Discretisation & Control of Irregularly Actuated and Sampled LTI systems

Christian Klauer[1] Thomas Schauer[1]
[1]Control Systems Group, Technische Universität Berlin, Germany

Abstract— **Motivated by control tasks in neuro-prosthetic systems, a method for the exact discretisation of continuous-time LTI systems in presence of irregular actuation and sampling times is presented. In this approach, sequences for the desired sampling and actuation times form additional inputs to the discrete-time systems. An analytical evaluation for a class of practically relevant systems is presented that yields discrete-time state-space systems with non-constant coefficients. Because of the non-linearity introduced by the irregular time intervals, a linearising controller is proposed that allows to re-use standard linear controllers. A simulation example demonstrates the feasibility of the presented control approach for a neuro-prosthetic system.**

I. Introduction

In neuro-prosthetic systems (NPS), a common procedure for artificial actuation of muscles is to apply Functional Electrical Stimulation (FES) [16] to achieve a certain goal, e.g. tracking a reference trajectory for a joint angle (e.g. [17]). FES relies on the application of short electrical pulses whose intensity in terms of pulse charge can be modulated, while the time intervals between the stimulation pulses form a second control input. Because the pulses are applied at discrete time points, discrete-time control systems are pre-destinated for the control of such NPS.

Compared to voluntary activation, FES activated muscles are subjected to a fast development of muscular fatigue. This fact can severely hinder functional tasks in the long term.

For achieving the same resulting muscular force, a higher stimulation frequency (and smaller intensity) leads to a faster proceeding of muscular fatigue, compared to a low frequency stimulation (and higher intensity) [11]. However, more frequently applied pulses typically enable a better performance in time-discrete control systems. Therefore, typically a compromise between the progressing of muscular fatigue and control performance is chosen. Here, fixed frequencies ranging from 20 to 60 Hz are used, whereby the stimulation intensity remains the only actuation variable.

For temporary increasing the control performance, an obvious approach could be to increase the actuation rate for periods during which it is beneficial (e.g. during fast movements), while for periods of low activity (e.g. holding a position) the actuation rate may be lowered. However, the effect of continuously adjusting the actuation times must be included in the discretised model.

Typical time-discretisation methods [7] assume a fixed duration of actuation- and sampling intervals. However, for the use case described above, a more general method is required that takes variable sampling and actuation times into account. Such a method is presented in this work (c.f. Fig. 1). Variable and alternating time-points for actuation (actuation time) and sampling (sampling time) are described by the sequences t_δ and t_Δ respectively that represent inputs to the discretised system additionally to the actuation sequence u.

In order to match an output sequence $y_{\delta\Delta}$ according to a reference r_y, typical combinations of the three input variables and their applications (not only in FES) include:

a) While t_δ and t_Δ are given sequences, find a sequence u that leads to $y_{\delta\Delta} = r_y$. Typical applications include: disturbed time periods in e.g. networked control, soft real-time systems.
b) While u and t_Δ are given sequences, find a sequence t_δ that leads to $y_{\delta\Delta} = r_y$. Application: Control of systems using PWM-modulation
c) Adjust u, t_δ and t_Δ such that $y_{\delta\Delta} = r_y$ and additional constraints on the relationship between u, t_δ and potentially t_Δ are fulfilled. Applications: Control of systems in which it is cost intensive to actuate or change the actuation intensity (e.g. FES, valves in process control).

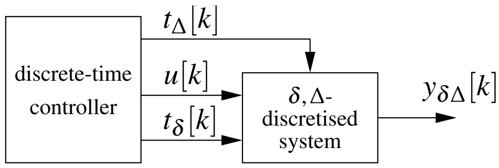

Fig. 1 - The discrete-time system as obtained by the transformation incorporating variable actuation ($t_\delta[k]$) and sampling ($t_\Delta[k]$) times of a continuous system. Additionally to the input sequence u, both times sequences are considered to be inputs to the discretised system. All variables may be adjusted by a control system.

Other potential applications include the estimation e.g. using Kalman-Filters under irregularly arriving measurements and the control of blood glucose concentration because the time intervals between glucose-concentration measurements and between the injections of insulin are typically not constant.

In the past, to obtain discrete-time models in FES, typically zero-order hold- or Euler-based approaches were used. We assume however, that models, which incorporate the application of Dirac-pulses to a continuous-time system are more suitable to describe the dynamics of the muscle's force to electrical pulses as observed e.g. in [18]. Therefore, unlike zero or first order hold discretisation methods, the influence of the actuation sequence is described by intensity-modulated Dirac-pulses that are applied to the given continuous dynamical system. Zero- or first order behaviour can still easily be realised by extending the presented discretisation method.

Typically, most methods in sampled-data systems rely on periodic sampling [7]. Irregular sampling times [15] are typically considered in the analysis of Networked Control Systems (NCS) by considering e.g. random time delays for the arrival of a sensor measurement at the controller [10]. An application for NCS is e.g. [1]. Such approaches typically consider variations in the sample time as external disturbances. In [19], stability properties for switching between multiple sampling rates are investigated, however only for a finite number of different rates. There are some approaches for determining an optimal sampling rate for each individual control loop in large scale control systems to optimise the usage of resources e.g. [9], [12]. Stability under random sampling rates is e.g. investigated in [6]. Applications that use variable sampling rates are e.g. [14], [20]. In event-based control [4], typically a control step is triggered by a condition on the plant's states that may cross defined borders or tolerance bands leading to irregular sampling times. In [13], a control scheme is presented that updates the control variable as a sensor measurement becomes available. In [5], several issues in event-based sampling in the context of non-linear filtering are discussed, e.g. anti-aliasing filters. An event based PID-controller is proposed in [3] without a formal analysis. First results for analysing event-based systems are presented in [8], whereby the focus is on regularly sampled control systems in which actuation variable updates can be skipped. In [2], the stability of a classically designed state-feedback controller applied to a plant is analysed, whereby the time for the next update of the controller output is decided on-line by an event generator.

In contrast to the above investigations, an easily usable, general framework for discretising LTI systems with variable times for sampling and additionally actuation is presented here and a general linearising controller for the obtained discrete-time models is proposed that compensates the effects introduced by the irregular time intervals.

II. Definition and Properties

A. Delta-actuation and Sampling

The continuous-time dynamical LTI SISO system

$$y(t) = \mathbb{S}[v(t)](t), \qquad t \geq 0$$

is considered that maps an input signal $v \in \mathbb{R}$ to an output signal $y \in \mathbb{R}$ for $t \geq 0$. The I/O relationship between v and y is described by the state-space representation

$$\mathbb{S} \equiv \left\{ \begin{array}{ll} \dot{\boldsymbol{x}}(t) = \mathcal{A}x(t) + \mathcal{B}v(t), & \boldsymbol{x} \in \mathbb{R}^n, v, y \in \mathbb{R} \\ y(t) = \mathcal{C}\boldsymbol{x}(t), & \boldsymbol{x}(t=0) = \boldsymbol{x_0} \end{array} \right\}. \tag{1}$$

The continuous-domain δ-actuation input signal is defined as follows:

$$v_\delta(t) := \sum_{j=0}^{\infty} u[j]\,\delta(t - t_\delta[j]), \qquad j \in \mathbb{N}, \tag{2}$$

whereby $\delta(t)$ is the Dirac delta distribution. The sequence $u = \{u[0], u[1], \cdots\}$ modulates the intensities of the δ-pulses. The time at which the k^{th} δ-pulse ($k \in \mathbb{N} \wedge k \geq 0$) is applied is denoted by the actuation time $t_\delta[k] \geq 0$ that is an element of the sequence $t_\delta = \{t_\delta[0], t_\delta[1], \cdots\}$. The first actuation takes place at time zero ($t_\delta[0] := 0$).

This signal is applied to $\mathbb{S}$ ($v(t) = v_\delta$), while the output $y(t)$ is sampled (denoted by Δ) at the sampling times $t_\Delta[k] > 0$, $k \geq 1$ ($t_\Delta = \{t_\Delta[1], t_\Delta[2], \cdots\}$). The actuation and sampling is performed alternating, which is denoted by

$$t_\delta[k] < t_\Delta[k+1] \leq t_\delta[k+1] < t_\Delta[k+2] \quad \forall k \geq 0 \wedge k \in \mathbb{N}. \tag{3}$$

The described procedure is illustrated in Fig. 2 and called time-variable δ-actuation. By a formal definition the transformation $\mathfrak{Z}_{\delta\Delta}$ is introduced:

$$\begin{aligned} y_{\delta\Delta}[k] &= \mathfrak{Z}_{\delta\Delta}[\mathbb{S}, u, t_\delta, t_\Delta][k] \\ &:= \mathbb{S}\left[\sum_{j=0}^{\infty} u[j]\delta(t - t_\delta[j])\right](t_\Delta[k]). \end{aligned} \tag{4}$$

The time interval between an actuation pulse $k \geq 0$ and its subsequent measurement $k+1$ is defined as

$$T_\Delta[k] := t_\Delta[k+1] - t_\delta[k] > 0, \qquad \forall k \geq 0. \tag{5}$$

Analog, the time interval between an actuation pulse k and a previous measurement k is defined (c.f. Fig 2) by

$$T_\delta[k-1] := t_\delta[k] - t_\Delta[k] \geq 0, \qquad \forall k \geq 1. \tag{6}$$

On the basis of these definitions, the following properties for calculating the time points $t_\Delta[k]$ and $t_\delta[k]$ from the time-differences derive: A relation for calculating the actuation time in dependence of sequences for the time intervals T_δ and T_Δ is given by

$$t_\delta[k] := t_\delta[0] + \sum_{j=0}^{k-1} (T_\delta[j] + T_\Delta[j]), \qquad \forall k \geq 1. \tag{7}$$

The sampling time t_Δ is then given by

$$t_\Delta[k+1] := T_\Delta[k] + t_\delta[k], \qquad \forall k \geq 0. \tag{8}$$

B. Properties for linear time-invariant systems

If the system $\mathbb{S}$ is a linear system in the sense of

$$\mathbb{S}[av_1(t) + bv_2(t)](t) = a\,\mathbb{S}[v_1(t)](t) + b\,\mathbb{S}[v_2(t)](t), \tag{9}$$

whereby $a, b \in \mathbb{R}$ then the property of linearity also holds true for the dependency of y on u of the sampled system:

$$\begin{aligned} & \mathfrak{Z}_{\delta\Delta}[\mathbb{S}, au_1 + bu_2, t_\delta, t_\Delta][k] \\ = \; & a\,\mathfrak{Z}_{\delta\Delta}[\mathbb{S}, u_1, t_\delta, t_\Delta][k] + b\,\mathfrak{Z}_{\delta\Delta}[\mathbb{S}, u_2, t_\delta, t_\Delta][k]. \end{aligned} \tag{10}$$

The prove is straightforward.

III. Discretisation of elementary Systems

In this section $\mathfrak{Z}_{\delta\Delta}$ is applied to 1^{st}- and 2^{nd}-order systems. Analytical representations in form of discrete-time state-space systems to describe the I/O behaviour result. For most higher-order systems, a decomposition into one or more of such basic systems can be performed (e.g. by means of a Jordan normal form or a partial fraction decomposition) to apply $\mathfrak{Z}_{\delta\Delta}$ separately according to (10).

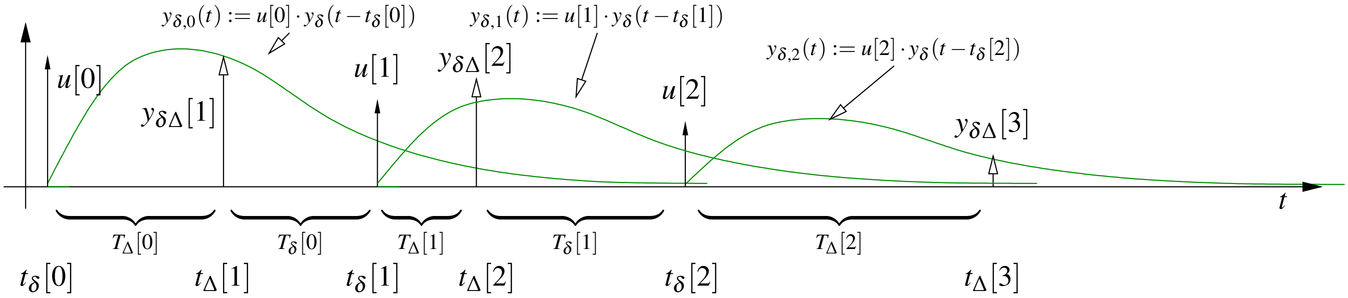

Fig. 2 - Actuation of a LTI system $\mathbb{S}$ using δ-actuation by applying δ-pulses at the actuation times $t_\delta[k]$ and sampling the output at $t_\Delta[k]$. The output of the continuous system is the superposition of the individual system responses $y_{\delta,i}$ $(i=0,1,2)$ to each δ-pulse in addition to the transient caused by the initial states (not shown here).

A. 1^{st}-order LTI System $\mathbb{S}^1$

Let the I/O-behaviour of $\mathbb{S}^1$ for $t \geq 0$ (the dependency of $y(t)$ on $v(t)$) be described by the 1^{st}-order state-space system

$$\mathbb{S}^1 \equiv \left\{ \begin{array}{ll} \dot{x}(t) = s_\infty x(t) + v(t), & x, y, v, s_\infty, x_o \in \mathbb{C} \\ y(t) = x(t), & x(t=0) = x_0. \end{array} \right\}.$$

Please note that the output, the state x, the initial state x_0 and the parameter s_∞ may be complex valued. In order to calculate the output y when applying the δ-actuation input $v_\delta(t)$ to v, the Laplace-transform

$$Y_\delta(s) = \underbrace{\frac{1}{s-s_\infty} x_0}_{:=Y_\delta^P(s)} + \underbrace{\frac{1}{s-s_\infty} V_\delta(s)}_{:=Y_\delta^H(s)}, \tag{11}$$

is obtained, whereby $Y_\delta^P(s)$ and $Y_\delta^H(s)$ are the Laplace-transformed output components $y_\delta^P(t)$ and $y_\delta^H(t)$ respectively. Further,

$$V_\delta(s) = \sum_{j=0}^{\infty} u[j] e^{-t_\delta[j]\, s}, \qquad j \in \mathbb{N} \tag{12}$$

is the Laplace-transform of the δ-actuation signal v_δ, while $G^1(s) = 1/(s-s_\infty)$ is the transfer function of $\mathbb{S}^1$.

Particular solution P: At first Y_δ^P is considered to calculate the output component y_δ^P. Therefore, the time-domain impulse response ($v(t) = \delta(t)$) of G^1 is calculated:

$$\mathcal{L}^{-1}[G^1(s)](t) = y_\delta(t) = \left\{ \begin{array}{ll} 0 & \text{if } t \leq 0 \\ e^{s_\infty t} & \text{if } t > 0 \end{array} \right. . \tag{13}$$

The time-domain response of Y_δ^P is then given by

$$y_\delta^P(t) = x_0 \cdot y_\delta(t) = \left\{ \begin{array}{ll} 0 & \text{if } t \leq 0 \\ x_0 e^{s_\infty t} & \text{if } t > 0 \end{array} \right. .$$

As mentioned in (4), the output signal is sampled at $t_\Delta[k]$, $\forall k \geq 1$. Because $\mathfrak{Z}_{\delta\Delta}$ is linear (c.f. Eq. (10)), sampling y_δ is equivalent to calculating the sum of the sampled output components of H and P. Sampling y_δ^P at $t_\Delta[k] > 0$ yields:

$$y_{\delta\Delta}^P[k] = y_{\delta a}^P(t_\Delta[k]) = x_0 e^{s_\infty t_\Delta[k]}, \qquad \forall k \geq 1. \tag{14}$$

Since the aim is to obtain a state-space representation of the sampled system, a recursive formulation of Eq. (14) is calculated, whereby $T_\delta[-1]$ is defined to be zero:

$$\begin{aligned} y_{\delta\Delta}^P[k+1] &= y_{\delta\Delta}^P[k] e^{s_\infty (T_\delta[k-1] + T_\Delta[k])} \qquad \forall k \geq 0 \\ y_{\delta\Delta}^P[0] &= x_0, \quad T_\delta[-1] := 0. \end{aligned} \tag{15}$$

Homogeneous solution H: Now Y_δ^H is considered. Because $\mathbb{S}^1$ is linear, the response $y_\delta^H(t)$ is the sum of all individual responses to each individual δ-pulse contained in v_δ:

$$y_\delta^H(t) = \sum_{j=0}^{\infty} \underbrace{u[j] y_\delta(t - t_\delta[j])}_{y_{\delta,j}(t)}.$$

Sampling the continuous signal y_δ^H at $t_\Delta[k]$ for $k \geq 1$ yields

$$y_{\delta\Delta}^H[k] = y_\delta^H(t_\Delta[k]) = \sum_{j=0}^{\infty} u[j] y_\delta(t_\Delta[k] - t_\delta[j])$$

Since $y_\delta(t)$ is zero for $t = t_\Delta[k] - t_\delta[j] \leq 0$ and because of (3), the elements $j \geq k$ of the sum are zero, yielding

$$\begin{aligned} y_{\delta\Delta}^H[k] &= \sum_{j=0}^{k-1} u[j] y_\delta(t_\Delta[k] - t_\delta[j]) = \sum_{j=0}^{k-1} u[j] e^{s_\infty (t_\Delta[k] - t_\delta[j])} \\ &= e^{s_\infty t_\Delta[k]} \cdot \sum_{j=0}^{k-1} u[j] e^{-s_\infty t_\delta[j]}, \qquad \forall k \geq 1. \end{aligned} \tag{16}$$

1) Recursion: To calculate a recursive representation of (16), an index shift is performed $\forall k \geq 0$:

$$y_{\delta\Delta}^H[k+1] = e^{s_\infty t_\Delta[k+1]} \cdot \left[u[k] e^{-s_\infty t_\delta[k]} + \sum_{j=0}^{k-1} u[j] e^{-s_\infty t_\delta[j]} \right]. \tag{17}$$

Decomposing the exponential factor using $t_\Delta[k+1] = t_\Delta[k] + T_\delta[k-1] + T_\Delta[k]$ following from Eq. (5) and (6) yields

$$e^{s_\infty t_\Delta[k+1]} = e^{s_\infty t_\Delta[k]} \cdot e^{s_\infty [T_\delta[k-1] + T_\Delta[k]]}.$$

Using this result, Eq. (17) can be rearranged yielding

$$\begin{aligned} y_{\delta\Delta}^H[k+1] &= e^{s_\infty [T_\delta[k-1] + T_\Delta[k]]} \cdot \underbrace{e^{s_\infty t_\Delta[k]} \cdot \sum_{j=0}^{k-1} u[j] e^{-s_\infty t_\delta[j]}}_{y_{\delta\Delta}^H[k]} \\ &+ e^{s_\infty t_\Delta[k]} \cdot e^{s_\infty [T_\delta[k-1] + T_\Delta[k]]} \cdot u[k] e^{-s_\infty t_\delta[k]}. \end{aligned} \tag{18}$$

By applying $t_\Delta[k] - t_\delta[k] = -T_\delta[k-1]$ following from Eq. (5) and Eq. (6), the resulting recursive equation for $y_{\delta,\Delta}^H$ is then:

$$y_{\delta\Delta}^H[k+1] = e^{s_\infty [T_\delta[k-1] + T_\Delta[k]]} y_{\delta\Delta}^H[k] + u[k] e^{s_\infty T_\Delta[k]}. \tag{19}$$

Evaluating equation (16) for $k = 1$ gives an initial condition

$$y^H_{\delta\Delta}[1] = u[0]y_\delta(t_\Delta[1] - t_\delta[0]) = u[0]e^{s_\infty T_\Delta[0]}. \quad (20)$$

By defining $T_\delta[-1] := 0$ it is observed that the initial condition (20) turns into the more usual initial condition

$$y^H_{\delta\Delta}[0] := 0. \quad (21)$$

The variable $y^H_{\delta\Delta}[0]$ is hereby assigned and $y^H_{\delta\Delta}[1]$ results from (19) for $k = 0$.

2) Result: The sampled output of $\mathbb{S}^1$ combines H and P:

$$y_{\delta\Delta}[k] = y^H_{\delta\Delta}[k] + y^P_{\delta\Delta}[k].$$

To obtain a recursive formulation, an index shift for k is performed and the partial results described by Eq. (15) and (19) are combined yielding

$$y_{\delta\Delta}[k+1] = \underbrace{\left(y^H_{\delta\Delta}[k] + y^P_{\delta\Delta}[k]\right)}_{y_{\delta\Delta}[k]} \cdot e^{s_\infty [T_\delta[k-1]+T_\Delta[k]]} + u[k]e^{s_\infty T_\Delta[k]}.$$

The initial condition for $y_{\delta\Delta}[0]$ is then obtained using (21) and (15) yielding

$$y_{\delta\Delta}[0] = y^H_{\delta\Delta}[0] + y^P_{\delta\Delta}[0] = x_0.$$

The resulting discretisation of $\mathbb{S}^1$ is then described by

$$\left\{\; y_{\delta\Delta} = \mathfrak{Z}_{\boldsymbol{\delta\Delta}}\left[\mathbb{S}^1, u, t_\delta, t_\Delta\right][k] \;\right\} \quad (22)$$
$$\equiv \left\{ \begin{array}{l} y_{\delta\Delta}[k+1] = y_{\delta\Delta}[k] \cdot e^{s_\infty [T_\delta[k-1]+T_\Delta[k]]} + u[k]e^{s_\infty T_\Delta[k]} \\ y_{\delta\Delta}[0] = x_0. \end{array} \right\}.$$

This relationship describes a non-linear discrete-time system with the inputs u, T_δ and T_Δ. A scheduling of the system's coefficients based of the inputs for the time differences T_δ and T_Δ is herein performed.

B. 2^{nd}-order LTI Systems with complex conjugate eigenvalues

For 2^{nd}-order systems, only the case of a system $\mathbb{S}^{cc}$ incorporating complex conjugated eigenvalues ($\lambda_1 = \lambda = \sigma + \mathrm{j}\omega$, $\lambda_2 = \overline{\lambda} = \sigma - \mathrm{j}\omega$, j is the imaginary unit) is considered. Systems with non-equal and real-valued eigenvalues ($\lambda_1, \lambda_2 \in \mathbb{R} \wedge \lambda_1 \neq \lambda_2$), can be decomposed into two 1^{st}-order systems with superposed outputs that may then be discretised separately by re-using (22).

For discretising $\mathbb{S}^{cc}$ this principle is also used in a first step, but as the resulting discrete-time systems have complex valued states, further effort for deriving real-valued systems is required. The I/O behaviour of a strictly proper 2^{nd}-order system $\mathbb{S}^{cc}$ with conjugate complex eigenvalues can always be described by the following state-space representation:

$$\mathbb{S}^{cc} \equiv \left\{ \begin{array}{l} \underbrace{\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix}}_{\dot{\boldsymbol{x}}(t)} = \underbrace{\begin{pmatrix} \lambda + \overline{\lambda} & -\lambda\overline{\lambda} \\ 1 & 0 \end{pmatrix}}_{\boldsymbol{A}} \underbrace{\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}}_{\boldsymbol{x}(t)} + \boldsymbol{B}v(t), \quad \boldsymbol{B} \in \mathbb{R}^2 \\ y(t) = \underbrace{\begin{pmatrix} 0 & 1 \end{pmatrix}}_{\boldsymbol{C}} \boldsymbol{x}(t) \qquad \boldsymbol{x}(0) = \begin{pmatrix} x_{1,0} \\ x_{2,0} \end{pmatrix} \end{array} \right\} \quad (23)$$

Ongoing, two different classes of systems are considered:

A) $\mathbb{S}^{cc}_A$ that derives from $\mathbb{S}^{cc}$ by choosing $\boldsymbol{B} = \boldsymbol{B_A} := [1, 0]^T$ and realises the transfer function $1/((s-\lambda)(s-\overline{\lambda}))$.

B) $\mathbb{S}^{cc}_B$ as derived from $\mathbb{S}^{cc}$ by choosing $\boldsymbol{B} = \boldsymbol{B_B} := [-a + \lambda + \overline{\lambda}, 1]^T$ that realises $(s-a)/((s-\lambda)(s-\overline{\lambda}))$.

In order to perform a separation into two 1^{st}-order systems, a diagonal form for describing $\boldsymbol{A}$ is calculated:

$$\boldsymbol{A} = \boldsymbol{PJP}^{-1}, \qquad \boldsymbol{J} = \mathrm{diag}(\overline{\lambda}, \lambda)$$
$$\boldsymbol{P} = \begin{pmatrix} \frac{\lambda}{2\omega} & \frac{\overline{\lambda}}{2\omega} \\ \frac{\mathrm{j}}{2\omega} & -\frac{\mathrm{j}}{2\omega} \end{pmatrix}, \qquad \boldsymbol{P}^{-1} = \begin{pmatrix} 1 & -\lambda \\ 1 & -\overline{\lambda} \end{pmatrix}. \quad (24)$$

By introducing the transformed state vector

$$\tilde{\boldsymbol{x}} = [\tilde{x}_1, \tilde{x}_2] = \boldsymbol{P}^{-1}\boldsymbol{x},$$

a representation with a diagonal system matrix $\boldsymbol{J}$ is obtained:

$$\mathbb{S}^{cc} \equiv \left\{ \begin{array}{l} \underbrace{\boldsymbol{P}^{-1}\dot{\boldsymbol{x}}}_{\dot{\tilde{\boldsymbol{x}}}} = \boldsymbol{J}\underbrace{\boldsymbol{P}^{-1}\boldsymbol{x}}_{\tilde{\boldsymbol{x}}} + \underbrace{\boldsymbol{P}^{-1}\boldsymbol{B}}_{\tilde{\boldsymbol{B}}}v(t), \quad \tilde{\boldsymbol{B}} = [\tilde{b}_1, \tilde{b}_2]^T \\ y(t) = \boldsymbol{CP}\tilde{\boldsymbol{x}} \qquad \tilde{\boldsymbol{x}}(0) = \boldsymbol{P}^{-1}\begin{pmatrix} x_{1,0} \\ x_{2,0} \end{pmatrix} = \begin{pmatrix} x_{1,0} - \lambda x_{2,0} \\ x_{1,0} - \overline{\lambda} x_{2,0} \end{pmatrix} \end{array} \right\}.$$

For the systems $\mathbb{S}^{cc}_A$ and $\mathbb{S}^{cc}_B$, the input matrix $\boldsymbol{B}$ becomes

$$\tilde{\boldsymbol{B}}_{\boldsymbol{A}} = \begin{bmatrix} \tilde{b}_{A1} \\ \tilde{b}_{A2} \end{bmatrix} = \boldsymbol{P}^{-1}\boldsymbol{B_A} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \vee \tilde{\boldsymbol{B}}_{\boldsymbol{B}} = \begin{bmatrix} \tilde{b}_{B1} \\ \tilde{b}_{B2} \end{bmatrix} = \boldsymbol{P}^{-1}\boldsymbol{B_B} = \begin{pmatrix} -a + \overline{\lambda} \\ -a + \lambda \end{pmatrix} \quad (25)$$

respectively, whereby the vector elements are conjugated complex to each other ($\tilde{b}_{A1} = \overline{\tilde{b}_{A2}}$ and $\tilde{b}_{B1} = \overline{\tilde{b}_{B2}}$). To avoid redundant calculations in the ongoing analysis, the system is transformed such that the input gains $\tilde{b}_1$ and $\tilde{b}_2$ appear in the obtained output matrix:

$$\mathbb{S}^{cc} \equiv \left\{ \begin{array}{l} \begin{pmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \end{pmatrix} = \begin{pmatrix} \overline{\lambda} & 0 \\ 0 & \lambda \end{pmatrix}\begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix}v(t), \quad y(t) = \begin{pmatrix} \frac{\mathrm{j}\tilde{b}_1}{2\omega} \\ -\frac{\mathrm{j}\tilde{b}_2}{2\omega} \end{pmatrix}^T \begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \end{pmatrix} \\ \begin{pmatrix} \hat{x}_1(0) \\ \hat{x}_2(0) \end{pmatrix} = \begin{pmatrix} (x_{1,0} - \lambda x_{2,0})/\tilde{b}_1 \\ (x_{1,0} - \overline{\lambda} x_{2,0})/\tilde{b}_2 \end{pmatrix} \end{array} \right\}.$$

Two decoupled 1^{st}-order systems with the respective eigenvalues λ and $\overline{\lambda}$ result. The original output y is a weighted sum of the states x_1 and x_2 using complex-valued parameters. Hence, the result for the 1^{st}-order system (19) is re-used:

$$y_{\delta\Delta}[k] = \underbrace{\mathrm{j}\tilde{b}_1/(2\omega)}_{:=r\mathrm{e}^{\mathrm{j}\phi}} \;\mathfrak{Z}_{\boldsymbol{\delta\Delta}}\left[\left\{ \begin{array}{l} \dot{y}(t) = \overline{\lambda}y(t) + v(t) \\ y(0) = (x_{1,0} - \lambda x_{2,0})/\tilde{b}_1 \end{array} \right\}, u, t_\delta, t_\Delta\right]$$
$$+ \quad \underbrace{-\mathrm{j}\tilde{b}_2/(2\omega)}_{:=r\mathrm{e}^{-\mathrm{j}\phi}} \;\mathfrak{Z}_{\boldsymbol{\delta\Delta}}\left[\left\{ \begin{array}{l} \dot{y}(t) = \lambda y(t) + v(t) \\ y(0) = (x_{1,0} - \overline{\lambda} x_{2,0})/\tilde{b}_2 \end{array} \right\}, u, t_\delta, t_\Delta\right].$$

A simpler notation is obtained by describing the weighting factors in the polar coordinates r and ϕ. For $\mathbb{S}^{cc}_A$ and $\mathbb{S}^{cc}_B$

$$\begin{array}{rcl} r_A\mathrm{e}^{\phi_A} & := & \mathrm{j}\tilde{b}_{A,1}/(2\omega) \Rightarrow r_A = 1/(2\omega),\ \phi_A = \pi/2 \\ r_B\mathrm{e}^{\phi_B} & := & \mathrm{j}\tilde{b}_{B,1}/(2\omega) = \frac{1}{2} + \mathrm{j}\frac{\sigma - a}{2\omega} \end{array}$$
$$\Rightarrow r_B = \frac{1}{2}\sqrt{1 + [(\sigma - a)/\omega]^2},\ \phi_B = \tan^{-1}[(\sigma - a)/\omega] \quad (26)$$

is obtained. The sampling interval $T_s[k] := T_\delta[k-1] + T_\Delta[k]$ is defined to compact the notation. Evaluating $\mathfrak{Z}_{\boldsymbol{\delta\Delta}}$ for both

1^{st}-order systems with the eigenvalues λ and $\overline{\lambda}$ by using (22) yields the discrete-time state-space representation

$$\underbrace{\begin{pmatrix} z_1[k+1] \\ z_2[k+2] \end{pmatrix}}_{} = \underbrace{\begin{pmatrix} e^{\overline{\lambda} T_s[k]} & 0 \\ 0 & e^{\lambda T_s[k]} \end{pmatrix}}_{\underline{\boldsymbol{A}}^{cc}} \begin{pmatrix} z_1[k] \\ z_2[k] \end{pmatrix} + \underbrace{\begin{pmatrix} e^{\overline{\lambda} T_\Delta[k]} \\ e^{\lambda T_\Delta[k]} \end{pmatrix}}_{\underline{\boldsymbol{B}}^{cc}} u[k]$$

$$y_{\delta\Delta}[k] = \underbrace{\begin{pmatrix} r e^{\mathrm{j}\phi} & r e^{-\mathrm{j}\phi} \end{pmatrix}}_{\underline{\boldsymbol{C}}^{cc}} \begin{pmatrix} z_1[k] \\ z_2[k] \end{pmatrix} \quad \underbrace{\begin{pmatrix} z_1[0] \\ z_2[0] \end{pmatrix}}_{\underline{\boldsymbol{z_0}}^{cc}} = \begin{pmatrix} (x_{1,0} - \lambda x_{2,0})/\tilde{b}_1 \\ (x_{1,0} - \overline{\lambda} x_{2,0})/\tilde{b}_2 \end{pmatrix}.$$

Please note that the states z_1 and z_2 relate conjugated complex to each other ($\bar{z}_1 = z_2$). A state-space representation with purely real-valued states is dersireable to reduce computational complexity in computer controlled systems. Therefore, a state transformation is performed such that the obtained states represent imaginary and real part of z_2 respectively:

$$\boldsymbol{z}^{cc}[k] = \begin{pmatrix} z^R[k] \\ z^I[k] \end{pmatrix} := \begin{pmatrix} \Re\{z_2[k]\} \\ \Im\{z_2[k]\} \end{pmatrix} = \underbrace{\begin{pmatrix} 0.5 & 0.5 \\ 0.5\mathrm{j} & -0.5\mathrm{j} \end{pmatrix}}_{:=\underline{\boldsymbol{T}}} \begin{pmatrix} z_1[k] \\ z_2[k] \end{pmatrix}. \quad (27)$$

By applying this state transformation, a real-valued state-space system is obtained and the resulting discretisation is

$$\left\{\; y_{\delta\Delta} = \mathfrak{Z}_{\boldsymbol{\delta\Delta}}[\mathbb{S}^{cc}, u, t_\delta, t_\Delta][k] \;\right\} \quad (28)$$

$$\equiv \left\{ \begin{array}{lcl} \boldsymbol{z}^{cc}[k+1] & = & \boldsymbol{A}^{cc} \cdot \boldsymbol{z}^{cc}[k] + \boldsymbol{B}^{cc} u[k] \\ y_{\delta\Delta}[k] & = & \boldsymbol{C}^{cc} \cdot \boldsymbol{z}^{cc}[k] \\ \boldsymbol{z_{cc}}[0] & = & \boldsymbol{z_0^{cc}}, \quad \boldsymbol{z_{cc}} \in \mathbb{R}^2 \end{array} \right\},$$

whereby the matrices $\boldsymbol{A}^{cc}$. $\boldsymbol{B}^{cc}$, $\boldsymbol{C}^{cc}$ and the initial state vector $\boldsymbol{z_0^{cc}}$ are given by

$$\begin{array}{lcl} \boldsymbol{A}^{cc} & = & \boldsymbol{T}\underline{\boldsymbol{A}}^{cc}\boldsymbol{T}^{-1} = e^{\sigma T_s[k]} \begin{bmatrix} \cos(\omega T_s[k]) & -\sin(\omega T_s[k]) \\ \sin(\omega T_s[k]) & \cos(\omega T_s[k]) \end{bmatrix}, \\ \boldsymbol{B}^{cc} & = & \boldsymbol{T}\underline{\boldsymbol{B}}^{cc} = e^{\sigma T_\Delta[k]} \begin{bmatrix} \cos(\omega T_\Delta[k]) \\ \sin(\omega T_\Delta[k]) \end{bmatrix}, \\ \boldsymbol{C}^{cc} & = & \underline{\boldsymbol{C}}^{cc}\boldsymbol{T}^{-1} = [2r\cos\phi, 2r\sin\phi], \\ \boldsymbol{z_0^{cc}} & = & \boldsymbol{T}\underline{\boldsymbol{z_0}}^{cc}, \qquad T_s[k] := T_\delta[k-1] + T_\Delta[k]. \end{array}$$

For case A) ($\tilde{b}_2 = 1$), the initial state vector $\boldsymbol{z_0^{cc}}$ becomes

$$\boldsymbol{z_{A,0}^{cc}} = \begin{bmatrix} x_{1,0} - \sigma x_{2,0} \\ \omega x_{2,0} \end{bmatrix} \quad (29)$$

and in case B) ($\tilde{b}_2 = \tilde{b}_{B2}$), using (26):

$$\boldsymbol{z_{B,0}^{cc}} = \begin{bmatrix} z^R[0] \\ z^I[0] \end{bmatrix} = \begin{bmatrix} \sin\phi_B & \cos\phi_B \\ -\cos\phi_B & \sin\phi_B \end{bmatrix} \begin{bmatrix} x_{1,0} - \sigma x_{2,0} \\ \omega x_{2,0} \end{bmatrix} /(2\omega r_B). \quad (30)$$

For the discretisation of $\mathbb{S}_A^{cc}$ by using (28), the parameters r, ϕ and $\boldsymbol{z_0^{cc}}$ are respectively substituted by r_A, ϕ_A (26) and $\boldsymbol{z_{A,0}^{cc}}$ (29), while using r_B, ϕ_B (26) and $\boldsymbol{z_{B,0}^{cc}}$ (30) for $\mathbb{S}_B^{cc}$.

IV. Simulation Results

A continuous system of second order (Eq. (23)), $\lambda = -40 + \mathrm{j}30$, $x_{1,0} = 2$, $x_{2,0} = 0.01$) is discretised using $\mathfrak{Z}_{\boldsymbol{\delta\Delta}}$. Predefined sequences for t_δ, t_Δ and u were applied as shown in Fig. 3. The continuous-time system output y in response to the δ-actuation was numerically calculated by superposing the responses to each individual δ-pulse and the dynamics caused by the initial states. The time-discrete state-space model (28) was evaluated and the sampled values are shown.

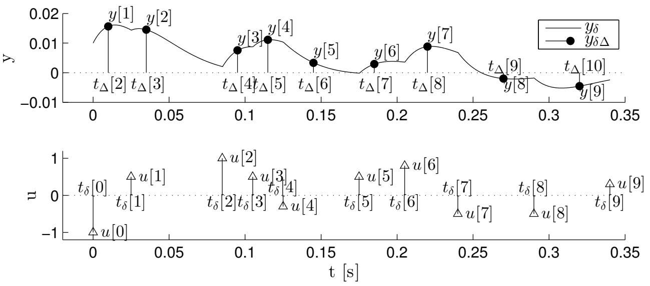

Fig. 3 - The I/O sequences of the discrete-time system along with the continuous-time output for a second-order system are shown.

V. Linearising feed-forward Controller

Since the resulting discrete-time representations are non-linear w.r.t. their coefficients, the classic linear control theory can not be directly applied. Therefore, an exact linearisation approach (system inversion) is presented that leads to a linear augmented plant. For simplicity reasons, T_δ is chosen to be zero $\forall k$ (actuation and sampling times are equal).

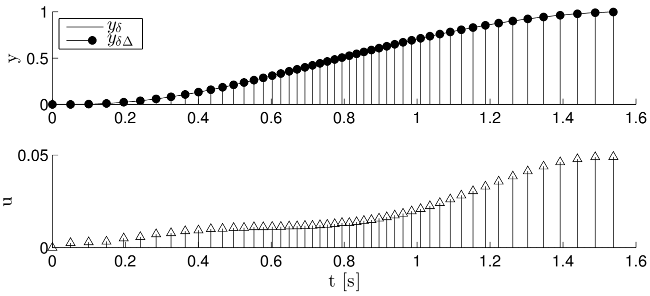

Fig. 4 - Results for an FES-system when applying the linearising feed-forward controller using irregular sampling-times. The sampled reference trajectory $\overline{v}[k] = r_y(t_\delta[k]) \forall k$ is matching the output $y_{\delta\Delta}$.

Most higher-order continuous-time systems can be decomposed and discretised using $\mathfrak{Z}_{\boldsymbol{\delta\Delta}}$ such that the state-space system

$$z[k+1] = \boldsymbol{A_{\delta\Delta}} z[k] + \boldsymbol{B_{\delta\Delta}} u[k] \quad (31)$$

$$y_{\delta\Delta}[k] = \boldsymbol{C_{\delta\Delta}} z[k] \quad (32)$$

is obtained, whereby the matrices are given by

$$\boldsymbol{A_{\delta\Delta}} = \mathrm{diag}\left[\boldsymbol{A_1^{cc}}, \boldsymbol{A_2^{cc}}, \cdots, a_1^1, a_2^1, \cdots\right] \quad (33)$$

$$z[k] := \begin{bmatrix} \boldsymbol{z_1^{cc}}[k] \\ \boldsymbol{z_2^{cc}}[k] \\ \vdots \\ z_1^1[k] \\ z_2^1[k] \\ \vdots \end{bmatrix} \qquad \boldsymbol{B_{\delta\Delta}} = \begin{bmatrix} \boldsymbol{B_1^{cc}} \\ \boldsymbol{B_2^{cc}} \\ \vdots \\ b_1^1 \\ b_2^1 \\ \vdots \end{bmatrix} \qquad \boldsymbol{C_{\delta\Delta}} = \begin{bmatrix} c_1^{cc}\boldsymbol{C_1^{cc}}^T \\ c_2^{cc}\boldsymbol{C_2^{cc}}^T \\ \vdots \\ c_1^1 \\ c_2^1 \\ \vdots \end{bmatrix}^T.$$

The matrices $\boldsymbol{A}_i^{cc}$, $\boldsymbol{B}_i^{cc}$, a_i^1 and b_i^1 are the coefficients of the discrete-time representations of all partial continuous-time systems obtained due to the decomposition. They depend on the time interval $T_\Delta[k]$. The discrete-time states are z_i^{cc} and z_1^1, while c_1^{cc} and c_1^1 are decomposition's weighting factors.

The overall output of the discrete-time system for the next time step is then

$$y_{\delta\Delta}[k+1] = \boldsymbol{C}_{\delta\Delta}\boldsymbol{A}_{\delta\Delta}z[k] + \boldsymbol{C}_{\delta\Delta}\boldsymbol{B}_{\delta\Delta}u[k]. \quad (34)$$

By applying the linearising feed-forward controller

$$u[k] = \frac{1}{\boldsymbol{C}_{\delta\Delta}\boldsymbol{B}_{\delta\Delta}}\left[-\boldsymbol{C}_{\delta\Delta}\boldsymbol{A}_{\delta\Delta}z[k] + \overline{v}[k]\right], \quad (35)$$

an exact linearisation $y_{\delta\Delta}[k+1] = \overline{v}[k]$ with regard to the virtual input (and new actuation variable) $\overline{v}$ is obtained. The therefor required state information $z[k]$ can be obtained either by Eq. (31) yielding in a feed-forward control (only possible for asymptotically stable plants) or by means of an observer. As the relative degree of the discrete-time system is always one whereby the system's order may be higher, a potentially instable zero-dynamics may be present. Hence stability must be investigated in the individual case.

1) Example for an FES-plant: A continuous-time model described by the transfer function (rise time 0.3 s, 0.25 % overshoot)

$$G(s) = \frac{c}{(s - s_\infty)(s^2 - 2\sigma s + \sigma^2 + \omega^2)}, \quad \begin{array}{c}\sigma = -3,\, \omega = 10, \\ c = 1090,\, s_\infty = -10\end{array}$$

is considered that describes the typical response of a joint angle (e.g. the shoulder abduction) to electrical stimulation. All initial values are assumed to be zero. The model is discretised using $\mathcal{Z}_{\delta\Delta}$ and transformed to match (31, 32) using partial fraction decomposition. Then, the linearising controller (35) is applied to u, whereby the required state information was calculated by (31). In this example, a fixed, cosine shaped sequence for the sampling frequency ranging from 20 to 50 Hz is defined yielding T_Δ ($T_\delta = 0$) and t_δ. Further, a continuous-time reference trajectory is defined by $r(t) = -(\cos(\pi t_\delta / T_r) - 1)/2$, $T_r = 1.5$ s and sampled at t_δ. The resulting sequence is applied to the linearising controller $\overline{v}[k] = r(t_\delta[k])$ and the obtained I/O sequences and the continuous-time output are shown in Fig. 4.

VI. Conclusions and Outlook

A method for the time-discretisation of continuous systems under variable actuation and sampling times is presented and analytically evaluated yielding discrete-time state-space systems with non-constant coefficients. By a simulation example relevant to the control of neuro-prosthetic systems, it could be demonstrated that the obtained gain-scheduled systems can be controlled by an inversion approach yielding an augmented linear dead-beat system. Time delays will be considered in the future by a time shift of the actuation pulse. The linearising feed-forward controller will be extended by a feedback controller and the stability of the zero-dynamics must be investigated in general. Further, the analysis of aliasing-effects and the inter-sampling behaviour are up to future investigations.

References

[1] Naif B Almutairi, Mo-Yuen Chow, and Yodyium Tipsuwan. Network-based controlled DC motor with fuzzy compensation. In *Industrial Electronics Society, 2001. IECON'01. The 27th Annual Conference of the IEEE*, volume 3, pages 1844–1849. IEEE, 2001.

[2] Adolfo Anta and Paulo Tabuada. Self-triggered stabilization of homogeneous control systems. In *American Control Conference, 2008*, pages 4129–4134. IEEE, 2008.

[3] Karl-Erik Årzén. A simple event-based PID controller. In *Proc. 14th IFAC World Congress*, volume 18, pages 423–428, 1999.

[4] Karl J Aström. Event based control. In *Analysis and design of nonlinear control systems*, pages 127–147. Springer, 2008.

[5] Mauricio G Cea and Graham C Goodwin. Event based sampling in non-linear filtering. *Control Engineering Practice*, 20(10):963–971, 2012.

[6] Huijun Gao and Tongwen Chen. Stabilization of nonlinear systems under variable sampling: a fuzzy control approach. *Fuzzy Systems, IEEE Transactions on*, 15(5):972–983, 2007.

[7] G.C. Goodwin, J.C. Agüero, M. Garrido, and J.I. Yuz. Sampling and sampled-data models. *IEEE control systems magazine*, 33(5):34–53, 2013.

[8] WPMH Heemels, JH Sandee, and PPJ Van Den Bosch. Analysis of event-driven controllers for linear systems. *International journal of control*, 81(4):571–590, 2008.

[9] Dan Henriksson and Anton Cervin. Optimal on-line sampling period assignment for real-time control tasks based on plant state information. In *Proc. of the 44th IEEE Conference on Decision and Control and in Proc. of the European Control Conference (CDC and ECC'05).*, pages 4469–4474. IEEE, 2005.

[10] Joao P Hespanha, Payam Naghshtabrizi, and Yonggang Xu. A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1):138–162, 2007.

[11] Trisha Kesar and Stuart Binder-Macleod. Effect of frequency and pulse duration on human muscle fatigue during repetitive electrical stimulation. *Experimental physiology*, 91(6):967–976, 2006.

[12] Zheng Li and Mo-Yuen Chow. Adaptive multiple sampling rate scheduling of real-time networked supervisory control system-part ii. In *IEEE Industrial Electronics, IECON 2006-32nd Annual Conference on*, pages 4615–4620. IEEE, 2006.

[13] Jan Lunze and Daniel Lehmann. A state-feedback approach to event-based control. *Automatica*, 46(1):211–215, 2010.

[14] Sébastien Mariéthoz and Manfred Morari. Explicit model-predictive control of a PWM inverter with an LCL filter. *Industrial Electronics, IEEE Transactions on*, 56(2):389–399, 2009.

[15] Farokh Marvasti. *Nonuniform sampling: theory and practice*, volume 1. Springer, 2001.

[16] P Hunter Peckham and Jayme S Knutson. Functional electrical stimulation for neuromuscular applications. *Annual Review of Biomedical Engineering*, 7:327–360, 2005.

[17] R. Riener. Model–based Development of Neuroprostheses for Paraplegic Patients. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 354(1385):877–894, 1999.

[18] Robert Riener and Jochen Quintern. A physiologically based model of muscle activation verified by electrical stimulation. *Bioelectrochemistry and Bioenergetics*, 43(2):257–264, 1997.

[19] Michael Schinkel, Wen-Hua Chen, and Anders Rantzer. Optimal control for systems with varying sampling rate. In *American Control Conference, 2002. Proceedings of the 2002*, volume 4, pages 2979–2984. IEEE, 2002.

[20] Rui N Silva, João M Lemos, and Luís M Rato. Variable sampling adaptive control of a distributed collector solar field. *Control Systems Technology, IEEE Transactions on*, 11(5):765–772, 2003.