Digital control of multiple discrete passive plants over networks

N. Kottenstette*

Institute for Software Integrated Systems,
Vanderbilt University,
P.O. Box 1829, Station B, Nashville, TN 37203, USA
E-mail: nkottens@isis.vanderbilt.edu
*Corresponding author

Joseph F. Hall III and X. Koutsoukos

Department of Electrical Engineering and Computer Science,
Vanderbilt University,
P.O. Box 1829, Station B, Nashville, TN 37203, USA
E-mail: joe.hall@vanderbilt.edu
E-mail: xenofon.koutsoukos@vanderbilt.edu

Panos Antsaklis

Department of Electrical Engineering,
University of Notre Dame,
Notre Dame, IN 46556
E-mail: antsaklis.1@nd.edu

J. Sztipanovits

Department of Electrical Engineering and Computer Science,
Vanderbilt University,
P.O. Box 1829, Station B, Nashville, TN 37203, USA
E-mail: janos.sztipanovits@vanderbilt.edu

Abstract: This paper provides a passivity based framework to synthesise l_2^m*-stable* digital control networks in which m *strictly-output passive* controllers can control $n-m$ *strictly-output passive* plants. The communication between the plants and controllers can tolerate time varying delay and data dropouts. In particular, we introduce a *power-junction-network*, a general class of *input-output-wave-variable-network* which allows even a single controller (typically designed to control a single plant) to accurately control the output of multiple plants even if the corresponding dynamics of each plant is different. In addition to the power-junction-network we also introduce a Passive Downsampler (PDS) and Passive Upsampler (PUS) in order to further reduce networking traffic while maintaining stability and tracking properties. A detailed (soft real-time) set of examples shows the tracking performance of the networked control system.

Keywords: power-junction-network; passivity; dissipative-systems; wave-variables; scattering theory; networked control; PDS; passive downsampler; PUS; passive upsampler.

Reference to this paper should be made as follows: Kottenstette, N., Hall III, J.F., Koutsoukos, X., Antsaklis, P.J. and Sztipanovits, J. (2011) 'Digital control of multiple discrete passive plants over networks', *Int. J. Systems, Control and Communications*, Vol. 3, No. 2, pp.194–228.

Biographical notes: Nicholas Kottenstette is currently a Research Scientist within ISIS at Vanderbilt University. A Senior Member of IEEE, he holds a MS from the Mechanical Engineering Department at MIT and a PhD in Electrical Engineering from The University of Notre Dame. He is a (co)-author of over 20 publications and (co)-inventor of numerous products resulting in 11 US patents related to design and control of (networked) embedded systems. Using passivity-based fundamentals to approach digital-networked control design of cyber-physical systems, he is tackling challenging problems including high confidence design and coordinated networked control of (quad-rotor) aircraft and robotic systems.

Joseph F. Hall III holds a BS in Engineering, and a BS in Computer Science from Union University and a MS Degree in Electrical Engineering from Vanderbilt University. His initial-graduate work focused on Cognitive Control in Humanoid Robotics which culminated in a thesis focused on an Internal Rehearsal System for the Central Executive Agent which was designed and implemented in the CIS Lab at Vanderbilt University. This allowed ISAC, the Cognitive Robot, to try certain behaviours internally and ascertain consequences before executing the behaviour in real life. His research interests include robotic cognitive control, manipulator kinematics and dynamics and digital control using passivity-based-techniques.

Xenofon D. Koutsoukos holds a Diploma in Electrical and Computer Engineering from the National Technical University of Athens, Greece, a MS Degree in Electrical Engineering and Applied Mathematics, and a PhD Degree in Electrical Engineering from the University of Notre Dame. He is an Associate Professor and Senior Research Scientist within ISIS at Vanderbilt University, his research interests include hybrid, real-time embedded and cyber-physical systems. He currently serves as an AE for the *ACM Transactions on Sensor Networks*, *Modelling Simulation Practice and Theory*, and the *International Journal of Social Computing and Cyber-Physical Systems*. He is a Senior Member of the IEEE.

Panos J. Antsaklis is the Brosey Professor of Electrical Engineering at the University of Notre Dame. He is a Graduate of the National Technical University of Athens, Greece, and holds MS and PhD Degrees from Brown University. His recent research focuses on networked embedded systems and addresses problems in the interdisciplinary research area of control, computing and communication networks, and on hybrid and discrete event dynamical systems. He is an IEEE Fellow and the 2006 recipient of the Engineering Alumni Medal of Brown University. He is currently the Editor-in-Chief of the *IEEE Transactions on Automatic Control*.

Janos Sztipanovits is the E. Bronson Ingram Distinguished Professor of Engineering at Vanderbilt University. The Founding Director of ISIS, his research interests include the foundations and applications of Model-Integrated Computing. He was the founding chair of the ACM Special Interest Group on Embedded Software (SIGBED). He is a Fellow of the IEEE. He won the National Prize in Hungary (1985) and the Golden Ring of the Republic (1982) for science and engineering achievements. He graduated (Summa Cum Laude) from the Technical University of Budapest and received his doctorate from the Hungarian National Academy of Sciences.

1 Introduction

The primary goal of our research is to develop reliable wireless control networks (Antsaklis and Baillieul, 2004, 2007). In the past we have shown numerous results related to the control of a single plant with a single controller over a network. In particular we have shown how to create a l_2^m-*stable* control network for a continuous *passive* plant (Kottenstette and Antsaklis, 2007, Theorem 4). The key is to transmit control and sensor data in the form of *wave variables* over networks similar to those depicted in Kottenstette and Antsaklis (2007, Figure 2). The use of *wave variables* allows the network to remain l_2^m-*stable* when subject to both fixed time delays and data dropouts (Kottenstette and Antsaklis, 2007, Lemma 2). In addition, if duplicate wave variable transmissions are dropped, then the network will remain l_2^m-*stable* in spite of time varying delays (Kottenstette and Antsaklis, 2007, Lemma 3). *It is not immediately clear how to apply these results to the control of multiple plants with* (*possibly multiple*) *controller*(s).

The main research challenge is to develop a formal way to *construct* a control network in which multiple plants and controllers can be interconnected such that the overall system remains stable and can change how the plants behave. This stability should be guaranteed in spite *random* time delays and data dropouts which are inherent to wireless networks. Furthermore we would like our statement on stability to have a *deterministic* characteristic such as either L_2^m or l_2^m stability (see (Kottenstette and Antsaklis, 2008c) in regards to how l_2^m stability and (Kottenstette et al., 2008) in regards to how L_2^m stability can be achieved in spite of random time delays and data dropouts for a single-plant-single-controller architecture). In regards to changing the plants behaviour we would like to show that the plants can tolerate disturbances and track a desired set-point as quickly and as closely as possible. This paper shows how a *power-junction-network* can address this problem.

The *power-junction-network* is a networking abstraction to interconnect wave variables from multiple controllers and plants such that the total wave-power-input is always greater than or equal to the total wave-power-output. Interconnecting wave variables in a 'power preserving' manner has appeared in the telemanipulation literature to augment potential position drift by modifying one of the waves u_m in a *passive* manner (Niemeyer and Slotine, 2004, Figure 9). Other abstractions to interconnect wave variables have also appeared in the wave digital filtering

literature which is primarily-concerned with structural synthesis rules to take a continuous-time reference filter in order to construct a discrete-time digital filter which possesses good properties concerning coefficient accuracy requirements, dynamic range, and stability properties in regards to finite-arithmetic (Fettweis, 1986). In Fettweis (1986) it is shown how through applying the bilinear-transform to a small set of continuous-time LTI system models (inductor, capacitor, resistor) that various stable-wave-digital-filters can be realised via networks involving wave ports. For example, in Kottenstette and Antsaklis (2007, Figure 2) the waves $u_{op} \in \mathbb{R}^m$ and $u_{oc} \in \mathbb{R}^m$ are each computed in a manner similar to a voltage *incident* wave (a), and the waves $v_{op} \in \mathbb{R}^m$ and $v_{oc} \in \mathbb{R}^m$ are each computed in a manner similar to a voltage *reflective* wave (b) Fettweis (1986). For wave digital filters a voltage *incident* waves can be thought of as a wave travelling into a two port junction, likewise a *reflective* wave travels out of a two port junction. When interconnecting two port elements for a wave digital filter, a voltage *incident* wave should connect to a voltage *reflective* wave or vice versa (Fettweis, 1986, Section IV-A-2). If we denote u_{op} and v_{oc} as *reflective* waves (with outgoing arrows) and denote u_{oc} and v_{op} as *incident* waves (with incoming arrows), then the interconnection rules appear to be in agreement. In Fettweis (1986, Section IX-H) it is noted that the use of power-waves for linear wave-digital-filter synthesis is equivalent to using voltage waves. However, the use of voltage waves does not allow one to study the interconnection of nonlinear passive systems, which this work does address. It should be appreciated that unlike wave-digital-filtering literature, we do not attempt to study special cases involving constructive rules to realise a high-Q filter, for example. On the contrary, we are concerned with how passive (non) linear discrete plants can be interconnected to passive (non) linear discrete controllers while guaranteeing tracking and stability inspite of time-(varying-)delays and data loss. Some work has appeared as it relates to Lyapunov stability in regards to consensus networks involving wave variables, continuous-time feed back among passive continuous-time plants (Chopra and Spong, 2006). To the best of our knowledge, this is the first work of its kind as it pertains to interconnecting digital controllers to multiple discrete time plants over a wave-variable network in a *negative feed-back* manner in which *weak* time varying delay conditions are only needed in order to guarantee l_2^m-stability in-spite of data-loss, in addition, tracking performance for LTI systems is verified. In this paper we show how power-junction-networks make it possible to allow m controllers to control up to $n - m$ plants. We prove that such a network can be shown to be l_2^m-*stable* if all the interconnected plants and controllers are *strictly-output passive*. This paper is a significant refinement of our earlier work in which we initially presented the power-junction-network (Kottenstette and Antsaklis, 2008a). In particular, Definition 2 is formally stated to handle the interconnection of m_s-dimensional waves. We also present the *averaging-power-junction-network* (Definition 3) and formally show how it satisfies the conditions required to be a power-junction-network (Lemma 2). Such a presentation is done to encourage others to create their own specific power-junction-network implementation and show how it satisfies Definition 2. In addition, this paper further introduce a *Passive Upsampler* (PUS) and *Passive Downsampler* (PDS) in order to further reduce the amount of digital control traffic, while maintaining a stable system. In order to simplify discussion with this particular paper, we will focus our presentation to the discrete form of stability (l_2^m-stability). However, remarks will be made which

show how continuous time plants can be integrated into a power-junction-control-network using a Passive Sampler (PS) and Passive Hold (PH) which is L_2^m-stable (Kottenstette et al., 2008).

Other refinements of this paper include a detailed set of soft real-time experimental results. In which multiple discrete time passive plants are controlled by a single controller over an ad-hoc wireless network. In particular, each plant is the passive-discrete-time equivalent of a simple mass (of different weight) which was transformed from the continuous time model using the *IPESH-Transform* (Definition 5) which consists of using an *Inner-Product Equivelant Sampler* (*IPES*) and Zero-Order Hold (*ZOH*) (Kottenstette and Antsaklis, 2007, Definition 4). The timing for each discrete time plant is maintained by a (soft) real time timer which is part of an advanced passivity based control library which runs on MATLAB/Simulink (MathWorks, 2008a, 2008b). Each plant can be thought of as a client which connects to the power-junction-network-server. The overall client server architecture used the UDP protocol because of its connectionless nature so that plants could easily connect and disconnect without 'stopping' the system. This convenient architecture was easily adapted to use a secure shell `ssh`-tunnelling mechanism (Ylonen and Lonvick, 2006), such that we could evaluate running the system in which the plants and controller were located in different areas throughout the world. Finally, we evaluated the system when subject to network attacks. Although multiple controllers can be used in this frame-work we chose not to focus on this case so as to establish a more complete simulation, the interested reader is referred to Kottenstette and Antsaklis (2008a) and Kottenstette et al. (2009) for additional results related to interconnecting multiple controllers over either an averaging-power-junction-network or *resilient-power-junction-network* respectively.

The rest of the paper is organised as follows:

- Section 2 presents all that is required to design network control systems for multiple-plants and multiple-controllers over a power-junction-network (Section 2.1) and the PUS and PDS (Section 2.2) which are l_2^m stable (Section 2.3)
- Section 3 presents a detailed experiment in which two 'soft-real-time' simulated plants are controlled over an ad-hoc wireless network by a single controller which is connected over an averaging-power-junction-network
- Section 4 provides our conclusions and a more specific summary of our contributions
- Appendix A provides a review on passivity while Appendix B provides detailed proofs for many of the results presented in this paper.

2 Networked control design

2.1 Power-junction-networks

Networks of a *passive* plant and controller are typically interconnected using *power variables. Power variables* are generally denoted with an *effort* and *flow* pair (e_*, f_*) whose product is power. They are typically used to show the exchange

of energy between two systems using *bond graphs* (Breedveld, 2006; Golo et al., 2003). However, when these *power variables* are subject to communication delays the communication channel ceases to be *passive* which leads to network instabilities. Wave variables allow *effort* and *flow* variables to be transmitted over a network while remaining *passive* when subject to arbitrary fixed time delays and data dropouts (Niemeyer and Slotine, 2004)

$$u_{pk}(i) = \frac{1}{\sqrt{2b}}(bf_{opk}(i) + e_{dock}(i)), \quad k \in \{m+1, \ldots, n\} \tag{1}$$

$$v_{cj}(i) = \frac{1}{\sqrt{2b}}(bf_{opdj}(i) - e_{ocj}(i)), \quad j \in \{1, \ldots, m\}. \tag{2}$$

Equation (1) can be thought of as each sensor output in a *wave variable* form for each plant G_{pk}, $k \in \{m+1, \ldots, n\}$ depicted in Figure 2. Likewise, equation (2) can be thought of as each command output in a *wave variable* form for each controller G_{cj}, $j \in \{1, \ldots, m\}$ depicted in Figure 2. The symbol $i \in \{0, 1, \ldots\}$ depicts discrete time. Denote $I \in \mathbb{R}^{m_s \times m_s}$ as the identity matrix. When actually implementing the wave variable transformation the 'outputs' (u_{pk}, e_{dock}) are related to the corresponding 'inputs' (v_{pk}, f_{opk}) as follows (see (Kottenstette, 2007, Figure 2.2)):

$$\begin{bmatrix} u_{pk}(i) \\ e_{dock}(i) \end{bmatrix} = \begin{bmatrix} -I & \sqrt{2b}I \\ -\sqrt{2b}I & bI \end{bmatrix} \begin{bmatrix} v_{pk}(i) \\ f_{opk}(i) \end{bmatrix} \tag{3}$$

likewise the 'outputs' (v_{cj}, f_{opdj}) are related to the corresponding 'inputs' (u_{cj}, e_{ocj}) as follows:

$$\begin{bmatrix} v_{cj}(i) \\ f_{opdj}(i) \end{bmatrix} = \begin{bmatrix} I & -\sqrt{\frac{2}{b}}I \\ \sqrt{\frac{2}{b}}I & -\frac{1}{b}I \end{bmatrix} \begin{bmatrix} u_{cj}(i) \\ e_{ocj}(i) \end{bmatrix}.$$

The *power-junction-network*, a special type of *io-wave-variable-network*, indicated in Figures 1 and 2 by the symbol **PJ** has waves both entering and leaving the power-junction-network as indicated by the arrows. Waves leaving the controllers v_{cj} and entering the power-junction-network v_j in which $j \in \{1, \ldots, m\}$ have the following relationship

$$v_j(i) = v_{cj}(i - pj(i))$$

in which $pj(i)$ denotes the time varying delay in transmitting the control wave from 'controller-j' to the power-junction-network. Next, the input wave to the plant v_{pk} is a delayed version of the outgoing wave from the power-junction-network v_k, $k \in \{m+1, \ldots, n\}$ such that

$$v_{pk}(i) = v_k(i - pk(i)), \quad k \in \{m+1, \ldots, n\}$$

in which $pk(i)$ denotes the discrete time varying delay in transmitting the outgoing wave to 'plant-k'. In Figure 2 the delays are represented as fixed for the discrete

Figure 1 An *io-wave-variable-network* of $m = 2$ pairs of *power-output-waves* and $n - m = 4 - 2 = 2$ pairs of *power-input-waves* depicted by the symbol **PJ** indicating it satisfies (4) in order to be a *power-junction-network*

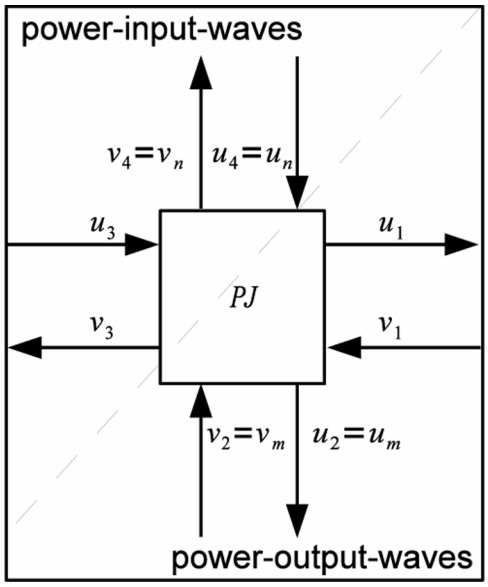
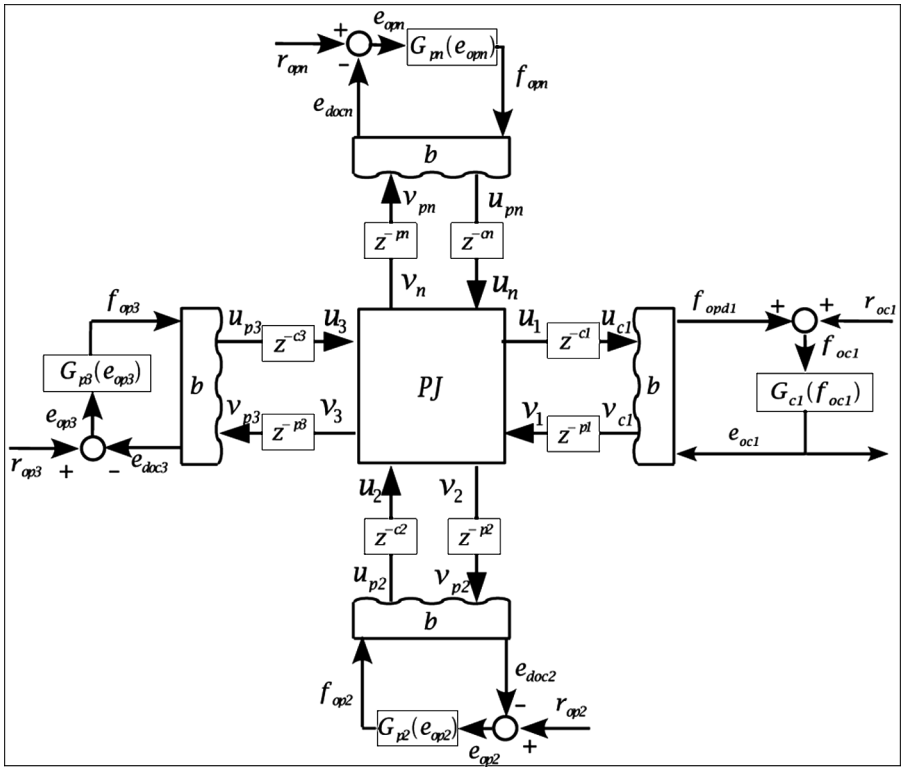


Figure 2 An example of a *power-junction-control-network*



time case (i.e., z^{-pk}). Next, the outgoing wave from each plant u_{pk} is related to the wave entering the power-junction-network u_k, $k \in \{m+1, \ldots, n\}$ as follows:

$$u_k(i) = u_{pk}(i - ck(i)), \quad k \in \{m+1, \ldots, n\}$$

in which $ck(i)$ denotes the discrete time varying delay in transmitting the wave from 'plant-k' to the power-junction-network. Last, the input wave to the controller u_{cj} is a delayed version of the outgoing wave from the power-junction-network u_j, $j \in \{1, \ldots, m\}$ such that

$$u_{cj}(i) = u_j(i - cj(i)), \quad j \in \{1, \ldots, m\}$$

in which $cj(i)$ denotes the discrete time varying delay in transmitting the wave from the power-junction-network to 'controller-j'. In Figure 2 the delays are represented as fixed for the discrete time case (i.e., z^{-cj}). Before, providing a formal definition for a power-junction-network, we define *input-output-wave-variable-networks*, a special class of wave-variable-networks.

Definition 1: An *input-output-wave-variable-network* (io-wave-variable-network) is any network (such as the network depicted in Figure 1) which interconnects n systems (in which $1 \leq m < n < \infty$) with the corresponding wave variable pairs $(u_1, v_1), (u_2, v_2), \ldots, (u_n, v_n)$ such that the *power-output-wave* pairs are denoted (u_j, v_j), $j \in \{1, \ldots, m\}$ (in which $u_j \in \mathbb{R}^{m_s}$ is an outgoing-power-output-wave and $v_j \in \mathbb{R}^{m_s}$ is an incoming-power-output-wave) and the *power-input-wave* pairs are denoted (u_k, v_k), $k \in \{m+1, \ldots, n\}$ (in which $u_k \in \mathbb{R}^{m_s}$ is an incoming-power-input-wave and $v_k \in \mathbb{R}^{m_s}$ is an outgoing-power-input-wave from the network).

Wave-variables in these networks denoted by the symbol u_* (v_*) will sometimes be referred to as power-output-u (v)-waves or power-input-u (v)-waves. We now provide a formal definition for the *power-junction-network*.

Definition 2: A *power-junction-network* is any io-wave-variable-network (Definition 1) such that the *passive* inequality

$$\sum_{k=m+1}^{n} \left(u_k^{\mathsf{T}} u_k - v_k^{\mathsf{T}} v_k\right) \geq \sum_{j=1}^{m} \left(u_j^{\mathsf{T}} u_j - v_j^{\mathsf{T}} v_j\right) \tag{4}$$

always holds. In other words, a power-junction-network is an io-wave-variable-network in which the total *wave-power-input* is always greater than or equal to the total *wave-power-output*. A *lossless-power-junction network* is a power-junction-network in which (4) is always satisfied with an equality.

Power-junction-networks provide a new way to interconnect multiple plants to multiple controllers. Figure 2 depicts $m = 1$ controller G_{c1} with the corresponding wave variables (u_{c1}, v_{c1}), and each plant G_{pk}, $k \in \{2, \ldots, n = 4\}$ has the corresponding wave variables (u_{pk}, v_{pk}). v_{c1} represents the wave-variable-control-output. u_{c1} represents a delayed feedback term which depends on the type of power-junction-network implemented and the corresponding wave-variable sensor

outputs u_{pk} from the remaining $n-1$ plants. Finally, for each plant v_{pk} represents the corresponding delayed control-command which depends on the type of power-junction-network implemented and v_{c1}.

There are many ways to realise a power-junction-network, in order to focus our discussion to a particular realisation of a power-junction-network we present Lemma 1 which allows us to focus on satisfying two respective inequalities relating to the scalar components of a given set of u-waves and a given set of v-waves which are sufficient to create a power-junction-network.

Lemma 1: *Any io-wave-variable-network* (*Definition* 1) *in which the power-output-waves* (u_j, v_j), $j \in \{1, \ldots, m\}$ *and power-input-waves* (u_k, v_k), $k \in \{m+1, \ldots, n\}$ *are combined in such a manner such that each* l*th scalar component* (*in which* $l \in \{1, \ldots, m_s\}$) *of the outgoing* m_s*-dimensional power-output-u-waves* u_{j_l} *are related to their respective incoming components of the power-input-u-waves* u_{k_l} *such that*

$$\sum_{j=1}^{m} u_{j_l}^2 \leq \sum_{k=m+1}^{n} u_{k_l}^2 \quad \forall l \in \{1, \ldots, m_s\} \tag{5}$$

always holds in addition each l*th scalar component of the outgoing power-input-v-waves* v_{k_l} *are related to the incoming components of the power-output-v-waves* v_{j_l} *such that*

$$\sum_{k=m+1}^{n} v_{k_l}^2 \leq \sum_{j=1}^{m} v_{j_l}^2 \quad \forall l \in \{1, \ldots, m_s\} \tag{6}$$

always holds then Definition 2 is satisfied.

The proof of Lemma 1 is in Appendix B.1.

Definition 3: An *averaging-power-junction-network* is any io-wave-variable-network (Definition 1) such that each lth component ($l \in \{1, \ldots, m_s\}$) of the outgoing-power-input-wave v_k (denoted v_{k_l}) are computed from the respective lth component of the incoming-power-output-wave v_j (denoted v_{j_l}) as follows:

$$v_{k_l} = \mathtt{sgn}\left(\sum_{j=1}^{m} v_{j_l}\right) \frac{\sqrt{\sum_{j=1}^{m} v_{j_l}^2}}{\sqrt{n-m}}, \quad k \in \{m+1, \ldots, n\}. \tag{7}$$

Similarly, each lth component ($l \in \{1, \ldots, m_s\}$) of the outgoing-power-output-wave u_j (denoted u_{j_l}) are computed from the respective lth component of the incoming-power-input-wave u_k (denoted u_{k_l}) as follows:

$$u_{j_l} = \mathtt{sgn}\left(\sum_{k=m+1}^{n} u_{k_l}\right) \frac{\sqrt{\sum_{k=m+1}^{n} u_{k_l}^2}}{\sqrt{m}}, \quad j \in \{1, \ldots, m\}. \tag{8}$$

Note, that for the *special case* when $m = 1$ then equations (7) and (8) respectively simplify to

$$v_{k_l} = \frac{v_{1_l}}{\sqrt{n-1}}, \quad k \in \{2, \ldots, n\}$$

$$u_{1_l} = \mathtt{sgn}\Big(\sum_{k=2}^{n} u_{k_l}\Big)\sqrt{\sum_{k=2}^{n} u_{k_l}^2}.$$

Lemma 2: *The averaging-power-junction-network* (*Definition* 3) *satisfies the inequality in* (4) *in order to be a power-junction-network* (*Definition* 2)*, furthermore it satisfies* (4) *as an equality and is therefore a lossless-power-junction-network.*

The proof of Lemma 2 is in Appendix B.2.

The engineer will need to scale the control input r_{ocj} in an appropriate manner, in order for the outputs f_{opk} of each plant to track the desired control input r_{osj} at steady-state. The following scaling relationship is proposed in which the scalar gain k_{pj} is used to account for the affects of a given power-junction-network implementation, and the scalar gain K_M is used to account for the scaling effects of the PUS and PDS.

$$r_{ocj} = -k_s r_{osj} = -(k_{\mathsf{pj}} K_M) r_{osj}. \tag{9}$$

When using the averaging-power-junction, the relationships can be quite complex, however, it is indeed possible to formulate a recursive structure to determine steady-state responses based on steady-state gains and steady-state inputs for a given plant-controller structure, as was done recently for averaging-power-junction-networks which interconnected continuous-time-plants to digital controllers (Kottenstette and Chopra, 2009, Theorem 16). In general we would like to consider the case when m identical controllers with identical references are used to command $n - m$ plants with identical steady-state gains. Assuming that the product of the steady-state gains for one plant and one controller is large then the scaling-gain k_{pj} should be computed such that

$$k_{\mathsf{pj}} = \sqrt{\frac{n-m}{m}} \tag{10}$$

in order for $r_{osj} = f_{opk}$ at steady-state when no PUS or PDS are used ($K_M = 1$). Note, that it is indeed the case that when the number of controllers equals the number of plants $k_{\mathsf{pj}} = 1$. In other words, k_{pj} equals the square-root of the ratio of the number of plants to the number of controllers. Such a relationship implies some resiliency to controller loss as was studied in Kottenstette et al. (2009) for the special-case when m redundant controllers, controlled a single plant over a *resilient-power-junction-network*.

Remark 1: For simplicity we will consider the case in which $r_{opk} = 0$ and all plants G_{pk} are single-input single-output satisfying:

$$f_{opk}(i) = -k_{pk} e_{dock}(i), \quad k_{pk} > 0$$

from equation (3) we see that:

$$e_{dock}(i) = -\sqrt{2b}v_{pk}(i) - bk_{pk}e_{dock}(i)$$

therefore,

$$f_{opk}(i) = -k_{pk}(i)e_{dock}(i) = \frac{k_{pk}\sqrt{2b}}{1 + bk_{pk}}v_{pk}(i).$$

If $(bk_{pk} >> 1)$, $\forall k \in \{m+1, m+2, \ldots, n\}$ then

$$f_{opk}(i) \approx \sqrt{\frac{2}{b}}v_{pk}(i).$$

This implies that as long as each plant processes the average wave commands from the controllers satisfying (8) for example, then as the system reaches a steady state $v_{pk}(i) = 0$, $\forall i > i_S$ and the delays are fixed then the following will approximately hold for some real constant C:

$$\sqrt{\frac{b}{2}}\sum_{i=0}^{i_s} f_{opk}(i) \approx \sum_{i=0}^{i_s} v_{pk}(i) = C.$$

Furthermore these tracking-like properties of each system connected to a *power-junction-network* can be extended to consider LTI systems in the frequency-domain in which the frequency content of $v_{pk}(e^{j\omega})$ is bandwidth limited such that

$$v_{pk}(e^{j\omega}) \approx 0, \quad \text{when } \omega_M < \omega \leq \pi$$
$$bH_{pk}(e^{j\omega}) >> 1, \quad \text{when } 0 \leq \omega \leq \omega_M.$$

Remark 2: Power-junction-networks complement prior work related to telemanipulation as summarised in Niemeyer and Slotine (2004, Section 6.4). In particular, a method is described showing how to augment potential position drift by modifying one of the waves u_m in a *passive* manner (Niemeyer and Slotine, 2004, Figure 9).

2.2 The Passive Up/Downsamplers

In Kottenstette et al. (2008) it was shown how a Passive Sampler (PS) and Passive Hold (PH) could be used to achieve a L_2^m-stable system for a passive robot and a digital controller. Clearly, these devices could be introduced into Figure 2 to create an overall L_2^m-stable system. In fact, this initial observation presented in this paper resulted in the L_2^m-stability and passivity theorem for digital control of continuous-time plants interconnected over power-junction-networks (Kottenstette and Chopra, 2009, Theorem 12). However, since our discussion is focused on discrete-time systems, we will now introduce the Passive Upsampler (PUS) and Passive Downsampler (PDS).

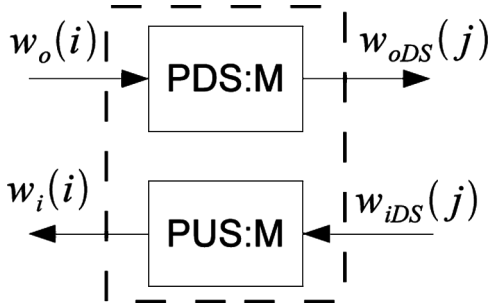
Definition 4: Figure 3 represents the Passive Upsampler (PUS) and Passive Downsampler (PDS) construction. $w_o(i)$ denotes a discrete wave variable going

out of a wave transform block, for example in Figure 2 $v_{c1}(i)$, $u_{p2}(i)$, $u_{p3}(i)$, $u_{pn}(i)$ are all unique $w_o(i)$'s. Similarly, $w_i(i)$ represents the respective discrete wave variable going in to a wave transform block, for example in Figure 2 $u_{c1}(i)$, $v_{p2}(i)$, $v_{p3}(i)$, $v_{pn}(i)$ are all unique $w_i(i)$'s. Downsample index $j = \lfloor \frac{i}{M} \rfloor$, therefore, we use the notation, $w_{oDS}(j)$ to represents the effective downsampled wave version of $w_o(i)$ and $w_i(i)$ can be thought of as the respective upsampled version of $w_{iDS}(j)$. Therefore, a valid PDS PUS pair is one which satisfies the following inequality:

$$\begin{aligned} &\langle w_o(i), w_o(i)\rangle_{MN} - \langle w_i(i), w_i(i)\rangle_{MN} \\ &\quad \geq \langle w_{oDS}(j), w_{oDS}(j)\rangle_N - \langle w_{iDS}(j), w_{iDS}(j)\rangle_N \quad \forall N > 0. \end{aligned} \tag{11}$$

There are many ways to satisfy (11), we chose to implement the PDS PUS pairs as indicated in Figure 4. Lemma 3 states this more formally.

Figure 3 The Passive Downsampler and Passive Upsampler construction



Lemma 3: *The following nonlinear-averaging-PDS* (*NLA-PDS*) *and hold-PUS satisfies the inequality* (11) *required of Definition* 4*:*

- *NLA-PDS: Let* $w_o, w_{oDS} \in \mathbb{R}^m$*, in which each kth element within their respective vectors* w_o, w_{oDS} *are denoted* w_{o_k}, w_{oDS_k} $k \in \{1, \ldots, m\}$*. Therefore the NLA-PDS is implemented as follows:*

$$w_{oDS_k}(j) = \sqrt{\sum_{i=M(j-1)}^{Mj-1} w_{o_k}^2(i)}\,\mathtt{sgn}\Big(\sum_{i=M(j-1)}^{Mj-1} w_{o_k}(i)\Big) \tag{12}$$

- *hold-PUS: Similarly let* $w_i, w_{iDS} \in \mathbb{R}^m$*, in which each kth element within their respective vectors* w_i, w_{iDS} *are denoted* w_{i_k}, w_{iDS_k} $k \in \{1, \ldots, m\}$*. Therefore the hold-PUS is implemented as follows:*

$$w_{i_k}(i) = \sqrt{\frac{1}{M}} w_{iDS_k}(j-1), \quad i = Mj, \ldots, M(j+1) - 1. \tag{13}$$

The proof of Lemma 3 is in Appendix B.3. Figure 5 shows a Single-Input Single-Output (SISO) controller with steady state gain K_{c1} controlling a SISO plant with

Figure 4 The NLA-PDS and hold-PUS implementation

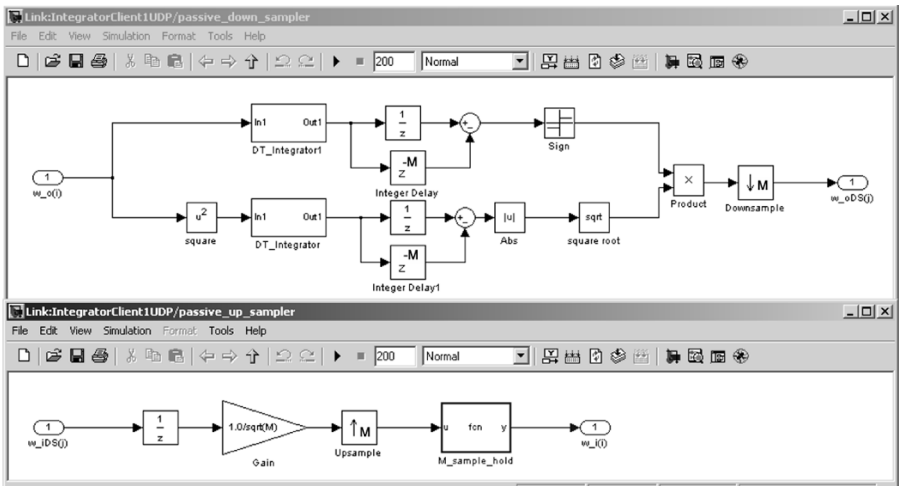
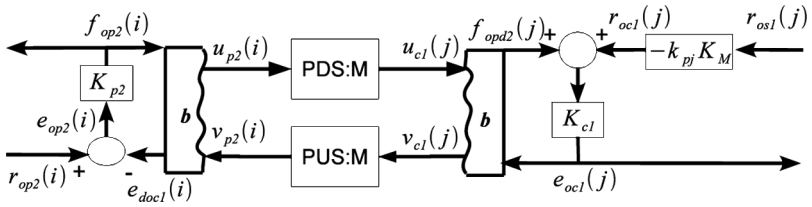


Figure 5 Simplified controller/plant with PDS/PUS in order to determine K_M



steady state gain K_{p2}. The steady state gain K_{ss} for any system with input $u(i)$ and output $y(i)$ is computed as follows

$$K_{ss} = \lim_{i \to \infty} \frac{y(i)}{u(i)}.$$

Recall, that since $n - m = 2 - 1 = m = 1$ then $k_{\mathrm{pj}} = 1$, in addition knowing the corresponding steady state gains K_{c1} and K_{p2} we can compute the appropriate scaling gain K_M so that $f_{op2}(i) = r_{os1}(j)$ in the limit as $i, j \to \infty$. The SISO case is treated for simplicity of discussion, however, if the controller-plant-steady-state-gain-matrix-product is much larger along the diagonal component and small elsewhere then the scaling matrix can be replaced with a scalar scaling term $K_M \in \mathbb{R}$.

Lemma 4: *In order for the steady state output of the SISO plant $f_{op2}(i)$ with steady state gain K_{p2} to equal the desired reference $r_{os1}(j)$ to the SISO controller with steady state gain K_{c1} depicted in Figure 5. The reference scaling gain K_M should be computed as follows*

$$\begin{aligned} K_M &= \frac{1 + K_{c1}K_{p2}}{K_{c1}K_{p2}}\sqrt{M} \\ &\approx \sqrt{M} \quad (when\ K_{c1}K_{p2}\ is\ large) \end{aligned}$$

in which M relates the downsample/upsample rates for the respective NLA-PDS/hold-PUS described in Lemma 3 in which $i = Mj$.

The proof of Lemma 4 is in Appendix B.4.

Remark 3: Although we chose to implement and investigate the NLA-PDS and hold-PUS there are indeed linear implementations which satisfy Definition 4. Noting that (11) can be written in the following compact form:

$$\|(w_o)_{MN}\|_2^2 - \|(w_i)_{MN}\|_2^2 \geq \|(w_{oDS})_N\|_2^2 - \|(w_{iDS})_N\|_2^2 \tag{14}$$

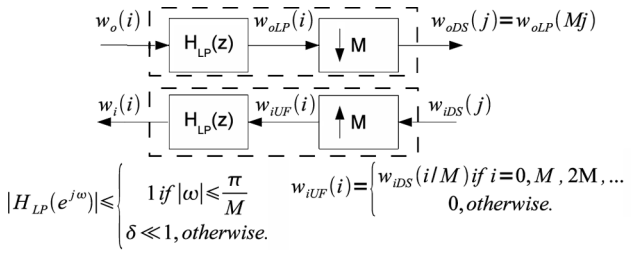
and denoting the respective upsampling (downsampling)-gains as $g_{PUS}(M)$ and $g_{PDS}(M)$ which are determined as follows:

$$g_{PUS}(M) = \sup_{\|(w_{iDS})_N\|_2^2 \neq 0} \frac{\|(w_i)_{MN}\|_2^2}{\|(w_{iDS})_N\|_2^2} \tag{15}$$

$$g_{PDS}(M) = \sup_{\|(w_o)_{MN}\|_2^2 \neq 0} \frac{\|(w_{oDS})_N\|_2^2}{\|(w_o)_{MN}\|_2^2} \tag{16}$$

After careful inspection of equations (14)–(16) it is clear that for a proposed-PUS if $g_{PUS}(M) \leq 1$ then it is a PUS, likewise if for a proposed-PDS if $g_{PDS}(M) \leq 1$ then it is a PDS. Therefore, traditional anti-aliasing up-sampling and down-sampling configurations (Proakis and Manolakis, 1996, Chapter 10), such as those depicted in Figure 6, in which the low-pass-filters ($H_{\mathsf{LP}}(z)$) l_2^m-gains are less-than or equal to one satisfy Definition 4.

Figure 6 Standard anti-aliasing down-sampler/up-sampler which are also a suitable PDS and PUS pair



2.3 l_2^m stable power junction control networks

Figure 2 depicts m controllers interconnected to $n-m$ plants over a power-junction-network. It can be shown that this power-junction-control-network will remain l_2^m/L_2^m-*stable* when subject to either fixed delays and/or data dropouts. For the discrete time case we can show how to safely handle time varying delays by dropping duplicate transmissions from the power-junction-network. Please refer to Appendix A for corresponding definitions or nomenclature.

Theorem 1: *The power-junction-control-network depicted in Figure* 2 *is* l_2^m*-stable if all plants* $G_{pk}(e_{opk}(i))$*,* $k \in \{m+1, \ldots, n\}$ *and all controllers* $G_{cj}(f_{ocj}(i))$*,* $j \in \{1, \ldots, m\}$ *are strictly-output passive and*

$$\sum_{k=m+1}^{n} \langle f_{opk}, e_{dock} \rangle_N \geq \sum_{j=1}^{m} \langle e_{ocj}, f_{opdj} \rangle_N \tag{17}$$

holds for all $N \geq 1$.

Proof: Each *strictly-output passive* plant for $k \in \{m+1, \ldots, n\}$ satisfies

$$\langle f_{opk}, e_{opk} \rangle_N \geq \epsilon_{opk} \|(f_{opk})_N\|_2^2 - \beta_{opk} \tag{18}$$

while each *strictly-output passive* controller for $j \in \{1, \ldots, m\}$ satisfies (19).

$$\langle e_{ocj}, f_{ocj} \rangle_N \geq \epsilon_{ocj} \|(e_{ocj})_N\|_2^2 - \beta_{ocj}. \tag{19}$$

Substituting, $e_{dock} = r_{opk} - e_{opk}$ and $f_{opdj} = f_{ocj} - r_{ocj}$ into equation (17) yields

$$\sum_{k=m+1}^{n} \langle f_{opk}, r_{opk} - e_{opk} \rangle_N \geq \sum_{j=1}^{m} \langle e_{ocj}, f_{ocj} - r_{ocj} \rangle_N$$

which can be rewritten as

$$\begin{aligned} &\sum_{k=m+1}^{n} \langle f_{opk}, r_{opk} \rangle_N + \sum_{j=1}^{m} \langle e_{ocj}, r_{ocj} \rangle_N \\ &\quad \geq \sum_{k=m+1}^{n} \langle f_{opk}, e_{opk} \rangle_N + \sum_{j=1}^{m} \langle e_{ocj}, f_{ocj} \rangle_N \end{aligned} \tag{20}$$

so that we can then substitute equations (18) and (19) into equation (20) to yield

$$\begin{aligned} &\sum_{k=m+1}^{n} \langle f_{opk}, r_{opk} \rangle_N + \sum_{j=1}^{m} \langle e_{ocj}, r_{ocj} \rangle_N \\ &\quad \geq \epsilon \left[\sum_{k=m+1}^{n} \|(f_{opk})_N\|_2^2 + \sum_{j=1}^{m} \|(e_{ocj})_N\|_2^2 \right] - \beta \end{aligned} \tag{21}$$

in which $\epsilon = \min(\epsilon_{opk}, \epsilon_{ocj})$, $k \in \{m+1, \ldots, n\}$ $j \in \{1, \ldots, m\}$ and $\beta = \sum_{k=m+1}^{n} \beta_{opk} + \sum_{j=1}^{m} \beta_{ocj}$. Thus equation (21) satisfies Definition 8-iii for *strictly-output passivity* in which the input is the row vector of all controller and plant inputs $[r_{oc1}, \ldots, r_{ocm}, r_{op(m+1)}, \ldots, r_{opn}]$, and the output is the row vector of all controller and plant outputs $[e_{oc1}, \ldots, e_{ocm}, f_{op(m+1)}, \ldots, f_{opn}]$. □

Remark 4: When we let $\epsilon_{opk} = \epsilon_{ocj} = 0$ we see that all the plants and controllers are *passive*, therefore the system depicted in Figure 2 is *passive* if it satisfies (17).

With these proofs complete, it is a fairly simple exercise to use Definition 2 and use the techniques shown in the proof for Kottenstette and Antsaklis (2007, Lemma 2) in order to prove the following:

Corollary 1: *If all of the discrete time varying delays in the network depicted in Figure* 2 *are fixed* $pl(i) = pl, cl(i) = cl,\ l \in \{1, \ldots, n\}$ *and/or data packets are dropped then* (17) *holds.*

Corollary 2: *The discrete time varying delays* $pl(i), cl(i),\ l \in \{1, \ldots, n\}$ *depicted in Figure* 2 *can vary arbitrarily as long as* (17) *holds. The main assumption* (17) *will hold if duplicate transmissions to the power-junction-network are dropped when received, and duplicate transmissions from the power-junction-network to the receivers are dropped. This can be accomplished for example by transmitting the tuple* $(i, u_{pk}(i))$ *to the power-junction-network, if* $i \in \{$*the set of received indexes*$\}$ *then set* $u_{pk}(i) = 0$ *before computing* $u_j(i)$ *to transmit to the controllers, etc.*

Using a averaging-power-junction-network, we shall use a single controller to control the velocity (and indirectly the position) of two masses using an idealised force source to actuate each mass. We chose this simple example in order to focus on implementing a more complete network control example and to simplify the discussion, the interested reader is referred to Kottenstette and Antsaklis (2008a) in which we studied the control of $n - m$ motors over a token network with perturbed dynamics.

Each plant with respective mass $M_{p2} = 2\,\text{kg}$ and $M_{p3} = 0.25\,\text{kg}$ has the following transfer function

$$H_{pk}(s) = \frac{1}{M_{pk}s}. \tag{22}$$

We will transform each plant to its discrete time passive equivalent using the *inner-product equivelant sample and hold-transform* (*IPESH-transform*) as defined by Definition 5.

Definition 5: Let $H_p(s)$ and $H_p(z)$ denote the respective continuous and discrete time transfer functions which describe a plant. Furthermore, let T_s denote the respective sample and hold time. Finally, denote $\mathcal{Z}\{F(s)\}$ as the z-transform of the sampled time series whose Laplace transform is the expression of $F(s)$, given on the same line in Franklin et al. (2006, Table 8.1 p.600). $H_p(z)$ is generated using the following IPESH-transform

$$H_p(z) = \frac{(z-1)^2}{T_s z}\mathcal{Z}\left\{\frac{H_p(s)}{s^2}\right\}.$$

The *IPESH-transform* is a result from applying the *inner-product equivelant sample and hold* (see Definition 9 in Appendix B.5) which is formally stated as Lemma 5 with the corresponding proof provided in Appendix B.5.

Lemma 5: *Applying the inner-product equivelant sample and hold to a Single-Input-Single-Output* (*SISO*) *passive Linear-Time Invariant* (*LTI*) *plant with*

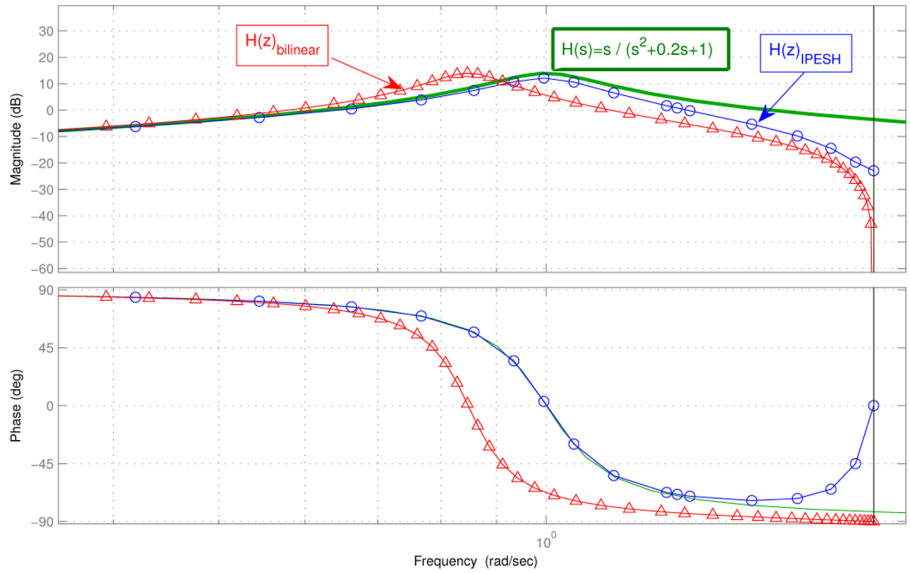
transfer function $H_p(s)$ results in a corresponding passive LTI plant $H_p(z)$ which results from the IPESH-transform.

Therefore, the respective discrete time passive model for each mass is

$$H_{pk}(z) = \frac{T_s}{2M_{pk}} \frac{z+1}{z-1}.$$

Remark 5: For this example, the exact transfer function would have been obtained if we had chosen instead to use the bilinear transform and substituted $s = \frac{2}{T_s}\frac{z-1}{z+1}$. It has been well known that the bilinear transformation preserves passivity (Fettweis, 1986), however the two transforms are not equivalent as can be appreciated by viewing Figure 7. Figure 7 clearly shows that the bilinear transformation for the plant $H(s) = \frac{s}{s^2+0.2s+1}$, while still passive, suffers from significant warping in amplitude and phase shift, which the *IPESH-transform* is much less sensitive to the low sampling rate.

Figure 7 Bode-plot comparing bilinear transform ($H(z)_{\text{bilinear}}$) to *IPESH-transform* ($H(z)_{\text{IPESH}}$), $T_s = \frac{\pi}{2}$ (see online version for colours)



Each plant is next rendered *strictly-output passive* by adding a small amount of damping using velocity feedback, such that the strictly output passive plants will have the following form:

$$H_{spk}(z) = \frac{H_{pk}(z)}{1 + \epsilon H_{pk}(z)}.$$

Since the plants are essentially integrators we will simply use a proportional feedback controller with gain K. Using loop-shaping techniques we choose $K = \frac{M_{p2}\pi}{2T_s M}$. This will provide a reasonable crossover frequency at roughly one half

the controllers-Nyquist frequency ($\omega_n = \frac{\pi}{T_s M}$) and maintain a 90° degree phase margin. Note, that we chose the largest mass to dictate the gain limit, as the system tolerated the larger overall system gain. In fact, the gain can be arbitrarily larger since this system will always have 90° phase margin, however the trade-off is a more oscillatory response which is verified in simulation.

Remark 6: The proof of Lemma 5 given in Appendix B.5 shows that causality is preserved when applying the *IPESH*-transform to a causal transfer function $H_p(s)$. But, can the *IPESH* formulation be applied to an actual physical system? Since a *ZOH* is applied to the input, it should be clear that a causal prediction can indeed be made if exact knowledge of the plant is known through the use of an observer structure. This has indeed been shown by Costa-Castello and Fossas (2007) using dissipative-systems theory which resulted in an observer structure which used the measured output of the plant. In addition, we showed that by simply applying the *IPESH* definition, it is a straight forward exercise to synthesise a passive observer structure which uses the integrated output of the plant (Kottenstette, 2007, Section 2.3.1). It should also be noted, that although the synthesis arguments required precise knowledge of the plant in order to make a prediction in order to implement a causal observer, passivity is still typically preserved even when exact knowledge of the plant is unknown. The reason for this robustness to uncertainty lies in the observer structure which includes a feed-forward term whose magnitude typically increases as sampling time increases. Therefore, the engineer should be careful that her implementation is well-posed (Willems, 1971) (all instantaneous feed-back loop-gains are less than one, $(bH_{pk}(z)|_{z=\infty} < 1$, since the controller is linear and known, the inherent feed-back loop resulting from the wave-transform can be precomputed so as to avoid any explicit loops (Kottenstette, 2007, (2.62) p.37))). It is a much more challenging problem to design observers for nonlinear systems in this framework, however, as such the *causal* PS, PH combined with the power-junction-network framework presented here does indeed apply (Kottenstette and Chopra, 2009). For an account on how the robotics community, in which the *IPESH*-like formulation first originated from as it applies to Port-Controlled-Hamiltonian Systems, has applied it with much success in an approximately passive manner by using energy dissipation techniques see Secchi et al. (2007, Sections 3.4, 4.4).

3 Experiments

In this section we present a detailed experiment in which two 'soft-real-time' simulated plants are controlled over an ad-hoc wireless network by a single controller which are connected over an averaging-power-junction-network.

3.1 Experimental setup

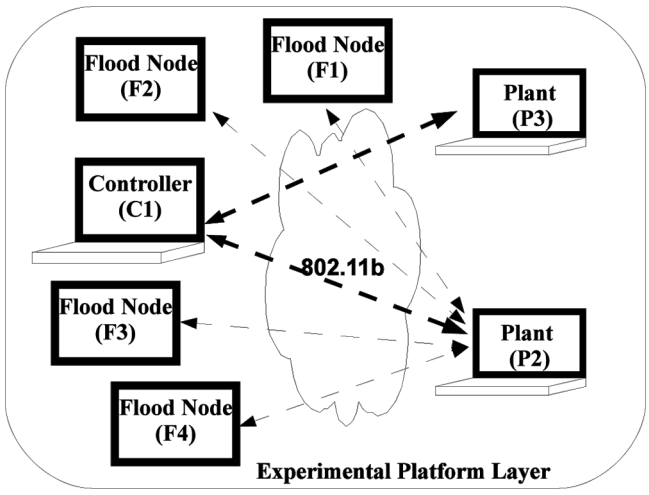
Table 1 summarises the respective properties and assumptions related to the controller and plants. In particular each plant is connected to a hold-PUS/NLA-PDS pair so that their respective velocity measurements are only transmitted every $T_s M = 0.1\,\mathrm{s}$ over the network. The controller, connected to the averaging-power-junction-network is implemented in an event driven manner as new data

arrives over the network. Such an asynchronous controller is possible and can be justified using a construct similar to the *Passive* Asynchronous Transfer Unit (PATRU) (Kottenstette and Antsaklis, 2008c, Definition 4). Furthermore a network flood attack will be initiated from four nodes which is directed towards the simulated plant G_{p2}. This network attack creates both an asymmetric delay and loss of data which allows us to evaluate these effects on the overall system. As indicated in Figure 8 each plant was simulated on a unique laptop, the controller which was implemented on its own personal laptop as well. Each Flood Node ran on a unique embedded 'brick' to launch its ping-flood attacks from.

Table 1 Simulation summary

Plant/Controller	*Assumptions*
G_{c1}	$K = \frac{M_{p2}\pi}{2T_s M}$, event-driven controller
G_{p2}	$M_{p2} = 2.0\,\text{kg}$, $\epsilon = 0.01$, $M = 10$, $T_s = 0.01\,\text{s}$
G_{p3}	$M_{p3} = 0.25\,\text{kg}$ $\epsilon = 0.01$, $M = 10$, $T_s = 0.01\,\text{s}$

Figure 8 Platform layer used for experiment



3.2 Software implementation

Each plant was simulated using Simulink which included a 'soft-real-time' timer which we denote as `rt_clock`. The development of `rt_clock` resulted from the need to pace Simulink simulations which required a variable step solver in order to be executed. We have refined our implementation such that we can pace our simulations to run at around 98% real-time. The key was to use MATLAB's non-blocking `pause` command and a moving time window indexed by `i` as show in Listing 1.

Listing 1: Snippet from rt_clock.m.

```
if dT < rt_timers.T(id)*rt_timers.i(id)
```

```
  p_t = rt_timers.T(id)*rt_timers.i(id) - dT;
  pause(p_t);
end
```

The basic networking interface for each plant and controller was built around a simple UDP client-server model in which the power-junction-network server (**PJ**) was listening to ports 6000 and 6001 and each plant (P2, P3) would send data to their respective port. However, to simulate running the system in a potentially hostile environment we used an SSH server running on the controller platform to permit secure tunnels from the respective plants on ports 7000 and 7001 respectively. In order to use SSH, we have to use a TCP/IP protocol which our initial UDP client-server interface did not support. Therefore we used `netcat` in order to create a UDP to TCP/IP bridge between the SSH tunnel and the respective plants and clients (Giacobbi et al., 2008). `nc_bridge` is a utility we created in order to establish the respective tunnels and bridges. In order to redirect connections on port 7000 from P2's host (`192.168.1.111`) to the power-junction-network-server on port 7000 (`192.168.1.110`) `nc_bridge` does the following from P2's host:

```
ssh -L 7000:127.0.0.1:7000 192.168.1.110 nc_s_0
```

`nc_s_0` is run on the power-junction-network-servers host to establish the netcat bridge which serves TCP/IP clients on port 7000 and relays these packets back and forth as UDP packets via port 6000.

```
nc -l 7000 < /t/fifo0 | nc -u 127.0.0.1 6000 > /t/fifo0.
```

Finally a `netcat` bridge is established on P2's host which serves UDP clients (from Simulink) locally which connect to port 6000 and relays these packets back and forth as TCP/IP packets via port 7000.

```
nc -u -l 6000 < /t/fifo | nc 127.0.0.1 7000 > /t/fifo.
```

The power-junction-network-server is a C-based server which ran in a completely event driven manner, as we highlight the main parts in Listing 2.

Listing 2: Snippet from powerjuncudp.c.

```
while(1){
 if (tick_flag){
  t_s += TS*DOWNSAMPLE;
  tick_flag=0;
 }
 r[0] = AMPLITUDE*sin(omega*t_s);
 FD_ZERO(&pl);
 for (i=0; i<N_P; i++)
  FD_SET(socketmatlab[i],&pl);
 //Block until data arrives from any N_P plant
 socketchosen = select(nfds, &pl, 0, 0, 0);
 for (i=0; i<N_P; i++){
```

```
  if ( FD_ISSET(socketmatlab[i],&pl) ) {
   if ( state[i] ){
    /* For all i in { state[i] == 1 :
     * calculate v_out from u_in[i],
     * v_fifo[i] = v_out,
     * state[i] = 0. */
    calc_v_out(state, u_in, r, v_fifo, N_P, WAVE_N);
    tick_flag=1;
   }
   if ( recvfrom(socketmatlab[i], u_in[i],...) )
    state[i]=1;
  }
 }
 for (i=0; i<N_P; i++){
  if ( ! state[i] )
   break;
 }
 if (i == N_P){
  calc_v_out(state, u_in, r, v_fifo, N_P, WAVE_N);
  tick_flag=1;
 }
 // Send out data from pending v_fifo[i]'s
 for (i=0; i<N_P; i++){
  if (!v_fifo[i].empty()){
   v_out=pop_v_fifo(v_fifo,i);
   sendto(socketmatlab[i], v_out,...);
  }
 }
}
```

3.3 Experimental results

Three experiments were performed. The first experiment established a nominal system response of the system when operating over a wireless network. Both velocity and position tracking performed quite well as indicated in Figures 10 and 11 respectively. The nominal round trip time delay is indicated in Figure 12, it can vary quite substantially over long periods of time, however under nominal conditions it is roughly the same for each respective plant. The substantial variance during normal operation in the time delay is captured in our second experiment and is displayed in Figure 14, where in a controlled manner we took plant-two 'off-line' at around 30 s and then brought plant-two back 'online' at 60 s. As Figure 13 indicates, when plant-two is 'off-line' the velocity of the plant goes to zero (m/s) until it goes back 'online' and receives additional commands from the controller. These results lead us to our discussion of our final experiment in which a flood attack is commenced on plant-two.

However, when a substantial network attack is commenced at around 50 s, as indicated in Figure 17 an asymmetric round-trip delay pattern results in which ΔT_{p2} grows to over 3 s while ΔT_{p3} slowly grows to around 1 s. As can be seen in Figure 15

Figure 9 Computational layer used to implement controllers and plants

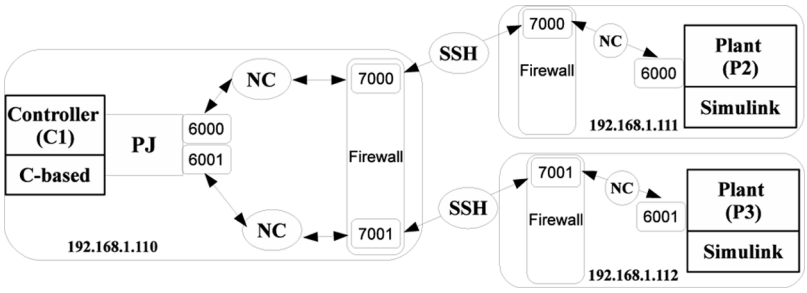


Figure 10 Nominal velocity response over wireless network (see online version for colours)

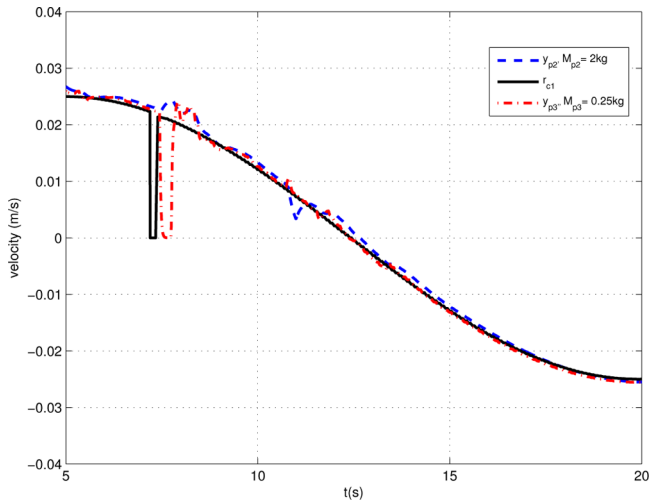


Figure 11 Nominal position response over wireless network (see online version for colours)

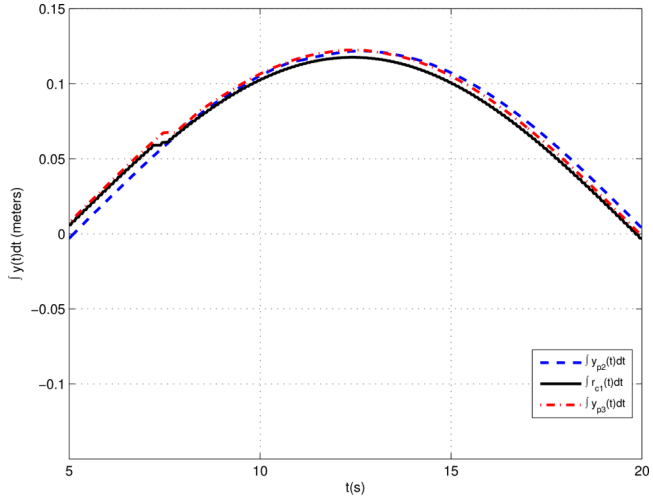


Figure 12 Nominal time delay over wireless network (see online version for colours)

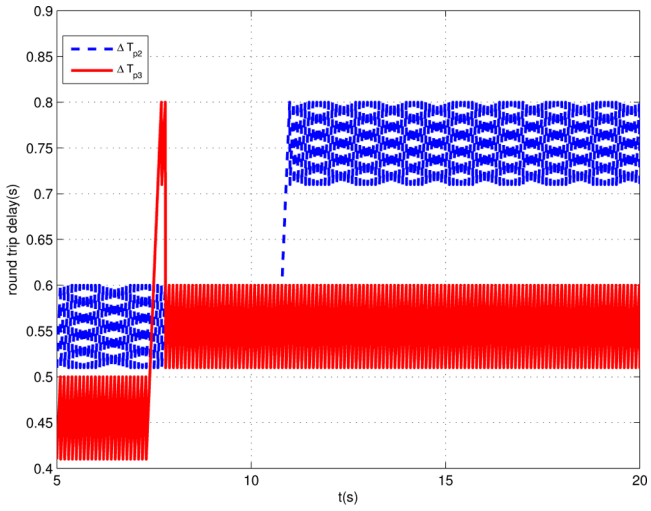
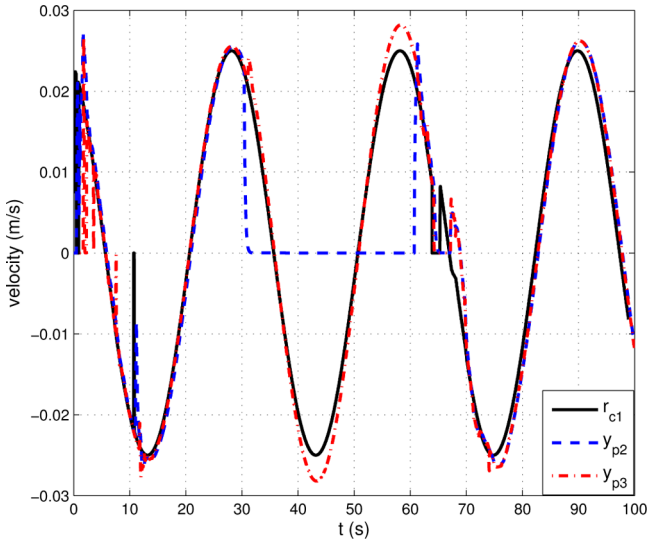


Figure 13 Velocity response for plant being removed and added from network (see online version for colours)



a substantial amount of data is lost which forces the velocity of plant-two to stay near 0 while the velocity profile for plant-three is just a bit more oscillatory and unbiased relative to the desired trajectory. As a result, an overall position drift occurs with plant-two relative to plant-three as indicated in Figure 16. We noticed that as the asymmetry in the delay between the two plants round-trip delays grew, there was a bit more oscillatory behaviour for plant-three in attempting to follow the same profile (which is intuitive). Therefore, we limited our input FIFO to hold only up to a maximum of 2 s (20 samples) worth of data. Note that only by dropping or compressing data can the overall round-trip delay be reduced. The effect of limiting

Figure 14 Time delay for plant being removed and added from network (see online version for colours)

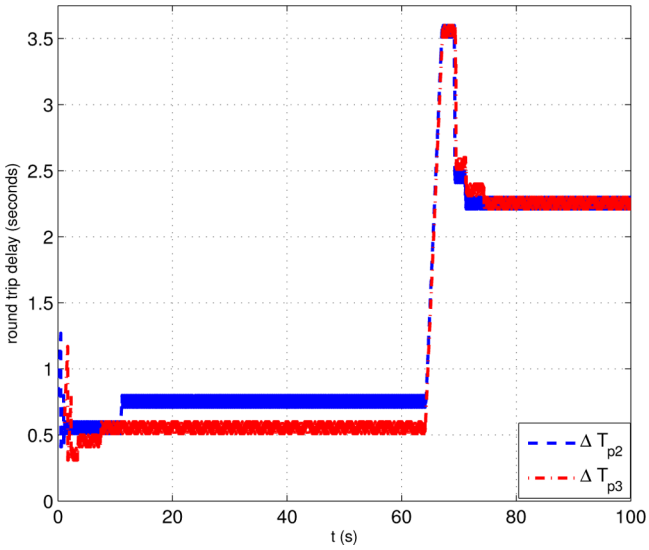
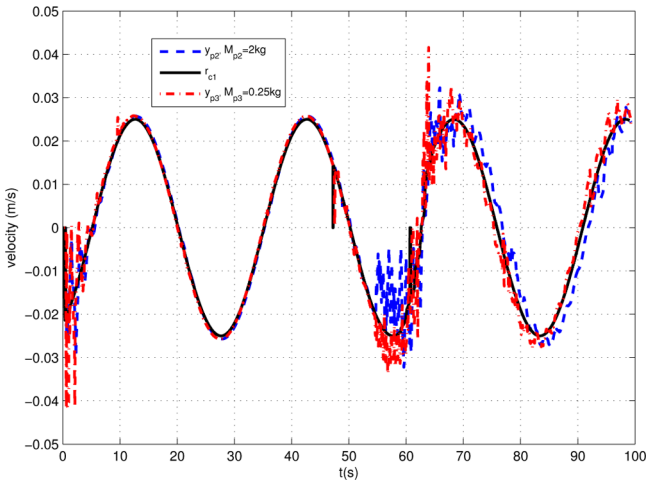


Figure 15 Velocity response over wireless network when subject to network attack (see online version for colours)

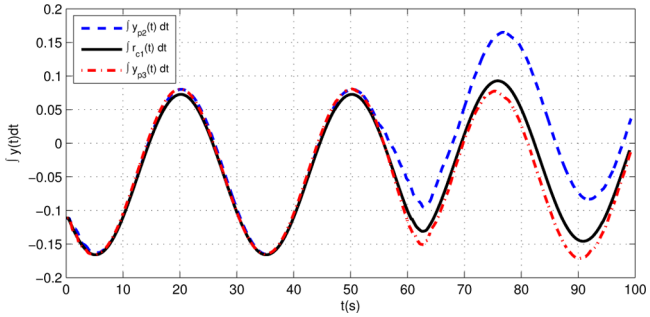


the size of the FIFO causes the delay to reduce from roughly 3.25 s to 2.25 s for plant-two.

4 Conclusions

We have shown how to interconnect multiple *passive* plants and controllers (*systems*) over a *passive* power-junction-network (Definition 2). In addition we showed how to implement an averaging-power-junction-network (Definition 3) and

Figure 16 Position response over wireless network when subject to network attack (see online version for colours)



proved that it satisfied the conditions required to be a power-junction-network (Lemma 2). Remark 1 provides sufficient conditions required in order for different LTI *passive* plants G_{pk} to track each other when interconnected over a power-junction-network. Theorem 1 states that if each plant and controller are connected to a power-junction-network as illustrated in Figure 2 are *strictly-output passive* then a l_2^m*-stable* power-junction-control-network is created which can tolerate both fixed delays and data dropouts (Corollary 1) as well as time-varying delays which do not generate additional power in the network (Corollary 2). The TCP/IP protocol does not duplicate data transmissions therefore power-junction-control-networks which transmit wave variables using TCP/IP will satisfy Corollary 2. The UDP protocol can potentially duplicate packets, however if the user is careful not to process these duplicate transmissions then Corollary 2 will be satisfied.

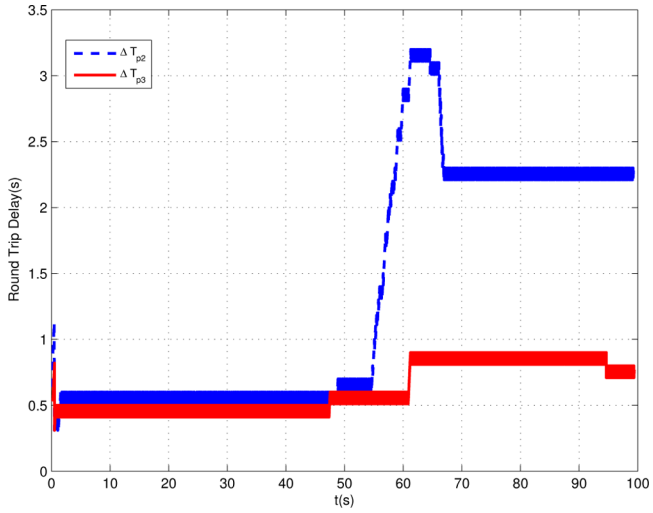
In order to reduce networking traffic and computational demands on a controller we introduce the PUS and PDS (Definition 4) and showed how a NLA-PDS satisfied the PDS requirements while a hold-PUS satisfied the PUS requirements (Lemma 3). Lemma 4 showed that a set-point scaling gain $K_M \approx \sqrt{M}$ should be used in conjunction with a hold-PUS and NLA-PDS networked control system such as those depicted in Figure 5. Remark 3 shows that traditional up-sampling/down-sampling schemes with an anti-aliasing filter can be implemented which satisfy the PUS PDS requirements, and warrants further investigation.

In order to simulate a continuous time plant $H_{pk}(s)$ we presented and used the *IPESH-Transform* (Definition 5) as depicted in Figure 18 in which we showed this to be a direct result of applying the *inner-product equivelant sample and hold* (Definition 9) to a continuous time SISO LTI plant (Lemma 5). In addition, a detailed set of experiments were conducted to evaluate the averaging-power-junction-network in conjunction with the hold-PUS/NLA-PDS system.

We evaluated a secure networked control system over an ad-hoc wireless network as described in Section 3. In particular the plants and clients could communicate over a connectionless UDP client-server architecture in which the averaging-power-junction-network was implemented around an event driven UDP server architecture. In order to establish secure connections, however each plant and controller were placed behind a firewall and communications were established over a TCP/IP-based `ssh`-tunnel in which `netcat` provided a UDP-to-TCP/IP bridge (Figure 9). Both velocity and position tracking were verified under normal operations as indicated

in Figures 10 and 11 respectively. When taking one plant entirely 'off-line' the other plant which is still 'online' has a different tracking-scale-factor than the one when both plants are online as shown in Figure 13. Even when the round trip delay exceeds 3.5 s (Figure 14) both controlled plants exhibit exceptional velocity tracking. However, when a significant network flooding attack is directed at an individual plant an asymmetry results in the velocity profile which results in position drift (Figures 15 and 16). Figures 15 and 17 also indicate that the velocity output remains slightly oscillatory when the round trip delay for each plant is significantly different.

Figure 17 Time delay over wireless network when subject to network attack (see online version for colours)



Acknowledgements

Contract/grant sponsor (number): NSF (NSF-CCF-0820088), Contract/grant sponsor (number): DOD (N00164-07-C-8510), Contract/grant sponsor (number): Air Force (FA9550-06-1-0312) and Contract/grant sponsor (number): NSF (NSF-CCF-0819865).

References

Antsaklis, P. and Baillieul, J. (Eds.) (2004) *Special Issue on Networked Control Systems*, Volume 49 number 9 of *IEEE Transactions on Automatic Control*. IEEE.

Antsaklis, P. and Baillieul, J. (Eds.) (2007) *Special Issue: Technology of Networked Control Systems*, Volume 95 number 1 of *Proceedings of the IEEE*. IEEE.

Breedveld, P.C. (2006) 'Port-based modeling of dynamic systems in terms of bond graphs', in Troch, I. (Ed.): *5th Vienna Symposium on Mathematical Modelling*, Vienna, Volume ARGESIM Report no. 30, Vienna. ARGESIM and ASIM, Arbeitsgemeinschaft Simulation (see http://eprints.eemcs.utwente.nl/6141/)

Chopra, N. and Spong, M. (2006) 'Passivity-based control of multi-agent systems', *Advances in Robot Control: From Everyday Physics to Human-Like Movements*, pp.107–134.

Costa-Castello, R. and Fossas, E. (2006) 'On preserving passivity in sampled-data linear systems', *2006 American Control Conference (IEEE Cat. No. 06CH37776C)*, Minneapolis, MN, USA, p.6.

Costa-Castello, R. and Fossas, E. (2007) 'On preserving passivity in sampled-data linear systems', *European Journal of Control*, Vol. 13, No. 6, pp.583–590.

Desoer, C.A. and Vidyasagar, M. (1975) *Feedback Systems: Input-Output Properties*, Academic Press, Inc., Orlando, FL, USA.

Fettweis, A. (1986) 'Wave digital filters: theory and practice', *Proceedings of the IEEE*, Vol. 74, No. 2, pp.270–327.

Franklin, G.F., Powell, J.D. and Emami-Naeini, A. (2006) *Feedback Control of Dynamic Systems*, 5th ed., Prentice-Hall, Upper Saddle River, New Jersey, USA.

Giacobbi, G., Baskin, B., Connelly, D., Schearer, M. and Seagren, E. (2008) *Netcat Power Tools*, Syngress Press, Burlington, MA, USA.

Golo, G., van der Schaft, A.J., Breedveld, P. and Maschke, B. (2003) 'Hamiltonian formulation of bond graphs', *Nonlinear and Hybrid Systems in Automotive Control*, Springer-Verlag, London, UK, pp.351–372.

Haddad, W.M. and Chellaboina, V.S. (2008) *Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach*, Princeton University Press, Princeton, New Jersey, USA.

Kottenstette, N. (2007) *Control of Passive Plants with Memoryless Nonlinearitites Over Wireless Networks*, PhD thesis, University of Notre Dame, Notre Dame, IN, USA.

Kottenstette, N. and Antsaklis, P. (2007) 'Stable digital control networks for continuous passive plants subject to delays and data dropouts', *Decision and Control, 2007 46th IEEE Conference*, New Orleans, Louisiana, USA, pp.4433–4440.

Kottenstette, N. and Antsaklis, P. (2008a) 'Control of multiple networked passive plants with delays and data dropouts', *American Control Conference*, Seattle, Washington, USA, pp.3126–3132.

Kottenstette, N. and Antsaklis, P. (2008b) 'Wireless control of passive systems subject to actuator constraints', *47th IEEE Conference on Decision and Control, CDC 2008*, Cancun, Mexico, pp.2979–2984.

Kottenstette, N. and Antsaklis, P.J. (2008c) 'Wireless digital control of continuous passive plants over token ring networks', *International Journal of Robust and Nonlinear Control*, Vol. 19, No. 18, pp.2016–2039 (see also http://www3.interscience.wiley.com/journal/121516295/abstract?CRETRY=1&SRETRY=0).

Kottenstette, N., Koutsoukos, X., Hall, J., Sztipanovits, J. and Antsaklis, P. (2008) 'Passivity-based design of wireless networked control systems for robustness to time-varying delays', *Real-Time Systems Symposium*, Barcelona, Spain, pp.15–24.

Kottenstette, N. and Chopra, N. (2009) 'Lm2-stable digital-control networks for multiple continuous passive plants', *1st IFAC Workshop on Estimation and Control of Networked Systems (NecSys'09)*, Venice, Italy, pp.120–125 (see also http://www.ifac-papersonline.net/Detailed/40538.html).

Kottenstette, N., Karsai, G. and Sztipanovits, J. (2009) 'A passivity-based framework for resilient cyber physical systems', *ISRCS 2009 2nd International Symposium on Resilient Control Systems*, Idaho Falls, ID, USA, pp.43–50 (DOI: 10.1109/ISRCS.2009.5251370).

MathWorks, I.T. (2008a) 'Matlab', *The Language of Technical Computing, Version 7.6*.

MathWorks, I.T. (2008b). 'Simulink', *Dynamic System Simulation for MATLAB, Version 7.1*.

Niemeyer, G. and Slotine, J-J.E. (2004) 'Telemanipulation with time delays', *International Journal of Robotics Research*, Vol. 23, No. 9, pp.873–890.

Proakis, J. and Manolakis, D. (1996) *Digital Signal Processing: Principles, Algorithms, and Applications*, Prentice-Hall, Inc. Upper Saddle River, NJ, USA.

Ryu, J-H., Kim, Y.S. and Hannaford, B. (2004) 'Sampled- and continuous-time passivity and stability of virtual environments', *IEEE Transactions on Robotics*, Vol. 20, No. 4, pp.772–776.

Secchi, C., Stramigioli, S. and Fantuzzi, C. (2007) *Control of Interactive Robotic Interfaces: A Port-Hamiltonian Approach*, Springer-Verlag, Springer Berlin/Heidelberg.

Stramigioli, S., Secchi, C., van der Schaft, A. and Fantuzzi, C. (2002) 'A novel theory for sampled data system passivity', *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (Cat. No. 02CH37332C)*, Vol. 2, pp.1936–1941.

Stramigioli, S., Secchi, C., van der Schaft, A.J. and Fantuzzi, C. (2005) 'Sampled data systems passivity and discrete port-hamiltonian systems', *IEEE Transactions on Robotics*, Vol. 21, No. 4, pp.574–587.

van der Schaft, A. (1999) *L2-Gain and Passivity in Nonlinear Control*, Springer-Verlag, New York, Inc., Secaucus, NJ, USA.

Willems, J. (1971) *The Analysis of Feedback Systems*, MIT Press, Cambridge, MA, USA & London, UK.

Ylonen, T. and Lonvick, C. (2006) The Secure Shell (SSH) Protocol Architecture, Technical Report, RFC 4251, January.

Appendix

A *Passive* systems

The following is a brief summary on *passive* systems. The interested reader is referred to Desoer and Vidyasagar (1975), van der Schaft (1999) and Haddad and Chellaboina (2008) for additional information. Let $\mathcal{T}$ represent a set indicating time in which $\mathcal{T} = \mathbb{R}^+$ for continuous time signals and $\mathcal{T} = \mathbb{Z}^+$ for discrete time signals. Let $\mathcal{V}$ be a linear space $\mathbb{R}^m$ and denote the space of all functions $u : \mathcal{T} \to \mathcal{V}$ by the symbol $\mathcal{H}$ which satisfy the following:

$$\|u\|_2^2 = \int_0^\infty u^\mathsf{T}(t)u(t)\mathrm{d}t < \infty, \tag{23}$$

for continuous time systems (L_2^m), and

$$\|u\|_2^2 = \sum_0^\infty u^\mathsf{T}(i)u(i) < \infty, \tag{24}$$

for discrete time systems (l_2^m). Similarly we will denote the extended space of functions $u : \mathcal{T} \to \mathcal{V}$ in $\mathcal{H}_e$ which satisfy the following:

$$\|u_T\|_2^2 = \langle u, u\rangle_T = \int_0^T u^\mathsf{T}(t)u(t)\mathrm{d}t < \infty; \quad \forall T \in \mathcal{T} \tag{25}$$

for continuous time systems (L_{2e}^m), and

$$\|u_T\|_2^2 = \langle u, u\rangle_T = \sum_0^{T-1} u^\mathsf{T}(i)u(i) < \infty; \quad \forall T \in \mathcal{T} \tag{26}$$

for discrete time systems (l_{2e}^m). Furthermore let $y = Hu$ describe a relationship of the function y to the function u in which the instantaneous output value at continuous time t is denoted $y(t) = Hu(t)$ and respectively $y(i) = Hu(i)$ at discrete time i.

Definition 6: A continuous time dynamic system $H : \mathcal{H}_e \to \mathcal{H}_e$ is L_2^m stable if

$$u \in L_2^m \implies Hu \in L_2^m. \tag{27}$$

Definition 7: A discrete time dynamic system $H : \mathcal{H}_e \to \mathcal{H}_e$ is l_2^m stable if

$$u \in l_2^m \implies Hu \in l_2^m. \tag{28}$$

Definition 8: Let $H : \mathcal{H}_e \to \mathcal{H}_e$. We say that H is

(i) *passive* if $\exists \beta \geq 0$ s.t.

$$\langle Hu, u\rangle_T \geq -\beta, \quad \forall u \in \mathcal{H}_e, \quad \forall T \in \mathcal{T} \tag{29}$$

(ii) *strictly-input passive* if $\exists \delta > 0$ and $\exists \beta \geq 0$ s.t.

$$\langle Hu, u\rangle_T \geq \delta \|u_T\|_2^2 - \beta, \quad \forall u \in \mathcal{H}_e, \quad \forall T \in \mathcal{T} \tag{30}$$

(iii) *strictly-output passive* if $\exists \epsilon > 0$ and $\exists \beta \geq 0$ s.t.

$$\langle Hu, u\rangle_T \geq \epsilon \|Hu_T\|_2^2 - \beta, \quad \forall u \in \mathcal{H}_e, \quad \forall T \in \mathcal{T} \tag{31}$$

(iv) *non-expansive* if $\exists \hat{\gamma} > 0$ and $\exists \hat{\beta} \geq 0$ s.t.

$$\|Hu_T\|_2^2 \leq \hat{\beta} + \hat{\gamma}^2 \|u_T\|_2^2, \quad \forall u \in \mathcal{H}_e, \quad \forall T \in \mathcal{T}. \tag{32}$$

Remark 7: A *non-expansive* system H is equivalent to any system which has finite L_2^m (l_2^m) gain in which there exists constants γ and $\beta \geq 0$ s.t. $0 < \gamma < \hat{\gamma}$ and satisfy

$$\|Hu_T\|_2 \leq \gamma \|u_T\|_2 + \beta, \quad \forall u \in \mathcal{H}_e, \quad \forall T \in \mathcal{T}. \tag{33}$$

Furthermore a *non-expansive* system implies L_2^m (l_2^m) stability (van der Schaft, 1999, p.4; Kottenstette and Antsaklis, 2007, Remark 1).

B Additional proofs

B.1 Proof of Lemma 1

Proof: Summing both sides of equation (5) with respect to $l \in \{1, \ldots, m_s\}$ we have

$$\begin{aligned}
\sum_{l=1}^{m_s} \sum_{j=1}^{m} u_{j_l}^2 &\leq \sum_{l=1}^{m_s} \sum_{k=m+1}^{n} u_{k_l}^2 \\
\sum_{j=1}^{m} u_j^{\mathsf{T}} u_j &\leq \sum_{k=m+1}^{n} u_k^{\mathsf{T}} u_k \\
\sum_{k=m+1}^{n} u_k^{\mathsf{T}} u_k &\geq \sum_{j=1}^{m} u_j^{\mathsf{T}} u_j.
\end{aligned} \tag{34}$$

Likewise, summing (6) with respect to $l \in \{1, \ldots, m_s\}$ we have

$$\begin{aligned}
\sum_{l=1}^{m_s} \sum_{k=m+1}^{n} v_{k_l}^2 &\leq \sum_{l=1}^{m_s} \sum_{j=1}^{m} v_{j_l}^2 \\
\sum_{k=m+1}^{n} v_k^{\mathsf{T}} v_k &\leq \sum_{j=1}^{m} v_j^{\mathsf{T}} v_j \\
\sum_{k=m+1}^{n} -v_k^{\mathsf{T}} v_k &\geq \sum_{j=1}^{m} -v_j^{\mathsf{T}} v_j. \qquad (35)
\end{aligned}$$

The sum of the left sides of equations (34) and (35) and the respective sum of the right sides of equations (34) and (35) results in

$$\sum_{k=m+1}^{n} \left(u_k^{\mathsf{T}} u_k - v_k^{\mathsf{T}} v_k\right) \geq \sum_{j=1}^{m} \left(u_j^{\mathsf{T}} u_j - v_j^{\mathsf{T}} v_j\right)$$

which is the required power-junction-network inequality given by equation (4). □

B.2 Proof of Lemma 2

Proof: First we square both sides of equations (7) and (8), which results in

$$v_{k_l}^2 = \frac{\sum_{j=1}^{m} v_{j_l}^2}{n-m}, \quad k \in \{m+1, \ldots, n\} \quad \text{and} \qquad (36)$$

$$u_{j_l}^2 = \frac{\sum_{k=m+1}^{n} u_{k_l}^2}{m}, \quad j \in \{1, \ldots, m\} \text{ respectively.} \qquad (37)$$

Next we solve for the sum of $v_{k_l}^2$ for $k \in \{m+1, \ldots, n\}$ using equation (36) which results in

$$\sum_{k=m+1}^{n} v_{k_l}^2 = \frac{(n-m)\sum_{j=1}^{m} v_{j_l}^2}{n-m} = \sum_{j=1}^{m} v_{j_l}^2. \qquad (38)$$

Similarly we solve for the sum of $u_{j_l}^2$ for $j \in \{1, \ldots, m\}$ using equation (37) which results in

$$\sum_{j=1}^{m} u_{j_l}^2 = \frac{m\sum_{k=m+1}^{n} u_{k_l}^2}{m} = \sum_{k=m+1}^{n} u_{k_l}^2. \qquad (39)$$

Since equations (38) and (39) are satisfied for all $l \in \{1, \ldots, m_s\}$, then they also satisfy (6) and (5) respectively for Lemma 1 therefore they satisfy the conditions for the lossless-power-junction-network. □

B.3 Proof of Lemma 3

Proof: It is assumed that we are referring to Figure 3 during this discussion. Our approach will be to show that the NLA-PDS satisfies $\forall N > 0$:

$$\langle w_o(i), w_o(i)\rangle_{MN} \geq \langle w_{oDS}(j), w_{oDS}(j)\rangle_N, \tag{40}$$

and that the hold-PUS satisfies $\forall N > 0$:

$$\begin{aligned} &-\langle w_i(i), w_i(i)\rangle_{MN} \geq -\langle w_{iDS}(j), w_{iDS}(j)\rangle_N \text{ equivalently} \\ &\langle w_i(i), w_i(i)\rangle_{MN} \leq \langle w_{iDS}(j), w_{iDS}(j)\rangle_N, \end{aligned} \tag{41}$$

since when both equations (40) and (41) are satisfied then equation (11) is also satisfied. Next, we will focus on each kth element of $w_i, w_{iDS}, w_o, w_{oDS} \in \mathbb{R}^m$ such that if $\forall N > 0$ and $k \in \{1, \ldots, m\}$ that both

$$\sum_{i=0}^{MN-1} w_{o_k}(i)^2 \geq \sum_{j=0}^{N-1} w_{oDS_k}(j)^2 \tag{42}$$

and

$$\sum_{i=0}^{MN-1} w_{i_k}(i)^2 \leq \sum_{j=0}^{N-1} w_{iDS_k}(j)^2, \tag{43}$$

are respectively satisfied, then so too will equations (40) and (41) be also satisfied. Therefore, we will show that the proposed NLA-PDS satisfies (42) and hold-PUS satisfies (43) respectively.

- *NLA-PDS*: Substituting equation (12) into the right-hand side of equation (42) results in:

$$\begin{aligned} \sum_{i=0}^{MN-1} w_{o_k}(i)^2 &\geq \sum_{j=0}^{N-1} \left(\sqrt{\sum_{i=M(j-1)}^{Mj-1} w_{o_k}(i)^2} \right)^2 \\ &\geq \sum_{j=0}^{N-1} \sum_{i=M(j-1)}^{Mj-1} w_{o_k}(i)^2 \\ &\geq \sum_{i=0}^{M(N-1)-1} w_{o_k}(i)^2 \\ \sum_{i=M(N-1)}^{MN-1} w_{o_k}(i)^2 &\geq 0 \end{aligned}$$

  in which the final inequality is clearly always satisfied therefore equation (42) is always satisfied for this type of PDS.

- *hold-PUS*: Substituting equation (13) into the left-hand side of equation (43) results in:

$$\begin{aligned}
\sum_{j=0}^{N-1} \frac{1}{M} \sum_{i=Mj}^{M(j+1)-1} w_{iDS_k}(j-1)^2 &\leq \sum_{j=0}^{N-1} w_{iDS_k}(j)^2 \\
\sum_{j=0}^{N-1} w_{iDS_k}(j-1)^2 &\leq \sum_{j=0}^{N-1} w_{iDS_k}(j)^2 \\
\sum_{j=0}^{N-2} w_{iDS_k}(j)^2 &\leq \sum_{j=0}^{N-1} w_{iDS_k}(j)^2 \\
0 &\leq w_{iDS_k}(N-1)^2
\end{aligned}$$

  in which the second to last inequality results from the obvious assumption that $w_{iDS_k}(-1) = 0$, and the final inequality is obviously always satisfied therefore equation (43) is always satisfied for this type of PUS. □

B.4 Proof of Lemma 4

Proof: Since we treat each component as the system reaches steady state we have the following relationships at steady state:

$$u_{c1}(j) = \sqrt{M} u_{p2}(i) \tag{44}$$

$$v_{p2}(i) = \frac{1}{\sqrt{M}} v_{c1}(j) \tag{45}$$

$$\begin{bmatrix} v_{c1}(j) \\ f_{opd2}(j) \end{bmatrix} = \begin{bmatrix} 1 & -\sqrt{\frac{2}{b}} \\ \sqrt{\frac{2}{b}} & -\frac{1}{b} \end{bmatrix} \begin{bmatrix} u_{c1}(j) \\ e_{oc1}(j) \end{bmatrix} \tag{46}$$

$$\begin{bmatrix} u_{p2}(i) \\ e_{doc1}(i) \end{bmatrix} = \begin{bmatrix} -1 & \sqrt{2b} \\ -\sqrt{2b} & b \end{bmatrix} \begin{bmatrix} v_{p2}(i) \\ f_{op2}(i) \end{bmatrix}. \tag{47}$$

Substituting equation (44) into equation (46), and equation (45) into equation (47) results in

$$\begin{bmatrix} v_{c1}(j) \\ f_{opd2}(j) \end{bmatrix} = \begin{bmatrix} \sqrt{M} & -\sqrt{\frac{2}{b}} \\ \sqrt{\frac{2M}{b}} & -\frac{1}{b} \end{bmatrix} \begin{bmatrix} u_{p2}(i) \\ e_{oc1}(j) \end{bmatrix} \tag{48}$$

$$\begin{bmatrix} u_{p2}(i) \\ e_{doc1}(i) \end{bmatrix} = \begin{bmatrix} -\sqrt{\frac{1}{M}} & \sqrt{2b} \\ -\sqrt{\frac{2b}{M}} & b \end{bmatrix} \begin{bmatrix} v_{c1}(j) \\ f_{op2}(i) \end{bmatrix} \tag{49}$$

respectively. Which can be written in the following form:

$$\begin{bmatrix} e_{doc1}(i) \\ f_{opd2}(j) \end{bmatrix} = C_1 \begin{bmatrix} v_{c1}(j) \\ u_{p2}(i) \end{bmatrix} + C_2 \begin{bmatrix} e_{oc1}(j) \\ f_{op2}(i) \end{bmatrix}$$

$$C_1 = \begin{bmatrix} -\sqrt{\frac{2b}{M}} & 0 \\ 0 & \sqrt{\frac{2M}{b}} \end{bmatrix}, \quad C_2 = \begin{bmatrix} 0 & b \\ -\frac{1}{b} & 0 \end{bmatrix} \tag{50}$$

$$\begin{bmatrix} v_{c1}(j) \\ u_{p2}(i) \end{bmatrix} = C_3 \begin{bmatrix} v_{c1}(j) \\ u_{p2}(i) \end{bmatrix} + C_4 \begin{bmatrix} e_{oc1}(j) \\ f_{op2}(i) \end{bmatrix}$$

$$C_3 = \begin{bmatrix} 0 & \sqrt{M} \\ -\sqrt{\frac{1}{M}} & 0 \end{bmatrix}, \quad C_4 = \begin{bmatrix} -\sqrt{\frac{2}{b}} & 0 \\ 0 & \sqrt{2b} \end{bmatrix}. \tag{51}$$

Solving for the wave variables in terms of the effort and flow variables in equation (51) results in

$$\begin{bmatrix} v_{c1}(j) \\ u_{p2}(i) \end{bmatrix} = (I - C_3)^{-1} C_4 \begin{bmatrix} e_{oc1}(j) \\ f_{op2}(i) \end{bmatrix}. \tag{52}$$

Substituting equation (52) into equation (50) results in the following expression which relates effort-flow inputs to their delayed counterparts

$$\begin{bmatrix} e_{doc1}(i) \\ f_{opd2}(j) \end{bmatrix} = [C_2 + C_1(I - C_3)^{-1} C_4] \begin{bmatrix} e_{oc1}(j) \\ f_{op2}(i) \end{bmatrix}. \tag{53}$$

Solving for equation (53) results in

$$\begin{bmatrix} e_{doc1}(i) \\ f_{opd2}(j) \end{bmatrix} = \begin{bmatrix} \sqrt{1/M} & 0 \\ 0 & \sqrt{M} \end{bmatrix} \begin{bmatrix} e_{oc1}(j) \\ f_{op2}(i) \end{bmatrix}. \tag{54}$$

Knowing equation (54) it is a simple exercise to show that

$$f_{op2}(i) = K_M \sqrt{\frac{1}{M}} \frac{K_{c1}K_{p2}}{1 + K_{c1}K_{p2}} r_{os1}(j) + \frac{K_{p2}}{1 + K_{p2}K_{c1}} r_{op2}(i) \lim_{i \to \infty}.$$

Therefore when $r_{op2}(i) = 0$ (no steady state disturbance) then $f_{op2}(i) = r_{os1}(j)$ at steady state when

$$K_M = \frac{1 + K_{c1}K_{p2}}{K_{c1}K_{p2}} \sqrt{M}$$
$$\approx \sqrt{M} \text{ when } K_{c1}K_{p2} \text{ is large.}$$

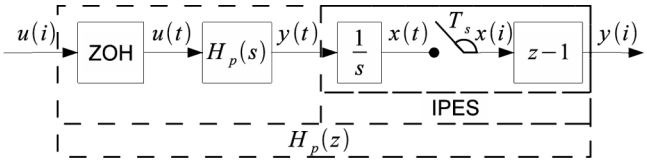
□

B.5 Proof of Lemma 5

In order to prove Lemma 5 we recall the formal Definition 9 for the *inner-product equivelant sample and hold* which is graphically illustrated for the SISO LTI case in Figure 18. The *inner-product equivelant sample and hold* is based on earlier work by Stramigioli et al. (2002) and Ryu et al. (2004).

Definition 9 (Kottenstette and Antsaklis, 2007, Definition 4)**:** Let a continuous one-port plant be denoted by the input-output mapping $H_{ct} : L_{2_e}^m \to L_{2_e}^m$. Denote

Figure 18 A representation of the *IPESH* for SISO LTI systems



continuous time as t, the discrete time index as i, the sample and hold time as T_s, the continuous input as $u(t) \in L_{2_e}^m$, the continuous output as $y(t) \in L_{2_e}^m$, the transformed discrete input as $u(i) \in l_{2_e}^m$, and the transformed discrete output as $y(i) \in l_{2_e}^m$. The inner-product equivalent sample and hold (*IPESH*) is implemented as follows:

I $\quad x(t) = \int_0^t y(\tau)\mathrm{d}\tau$

II $\quad y(i) = x((i+1)T_s) - x(iT_s)$

III $\quad u(t) = u(i), \forall t \in [iT_s, (i+1)T_s).$

As a result

$$\langle y(i), u(i)\rangle_N = \langle y(t), u(t)\rangle_{NT_s}, \quad \forall N \geq 1 \tag{55}$$

holds.

It should be obvious from the *IPESH* definition that it is indeed causal as the output $y(i)$ does not depend on any future inputs $u(i+n)$, $n \geq 1$. To be clear, when people speak of passive systems such as $H_p(s)$ for example, it is implicitly assumed to be causal. It is sufficient therefore to show that for the case when

$$y(t) = \frac{1}{T_s}u(t) = \frac{1}{T_s}u(i), \quad \forall t \in [iT_s, (i+1)T_s) \text{ that } y(i) \text{ is indeed causal}$$

$$y(i) = \int_{iT_s}^{(i+1)T_s} \frac{u(t)}{T_s}\mathrm{d}t = u(i)\frac{1}{T_s}\int_{iT_s}^{(i+1)T_s} \mathrm{d}t = u(i)$$

so even for the feed-through case $y(i) = u(i)$ obviously does not depend on any future input $u(i+n)$. All that remains to be shown therefore is that the *IPESH*-transform satisfies Definition 9 and recall that the *IPESH* preserves passivity (Kottenstette and Antsaklis, 2007, Theorem 3-I), noting that the preservation of passivity has also been shown by both (Stramigioli et al., 2002; Ryu et al., 2004) later in Stramigioli et al. (2005) and, apparently not realising that they had formulated a problem which satisfied the *IPESH* which leads to a trivial proof for preservation of passivity, resulted in an extremely involved dissipative systems-theory proof (Costa-Castello and Fossas, 2006). Both Kottenstette and Antsaklis (2007, Theorem 3) and later a slightly corrected (Kottenstette and Antsaklis, 2008b, Theorem 1) show that in general the *IPESH* preserves stronger forms of passivity when transforming from the continuous-time model to the discrete-time model including strictly-input passive and strictly-output passive systems.

Proof: For simplicity of discussion it is assumed that $y(0) = 0$. Definition 9-I describes an integration operation, therefore the corresponding transfer function

$\frac{X(s)}{Y(s)} = \frac{1}{s}$ as indicated in Figure 18. Next we denote the transfer function from $\frac{X(s)}{U(s)}$ as $H_{pI}(s)$ which has the following form

$$H_{pI}(s) = \frac{H_p(s)}{s}.$$

Definition 9-II can be described using a periodic sampling operation in which $x(t) = x(iT_s)$ and applying the respective z-transforms to $x(i)$ (denoted $X(z)$) and $y(i)$ (denoted $Y(z)$) in which

$$\frac{Y(z)}{X(z)} = (z-1)$$

as indicated in Figure 18. It is well known that an *exact* discrete equivalent transfer function can be used to describe the *ZOH* and periodic sampling operation such that

$$\frac{X(z)}{U(z)} = \frac{(z-1)}{z}\mathcal{Z}\left\{\frac{H_{pI}(s)}{s}\right\}$$
$$H_{pI}(z) = \frac{(z-1)}{z}\mathcal{Z}\left\{\frac{H_p(s)}{s^2}\right\}$$

as discussed in Franklin et al. (2006, Section 8.6.1). This naturally leads to the final expression describing the transfer function for the discrete time passive plant $H_p(z)$ (in which $\frac{1}{T_s}$ is used as a typical scaling term)

$$H_p(z) = \frac{(z-1)}{T_s}H_{pI}(z) = \frac{(z-1)^2}{T_s z}\mathcal{Z}\left\{\frac{H_p(s)}{s^2}\right\}.$$

The preservation of passivity of the transform is a direct consequence of using the *IPESH* as stated in Kottenstette and Antsaklis (2007, Theorem 3-I). □