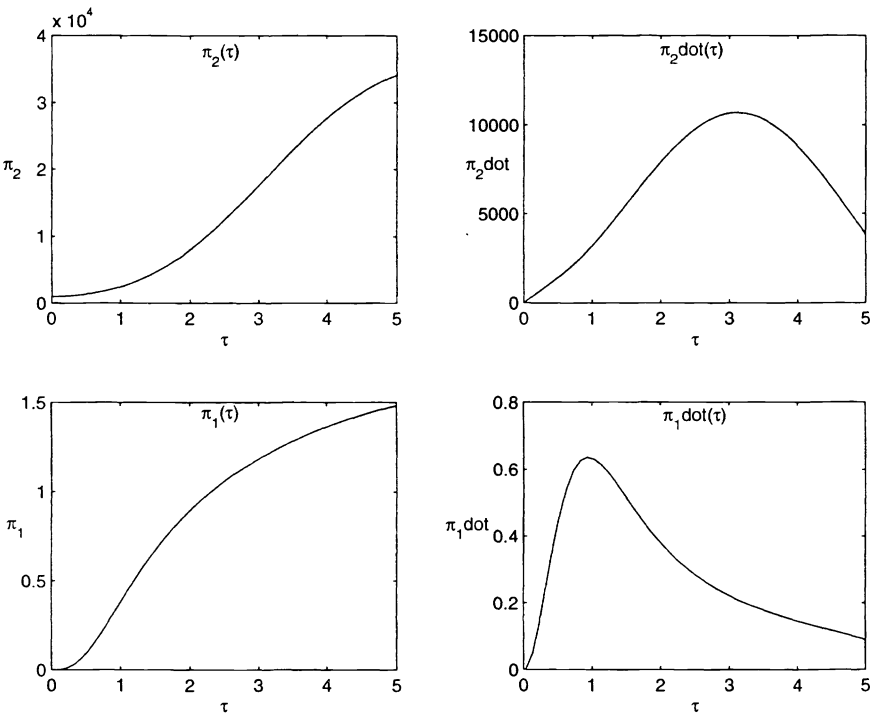
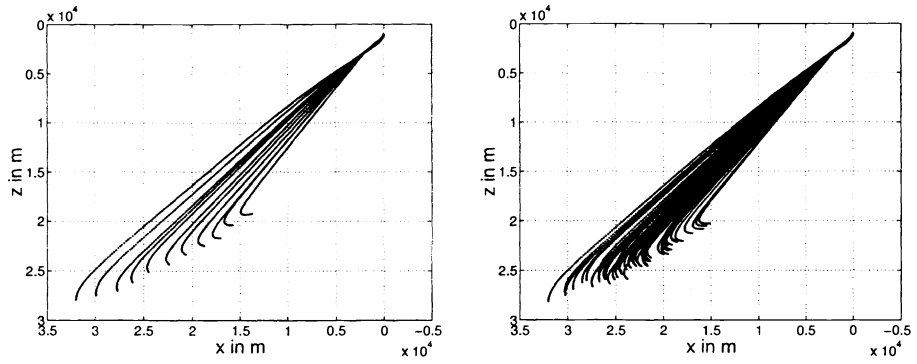



**Figure 7.** Tether states $\pi_1, \pi_2, \dot{\pi}_1, \dot{\pi}_2$ drawn against dimensionless time $\tau$

This system has been trained into a feedforward neural network with external feedback representing the nonlinear relationship for the external input $\pi_3$ and the state vector $\underline{x} = [\pi_1, \pi_2, \dot{\pi}_1, \dot{\pi}_2]$

$$\underline{x}^*(t) = f\left[\pi_3(t), \ldots, \pi_3(t - n \cdot \Delta t), \underline{x}(t - \Delta t), \ldots, \underline{x}(t - n \cdot \Delta t)\right] \tag{28}$$

where $\underline{x}^*(t)$ is the estimated actual state, depending on the $n$ preceding states and the actual and $n$ preceding inputs $\pi_3 = F/(m \cdot \Omega^2 \cdot l_0)$. The sampling frequency used is $f_s = 1/\Delta t$. The training data consist of 11 different perturbed deployment paths (selected out of 50 simulations). The validation data were generated simulating deployment paths using different sampling times. Figure 8 shows the training data (on the left) and the validation data (on the right).




training data validation data

**Figure 8.** The deployment paths used for training and validation of the neural network

In order to create training data for the neural network plant model, the initial values were disturbed. The disturbances were chosen to be within a reasonable magnitude, as indicated in the following table.

| **Parameter** | **Perturbation** |
|---|---|
| in-plane angle $\theta_0$ | $\pm 2.5°$ |
| initial tether length $l_0$ | $\pm 20\ m$ |
| tether angular velocity $\dot{\theta}_0$ | $\pm 0.005°/s$ |
| initial tether velocity $\dot{l}_0$ | $\pm 0.05\ m/s$ |
| braking force | $\pm 10\%$ |

In figure 9 a zoomed view onto the training and validation data is given. The neural network model is used to estimate the next state $\underline{x}^*$ ("Output") using the last $n$ states. In figure 9 a neural network using the last three states $(\underline{x}(t-\Delta t), \underline{x}(t-2\Delta t), \underline{x}(t-3\Delta t))$ to estimate the next state $(\underline{x}^*(t))$ is shown, the fourth dashed line represents possible additional states $\underline{x}(t-n\Delta t)$ that could be used as additional inputs. Numerical simulations[15] showed that in the case of the tether deployment system the use of two preceding states $\underline{x}(t-\Delta t), \underline{x}(t-2\Delta t)$ is sufficient for the prediction of the actual state $\underline{x}^*(t)$.

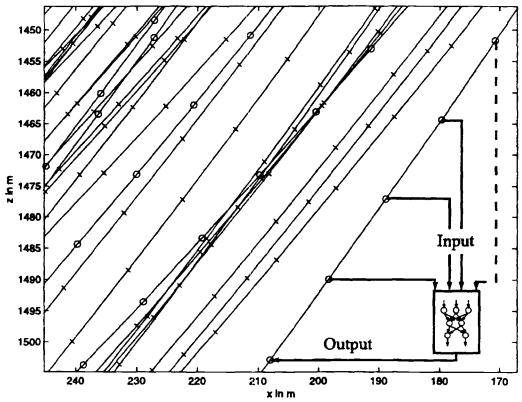



**Figure 9.** Zoomed view of the training and validation data

The results of the identification using a nonlinear feedforward neural network with 11 input neurons, four output neurons, and 10 hidden neurons are shown in figure 10. The input and output neurons used the identity function as activation function, the hidden neurons use a sigmoidal activation function. For the plot (fig. 10) the data have been transformed back from dimensionless coordinates $\pi_1$ and $\pi_2$ into dimensional coordinates $\theta$ and $l$.

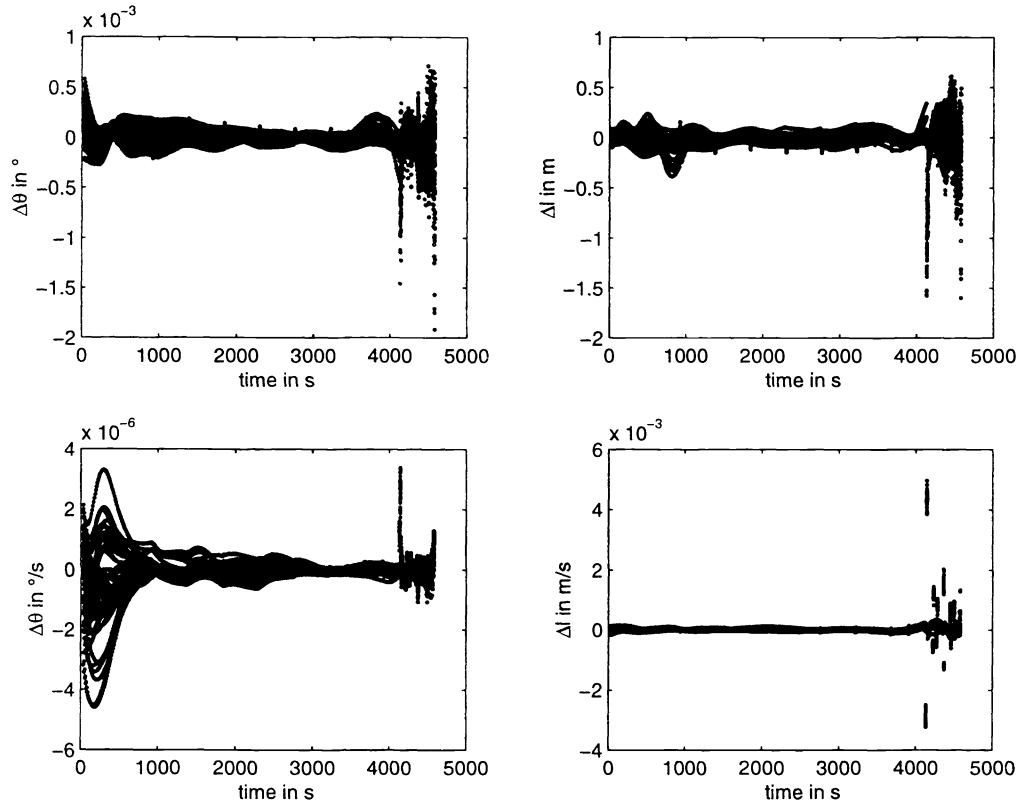



**Figure 10.** Errors in the network output for a nonlinear two-layer neural network with external feedback used to identify the tether deployment system (50 perturbed deployment paths are drawn to show the error envelope)

## 6. SUMMARY

It has been shown that the equations of motion of a tethered deorbit system for the International Space Station ISS can successfully be transformed into a dimensionless formulation using dimensional analysis. The number of parameters involved in the formulation has been substantially reduced in the dimensionless formulation compared to the dimensional relation. The dimensionless formulation and the dimensionless groups allow for an easy scaling of optimal deployment trajectories onto different orbit radii ($r_0$) or different masses of the reentry-capsules $m$. This is simply done by keeping the optimal curves of the $\pi_i$ constant and by transforming these optimal trajectories back onto the different physical realizations. The neural network designed using dimensional analysis provides a good plant model that can successfully be used for the training of a neural network controller for the tether deployment system. The system identification technique and neural network controller design methodology shown in this work can easily be adapted and extended to other smart structure applications. The extension onto distributed control systems can easily be accomplished due to the parallel nature of the neural networks used. The neural network for system identification presented in this work is an further step towards successful future indirect neural network control. The research on indirect neural network controllers for the tether deployment system is still ongoing, the first controllers which have been investigated and simulated show promising results and are continuously improved.

## ACKNOWLEDGMENTS

The financial support of the Deutsche Forschungsgemeinschaft (DFG) in the Collaborative Research Project SFB 409 is acknowledged. The authors want to express their thanks to Dipl.-Ing. Frank Zimmermann and Dr.-Ing. Ulrich Schöttle from the Space Systems Institute of the University of Stuttgart,[11,12,14] for the collaboration on the neural network control of the tether deployment system. Special thanks go to Marc Ueberle for performing many of the simulations.[15]

## REFERENCES

1. G. Irwin, K. Warwick, and K. Hunt, eds., *Neural Network Applications in Control*, Control Engineering Series 53, The Institution of Electrical Engineers, Stevenage, 1995.
2. W. Miller III., R. Sutton, and P. Werbos, eds., *Neural Networks for Control*, MIT Press, Cambridge, MA, 1995.
3. W. Baker and J. Farell, "An introduction to connectionist learning control systems," in *Handbook of Intelligent Control*, D. White and D. Sofge, eds., ch. 2, pp. 35–64, Van Nostrand Reinhold, New York, 1992.
4. A. Barto, "Connectionist learning for control: An overview," in *Neural Networks for Control*, W. Miller, R. Sutton, and P. Werbos, eds., ch. 1, pp. 5–58, MIT Press, Cambridge, MA, 1990.
5. T. Szirtes, *Applied dimensional analysis and modeling*, McGraw-Hill, 1997.
6. W. Baker, P. Westine, and F. Dodge, *Similarity Methods in Engineering Dynamics*, Elsevier, New York, 1991.
7. R. Rojas, *Theorie der neuronalen Netze*, Springer, Berlin, 1996.
8. The MathWorks, Natick, MA, *MATLAB: Neural Network Toolbox*, 1998.
9. K. Hunt, D. Sbarbaro, R. Zbikowski, and P. Gawthrop, "Neural networks for control systems - a survey," *Automatica* **28**(6), pp. 1083–1112, 1992.
10. K. Narendra, "Adaptive control using neural networks," in *Neural Networks for Control*, T. Miller, R. Sutton, and P. Werbos, eds., ch. 5, pp. 115–142, MIT Press, Cambridge, MA, 1990.
11. F. Zimmermann, E. van der Heide, E. Messerschmid, and U. Schöttle, "Application of tethers to space station utilization," in *ESA Symposium Proceedings on Space Station Utilization*, ESOC, ESOC, (Darmstadt, Germany), 1996.
12. F. Farachi, F. Zimmermann, and U. Schöttle, "Impact of a tether-assisted deorbitation on the performance of guided capsule re-entry, DGLR-JT99-203," in *Deutscher Luft- und Raumfahrtkongreß*, DGLR, (Berlin), September 1999.
13. J. Walls and M. Greene, "Adaptive control of an orbiting single tether system," in *Proceedings of the 21st Southeastern Symposium on System Theory*, 1989.
14. F. Zimmermann, U. Schöttle, and E. Messerschmid, "Optimal deployment and return trajectories for a tether-assisted re-rentry mission, AIAA-99-4168," in *AIAA Atmospheric Flight Mechanics Conference*, AIAA, AIAA, (Portland, OR), August 1999.
15. M. Ueberle, "Analyse quasistatischer Neuronaler Netze zur Abbildung der Dynamik orbitaler Seilstrukturen während der Ausspulphase," Master's thesis, Space Systems Institute, Stuttgart University, Stuttgart, April 1999.