

Real-time Modelling of Complex Dynamical Systems

Christfried Webers & Uwe R. Zimmer

Australian National University
Research School for Information Science and Engineering
and the Faculty for Engineering and Information Technology
Autonomous Underwater Robotics Research Group
Canberra, ACT 0200, Australia
christfried.webers@ieee.org | uwe.zimmer@ieee.org

Dynamical systems modelling for physical, real-world systems is a recurring task in many robotics applications. It is often preferable or even required that the creation/adaptation of such a model is performed on-line with the actual stream of sensor samples and also under realistic real-time constraints. This article suggests a new method for real-time modelling of complex dynamical systems, overcoming the 'curse' of dimensionality which applies to any complete modelling system known up to now and which prevented any real-time implementation in even moderately complex environments.

The key concept of the proposed method (RTDSM) is to define a trade-off between the dimensionality of the underlying delay-vector space and local delay-vector statistics (called 'histories'). Combined with locally limited adaptations in each step and the exclusive employment of each sensor sample once and in sampling order, the method recommends itself for on-line and real-time applications, while keeping the modelling power of off-line analysis methods. Restrictions apply, and so the understanding of the trade-off introduced by this method is crucial for its successful application. Simulations and a comparison to an established related method are given.

1. Motivation

Modelling real-world complex dynamical systems is the theoretical foundation which most physical robotics systems are based upon. This task poses a challenge in itself already. Performing this modelling process on-line and under real-time constraints provides yet another level of demanding requirements – but it is also requested by all robotics applications, where continuous and on-line adaptation is essential. Reviewing the dynamical systems modelling literature in this field (see e.g. [5][6] for good starting points) leaves little hope that the complexity of this task can be tackled under real-time constraints at all. So the question what exactly gives us enough theoretical perspectives in order to address this problem, needs to be answered first.

Considering standard dynamical systems modelling methods, two aspects seem to eliminate any chance to find a real-time version. First the usually very high dimensions of the delay-vector spaces [6] imply very high computational costs, usually exploding exponentially with the dimensions. Second and even more striking: since this high dimensional space can only be populated very sparsely with real-world measurements, a large number of measurements need to be employed in order to achieve a reasonable statistical basis for generalization and modelling. Traditionally these measurements are taken off-line (or generated by a simulator) and are presented to the modelling system in a recurring and randomly drawn fashion. Obviously any of those constraints will prevent an on-line implementation.

If the delay-vector space is on the other hand intentionally restricted to a relatively small dimensionality – with the idea to enable the chance for a real-time implementation – risks of losing differentiation capabilities are increased. If the delay-vector space is reduced enough the computational complexity might become reasonable for on-line adaptation and at the same time the relative density of samples in this space is significantly increased without employing more samples.

While the system might now become real-time capable, it might also no longer capture essential properties of the original system. The suggestion for a solution of this dilemma is the central contribution of this article. Instead of increasing the delay-vector space until all aspects of the original system are covered, another way of detecting possible losses of differentiation in a local fashion based on locally stored additional temporal and sequential attributes is proposed. The classical method would be to extend the delay-vector space globally and hence increasing the complexity and sparseness for the whole system, even if this dimensionality is only required in one single spot of the whole dynamical system. This article proposes to overcome this problem by a slight ex-

tension of the system locally, only at this spot – while the rest of the system is untouched.

The key to a useful on-line and real-time capable dynamical systems modelling method lies thus in the right balance between global delay-vector space embedding which covers the majority of ambiguities on the one hand and spontaneous local extensions to the system in order to overcome the small number of remaining ambiguities on the other hand. Furthermore a usable model will need to allow for pure or mostly local adaptations in order to keep computational costs constant in each up-date step, i.e. (mostly) independent of the overall size of the system.

The proposed real-time modelling method RTDSM is introduced below followed by a brief reasoning why all equations determining the system can be implemented on-line and under real-time constraints. Finally the capabilities of the actual implementation with respect to a well-known dynamical system are demonstrated and compared to a classical off-line modelling system (of the same ‘size’, but with a much higher computational complexity).

2. Dynamical systems model: RTDSM

The **original dynamical system** X is assumed to produce discrete samples

$$X_t \in \mathfrak{R}^{\chi_X}; t > 0 \quad (1)$$

at discrete time steps t and a real-time-stamp $\theta_t \in \mathfrak{R}$. The following relation between time step indices and time-stamps follows if the continuity assumption for the real-time-stamps holds

$$\theta_i < \theta_j; \forall i < j \quad (2)$$

This system X is assumed not to be accessible in real-world production modes, but only in simulations.

The sequence of **observations** $Y_t \in \mathfrak{R}^{\chi_Y}$ with their real-time-stamps Y_t^θ is the only accessible information about the dynamical system to be modelled. Although the actual relation between X and Y cannot be known, it might be useful in some cases (see remark 2) to assume this relation as a linear projection from the original space \mathfrak{R}^{χ_X} onto the subspace \mathfrak{R}^{χ_Y} which is spanned by the first χ_Y components e_i of an arbitrary system of linearly independent vectors in \mathfrak{R}^{χ_X} .

$$Y_t = PX_t = \sum_{i=1}^{\chi_Y} (X_t^T e_i) e_i \quad (3)$$

The time-stamp Y_t^θ for the observation Y_t is the same θ_t as the time-stamp X_t^θ for the sample X_t .

Remark 1: Note, that this projection is considered given and fixed by an external real-world system. This differs from the many applications of (3) in pattern recognition which assume access to the original system X , and try to find an optimal subspace according to some minimisation

criteria (like minimum approximation error, least representation entropy, uncorrelated coefficients, maximum variances of the coefficients, maximum separation of two classes in terms of distance in a single subspace, or maximum separation of two classes in terms of approximation errors in two subspaces, statistical independence of the coefficients) [1][4].

Remark 2: For all practical purposes the observations Y_t are considered the only information accessible to a real-world dynamical systems modelling system. Still the fact that in case the projections are all linear and the original system is non-linear, the information from the original system is preserved in its projections [6]. Even if this might not hold completely true in real systems, there is still hope that a number of characteristics of X will still be recognizable in Y .

In order to reconstruct the original state space, a number L of past observations taken at fixed lag times $\tau_i \in \mathfrak{R}$ are stacked together to create a **delay-vector** $Z_t \in \mathfrak{R}^{\chi_Z}$. Obviously, the dimension of the delay-vector is then a multiple of the dimension of the observation vector $\chi_Z = \chi_Y L$. The lag times τ_i are constants (see [5] w.r.t. choice of these constants). In order to avoid capturing the same information from the time-series multiple times, the lag times are all chosen different from each other $\tau_i - \tau_j \neq 0, \forall i, j \in 1 \dots L$ and $i \neq j$.

Without restriction to the generality, we further assume: $\tau_i > \tau_{i-1}; \tau_1 = 0$

The delay-vector Z_t is now defined as

$$Z_t = (Z_t^{\tau_1}, \dots, Z_t^{\tau_L}) \quad (4)$$

where each component $Z_t^{\tau_i}$ is identical to one of the observation vectors $o_u \in O$, such that

$$Z_t^{\tau_i} = o_u |_{\min(|(\theta_t - \tau_i) - \theta_u|)} \quad (5)$$

which addresses the observation vector with its time-stamp closest to $\theta_t - \tau_i$.

With $\tau_1 = 0$ follows that the time-stamp of $Z_t^{\tau_1} = o_t$ is θ_t , which is also assumed to be the time-stamp for the whole delay-vector Z_t .

The series of delay-vectors $Z_t \in \mathfrak{R}^{\chi_Z}$ are used to adapt a **network** structure N_t consisting of a set of cells $C_t = \{c_{i,t}\}$, a set of directed edges $E_t = \{e_{k,t}\}$ where $e_{k,t} = \langle c_{k_1,t}, c_{k_2,t} \rangle$, and a set of parameters Π

$$N_t = (C_t, E_t, \Pi) \quad (6)$$

$c_{i,t}^r \in \mathfrak{R}^{\chi_Z}$ denotes the *representative* for the specific cell $c_{i,t}$, and $c_{i,t}^a \in \mathfrak{R}$ is a measure of the *local adaptation rate* attached to cell $c_{i,t}$. Properties of those two cell attributes and the specification of the edge-set will be introduced below, after the notion of the currently ‘active’ cell and multiple notions of neighbourhood are available.

The initial network is set to $N_1 = (\emptyset, \emptyset, \Pi)$.

For the **metrical distances in the delay-vector** space, the euclidian norm in \mathfrak{X}^{Z} will be used

$$|Z_i, Z_j|_d = |Z_i - Z_j|_{\text{euclid}} = \sqrt{\sum_{k=1}^{Z} (Z_{i_k} - Z_{j_k})^2} \quad (7)$$

where Z_{i_k} is the k -th component of the vector Z_i . Any other norm might be considered here as well.

Given the metric distance measure and an arbitrary time step t , the **set of metrically closest cells** V_t^p can be defined as the cells with a representative $c_{i,t}^r$ inside a sphere of radius ρ around sample Z_t

$$V_t^p = \{c_{i,t} \in C_t \mid |c_{i,t}^r, Z_t|_d \leq \rho\} \quad (8)$$

For each sample Z_t , the set of metrically closest cells V_t^p contains all candidate cells which may approximate the current sample given the granularity ρ of the network.

A **path** $p \in P$ between two cells $c_{i,t}$, and $c_{j,t}$ is specified if $\exists n > 1$ such that

$$p(c_{i,t}, c_{j,t}) = (c_{k_1,t}, \dots, c_{k_n,t}) \quad (9)$$

where

$$\begin{aligned} & (\langle c_{k_l,t}, c_{k_{l+1},t} \rangle \in E_t) \\ & \wedge (c_{k_1,t} = c_{i,t}) \wedge (c_{k_n,t} = c_{j,t}) \\ & \wedge ((a = b) \vee (c_{k_a,t} \neq c_{k_b,t})) \\ & \forall l \in 1, \dots, n-1, \forall a, b \in 1, \dots, n \end{aligned} \quad (10)$$

The edges along a path p are denoted as p^e

$$\begin{aligned} p^e(c_{i,t}, c_{j,t}) &= (e_{l_1,t}, \dots, e_{l_{n-1},t}) \\ \text{with } e_{l_i,t} &= \langle c_{k_{l_i},t}, c_{k_{l_i+1},t} \rangle \end{aligned} \quad (11)$$

If such an n cannot be found, a path $p(c_{i,t}, c_{j,t})$ is specified non-existent, otherwise

$$|p(c_{i,t}, c_{j,t})|_s = n - 1 \quad (12)$$

denotes the *number of (topological) path-steps*, and

$$\begin{aligned} |p(c_{i,t}, c_{j,t})|_m & \\ &= |(c_{k_1,t}, \dots, c_{k_n,t})|_m = \sum_{l=1}^{n-1} |c_{k_l,t}, c_{k_{l+1},t}|_d \end{aligned} \quad (13)$$

denotes the *(metrical) path-length*, while

$$p_{s\text{-min}}(c_{i,t}, c_{j,t}) = (p \in P \mid \forall p_i \in P: |p|_s \leq |p_i|_s) \quad (14)$$

is the *shortest topological path*,

$$p_{m\text{-min}}(c_{i,t}, c_{j,t}) = (p \in P \mid \forall p_i \in P: |p|_m \leq |p_i|_m) \quad (15)$$

is the *shortest metrical path*,

$$|p_{s\text{-min}}(c_{i,t}, c_{j,t})|_s \quad (16)$$

is the *smallest number of (topological) path-steps*, and

$$|p_{m\text{-min}}(c_{i,t}, c_{j,t})|_m \quad (17)$$

is the *shortest metrical path-length*.

The **topological neighbourhood** $N_{i,t}^s$ is given as

$$\begin{aligned} N_{i,t}^s &= \{c_{j,t} \in C_t \mid (\exists p = (c_{i,t}, c_{j,t}) \wedge |p|_s \leq s) \\ &\quad \vee (\exists p = (c_{j,t}, c_{i,t}) \wedge |p|_s \leq s)\} \end{aligned} \quad (18)$$

While the **metrically ordered topological neighbourhood** is given as an ordered set $W_{i,t}^s$

$$\begin{aligned} W_{i,t}^s &= \{c_{j,t} \in N_{i,t}^s \mid \\ &\quad |p_{m\text{-min}}(c_{i,t}, c_{j,t})|_m \leq |p_{m\text{-min}}(c_{i,t}, c_{j+1,t})|_m\} \end{aligned} \quad (19)$$

which is a metrically ordered version of $N_{i,t}^s$.

The (best) **matching cell** b_t denotes the currently 'active' cell in the network.

$$\forall t: \exists b_t \in C_t \quad (20)$$

The selection of this cell b_t is based on a number of dynamical properties, which are defined in the following. The actual specification of b_t will need to be given below for that reason.

The ordered **multi-set of matching cells** is $B_t = \{b_i\}$, $i = 1, \dots, t$

The ordered **multi-set of changing matching cells** R_t is a subset of B_t , where

$$R_t = \{r_1, \dots, r_{m_t}\}; \text{ and } r_i \in C_t \quad (21)$$

All successive elements in R_t are different

$$r_i \neq r_{i+1} \quad \forall i \in 1, \dots, m_t - 1 \quad (22)$$

The following relation between the elements of the matching cell multi-set B_t and R_t exist: All elements of R_t are also elements of B_t , appear in the same order, and denote the first element out of B_t after a change of matching cells over t :

- $\forall r_i \in R_t, r_i \in B_t$, the corresponding element in B_t is b_{m_i} so that $r_i = b_{m_i}$. The time-stamps are identical such that $r_i^0 = \theta_{m_i}$.
- $\forall r_i, r_j \in R_t, i < j$ the indices for the corresponding $b_{m_i}, b_{m_j} \in B_t$ are also ordered: $m_i < m_j$ and
- $\forall r_i \in R_t, r_i \neq r_1$ the corresponding $b_{m_i}, b_{m_{i-1}} \in M_t$ are different $b_{m_i} \neq b_{m_{i-1}}$

The set of **edges** $E_t = \{e_{k,t}\}$ with $e_{k,t} = \langle c_{k_1,t}, c_{k_2,t} \rangle$ can now be specified as

$$\forall e_{k,t} \in E_t, \exists r_{i,t} \in R_t \mid e_{k,t} = \langle r_{i,t}, r_{i+1,t} \rangle \quad (23)$$

i.e. the set of edges represents the observed transitions between cells.

The **trace** $T_t = \{t_1, \dots, t_h\}$ is a subset of R_t with the following properties

$$|T_t| \leq n, \text{ e.g. } h \leq n \quad (24)$$

$$\forall t_i \in T_t: t_i = r_{m_t - i} \quad (25)$$

i.e. the trace consists of the last, maximal h elements of R_t , excluding the very last (current) element r_{m_t} . Therefore the trace T_t represents the immediate short-term history in the sequence of matching cell changes R_t .

Accumulative information, observed in conjunction with a specific cell or edge is represented as **histories**

$$c_{i,t}^H = \left\{ c_{i,t}^{H_1}, \dots, c_{i,t}^{H_n} \right\}, e_{i,t}^H = \left\{ e_{i,t}^{H_1}, \dots, e_{i,t}^{H_n} \right\} \quad (26)$$

where

$$c_{i,t}^{H_k} = \{r_j \in R_t | r_{j+k} = c_i\} \text{ and} \quad (27)$$

$$e_{i,t}^{H_k} = \{r_j \in R_t | (r_{j+k} = c_i) \wedge (e_{i,t} = \langle c_i, c_* \rangle)\} \quad (28)$$

While the above sets represent the actual cells observed as matching units in the immediate neighbourhood ('past') of a specific cell c_i , the function μ counts the number of occurrences of a cell c in the histories of $c_{i,t}$ attached to a specific topological distance k

$$\mu(c, c_{i,t}^{H_k}) = \left| \left\{ c_i \in c_{i,t}^{H_k} | c_i = c \right\} \right| \quad (29)$$

and

$$\mu(c, c_{i,t}^H) = \sum_{k=1}^n \mu(c, c_{i,t}^{H_k}) \quad (30)$$

denotes the number of occurrences of a specific cell c in $c_{i,t}^H$ (analog definitions for μ with respect to $e_{i,t}^H$).

Due to the fact that there is no direct access to the actual dynamical system X in any real-world scenario and the system as described here is thus constructed based on the observable space Y and its delay-vector space Z there is no principal way to avoid projections of uncorrelated parts of X onto the same spaces in Y or even in Z (the delay-vector has a finite length). But there is a way to further reduce ambiguities in the system presented here – without expanding the underlying dimensionality or the delay-vector length. In case that the same cell $c_{i,t}$ is employed in two uncorrelated parts of X then all observed paths through $c_{i,t}$ (as recorded in the associated $e_{i,t}^H$'s) can be divided into a number of disjunct sets of paths.

Two edges $e_k = \langle c_{i,t}, c_* \rangle$, and $e_l = \langle c_{i,t}, c_* \rangle$ originating from the same cell $c_{i,t}$ are defined compatible to each other if

$$\gamma(e_k, e_l) = (\exists c_{j,t} | (c_{j,t} \in e_{i,t}^H) \wedge (c_{j,t} \in e_{i,t}^H)) \quad (31)$$

which means, the histories of the two edges share at least one 'predecessor' cell. In a stricter form one can also request that these histories need to share one immediate predecessor

$$\gamma^1(e_k, e_l) = (\exists c_{j,t} | (c_{j,t} \in e_{i,t}^{H_1}) \wedge (c_{j,t} \in e_{i,t}^{H_1})) \quad (32)$$

The set of compatible edges $\Gamma(e_k)$ for an edge $e_k = \langle c_{i,t}, c_* \rangle$ is then defined as

$$\Gamma(e_k) = \{(e_l = \langle c_{i,t}, c_* \rangle) \in E | \gamma^*(e_k, e_l)\} \quad (33)$$

where γ^* is the transitive closure of the relation γ or γ^1 .

The constraint which forces the system to separate uncorrelated regions in X (the *splitting constraint*) can now be formulated as

$$\forall (e_k = \langle c_{i,t}, c_* \rangle) : \Gamma(e_k) = \{e_k = \langle c_{i,t}, c_* \rangle\} \quad (34)$$

i.e. all edges emerging from $c_{i,t}$ need to be compatible with each other. Alternatively the same constraint can also be formulated as

$$\forall c_{i,t} \in C_t, \forall e_{k,t} = \langle c_{i,t}, c_* \rangle, \forall c_{j,t} \in e_{i,t}^{H_1} : \sum_{e_{l,t} = \langle c_{i,t}, c_* \rangle, l \neq k} \mu(c_{j,t}, e_{l,t}^{H_1}) \neq 0 \quad (35)$$

Enforcing (34) or (35) might require to 'split' an existing cell (hence the name *splitting constraint*) into multiple cells, by dividing an incompatible edge-set into compatible sub-sets and attach those to the new cells such that (34) and (35) will hold for each individual cell.

The **topological distance** $|c_{i,t}, T_t|_t \in [0, 1]$ is defined for each cell $c_{i,t}$ and the trace T_t in the following way

$$|c_{i,t}, T_t|_t = 1 - \frac{1}{S_N} \sum_{c_{j,t} \in T_t} \mu(c_{j,t}, c_{i,t}^H) \quad (36)$$

where S_N is the normalisation

$$S_N = |T_t| \sum_{j=1}^n |H_{j,t}|. \quad (37)$$

At this point the **matching cell** b_t can be defined by the constraint

$$b_t = \begin{cases} c_{j,t} \in V_t^p | |c_{j,t}, T_t|_t \leq |c_{i,t}, T_t|_t, \forall c_{i,t} \in V_t^p ; V_t^p \neq \emptyset \\ c_{*,t+1} \in C_{t+1} | c_{*,t+1}^r = Z_t \wedge c_{*,t+1}^a = \Pi^\alpha ; V_t^p = \emptyset \end{cases} \quad (38)$$

Note that the second case implies that the network is extended by one addition cell

$$c_{*,t+1} \in C_{t+1} \wedge c_{*,t+1} \notin C_t. \quad (39)$$

The **topological neighbourhood** N_t^s around the matching cell b_t is given analog to its general form (18) as

$$N_{i,t}^s = \{c_{j,t} \in C_t | (\exists p = (b_t, c_{j,t}) \wedge |p|_s \leq s) \vee (\exists p = (c_{j,t}, b_t) \wedge |p|_s \leq s)\} \quad (40)$$

The **metrically ordered topological neighbourhood** W_t^s around the matching cell b_t is defined accordingly

$$W_{i,t}^s = \{c_{j,t} \in N_t^s | |p_{m-\min}(b_t, c_{j,t})|_m \leq |p_{m-\min}(b_t, c_{j+1,t})|_m\} \quad (41)$$

which is again a metrically ordered version of N_t^s .

All cells can now be adapted according to their ranks (indices) k in W_t^s , their adaptation rates $c_{k,t}^\alpha$, and the

distance vector between the cell representative $c_{k,t}^r$ and the current delay-vector Z_t

$$\forall c_{k,t} \in W_t^s: \quad (42)$$

$$c_{k,t+1}^r = \begin{cases} c_{k,t}^r + e^{-k/\Pi^\lambda} c_{k,t}^\alpha (Z_t - c_{k,t}^r); V_t^p \neq \emptyset \\ c_{k,t}^r; V_t^p = \emptyset \end{cases}$$

In case that $V_t^p \neq \emptyset$, i.e. the existing network could be employed and adapted in order to cover the current delay-vector Z_t , the adaptation rates $c_{k,t}^\alpha$ are decreased, or otherwise (i.e. the network needed to be expanded) increased according to

$$c_{k,t+1}^\alpha = \begin{cases} c_{k,t}^\alpha \left(1 - e^{-k/\Pi^\lambda} \left(1 - (1/2)^{1/\Pi^h \alpha} \right) \right); V_t^p \neq \emptyset \\ c_{k,t}^\alpha + (\Pi^\alpha - c_{k,t}^\alpha) e^{-\frac{k}{4\Pi^\lambda}}; V_t^p = \emptyset \end{cases} \quad (43)$$

For the sake of prediction the following edge attributes are introduced. First the **number of occurrences of a specific transition** between two cells, connected by $e_{k,t}$ are counted

$$e_{k,t}^c = \{(r_{i,t}, r_{i+1,t}) | r_{i,t} = c_{k_1,t} \wedge r_{i+1,t} = c_{k_2,t}\} \quad (44)$$

$$e_{k,t}^o = |e_{k,t}^c| \quad (45)$$

where

$$e_{k,t} = \langle c_{k_1,t}, c_{k_2,t} \rangle, \text{ and } r_{i,t}, r_{i+1,t} \in R_t. \quad (46)$$

Then the sums of real-time spans taken to traverse this edge is accumulated

$$e_{k,t}^\delta = \sum_{(r_{i,t}, r_{i+1,t}) \in e_{k,t}^c} r_{i+1,t}^\theta - r_{i,t}^\theta \quad (47)$$

The mean time spent while traversing edge $e_{k,t}$ can now be expressed as

$$e_{k,t}^{\bar{\delta}} = e_{k,t}^\delta / e_{k,t}^o \quad (48)$$

The **mean time spent along a path** is thus defined as

$$p^{\bar{\delta}} = \sum_{e \in p} e^{\bar{\delta}} \quad (49)$$

3. Prediction

Given the observation Y_t (in on-line real-time systems, this is the *current* observation), with its time-stamp Y_t^θ , prediction gives an observation $\hat{Y}_t^{\theta+\delta_p}$ with $\delta_p > 0$, employing the Network N_t at time N_t^θ . Since all network structures are based on the delay-vectors Z rather than the actual observations, the prediction process produces a future delay-vector $\hat{Z}^{\theta+\delta_p}$ as an approximation of the exact, unknown $Z^{\theta+\delta_p}$ first, from which the first component is extracted as an approximation $\hat{Y}_t^{\theta+\delta_p}$ for $Y_t^{\theta+\delta_p}$.

Prediction as introduced here is restricted to the granularity of the underlying clustering, i.e. a single

discrete $c_{p,t}^r = \hat{Z}^{\theta+\delta_p}$ is produced as the predicted delay-vector and no interpolation between cells is employed, i.e. the prediction optimum is restricted to

$$|c_{p,t}^r, \hat{Z}^{\theta+\delta_p}|_d \leq |c_{i,t}^r, \hat{Z}^{\theta+\delta_p}|_d \quad \forall c_{i,t} \in C_t \quad (50)$$

The introduced prediction method is based on the determination of the optimal path p^{δ_p} between the current b_t and $c_{p,t}^r$

$$p^{\delta_p} = p(b_t, c_{p,t}^r) = (c_{k_1,t}, \dots, c_{k_n,t}) \quad (51)$$

with the following constraints. First the set of paths ending at the correct temporal distance P^{δ} is defined as

$$P^{\delta_p} = \{p(b_t, c_{p,t}^r)\} = \{(c_{k_1,t}, \dots, c_{k_n,t})\} \quad (52)$$

where

$$|p^{\bar{\delta}} - \delta_p| \leq |p^{\bar{\delta}} - \langle c_{k_{n-1,t}}, c_{k_n,t} \rangle^{\bar{\delta}} - \delta_p| \quad \text{and} \quad (53)$$

$$|p^{\bar{\delta}} - \delta_p| \leq |p^{\bar{\delta}} + \langle c_{k_n,t}, c_{k_{n-1,t}} \rangle^{\bar{\delta}} - \delta_p| \quad (54)$$

i.e. neither the immediately connected predecessor $c_{k_{n-1,t}}$ nor any successor $c_{*,t}$ of the end-cell of those paths has a predicted real-time stamp closer to $\theta + \delta_p$ than the end-point $c_{k_n,t}$ itself. Furthermore it is required that the number of self-references from the histories along the path p^{δ_p} to this path itself (and including the current trace T_t) is maximized. The number of self-references $\Xi(p_t)$ along any path at time t is given as

$$\Xi(p_t) = \sum_{c_{i,t} \in p_t \cup T_t} \sum_{c_{j,t} \in p_t} \mu(c_{i,t}, c_{j,t}^H) \quad (55)$$

The only missing constraint for the optimal prediction path is thus

$$\Xi(p^{\delta_p}) \geq \Xi(p) \quad \forall p \in P^{\delta_p} \quad (56)$$

Now the prediction $\hat{Y}_t^{\theta+\delta_p}$ is given simply by extracting the appropriate values at the end-point of the optimal prediction path

$$\hat{Z}^{\theta+\delta_p} = c_{p,t}^r \quad \text{and} \quad \hat{Y}_t^{\theta+\delta_p} = \hat{Z}^{\theta+\delta_p, \tau_1} \quad (57)$$

4. Complexities for real-time operation

The set of predicates and equations in section 2 does not indicate whether there is a possible on-line and real-time implementation. This aspect of the RTDSM will be discussed here.

The method has been designed to rely on local adaptations only, i.e. with every new sample there is supposed to be only a limited number of cells involved. Once a reasonable upper limit for those cells which need to be touched with every new sample can be established, the calculations for the computational complexities for the individual equations will finalize the real-time considerations. Simulation (next section) will reassure the principal findings.

As seen in section 2 all adaptation steps adapting c_i^f , and c_i^a in order to integrate the new sample into the existing model are related to the metrical neighbourhood V_i^p or more specifically to the topological neighbourhood only. Each adaptation step (42)(43) is bound by a constant computational complexity per cell c_i . Two points need to be discussed here:

- a. Are the number of elements in each neighbourhood bound by a constant number?
- b. Can the neighbourhoods be identified in constant complexity?

Point *a* has been extensively investigated in the context of self-organizing maps and there it can be guaranteed that a set of cells will uniformly distribute in a given sampling space – assuming an infinite amount of time. A proof for systems which adapt only one part of the system at a time and process the data in sampling order (as the method under consideration here) is not (yet) available. Simulations indicate that sequence-bound, local methods perform very similar to out-of-sequence, and globally adapting methods – as the following section will demonstrate again (see also [7] for another real-world example). Still a formal proof is open.

Point *b* is a harder one and in the general case this one needs to be answered with ‘no’ currently. But not all is lost: As for any mass-affected, physical dynamical system the vast majority of samples will follow some continuity assumption. So the chance that a new sample will relate to the same or a close metrical and topological neighbourhood is very high. This leads to simple forms of caching mechanisms for the cells. As for the cells in the local neighbourhood an ordered set needs to be specified so that the adaptations can be done according to the ranks in those index sets, this ordered set can be preserved and re-used in the search process for the next sample. In fact all cells in N are in global order, with the cells in the local neighbourhood always dragged to the top of this ordered set. Now as we assume that a uniform distribution of cells establishes itself, the search process for local cells can be stopped after a certain number of close cells has been found. Assuming some form of locality in the sampling sequence (continuity assumption, or frequent trajectories for instance) this caching mechanism is effective. Still for point *b* there will always be practical cases where the whole model needs to be searched in order to identify the local neighbourhoods.

Besides those two critical issues (which can be reasonably addressed in practical systems) the set of predicates and equations are all bound by constant complexities or depend on the number of cells in the local neighbourhood V_i^p . If point *a* can be answered with ‘yes’ (as indicated by simulations) then also those equations ((34), (35), (42), and (43)) translate into constant complexities.

The actual translation into algorithmic form can not be given here due to space constraints, but an efficient implementation (as also used in the next section) is freely available on request.

5. Simulations

In the following simulations, both the method described above (RTDSM) and a neural gas model [3] are applied to the prediction of time series produced by a numerical solution of the Mackey-Glass equation [2]

$$\frac{d}{dt}x_t = \beta x_t + \frac{\alpha x_{t-\tau}}{1 + x_{t-\tau}^{10}} \quad (58)$$

with parameters $\alpha = 0.2$, $\beta = -0.1$, $\tau = 17$, $x_0 = 1.2$, and $x_t = 0$ for $t < 0$. With these parameters, the dimensionality of the attractor is $d \approx 2.1$. The time resolution for the discretization is $\Delta t = 0.1$.

As the characteristic time constant for the Mackey-Glass system is $t_{\text{char}} \approx 50$, the goal for the prediction is set to forecast x_{t+48} . All prediction errors are determined by the *rms* value of the absolute prediction error at x_{t+48} divided by the standard deviation of x_t .

It can be shown, that for the Mackey-Glass system (58) an embedding with a 4-dimensional delay-vector is sufficient to allow a smooth prediction of future values. In practice, any numerical solution of (58) has only a limited resolution in the delay-vector space. Therefore, a simulation will sometimes map different parts of the trajectory into the same discrete volume of the delay-vector space. The simulated system will therefore differ from (58). In a simulation, this effect can not be avoided.

If the dimensionality of the delay-vector space is reduced, the trajectory of the system - now projected onto a subspace - will cross itself at additional locations. In a simulation, the regions around these points see multiple trajectories leaving and thereby rendering the system non-deterministic.

The approach described in this paper focuses exactly on this latter case. For the simulation, an embedding dimension of three leading to a state of

$$Z_t = (Y_t, Y_{t-6}, Y_{t-12}) \quad (59)$$

was chosen. The parameters of the network were

$$\Pi^\alpha = 0.04, \Pi^\lambda = 1.0 \quad (60)$$

the length of the trace $n = 5$, and the neighbourhood distance was uniformly chosen as $\rho = \Pi^\alpha$.

Figure 1 shows the evolving topological structure after 100,000 training steps. Darker edges denote more frequently used trajectories throughout the system.

In a magnified part of the system taken after 30,000 steps (figure 3), one can clearly see single trajectories crossing each other without interference. See for example the area which is marked with a circle, where

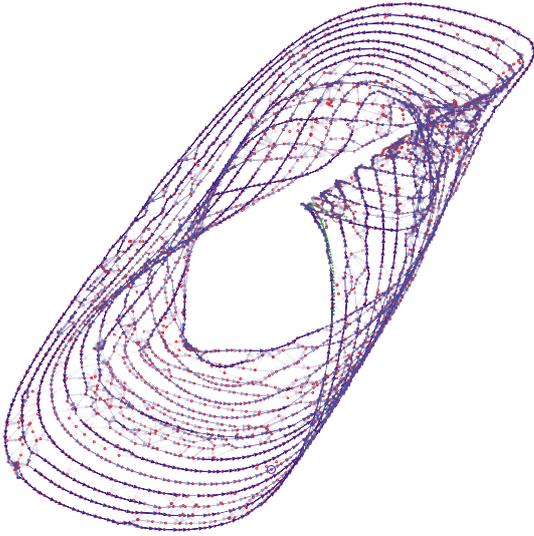


figure 1: 3-dim. system with RTDSM (100,000 steps)

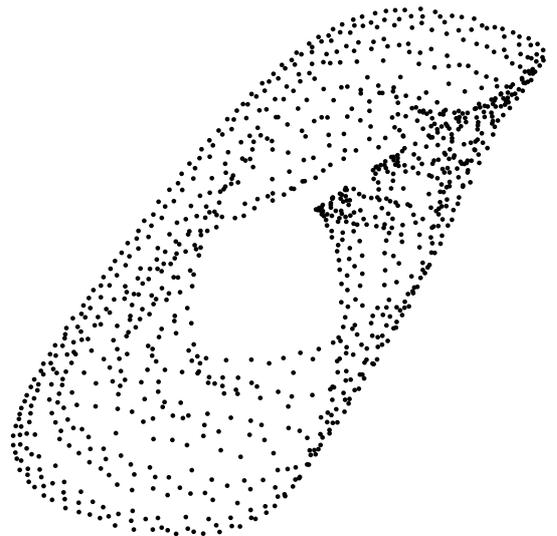


figure 2: 4-dim. system with neural gas (100,000 steps)

multiple metrically close cells have been separated as belonging to different trajectories.

For the neural gas method, a network with 1,000 cells was trained with 100,000 samples and the same parameter settings as in [3]. Here, a 4-dimensional delay-vector space

$$Z_t = (Y_t, Y_{t-6}, Y_{t-12}, Y_{t-18}) \quad (61)$$

had to be employed because the approach described in [3] is a function approximation and will fail if the trajectory in the delay-vector space does not describe a deterministic system. A 2-dimensional projection of the 4-dimensional cell distribution after the adaptation of the neural gas network in figure 2 shows the adaptation to the attractor manifold of the Mackey-Glass system.

The formal definition for the prediction (52) uses the set of all trajectories covering the forecast interval. As the size of this set grows exponentially, it is not feasible to apply it directly in simulations. An approximation iteratively searches for the best next cell as long as the remaining time interval is bounded by (54). The center of the final cell found is then used as prediction in the delay-vector space. No further extrapolation is applied. If the network still learns new regions of the data manifold, the matching cell is not connected to any other cell via an outgoing edge. In those cases, no prediction is provided.

The speed with which the state of a system moves through the phase space and accordingly through the delay-vector space varies for all but very simple systems. Therefore, the length of the mean transition times on the edges of the proposed system will vary, too (between 0.2s and 2.7s in the system at hand). When the best matching cell is in a region with high time resolution and the predicted cell in an area with

low time resolution, extra time jitter is introduced into the prediction. The jitter of the prediction error coming from time pace differences can be estimated as the ration of the longest and shortest transition time multiplied by the spatial resolution ρ .

The neural gas iterates over a number of time steps (e.g. 8 steps in the approach shown) to interpolate the next function approximation with the help of a learned local linear extrapolation. For each time step, all cells of the network must be searched in order to locate the closest cell for the given delay-vector.

After about 3,400 samples, RTDSM has explored a large part of the data manifold and the prediction error settles in the range shown in figure 4.

For the neural gas approach, the prediction error starts very high, because the system needs to be initialized randomly and with high mobility parameters (as described in [3]). When the neural gas adapts, the prediction error decreases. It reaches a level similar to RTDSM only after more than approximately

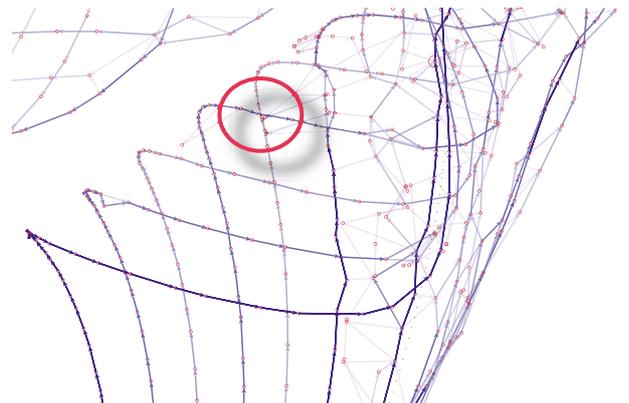


figure 3: Magnified 3-dimensional system (30,000 steps)

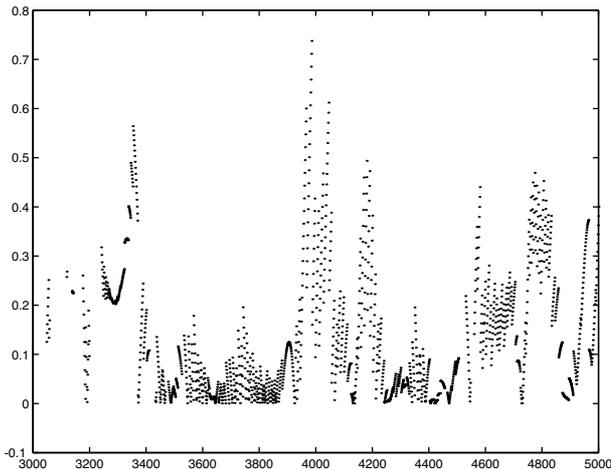


figure 4: Normalized prediction error (RTDSM)

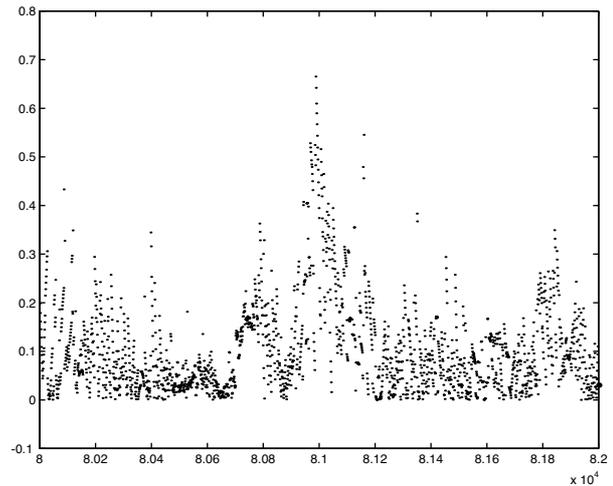


figure 5: Normalized prediction error (neural gas)

80,000 samples have been processed (figure 5) – compared to 3,000 to 4,000 samples in RTDSM, even though the computational costs for each sample is significantly higher in neural gas. The convergence of the neural gas method depends critically on the data samples to be drawn randomly from the unknown distribution over the manifold. This required randomness of the data samples is usually not given when on-line data which contains many short term correlations is employed.

6. Conclusions

As this publication is the introduction of the proposed real-time modelling method RTDSM for complex dynamical systems, it is focused on its precise specification and gives only brief examples of its actual performance. The principal findings are encouraging and more tests involving the physical systems in the authors' group (autonomous underwater vehicles) will be documented shortly. Those underwater systems are perfectly suited to supply highly complex non-stationary dynamical systems with even simple experimental setups and are actually the driving force for the work presented here.

References

- [1] Duda, Richard O., Hart, Peter E., Stork, David G., *Pattern Classification*, John Wiley & Sons, Inc, New York, 2000.
- [2] Mackey, Michael C., Glass, Leon, *Oscillation and Chaos in Physiological Control Systems*, Science, New Series, Vol. 197, No. 4300, 287-289, 1977.
- [3] Martinez, Thomas M., Berkovich, Stislav G., Schulten, Klaus J., "Neural-Gas" Network for Vector Quantization and its Application to Time-Series Prediction, IEEE Transactions on Neural Networks, Vol. 4, No. 4, July 1993, pp. 558-569.
- [4] Oja, Erkki, *Subspace Methods of Pattern Recognition*, Research Studies Press, Letchworth, England, 1983.
- [5] Small, M. and Tse, C.K., *Optimal selection of embedding parameters for time series modelling*, European Conference on Circuits Theory and Design (European Circuit Society and the Institute of Electrical and Electronic Engineers; Krakow, Poland, 1-4 September 2003)
- [6] Takens, Floris, *Detecting strange attractors in turbulence*, Lecture Notes in Mathematics, 898, p. 366, Springer-Verlag 1981.
- [7] Zimmer, Uwe R., *Embedding local metrical map patches in a globally consistent topological map*, Proceedings of Underwater Technologies, Tokyo, Japan, 2000