

control mechanisms are not reflected in this document. Changes will have been made to this work since it will be submitted for possible publication.

Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, 37235

Passivity-Based Design of Wireless Networked Control Systems Subject to Time-Varying Delays

Nicholas Kottenstette, Joseph Hall, Xenofon
Koutsoukos, Janos Sztipanovits, and Panos
Antsaklis

TECHNICAL REPORT

ISIS-08-904

Revised:2/15/2011

This is the authors version of a work that is under consideration in revised form for possible publication in IEEE Transactions on Control Systems Technology. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality

Passivity-Based Design of Wireless Networked Control Systems Subject to Time-Varying Delays

Nicholas Kottenstette, Joseph Hall, Xenofon Koutsoukos, Janos Sztipanovits, and Panos Antsaklis

Abstract—Real-life cyber-physical systems, such as automotive vehicles, building automation systems, and groups of unmanned vehicles are monitored and controlled by networked control systems. The overall system dynamics emerges from the interaction among physical dynamics, computational dynamics, and communication networks. Network uncertainties such as time-varying delay and packet loss cause significant challenges. This paper proposes a passive control architecture for designing networked control systems that are insensitive to network uncertainties. We describe the architecture for a system consisting of a robotic manipulator controlled by a digital controller over a wireless network and we show that the system is stable even in the presence of time-varying delays. Experimental results demonstrate the advantages of the passivity-based architecture with respect to stability and performance and show that the system is insensitive to network uncertainties.

I. INTRODUCTION

The heterogeneous composition of computing, sensing, actuation, and communication components has enabled a modern grand vision for real-world Cyber Physical Systems (CPSs). Real-world CPSs, such as automotive vehicles, building automation systems, and groups of unmanned air vehicles are monitored and controlled by networked control systems and the overall system dynamics emerges from the interaction among physical dynamics, computational dynamics, and communication networks. Design of CPSs requires controlling real-world system behavior and interactions in dynamic and uncertain conditions. This paper, in particular, is inspired by the rapidly increasing use of Networked Control System (NCS) architectures in constructing real-world CPSs that integrate computational and physical devices using wireless networks such as medical device networks, groups of unmanned vehicles, and transportation networks. NCS research has been recently a very active area investigating problems at the intersection of control systems, networking, and computer science [1].

CPSs are inherently heterogeneous not only in terms of their components but also in terms of essential design requirements. Besides functional properties, CPSs are subject to a wide range of physical requirements, such as dynamics, power, physical size, and fault tolerance in addition to system-level requirements, such as safety and security. This heterogeneity does not go well with current methods of compositional design for

several reasons. The most important principle used in achieving multi-objective compositionality is separation of concerns (in other words, defining design viewpoints). Separation of concerns works if the design views are orthogonal, i.e. design decisions in one view does not influence design decisions in other views. Unfortunately, achieving compositionality for multiple physical and functional properties simultaneously is a very hard problem because of the lack of orthogonality among the design views.

Fig. 1 represents a simplified model-based design flow of a CPS composed of a physical plant and a networked control system. In a conventional design flow, the controller dynamics is synthesized with the purpose of optimizing performance. The selected design platform (abstractions and tools used for control design in the design flow) is frequently provided by a modeling language and a simulation tool, such as MATLAB/Simulink [2], [3]. The controller specification is passed to the implementation design layer through a “Specification/Implementation Interface”. The implementation in itself has a rich design flow that we compressed here only in two layers: System-level design and Implementation platform design. The software architecture and its mapping on the (distributed) implementation platform are generated in the system-level design layer. The results - expressed again in the form of architecture and system models - are passed on through the next Specification and Implementation Interface to generate code as well as the hardware and network design. This simplified flow reflects the fundamental strategy in platform-based design [4]. Design progresses along precisely defined abstraction layers. The design flow usually includes top-down and bottom-up elements and iterations (not shown in the figure).

Effectiveness of the platform-based design largely depends on how much the design concerns (captured in the abstraction layers) are orthogonal, i.e., how much the design decisions in the different layers are independent. Heterogeneity causes major difficulties in this regard. The controller dynamics is typically designed without considering implementation side effects (e.g. numeric accuracy of computational components, timing accuracy caused by shared resource and schedulers, time varying delays caused by network effects, etc.). Timing characteristics of the implementation emerge at the confluence of design decisions in software componentization, system architecture, coding, and HW/network design choices. Compositionality in one layer depends on a web of assumptions to be satisfied by other layers. For example, compositionality on the controller design layer depends on assumptions that the effects of quantization and finite word-length can be neglected and the

N. Kottenstette, J. Hall, X. Koutsoukos, and J. Sztipanovits are with the Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN 37203.

P. Antsaklis is with the University of Notre Dame, Notre Dame, IN 46556.

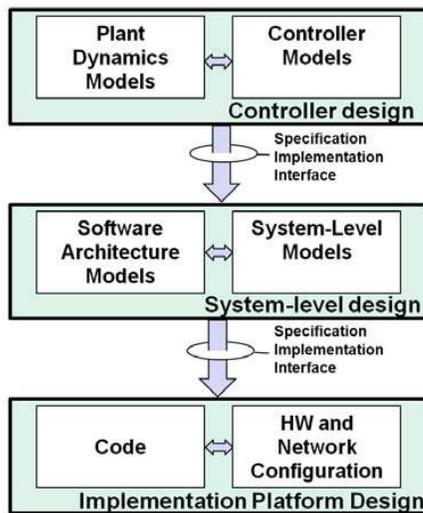


Fig. 1. Simplified CPS design flow.

discrete-time model is accurate. Since these assumptions are not satisfied by the implementation layer, the overall design needs to be verified after implementation - even worst - changes in any layer may require re-verification of the full system.

An increasingly accepted way to address these problems is to enrich abstractions in each layer with implementation concepts. An excellent example for this approach is TrueTime [5] that extends MATLAB/Simulink with implementation related modeling concepts (networks, clocks, schedulers) and supports simulation of networked and embedded control systems. While this is a major step in improving designers' understanding of implementation effects, it does not help in decoupling design layers and improving orthogonality across the design concerns. A controller designer can now factor in implementation effects (e.g., network delays), but still, if the implementation changes, the controller may need to be redesigned.

Decoupling the design layers is a very hard problem and typically introduces significant restrictions and/or over-design. For example, the Timed Triggered Architecture (TTA) orthogonalizes timing, fault tolerance, and functionality [6], but it comes on the cost of strict synchrony, and static structure. A new approach for decoupling between the control design and implementation layers has been proposed recently in [7]. The approach allows the design of state-feedback controllers that minimize a quadratic performance bound for a given level of timing jitter using linear matrix inequality methods.

This paper is motivated by the rapidly increasing use of network control system architectures in constructing real-world CPSs and aims at addressing fundamental problems caused by networks effects, such as time-varying delay, jitter, limited bandwidth, and packet loss. To deal with these implementation uncertainties, we propose a model-design flow on top of passivity, a very significant concept from system theory [8]. A precise mathematical definition requires many technical details, but the main idea is that a passive system cannot apply an infinite amount of energy to its environment. The inherent safety that passive systems provide is fundamental

in building systems that are insensitive to implementation uncertainties. Passive systems have been exploited for the design of diverse systems such as smart exercise machines [9], teleoperators [10]–[15], digital filters [16], networked control systems [17]–[19] and complex non-linear thermal and chemical based processes [20]–[22].

Our approach advocates a concrete and important transformation of model-based methods that can improve orthogonality across the design layers and facilitate compositional component-based design of CPSs. By imposing passivity constraints on the component dynamics, the stability of the NCS is guaranteed if the received data transmitted over a given network is only processed once at the respective receiving controller or plant nodes. This separation of concerns empowers the model-based design process to be applied for networked control systems. Information about the network effects needs not to be considered at the controller design layer because the theoretical guarantees for stability are independent of the remaining networking implementation uncertainties. Furthermore, stability is maintained even in the presence of disturbance traffic in the network.

The primary contributions of this paper are:

- We present a passive control architecture for a system consisting of a robotic manipulator controlled by a *digital controller* over a wireless network.
- We provide analytical results that prove that our architecture ensures stability of the networked control system in the presence of time varying delays assuming that the communication protocols do not process duplicate transmissions.
- We implement the passive control architecture on an experimental networked control system consisting of two computer nodes that realize the robotic manipulator and the digital controller respectively and communicate over an ad hoc 802.11b wireless network subject to additional traffic induced by disturbance nodes.
- We present experimental results that demonstrate the stable operation of the system in the presence of severe time-varying delays caused by network traffic generated by the disturbance nodes or by excessive computational load competing with the controller. Our experimental results validate that the passivity-based architecture ensures stability of the networked system and provides robustness to time varying delays.

The work presented in the paper demonstrates that passivity can be exploited to account for the effects of network uncertainties, thus improving orthogonality across the controller design and implementation design layers and empowering model-driven development. Part of this work has been presented in [23]. The main extensions are (1) experimental implementation and evaluation of the passivity-based architecture using a networked control system, (2) detailed design of the digital passive controller, and (3) theoretical analysis that includes the proofs of passivity and stability for the proposed architecture. It should be noted that passive structures offer additional advantages for robustness to finite length representations and saturation [16] but this paper focuses on network

effects which is one of the most significant concerns in the development of CPSs.

The rest of the paper is organized as follows: An overview of related work is presented in Section II followed by a summary of notation used and passivity definitions in Section III. Section IV presents the passive control architecture focusing on the technical details required for implementation. Analysis of our proposed networked control system is provided in Section V. Section VI describes the implementation and presents detailed experimental results. Finally, Section VII presents the main directions of our future work.

II. RELATED WORK

Our overall approach to designing networked control systems which can tolerate time-varying delay and data loss is constructive in nature in which we rely on *passivity* based networked control fundamentals [17]. Constructive approaches typically rely on a system or controller to be restricted to a given *sector* in order for the overall system to remain stable [24]–[26]. *Passive* systems are interior-conic systems which are *inside the sector* $[0, \infty]$ and when connected in either parallel or a negative feedback arrangement remain passive and Lyapunov stable. When time delays are introduced into a feedback arrangement involving two passive systems the overall passive structure is lost and stability can be lost. As a result those in the telemanipulation community who wanted to preserve stability for arbitrary fixed time-delays proposed using *wave variables* [27], [28]. Wave variables were originally introduced by Fettweis in order to circumvent the problem of delay-free loops and guarantee that the implementation of wave digital filters is both stable and realizable [16]. The wave variables which resulted from a bilinear scattering transformation allowed for a stable minimum phase continuous system to be mapped to a stable minimum phase discrete-time system. Stability is guaranteed because the wave variables allowed for the primitive discrete-time components derived from their passive continuous-time counterparts to remain passive while allowing for the overlying continuous-time networking structure to remain unaltered so as to preserve stability. The use of wave-variables for networked control has continued to advance.

In particular [19] provides constructive conditions for continuous-time plants and controllers which are interconnected with wave-variables derived from a *generalized scattering transform* in order to maintain L_2^m -stability when subject to fixed-time delays. It is further asserted that the results presented in [19] apply for the case in which the wave variables transmitted between the two continuous time systems are first compressed and converted to a discrete-time wave variable then transmitted over a network, received, and finally decompressed back to a continuous-time wave variable. Additional details on compression/ decompression techniques for wave variables are described in [15], [18], [29]. Both [11], [30] have shown that the discrete-time wave variables can tolerate both arbitrary fixed-delays and data-loss in which we clarified that only time-varying delays which do not replicate previously transmitted wave-variables can always be safely handled [17].

The precise definitions chosen for passivity shall be presented from the input-output perspective similar to the definition for positive systems given in [24]. Systems which will satisfy this input-output passivity definition include positive real and dissipative dynamical systems [31]. When a dissipative dynamical system can be described by a Hamiltonian (the sum of kinetic and potential energy, $\mathcal{H} = \mathcal{T} + \mathcal{V}$) a passive mapping typically exists in which the Hamiltonian serves as storage-function (β) [31]. This will be clearly illustrated in our discussion of the passive structure of robotic systems in relating the joint-velocities to their corresponding motor torques. However, there are some limitations with the study of passive systems. For example, systems which consist of cascades of passive systems (such as two integrators in series) are not necessarily a passive system.

The conditions for stability for our networked digital control system require the digital controller and continuous-time plant to be strictly-output passive. As a result this limits us to initially controlling the velocity output of a robot in order to indirectly control its corresponding position. Such indirect control frameworks can be subject to position drift and require an additional drift-compensation algorithm such as those described in [32]. However, using the notion of a *passivity index* [20], we demonstrate how to design low-complexity analog filters to place in parallel with an asymptotically stable minimum-phase linear-time invariant stable systems in order to render the combined system strictly output passive. Such an architecture allows us to achieve steady-state position control in our proposed framework. Finally, we have recently shown that our proposed framework is applicable to the control of a larger class of Lyapunov stable systems which possess the same number of inputs and outputs and are interior conic or equivalently inside the sector $[a, \infty]$ in which $|a| < \infty$ [33]. Certain classes of stable non-minimum phase systems are inside the sector $[a, \infty]$ in which $a < 0$ and can be controlled in our proposed framework.

III. PRELIMINARIES

We choose to use the following compact notation for continuous time (CT) and discrete time (DT) systems:

$$\begin{aligned} \langle G(u), u \rangle_{NT_s} &\triangleq \int_0^{NT_s} G(u(t))^T u(t) dt \text{ CT inner product} \\ \langle G(u), G(u) \rangle_{NT_s} &\triangleq \|(G(u))_{NT_s}\|_2^2 \\ \langle G(u), u \rangle_N &\triangleq \sum_{i=0}^{N-1} G(u[i])^T u[i] \text{ DT inner product} \\ \langle G(u), G(u) \rangle_N &\triangleq \|(G(u))_N\|_2^2. \end{aligned}$$

Note that in order to distinguish continuous time from discrete time the integral is taken to the limit NT_s while the summation is taken to $N - 1$ in which $N \in \{1, 2, \dots\}$ and T_s is a real non-negative number. We also denote $L_{2_e}^m(U)$ as the *extended* L_2^m space for the function $u(t) \in U$ in which $U \subset \mathbb{R}^m$ as all possible functions for a given $NT_s \geq 0$ which satisfy:

$$\|(u)_{NT_s}\|_2^2 < \infty.$$

In the limit as $NT_s \rightarrow \infty$, then $u \in L_2^m(U)$ is any function which satisfies

$$\int_0^\infty u^T(t)u(t)dt < \infty \text{ or more compactly, } \|u\|_2^2 < \infty.$$

Note also that $L_2^m(U) \subset L_{2e}^m(U)$.

Definition 1: [34] Let $G : L_{2e}^m(U) \rightarrow L_{2e}^m(U)$ then for all $u \in L_{2e}^m(U)$ and all real $NT_s \geq 0$:

I. G is *passive* if there exist a constant β such that (1) holds.

$$\langle G(u), u \rangle_{NT_s} \geq -\beta \quad (1)$$

II. G is *strictly-output passive* if there exists some constants β and $\epsilon > 0$ such that (2) holds.

$$\langle G(u), u \rangle_{NT_s} \geq \epsilon \| (G(u))_{NT_s} \|_2^2 - \beta \quad (2)$$

Definition 2: [34, Definition 1.2.3] Let $G : L_{2e}^m(U) \rightarrow L_{2e}^m(U)$, it is said to be L_2^m -stable if

$$u \in L_2^m(U) \implies y = G(u) \in L_2^m(U),$$

and G is said to have *finite- L_2^m -gain* if $\exists \gamma_q > 0, \beta_q$ s.t. for all $NT_s \geq 0$

$$u \in L_{2e}^m(U) \implies \| (y)_{NT_s} \|_2 \leq \gamma_q \| (u)_{NT_s} \|_2 + \beta_q.$$

Any $G : L_{2e}^m(U) \rightarrow L_{2e}^m(U)$ which has *finite- L_2^m -gain* is L_2^m -stable.

The following theorem will allow us to complete the proof of our main result (Theorem 2) in which it is shown that the network control system depicted in Fig. 2 is strictly-output passive for any passive robot (plant).

Theorem 1: [34, Theorem 2.2.14] Let $G : L_{2e}^m(U) \rightarrow L_{2e}^m(U)$ be strictly-output passive. Then G has *finite L_2^m -gain*.

For an asymptotically stable LTI systems $G : L_{2e}^m(U) \rightarrow L_{2e}^m(U)$ whose transfer function is denoted $G(s)$, a frequency dependent measure known as the *passivity index* $v_F(G(s), \omega) = -\frac{1}{2} [G(j\omega) + G(-j\omega)]$ is defined such that $G(s) + v_F(G(s), \omega)$ is positive real (equivalently passive) [20].¹ If the passivity index is negative for all $\omega \in \mathbb{R}$ then $G(s)$ is strictly-input passive such that $(G(s) + v)$ is positive real in which $v = \sup_{\omega \in \mathbb{R}} v_F(G(s), \omega)$. Finally, we recall the well known result that if $G(s)$ is both asymptotically stable and strictly-input passive then it is strictly-output passive [31, Proposition 5.2-x)].

IV. PASSIVE CONTROL ARCHITECTURE

This section presents the passive control architecture depicted in Fig. 2. Section IV-A describes the continuous time passive robotic system $G : \tau_u(t) \mapsto \dot{\Theta}(t)$. Section IV-B presents the wave-variables which result from a bilinear scattering transformation indicated by the boxes denoted b in Fig. 2. Section IV-C shows how the continuous-time robotic wave-variable $u_p(t)$ is converted to a discrete-time wave-variable $u_p[i]$ using a passive sampler denoted (PS, T_s) and discrete-time wave-variable $v_{ucd}[i]$ is converted to a

continuous-time wave variable $v_{ucd}(t)$ using a passive hold denoted (PH, T_s). Section IV-D describes the passive digital controller $G_{pc} : \dot{e}_1[i] \mapsto \tau_{uc}[i]$. Finally, Section IV-E demonstrates how the *inner-product equivalent sampler (IPES, T_s)* and zero-order-hold (ZOH, T_s) block can be used to relate the discrete-time variables $(\dot{\Theta}_{sr}[i], \tau_{uc}[i])$ to the respective continuous-time variables $(\dot{\Theta}_{sr}(t), \tau_{uc}(t))$.

A. Robotic System

Our control strategy takes advantage of the passive structure of a robotic system [35]. The robot dynamics which are denoted by $G_{robot}(\tau(t))$ in Fig. 2 are described by

$$\tau = M(\Theta)\ddot{\Theta} + C(\Theta, \dot{\Theta})\dot{\Theta} + g(\Theta). \quad (3)$$

The state variables Θ denote the robot joint angles, τ is the input torque vector, $M(\Theta)$ is the mass matrix, $C(\Theta, \dot{\Theta})$ is the matrix of centrifugal and Coriolis effects, and $g(\Theta)$ is the gravity vector. The inertia matrix $M(\Theta) = M(\Theta)^T > 0$ and the matrix C and \dot{M} are related as follows:

$$-(\dot{M} - 2C) = (\dot{M} - 2C)^T \implies x^T(\dot{M} - 2C)x = 0, \forall x \in \mathbb{R}^n. \quad (4)$$

It is the skew-symmetry property given by (4) which makes it possible for the robot to achieve a passive mapping. Despite the complexity of robotic manipulators, simple control laws can be used in a number of cases. A fundamental consequence of the passivity property is that a simple independent joint continuous-time proportional-derivative (PD) control can achieve global asymptotic stability for set-point tracking in the absence of gravity [36]. Therefore, we employ a PD controller but we consider a discrete-time equivalent implementation that communicates with the robotic system via a wireless network. To compensate gravity, we select as the control command $\tau_u = \tau - g(\Theta)$. Then the following supply rate $s(\tau_u(t), \dot{\Theta}(t)) = \dot{\Theta}^T(t)\tau_u(t)$ and corresponding storage function $V(\dot{\Theta}(t)) = \frac{1}{2}\dot{\Theta}^T(t)M(\Theta(t))\dot{\Theta}(t)$ can be used to show that the robot is a passive system which is also *lossless* in which all supplied energy is stored as kinetic energy in the robot [31]. Mathematically, this lossless property is characterized as follows $\int_0^{NT_s} \dot{\Theta}(t)^T \tau_u(t) dt = (V(\dot{\Theta}(t)) - V(\dot{\Theta}(0)))$. As a result $\int_0^{NT_s} \dot{\Theta}(t)^T \tau_u(t) dt \geq -V(\dot{\Theta}(0))$. $V(\dot{\Theta}(0))$ represents all the *available storage energy* which can be extracted from the robot at time $t = 0$.

Furthermore, the robot can be made to be strictly-output passive by adding negative velocity feedback [17]. If the joints of the robotic system have significant friction, then additional velocity feedback is *unnecessary* in order to render the robotic system strictly output passive. Therefore, we select the control command τ_u to have the following final form: $\tau_u = \tau - g(\Theta) + \epsilon\dot{\Theta}$, $\epsilon \geq 0$. The gravity compensation and the velocity damping are implemented locally at the robotic system and it can be shown that the gravity compensated system with velocity damping denoted $G : \tau_u \mapsto \dot{\Theta}$ is passive when $\epsilon = 0$ and strictly-output passive for any $\epsilon > 0$

¹For the multi-input multi-output (MIMO) case $v_F(G(s), \omega) = -\frac{1}{2}\lambda_{\min} [G(j\omega) + G^T(-j\omega)]$ in which $\lambda_{\min}[M]$ denotes the minimum eigenvalue of the matrix M .

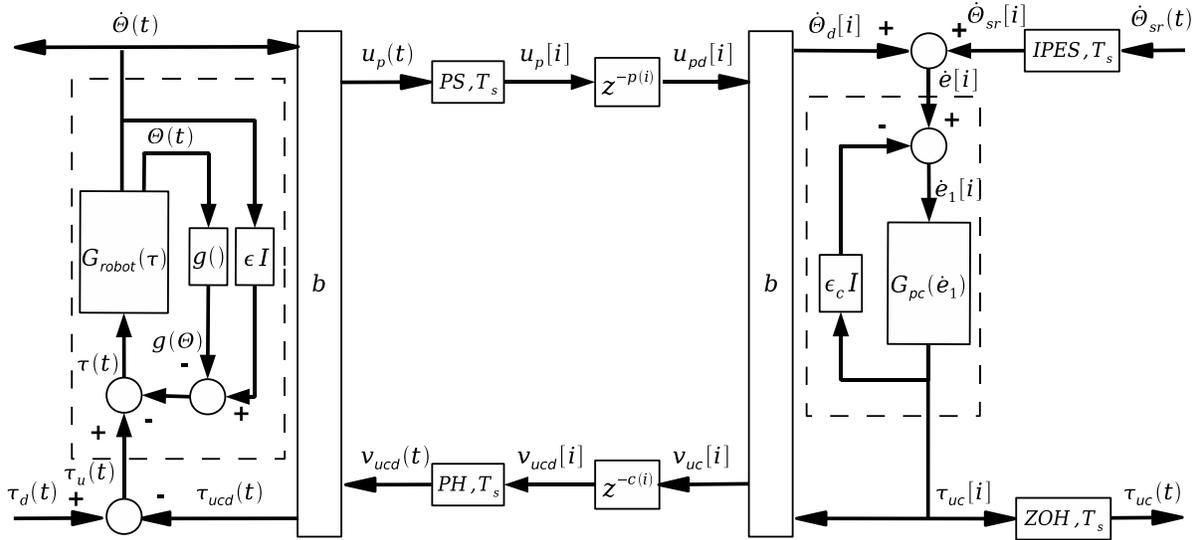


Fig. 2. Proposed Wireless Digital Control Scheme

respectively. Therefore, the following conditions are satisfied:

$$\int_0^{NT_s} \left[\dot{\theta}(t)^\top \tau_u(t) - \epsilon \dot{\theta}^\top(t) \dot{\theta}(t) \right] dt \geq V(\dot{\theta}(t)) - V(\dot{\theta}(0)) \quad (5)$$

$$\int_0^{NT_s} \dot{\theta}(t)^\top \tau_u(t) dt \geq \epsilon \int_0^{NT_s} \dot{\theta}^\top(t) \dot{\theta}(t) dt - V(\dot{\theta}(0)). \quad (6)$$

Note that the velocity damped robot is a strictly-output passive system which is a L_2^m -stable system. It is the robots' strictly-output passive property which allows us to interconnect a strictly-output passive controller over a wireless network using wave variables such that the overall system remains strictly-output passive and L_2^m -stable. The proof for Theorem 2 requires these properties in order to show that digital control system depicted in Fig. 2 is L_2^m stable.

B. Wave Variables

The continuous robot wave variables $v_{ucd}(t)$, $u_p(t) \in \mathbb{R}^m$ depicted in Fig. 2 are related to the corresponding torque and velocity vectors $\tau_{ucd}(t)$, $\dot{\theta}(t) \in \mathbb{R}^m$ as follows:

$$\frac{1}{2}(u_p^\top(t)u_p(t) - v_{ucd}^\top(t)v_{ucd}(t)) = \dot{\theta}^\top(t)\tau_{ucd}(t). \quad (7)$$

The wave variable $v_{ucd}(t)$ and velocity measurement $\dot{\theta}(t)$ determine the corresponding wave variable $u_p(t)$ and delayed control torque $\tau_{ucd}(t)$ which result from the following equation

$$\begin{bmatrix} u_p(t) \\ \tau_{ucd}(t) \end{bmatrix} = \begin{bmatrix} -I & \sqrt{2b}I \\ -\sqrt{2b}I & bI \end{bmatrix} \begin{bmatrix} v_{ucd}(t) \\ \dot{\theta}(t) \end{bmatrix}.$$

where $I \in \mathbb{R}^{m \times m}$ denotes the identity matrix and $0 < b < \infty$ is a real number.

The digital control input and output wave variables $u_{pd}[i]$, $v_{uc}[i] \in \mathbb{R}^m$ depicted in Fig. 2 are related to the corresponding discrete torque and velocity vectors $\tau_{uc}[i]$, $\dot{\theta}_d[i] \in \mathbb{R}^m$ as follows:

$$\frac{1}{2}(u_{pd}^\top[i]u_{pd}[i] - v_{uc}^\top[i]v_{uc}[i]) = \tau_{uc}[i]^\top \dot{\theta}_d[i]$$

The wave variable $u_{pd}[i]$ and control torque $\tau_{uc}[i]$ determine the corresponding wave variable $v_{uc}[i]$ and delayed velocity $\dot{\theta}_d[i]$ which result from the following equation

$$\begin{bmatrix} v_{uc}[i] \\ \dot{\theta}_d[i] \end{bmatrix} = \begin{bmatrix} I & -\sqrt{\frac{2}{b}}I \\ \sqrt{\frac{2}{b}}I & -\frac{1}{b}I \end{bmatrix} \begin{bmatrix} u_{pd}[i] \\ \tau_{uc}[i] \end{bmatrix}$$

. The received wave variables $u_{pd}[i]$, $v_{uc}[i]$ are delayed versions of the transmitted wave variables $u_p[i]$, $v_{uc}[i]$ such that $u_{pd}[i] = u_p[i - p(i)]$ and $v_{uc}[i] = v_{uc}[i - c(i)]$ in which $p(i)$, $c(i) \in \{0, 1, \dots, N\}$ are the respective delay at time i .

C. Passive Sampler and Passive Hold

The passive sampler denoted (PS, T_s) in Fig. 2 and the corresponding passive hold denoted (PH, T_s) must be designed such that the following inequality is satisfied $\forall N > 0$:

$$\int_0^{NT_s} (u_p^\top(t)u_p(t) - v_{ucd}^\top(t)v_{ucd}(t))dt - \sum_{i=0}^{N-1} (u_p^\top[i]u_p[i] - v_{ucd}^\top[i]v_{ucd}[i]) \geq 0. \quad (8)$$

It will be seen that (8) is a sufficient conditions for the PS, T_s and PH, T_s to satisfy in order to construct our L_2^m -stable network depicted in Fig. 2. It is obvious that a sufficient condition to satisfy (8) is to design the PS to satisfy the following inequality:

$$\sum_{i=0}^{N-1} u_p^\top[i]u_p[i] \leq \int_0^{NT_s} u_p^\top(t)u_p(t)dt \quad (9)$$

and for the PH to satisfy the following inequality:

$$\int_0^{NT_s} v_{ucd}^\top(t)v_{ucd}(t)dt \leq \sum_{i=0}^{N-1} v_{ucd}^\top[i]v_{ucd}[i]. \quad (10)$$

Therefore, we shall evaluate a PS and PH which satisfy (9) and (10) respectively. Denote each j^{th} element of the column

vectors $u_p(t), u_p[i]$ as $u_{p_j}(t), u_{p_j}[i]$ in which $j = \{1, \dots, m\}$ and assume that $u_{p_j}(t) = 0$, if $t < 0$. Our proposed implementation of the PS which satisfies condition (9) is as follows:

$$u_{p_j}[i] = \sqrt{\int_{(i-1)T_s}^{iT_s} u_{p_j}^2(t) dt} \operatorname{sgn}\left(\int_{(i-1)T_s}^{iT_s} u_{p_j}(t) dt\right). \quad (11)$$

Denote each j^{th} element of the column vectors $v_{\text{ucd}}(t), v_{\text{ucd}}[i]$ as $v_{\text{ucd}_j}(t), v_{\text{ucd}_j}[i]$ in which $j = \{1, \dots, m\}$ and assume that $v_{\text{ucd}_j}[i] = 0$, if $i < 0$. Our proposed implementation of the PH that satisfies condition (10) is as follows:

$$v_{\text{ucd}_j}(t) = \frac{1}{\sqrt{T_s}} v_{\text{ucd}_j}[i-1], \quad t \in [iT_s, (i+1)T_s]. \quad (12)$$

1) *Deriving a PS From Data Reduction Methods Used in Telemanipulation Systems*: The sufficient conditions given by (9) and (10) in order to implement a PS and PH for our proposed architecture are similar to conditions required to send continuous-time wave variables over a digital network between two continuous time robotic systems in a telemanipulation network. For example in [18, Theorem 3.1] the sufficient Conditions 1 and 2 required to achieve an asymptotically stable telemanipulation system required the slaves received wave variable $\hat{u}_s(t)$ to be bounded by the original wave variable transmitted from the master $u_m(t)$ s.t. $\int_0^t \hat{u}_s^\top(s) \hat{u}_s(s) ds \leq \int_0^t u_m^\top(s) u_m(s) ds$ (Condition 1) and conversely the masters received wave variable $\hat{v}_m(t)$ is to be bounded by the original wave variable transmitted from the slave $v_s(t)$ s.t. $\int_0^t \hat{v}_m^\top(s) \hat{v}_m(s) ds \leq \int_0^t v_s^\top(s) v_s(s) ds$ (Condition 2). In order to satisfy these two conditions [18] proposed to use an identity *function-generator* and a time-varying gain *signal reconstructor* to satisfy Conditions 1 and 2. The recent work of [15] describes elaborate compression/ decompression techniques known as *energy supervised data reconstruction* in order to satisfy Conditions 1 and 2. Finally, the work of [29] describes *passive encoder/decoder* algorithms which satisfy Condition 1 and 2. Although Conditions 1 and 2 is too general for our framework these algorithms can typically be modified with a scaling term in order to satisfy (9) and (10) or the weaker condition (8). For example in [29] the authors propose a *passive interpolative downsampler* to generate a discrete-time wave-variable $u_m[i]$ from its discrete-time counterpart $u_m(t)$ as follows:

$$u_m[i] = \frac{1}{T_s} \int_{(i-1)T_s}^{iT_s} u_m(t) dt$$

they then transmit $u_m[i]$ over a network to the slave and extrapolate it such that $\hat{u}_s(t) = u_m[i - p(i)]$, $t \in [iT_s, (i+1)T_s)$. For simplicity of discussion if we assume that $p(i) = 0$ and u_m is a scalar then we have that:

$$\begin{aligned} \int_0^{NT_s} \hat{u}_s^2(t) dt &= \frac{1}{T_s} \sum_{i=1}^{N-1} \left(\int_{(i-1)T_s}^{iT_s} u_m(t) dt \right)^2 \\ &\leq \frac{T_s}{T_s} \sum_{i=1}^{N-1} \int_{(i-1)T_s}^{iT_s} u_m^2(t) dt \quad \text{Schwarz Inequality} \end{aligned}$$

$$\begin{aligned} &\leq \sum_{i=0}^{N-1} \int_{iT_s}^{(i+1)T_s} u_m^2(t) dt \\ &\leq \int_0^{NT_s} u_m^2(t) dt \quad (\text{Condition 1}). \end{aligned}$$

The passive interpolative downsampler *must be rescaled* in order to satisfy (9), therefore we propose a *linear-PS* to be implemented as follows:

$$u_{p_j}[i] = \frac{1}{\sqrt{T_s}} \int_{(i-1)T_s}^{iT_s} u_{p_j}(t) dt, \quad \forall j \in \{1, \dots, m\}. \quad (13)$$

Finally, it should be obvious that a PS designed to satisfy (9) and a PH designed to satisfy (10) can be used in a cascade manner in order to satisfy Condition 1 and 2 for continuous-time telemanipulation applications; however, the converse may not always be the case as we have shown. Therefore the engineer can look to the telemanipulation literature for guidance when choosing to design an appropriate PS and PH being careful that such a configuration satisfies (9) and (10) respectively or the weaker condition (8).

2) *Accounting for scaling effects of PS and PH*: The consequence of using our proposed PS and PH interconnected to a digital controller is that $\dot{\Theta}(t) \neq \dot{\Theta}_d[i]$ at steady-state. Intuitively this is due to the scaling of $u_p[i]$ due to the PS; however, the relationships are also dependent on the properties of both the controller and plant. We can derive these relationships precisely using the techniques used to prove [37, Lemma 4] in order to scale the controller set-point for the discrete-time case when using a *passive downsampler*. Assuming that the system is at a steady-state operating point we can use steady-state relationships in order to compare continuous-time and discrete-time components. Precisely, if there is no data-loss, the disturbance $\tau_d(t)$ and reference $\dot{\Theta}_r(t)$ are held constant then the following steady-state relationships hold: $\dot{\Theta}(t) = k_p \tau_u(t) \leq \frac{1}{\epsilon} \tau_u(t)$, $\tau_{\text{uc}}[i] = k_c \dot{e}[i] = \frac{1}{\epsilon_c} \dot{e}[i]$, $v_{\text{ucd}}(t) = \frac{1}{\sqrt{T_s}} v_{\text{uc}}[i]$, $u_{\text{pd}}[i] = \sqrt{T_s} u_p(t)$, $\dot{\Theta}_{sr}[i] = T_s \dot{\Theta}_{sr}(t)$, $\dot{\Theta}_{sr}(t) = -k_s \dot{\Theta}_r(t)$. Using these relationships Lemma 4 shows that for the SISO case that at steady-state:

$$\dot{\Theta}(t) = k_s \frac{T_s}{\sqrt{T_s}} \frac{k_c k_p}{1 + k_c k_p} \dot{\Theta}_r(t) + \frac{k_p}{1 + k_p k_c} \tau_d(t).$$

Therefore in order for $\dot{\Theta}(t) = \dot{\Theta}_r(t)$ when $\tau_d(t) = 0$ and $\epsilon_c \epsilon \ll 1$ then $k_s = \frac{1 + k_c k_p}{k_c k_p \sqrt{T_s}} \approx \frac{1}{\sqrt{T_s}}$. N.B. that this result takes in to account the scaling effects of the *IPES*. For the case in which the *IPES* is not used and $\dot{\Theta}_{sr}[i] = \dot{\Theta}_{sr}(t)$ for $t \in [iT_s, (i+1)T_s)$ then $k_s = \sqrt{T_s}$.

D. Passive Digital Controller

Typically a passive continuous-time PD controller is implemented as

$$\begin{aligned} \dot{e}_1(t) &= (\dot{\Theta}_d(t) + \dot{\Theta}_{sr}(t)) \\ \tau_{\text{uc}}(t) &= K_p e_1(t) + K_d (\dot{\Theta}_d(t) + \dot{\Theta}_{sr}(t)). \end{aligned}$$

N.B. the ‘‘proportional’’ term K_p and ‘‘derivative’’ term K_d are with respect to the integrated velocity error term $e_1(t)$ therefore our proposed control structure includes an integrator

which results in an infinite steady-state gain. Furthermore K_p and K_d are real symmetric matrices so although we may refer to their structure in terms of scalar terms in order to simplify discussion in no way does this imply that better performing controllers can not be implemented which can exploit this general non-diagonal control structure. A state-space realization of the controller can be described by

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (14)$$

$$y(t) = Cx(t) + Du(t). \quad (15)$$

where $A = 0$, $B = I$, $C = K_p = K_p^\top > 0$, $D = K_d = K_d^\top > 0$ (all matrices are in $\mathbb{R}^{m \times m}$).

To obtain a digital controller, first we design the discrete-time equivalent passive controller $G_{pc} : \dot{e}_1[i] \mapsto \tau_{uc}[i]$ computed from the state-space realization (14)-(15) with sampling period T_s . The resulting controller is

$$\begin{aligned} x[i+1] &= \Phi_o x[i] + \Gamma_o u[i] \\ y[i] &= \mathbf{C}_p x[i] + \mathbf{D}_p u[i]. \end{aligned} \quad (16)$$

where $u[i] = (\dot{\Theta}_d[i] + \dot{\Theta}_{sr}[i])$. Note that in our derivation we scaled \mathbf{C}_p and \mathbf{D}_p by $\frac{1}{T_s}$ in order to closely match the frequency response of the discrete time implementation with that of the continuous time model it is derived from. A discrete passive controller can be synthesized using the method presented in [38] and is described by

$$\begin{aligned} \Phi_o &= e^{A_o T_s}, \quad A_o = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ K_p & 0 \end{bmatrix} \\ &= \begin{bmatrix} I & 0 \\ T_s K_p & I \end{bmatrix} \\ \Gamma_o &= \int_0^{T_s} e^{A_o \eta} d\eta B_o, \quad B_o = \begin{bmatrix} B \\ D \end{bmatrix} = \begin{bmatrix} I \\ K_d \end{bmatrix} \\ &= \begin{bmatrix} T_s I & 0 \\ \frac{T_s^2}{2} K_p & T_s I \end{bmatrix} \begin{bmatrix} I \\ K_d \end{bmatrix} = \begin{bmatrix} T_s I \\ \frac{T_s^2}{2} K_p + T_s K_d \end{bmatrix} \\ \mathbf{C}_p &= \frac{1}{T_s} \mathbf{C}_o (\Phi_o - I) = \frac{1}{T_s} \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} 0 & 0 \\ T_s K_p & 0 \end{bmatrix} \\ &= \begin{bmatrix} K_p & 0 \end{bmatrix} \\ \mathbf{D}_p &= \frac{1}{T_s} \mathbf{C}_o \Gamma_o = \frac{1}{T_s} \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} T_s I \\ \frac{T_s^2}{2} K_p + T_s K_d \end{bmatrix} \\ &= \frac{T_s}{2} K_p + K_d. \end{aligned}$$

It should be noted that this is not a minimal realization for this controller however, solving for

$$\begin{aligned} H(z) &= \mathbf{C}_p (zI - \Phi_o)^{-1} \Gamma_o + \mathbf{D}_p \\ &= \begin{bmatrix} K_p & 0 \end{bmatrix} \begin{bmatrix} (z-I)^{-1} & 0 \\ T_s(z-I)^{-1} K_p (z-I)^{-1} & (z-I)^{-1} \end{bmatrix} \\ &\quad \begin{bmatrix} T_s I \\ \frac{T_s^2}{2} K_p + T_s K_d \end{bmatrix} + \mathbf{D}_p \\ &= K_p (z-I)^{-1} T_s I + \mathbf{D}_p \end{aligned}$$

results in a minimal controller $\Sigma = \{\Phi_o, \Gamma_o, \mathbf{C}_p, \mathbf{D}_p\}$ such that: $\Phi_o = I \in \mathbb{R}^{m \times m}$, $\Gamma_o = T_s I \in \mathbb{R}^{m \times m}$, $\mathbf{C}_p = K_p \in \mathbb{R}^{m \times m}$, and $\mathbf{D}_p = \frac{T_s}{2} K_p + K_d \in \mathbb{R}^{m \times m}$.

In order to ensure that the mapping $G_{sp} : \dot{e}[i] \mapsto \tau_{uc}[i]$ is strictly-output passive (see Fig. 2) we chose $\epsilon_c > 0$ and denote $G = (I + \epsilon_c \mathbf{D}_p)$. Therefore the strictly-output passive controller has a discrete time realization $\Sigma_{sp} = \{\Phi_{sp}, \Gamma_{sp}, \mathbf{C}_{sp}, \mathbf{D}_{sp}\}$ which is described analogously to (16) in which

$$\begin{aligned} \Phi_{sp} &= \Phi_o - \epsilon_c \Gamma_o G^{-1} \mathbf{C}_p = I - \epsilon_c T_s G^{-1} K_p \\ \Gamma_{sp} &= \Gamma_o (I - \epsilon_c G^{-1} \mathbf{D}_p) = T_s (I - \epsilon_c G^{-1} \mathbf{D}_p) \\ \mathbf{C}_{sp} &= G^{-1} \mathbf{C}_p = G^{-1} K_p \\ \mathbf{D}_{sp} &= G^{-1} \mathbf{D}_p = G^{-1} \mathbf{D}_p. \end{aligned}$$

Finally the effects of the wave-variables (Section IV-B) need to be considered, therefore the final implementation of the strictly-output passive digital controller depicted in Fig. 2 with inputs $(u_{pd}[i], \dot{\Theta}_{sr}[i])$ and outputs $(\tau_{uc}[i], v_{uc}[i])$ is as follows:

$$\begin{aligned} x[i+1] &= \Phi_{fe} x[i] + \Gamma_{fe} \left(\sqrt{\frac{2}{b}} u_{pd}[i] + \dot{\Theta}_{sr}[i] \right) \\ \tau_{uc}[i] &= \mathbf{C}_{fe} x[i] + \mathbf{D}_{fe} \left(\sqrt{\frac{2}{b}} u_{pd}[i] + \dot{\Theta}_{sr}[i] \right) \\ v_{uc}[i] &= u_{pd}[i] - \sqrt{\frac{2}{b}} \tau_{uc}[i] \end{aligned}$$

in which $G_1 = I + \frac{1}{b} \mathbf{D}_{sp}$, $\mathbf{C}_{fe} = G_1^{-1} \mathbf{C}_{sp}$, $\mathbf{D}_{fe} = G_1^{-1} \mathbf{D}_{sp}$, $\Phi_{fe} = \Phi_{sp} - \frac{1}{b} \Gamma_{sp} \mathbf{C}_{fe}$ and $\Gamma_{fe} = \Gamma_{sp} (I - \frac{1}{b} \mathbf{D}_{fe})$.

E. Mapping Discrete-Time Controller Variables to Continuous Time

The *inner-product equivalent sampler (IPES)* and zero-order-hold (*ZOH*) blocks are introduced in order to relate the continuous-time robots inputs and outputs to the discrete-time controllers inputs and outputs. Specifically the *IPES* at the input of the digital controller and *ZOH* at the output can be used to ensure that the overall system $G_{net} : [\dot{\Theta}_{sr}^\top(t), \tau_d^\top(t)]^\top \mapsto [\tau_{uc}^\top(t), \dot{\Theta}^\top(t)]^\top$ is (*strictly output*) passive. We present the *non-causal* version of the *IPESH* which is based on the *causal* version of the *IPESH* presented in [17], [37, Definition 4] and based on the earlier work of [13], [39]. The *IPES* is implemented as follows:

$$\begin{aligned} x(t) &= \int_0^t \dot{\Theta}_{sr}(\tau) d\tau \\ \dot{\Theta}_{sr}[i] &= x((i+1)T_s) - x(iT_s). \end{aligned} \quad (17)$$

The *ZOH* is implemented as follows:

$$\tau_{uc}(t) = \tau_{uc}[i], \quad \forall t \in [iT_s, (i+1)T_s). \quad (18)$$

$\dot{\Theta}_{sr}(t)$ denotes an appropriately scaled velocity profile for the robot to follow such that $\dot{\Theta}_{sr}(t) = -k_s \dot{\Theta}_r(t)$ in which $\dot{\Theta}_r(t)$ is the desired trajectory to track and k_s is a positive real value which will be determined to account for the scaling effects which result from the use of the passive sampler described later in Section IV-B. N.B. the ordering of the *IPESH* is reversed from typical applications [13], [17], [37], [39] in which the *IPES* is located at the output of a continuous-time system and the *ZOH* is located at the input to the continuous-time system. In [37, Appendix E] a deeper discussion is provided showing

that such traditional *IPESH* arrangements are indeed causal and can be used to synthesize passive discrete-time SISO *LTI* filters. In fact the passive digital controllers presented in Section IV-D were derived using the *IPESH* in its causal framework [38] a different realization which is fundamentally based on the *IPESH* appeared in [40]. However, the causal *IPESH* not only can be used to generate passive digital controllers it can also be realized through the use of state-space observers for *LTI* systems [38], [41]. The difficulty in applying the *IPESH* in the causal framework is that it is not clear how to systematically apply it to continuous-time *non-linear* plants. Therefore implementations which attempt to apply the *IPESH* to a continuous-time system are only approximate realizations [13], [39] in which some non-passive behavior remains. The recent work of [42] shows promise of achieving improved discrete-time mappings which preserve passivity like-properties for certain non-linear continuous-time systems; however, these results are still sampling rate limited. Which leads us to conclude the novelty of our proposed framework we simply use the *IPESH* in this non-causal manner in order to relate our digital control inputs and outputs back to the continuous-time domain in order to derive an L_2^m -stable architecture involving a digital controller connected to a non-linear continuous-time plant which does not require a complex non-linear observer. It also allows us to precisely include a digital controller for the continuous-time system which is interconnected to wave-variables in which neither [18] nor [15] could fully address.

V. ANALYSIS OF THE NCS

This section presents the main results which provide sufficient conditions for our proposed networked control architecture to remain passive and stable. In addition our results show how minimum phase LTI asymptotically stable systems which are not passive can be rendered strictly-output passive, and thus, used in our proposed architecture.

A. Passivity Analysis

We first present the results that confirm the passivity of the proposed sampler and hold devices and derive the scaling conditions that account for the effects of the PS and PH. Then, we prove that the NCS is L_2^m -stable and we discuss practical networked delay conditions and weak synchronization requirements that satisfy the assumption of our framework.

Lemma 1: The proposed PS given by (11) satisfies the passive-sampling condition (9).

Proof: Substituting (11) into the left hand side of (9) it is clear that

$$\begin{aligned} \sum_{i=0}^{N-1} u_p^\top[i] u_p[i] &= \sum_{j=1}^m \sum_{i=0}^{N-1} \left(\sqrt{\int_{(i-1)T_s}^{iT_s} u_{p_j}^2(t) dt} \operatorname{sgn}\left(\int_{(i-1)T_s}^{iT_s} u_{p_j}(t) dt\right) \right)^2 \\ &= \sum_{j=1}^m \sum_{i=0}^{N-1} \int_{(i-1)T_s}^{iT_s} u_{p_j}^2(t) dt \\ &\leq \int_0^{NT_s} u_p^\top(t) u_p(t) dt \text{ holds.} \end{aligned}$$

Lemma 2: The proposed PH given by (12) satisfies the passive-hold condition (10). ■

Proof: Substituting (12) into the left hand side of (10) it is clear that

$$\begin{aligned} \int_0^{NT_s} v_{\text{ucd}}^\top(t) v_{\text{ucd}}(t) dt &= \sum_{j=1}^m \sum_{i=0}^{N-1} \int_{iT_s}^{(i+1)T_s} v_{\text{ucd}_j}^2(t) dt \\ &= \sum_{j=1}^m \sum_{i=0}^{N-1} \frac{1}{T_s} \int_{iT_s}^{(i+1)T_s} v_{\text{ucd}_j}^2[i-1] dt \\ &= \sum_{j=1}^m \sum_{i=0}^{N-2} \frac{T_s}{T_s} v_{\text{ucd}_j}^2[i] \\ &\leq \sum_{i=0}^{N-1} v_{\text{ucd}}^\top[i] v_{\text{ucd}}[i]. \end{aligned}$$

Lemma 3: The proposed *linear-PS* given by (13) satisfies the passive-sampler condition (9).

Proof: Substituting (13) into the left hand side of (9) results in

$$\begin{aligned} \sum_{i=0}^{N-1} u_p^\top[i] u_p[i] &= \sum_{j=1}^m \sum_{i=0}^{N-1} u_{p_j}^2[i] \\ &= \sum_{j=1}^m \sum_{i=1}^{N-1} \left(\frac{1}{\sqrt{T_s}} \int_{(i-1)T_s}^{iT_s} u_{p_j}(t) dt \right)^2 \\ &\leq \frac{T_s}{T_s} \sum_{j=1}^m \sum_{i=1}^{N-1} \int_{(i-1)T_s}^{iT_s} u_{p_j}^2(t) dt \\ &\leq \int_0^{NT_s} u_p^\top(t) u_p(t) dt. \end{aligned}$$

Lemma 4: If the SISO continuous-time plant and controller subsystems depicted in Fig. 2 are not subject to additional data-loss and the disturbance $\tau_d(t)$ and reference $\dot{\Theta}_{sr}(t) = -k_s \dot{\Theta}_r(t)$ are held constant then the following steady-state relationship holds

$$\dot{\Theta}(t) = k_s \frac{T_s}{\sqrt{T_s}} \frac{k_c k_p}{1 + k_c k_p} \dot{\Theta}_r(t) + \frac{k_p}{1 + k_p k_c} \tau_d(t)$$

in which $k_p = \frac{\dot{\Theta}(t)}{\tau_u(t)}$, $k_c = \frac{1}{\epsilon_c} = \frac{\dot{\Theta}[i]}{\tau_{\text{uc}}[i]}$.

Proof: We derive these relationships as follows:

$$u_{\text{pd}}[i] = \sqrt{T_s} u_p(t) \quad (19)$$

$$v_{\text{ucd}}(t) = \frac{1}{\sqrt{T_s}} v_{\text{uc}}[i] \quad (20)$$

$$\begin{bmatrix} v_{\text{uc}}[i] \\ \dot{\Theta}_d[i] \end{bmatrix} = \begin{bmatrix} 1 & -\sqrt{\frac{2}{b}} \\ \sqrt{\frac{2}{b}} & -\frac{1}{b} \end{bmatrix} \begin{bmatrix} u_{\text{pd}}[i] \\ \tau_{\text{uc}}[i] \end{bmatrix} \quad (21)$$

$$\begin{bmatrix} u_p(t) \\ \tau_{\text{ucd}}(t) \end{bmatrix} = \begin{bmatrix} -1 & \sqrt{2b} \\ -\sqrt{2b} & b \end{bmatrix} \begin{bmatrix} v_{\text{ucd}}(t) \\ \dot{\Theta}(t) \end{bmatrix}. \quad (22)$$

Substituting (19) into (21), and (20) into (22) results in

$$\begin{bmatrix} v_{uc}[i] \\ \dot{\Theta}_d[i] \end{bmatrix} = \begin{bmatrix} \sqrt{T_s} & -\sqrt{\frac{2}{b}} \\ \sqrt{\frac{2T_s}{b}} & -\frac{1}{b} \end{bmatrix} \begin{bmatrix} u_p(t) \\ \tau_{uc}[i] \end{bmatrix} \quad (23)$$

$$\begin{bmatrix} u_p(t) \\ \tau_{ucd}(t) \end{bmatrix} = \begin{bmatrix} -\sqrt{\frac{1}{T_s}} & \sqrt{2b} \\ -\sqrt{\frac{2b}{T_s}} & b \end{bmatrix} \begin{bmatrix} v_{uc}[i] \\ \dot{\Theta}(t) \end{bmatrix} \quad (24)$$

respectively. Which can be written in the following form:

$$\begin{bmatrix} \tau_{ucd}(t) \\ \dot{\Theta}_d[i] \end{bmatrix} = C_1 \begin{bmatrix} v_{uc}[i] \\ u_p(t) \end{bmatrix} + C_2 \begin{bmatrix} \tau_{uc}[i] \\ \dot{\Theta}(t) \end{bmatrix} \quad (25)$$

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

$$\begin{bmatrix} v_{uc}[i] \\ u_p(t) \end{bmatrix} = C_3 \begin{bmatrix} v_{uc}[i] \\ u_p(t) \end{bmatrix} + C_4 \begin{bmatrix} \tau_{uc}[i] \\ \dot{\Theta}(t) \end{bmatrix} \quad (26)$$

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

Solving for the wave variables in terms of $\tau_{uc}[i]$ and $\dot{\Theta}(t)$ in (26) results in

$$\begin{bmatrix} v_{uc}[i] \\ u_p(t) \end{bmatrix} = (I - C_3)^{-1} C_4 \begin{bmatrix} \tau_{uc}[i] \\ \dot{\Theta}(t) \end{bmatrix}. \quad (27)$$

Substituting (27) into (25) results in

$$\begin{bmatrix} \tau_{ucd}(t) \\ \dot{\Theta}_d[i] \end{bmatrix} = [C_2 + C_1(I - C_3)^{-1} C_4] \begin{bmatrix} \tau_{uc}[i] \\ \dot{\Theta}(t) \end{bmatrix}. \quad (28)$$

Solving for (28) results in

$$\begin{bmatrix} \tau_{ucd}(t) \\ \dot{\Theta}_d[i] \end{bmatrix} = \begin{bmatrix} \sqrt{1/T_s} & 0 \\ 0 & \sqrt{T_s} \end{bmatrix} \begin{bmatrix} \tau_{uc}[i] \\ \dot{\Theta}(t) \end{bmatrix}. \quad (29)$$

Using (29) it is a simple exercise to show that

$$\dot{\Theta}(t) = k_s \frac{T_s}{\sqrt{T_s}} \frac{k_c k_p}{1 + k_c k_p} \dot{\Theta}_r(t) + \frac{k_p}{1 + k_p k_c} \tau_d(t).$$

Finally, note that our analysis reveals that the controller attenuates the steady-state disturbances τ_d such that $\dot{\Theta} \approx \frac{1}{k_c} \tau_d$ when $k_p k_c \gg 1$ and $\dot{\Theta}_r = 0$, independent of T_s . ■

Lemma 5: The proposed IPESH has the following properties:

- i) equivalent inner-products between the continuous-time variables $(\dot{\Theta}_{sr}(t), \tau_{uc}(t))$ and the discrete-time variables $(\dot{\Theta}_{sr}[i], \tau_{uc}[i])$ s.t. $\langle \tau_{uc}, \dot{\Theta}_{sr} \rangle_N = \langle \tau_{uc}, \dot{\Theta}_{sr} \rangle_{NT_s}$ holds $\forall N \geq 1$.
- ii) proportional two-norms for the continuous-time variable $\tau_{uc}(t)$ and the discrete-time variable $\tau_{uc}[i]$ s.t. $\|(\tau_{uc})_{NT_s}\|_2^2 = T_s \|(\tau_{uc})_N\|_2^2$.

Proof:

i)

$$\begin{aligned} \langle \tau_{uc}, \dot{\Theta}_{sr} \rangle_N &= \sum_{i=0}^{N-1} \tau_{uc}^T[i] \dot{\Theta}_{sr}[i] \\ &= \sum_{i=0}^{N-1} \sum_{j=1}^m \tau_{uc_j}[i] \dot{\Theta}_{sr_j}[i] \\ &= \sum_{j=1}^m \sum_{i=0}^{N-1} \tau_{uc_j}[i] \int_{iT_s}^{(i+1)T_s} \dot{\Theta}_{sr_j}(\tau) d\tau \\ &= \sum_{j=1}^m \sum_{i=0}^{N-1} \int_{iT_s}^{(i+1)T_s} \tau_{uc_j}(\tau) \dot{\Theta}_{sr_j}(\tau) d\tau \\ &= \sum_{j=1}^m \int_0^{NT_s} \tau_{uc_j}(\tau) \dot{\Theta}_{sr_j}(\tau) d\tau \\ &= \int_0^{NT_s} \tau_{uc}^T(\tau) \dot{\Theta}_{sr}(\tau) d\tau \\ \langle \tau_{uc}, \dot{\Theta}_{sr} \rangle_N &= \langle \tau_{uc}, \dot{\Theta}_{sr} \rangle_{NT_s} \text{ holds } \forall N \geq 1. \end{aligned} \quad (30)$$

ii)

$$\begin{aligned} \|(\tau_{uc})_{NT_s}\|_2^2 &= \int_0^{NT_s} \tau_{uc}^T(t) \tau_{uc}(t) dt \\ &= \sum_{j=1}^m \int_0^{NT_s} \tau_{uc_j}(t) \tau_{uc_j}(t) dt \\ &= \sum_{j=1}^m \sum_{i=0}^{N-1} \int_{iT_s}^{(i+1)T_s} \tau_{uc_j}[i] \tau_{uc_j}[i] dt \\ &= T_s \sum_{j=1}^m \sum_{i=0}^{N-1} \tau_{uc_j}[i] \tau_{uc_j}[i] \\ &= T_s \sum_{i=0}^{N-1} \tau_{uc}^T[i] \tau_{uc}[i] \\ \|(\tau_{uc})_{NT_s}\|_2^2 &= T_s \|(\tau_{uc})_N\|_2^2. \end{aligned} \quad (31)$$

■
Theorem 2: For the wireless control architecture depicted in Fig. 2 which consists of the passive robot described by (3) and (4) and the passive digital controller described by (16), if the communication protocol ensures that both Condition i) $i - p(i) \neq j - p(j)$ and Condition ii) $i - c(i) \neq j - c(j)$ for all $j \neq i$ in which $j, i \in \{0, 1, \dots, N-1\}$ then

$$\int_0^{NT_s} \dot{\Theta}^T(t) \tau_{ucd}(t) dt \geq \sum_{i=0}^{N-1} \tau_{uc}^T[i] \dot{\Theta}_d[i] \quad (32)$$

always holds therefore if $\epsilon_c = \epsilon = 0$ then the system depicted in Fig. 2 is passive in addition if $\epsilon_c > 0$, and $\epsilon > 0$ then the system is both strictly-output passive and L_2^m stable.

Proof: The PS and PH satisfy (8) which can be compactly written as

$$\|(u_p)_{NT_s}\|_2^2 - \|(v_{ucd})_{NT_s}\|_2^2 \geq \|(u_p)_N\|_2^2 - \|(v_{ucd})_N\|_2^2. \quad (33)$$

Integrating both sides of (7) and substituting into (33) results in

$$\langle \dot{\Theta}, \tau_{ucd} \rangle_{NT_s} \geq \|(u_p)_N\|_2^2 - \|(v_{ucd})_N\|_2^2.$$

Condition i) ensures that $\|(u_{pd})_N\|_2^2 = \sum_{i=0}^{N-1} u_p^T[i - p(i)]u_p[i - p(i)] \leq \|(u_p)_N\|_2^2$ analogously Condition ii) ensures that $\|(v_{ucd})_N\|_2^2 = \sum_{i=0}^{N-1} v_{uc}^T[i - c(i)]v_{uc}[i - c(i)] \leq \|(v_{uc})_N\|_2^2$ therefore

$$\begin{aligned} \|(u_p)_N\|_2^2 - \|(v_{ucd})_N\|_2^2 &\geq \|(u_{pd})_N\|_2^2 - \|(v_{uc})_N\|_2^2 \\ &\geq \langle \dot{\Theta}_d, \tau_{uc} \rangle_N \end{aligned}$$

will always hold. Therefore, we can satisfy (32) which can be more compactly written as

$$\langle \dot{\Theta}, \tau_{uc} \rangle_{NT_s} \geq \langle \dot{\Theta}_d, \tau_{uc} \rangle_N. \quad (34)$$

Recall that the gravity compensated robot satisfies (6). Denoting $V(x(0))$ as β_r for the robot and $\beta_c > 0$ to account for non-zero initial conditions for the passive controller. Then the robot satisfies

$$\langle \dot{\Theta}, \tau_u \rangle_{NT_s} \geq \epsilon \|\langle \dot{\Theta} \rangle_{NT_s}\|_2^2 - \beta_r, \quad (35)$$

and the controller satisfies

$$\langle \tau_{uc}, \dot{\epsilon} \rangle_N \geq \epsilon_c \|\langle \tau_{uc} \rangle_N\|_2^2 - \beta_c. \quad (36)$$

We recall that

$$\tau_{ucd}(t) = \tau_d(t) - \tau_u(t), \text{ and} \quad (37)$$

$$\dot{\Theta}_d[i] = \dot{\epsilon}[i] - \dot{\Theta}_{sr}[i]. \quad (38)$$

Substituting (37) into the left hand side of (34) and (38) into the right hand side of (34) results in

$$\begin{aligned} \langle \dot{\Theta}, \tau_d \rangle_{NT_s} - \langle \dot{\Theta}, \tau_u \rangle_{NT_s} &\geq \langle \dot{\epsilon}, \tau_{uc} \rangle_N - \langle \dot{\Theta}_{sr}, \tau_{uc} \rangle_N \\ \langle \dot{\Theta}, \tau_d \rangle_{NT_s} + \langle \dot{\Theta}_{sr}, \tau_{uc} \rangle_N &\geq \langle \dot{\Theta}, \tau_u \rangle_{NT_s} + \langle \dot{\epsilon}, \tau_{uc} \rangle_N. \end{aligned} \quad (39)$$

Substituting (35) and (36) into (39) results in

$$\begin{aligned} \langle \dot{\Theta}, \tau_d \rangle_{NT_s} + \langle \dot{\Theta}_{sr}, \tau_{uc} \rangle_N &\geq \\ \epsilon \|\langle \dot{\Theta} \rangle_{NT_s}\|_2^2 + \epsilon_c \|\langle \tau_{uc} \rangle_N\|_2^2 &- (\beta_r + \beta_c) \end{aligned} \quad (40)$$

Next, we recall the first two properties listed for the *IPESH* in Section IV-E in which (30) is $\langle \tau_{uc}, \dot{\Theta}_{sr} \rangle_N = \langle \tau_{uc}, \dot{\Theta}_{sr} \rangle_{NT_s}$ and (31) is $T_s \|\langle \tau_{uc} \rangle_N\|_2^2 = \|\langle \tau_{uc} \rangle_{NT_s}\|_2^2$. Substituting (30) and (31) into (40) results in

$$\langle y, u \rangle_{NT_s} \geq \epsilon_s \|\langle y \rangle_{NT_s}\|_2^2 - \beta_s. \quad (41)$$

in which

$$\begin{aligned} y &= [\dot{\Theta}^T, \tau_{uc}^T]^T, \quad u = [\tau_d^T, \dot{\Theta}_{sr}^T]^T \\ \epsilon_s &= \min \left\{ \epsilon, \frac{\epsilon_c}{T_s} \right\}, \quad \beta_s = \beta_r + \beta_c. \end{aligned}$$

Therefore, (41) satisfies Definition 1-I for passivity when $(\epsilon_c, \epsilon) \geq 0$ and either $\epsilon_c = 0$ or $\epsilon = 0 \implies \epsilon_s = 0$. Furthermore (41) satisfies Definition 1-II when $\epsilon_c > 0$ and $\epsilon > 0 \implies \epsilon_s > 0$ in order for the system to be strictly-output passive, therefore from Theorem 1 the strictly-output passive system is also L_2^m -stable. ■

Obviously these results apply for the more general case when the robotic and controller subsystems are replaced with arbitrary passive systems.

Corollary 1: For the wireless control architecture depicted in Fig. 2 in which the robot ($G_{\text{robot}}(\tau(t))$) is replaced by any

passive system satisfying Definition 1-I (with gravity compensation disabled $g(\Theta(t)) = 0$) and the passive digital controller ($G_{pc}(\dot{\epsilon}_1[i])$) satisfies Definition 1-I, if the communication protocol ensures that both Condition i) $i - p(i) \neq j - p(j)$ and Condition ii) $i - c(i) \neq j - c(j)$ for all $j \neq i$ in which $j, i \in \{0, 1, \dots, N-1\}$ then

$$\int_0^{NT_s} \dot{\Theta}^T(t) \tau_{ucd}(t) dt \geq \sum_{i=0}^{(N-1)} \tau_{uc}^T[i] \dot{\Theta}_d[i] \quad (42)$$

always holds therefore if $\epsilon_c = \epsilon = 0$ then the system depicted in Fig. 2 is passive in addition if $\epsilon_c > 0$, and $\epsilon > 0$ then the system is both strictly-output passive and L_2^m stable.

1) *Communication Protocols Which Satisfy Condition i) and Condition ii) and Addressing Synchronization:* Condition i) and Condition ii) can be satisfied by communication protocols which prevent processing of duplicate transmissions of wave variables [17]. TCP is an appropriate protocol because it provides an unduplicated ordered stream of data unlike the User Datagram Protocol (UDP) protocol which *can duplicate datagrams* due to the effects of having potentially multiple routes for the data to travel. This *important detail* gets missed in the presentation of [13, Proposition 2] and [18] which appeared to have incorrectly assumed that if the transmitting node does not duplicate data when using either a ‘‘packet switching communication channel’’ or ‘‘UDP’’ respectively data will not get replicated at the receiver. By making Condition i) and Condition ii) explicit we hope to *clarify* this necessary assumption which needs to be satisfied when transmitting discrete-time wave variables. Note that Condition i) and Condition ii) *do not* require that the data needs to be ordered or for all the data to arrive as is guaranteed by the TCP protocol. Therefore, augmenting the transmitted wave variables with an index and using a table to enforce Condition i) and Condition ii) can allow an engineer to use the UDP protocol which should minimize delay and therefore improve performance. Furthermore, the controller can essentially be run as an asynchronous non-periodic task in which it only needs to compute and send a new control command as new data is received from the plant [43]. An asynchronous non-periodic task as defined in [43] requires the global digital clocks of the plant subsystem and the digital controller subsystem to be synchronized; however, the controller does not need to be executed in a synchronous manner as governed by a periodic schedule instead its execution schedule is determined by the arrival of sensor feedback data from the plant. The reason for the *relatively weak* global clock synchronization assumption becomes obvious when studying the proofs in [43] for stability because removing the synchronization assumption in order to relate input-output relationships between the controller and plant subsystems would render the task of proving stability *ad absurdum*. The *main* problem with not enforcing a global clock is that if the digital controller and the plant are executed at the same periodic rate and the clocks of the two systems drift apart then extremely large TCP communication delays will result due to buffering of data. From a practical standpoint enforcing a good global clock synchronization scheme tends to lead to more efficient system implementations and can be

enforced using readily available methods [44]–[46].

B. Passifying Asymptotically Stable LTI Systems

A direct result of Corollary 1 is that stable LTI passive systems with a corresponding square real-rational transfer function matrix $G_{\text{robot}}(s)$ can be rendered strictly-output passive when $\epsilon > 0$ which results in an L_2^m stable system. However, if $G_{\text{robot}}(s)$ is already strictly-output passive then the additional analog feedback loop can be eliminated by setting $\epsilon = 0$. We wish to consider the control of a minimum phase LTI asymptotically stable system which is not passive and has the corresponding real-rational transfer function matrix $G(s)$. We shall demonstrate how to design a low-complexity asymptotically stable analog filter $H_*(s)$ such that $G_{\text{robot}}(s) = G(s) + H_*(s)$ is rendered strictly output passive such that it can be incorporated into our networked digital control framework in which $\epsilon = 0$.

For simplicity of discussion we shall consider the control of SISO LTI minimum phase systems $G(s)$. The passivity index described in Section III can be used to design a low-complexity asymptotically stable high-pass filter $H_{\text{hp}}(s)$ such that if $v_f(G_{\text{robot}}(s), \omega) < 0$ ($v_f(H_{\text{hp}}(s), \omega) < -v_f(G(s), \omega)$) for all ω then $G_{\text{robot}}(s)$ is strictly-output passive. Furthermore if $H_{\text{hp}}(s)$ is designed such that $|H_{\text{hp}}(j0)| = 0$ and $|H_{\text{hp}}(j\omega)|_{\lim \omega \rightarrow \infty} \neq 0$ then the output of $G(s)$ can be directly controlled by our digital controller at steady-state. A typical high-pass filter which may satisfy these conditions is one of the form $H_{\text{hp}}(s) = \frac{vs}{s + \omega_{\text{hp}}}$ in which $v, \omega_{\text{hp}} > 0$ and can be easily diagonalized for the MIMO case. If instead we choose to design an asymptotically stable band-pass filter $H_{\text{bp}}(s)$ in which both $|H_{\text{bp}}(j0)| = |H_{\text{bp}}(j\omega)|_{\lim \omega \rightarrow \infty} = 0$ and the combined system $v_f(G_{\text{robot}}(s), \omega) < 0$ for all ω except in the limit such that $v_f(G_{\text{robot}}(s), \omega)_{\lim \omega \rightarrow \infty} = 0$ then we can only conclude that the combined system $G_{\text{robot}}(s) = (G(s) + H_{\text{bp}}(s))$ is passive and asymptotically stable; however, a necessary and sufficient test to determine if $G_{\text{robot}}(s)$ is strictly-output passive is to determine the feasibility of the following linear-matrix inequality (LMI) which results from the application of [47, Corollary 1].

Corollary 2: The real-rational transfer-function matrix $G_{\text{robot}}(s)$ with a corresponding minimal state-space realization $\Sigma : \{A, B, C, D\}$ s.t. $\dot{x} = Ax + Bu$, $y = Cx + Du$ in which $x \in \mathbb{R}^n$, $y, u \in \mathbb{R}^m$, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{m \times n}$, and $D \in \mathbb{R}^{m \times m}$ is strictly-output passive iff there exists a real symmetric positive definite matrix $P = P^T > 0$ and positive real constant $\epsilon > 0$ s.t. the following LMI is satisfied:

$$\begin{bmatrix} A^T P + PA + \epsilon C^T C & PB - \frac{1}{2} C^T + \epsilon C^T D \\ \left(PB - \frac{1}{2} C^T + \epsilon C^T D \right)^T & \epsilon D^T D - \frac{1}{2} (D^T + D) \end{bmatrix} \leq 0.$$

A typical band-pass filter which may render $G_{\text{robot}}(s)$ to be strictly output passive is one of the form $H_{\text{bp}}(s) = k_{\text{bp}} \frac{\omega_{\text{bp}}^2 s}{s^2 + 2\zeta_{\text{bp}} \omega_{\text{bp}} s + \omega_{\text{bp}}^2}$ in which $k_{\text{bp}}, \omega_{\text{bp}} > 0$ and $0 < \zeta_{\text{bp}} \leq 1$. This band-pass filter will be used in the digital control of a mass-spring damper system described in the following subsection.

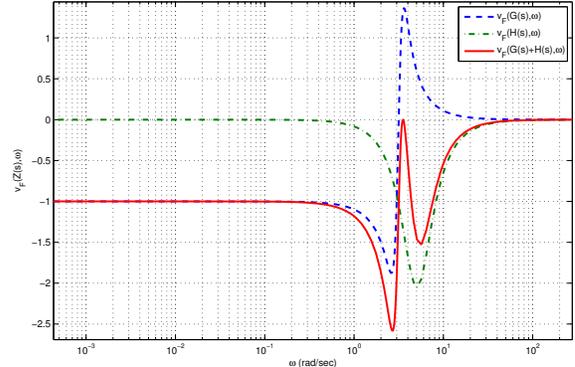


Fig. 3. Passivity indexes for $G(s)$, $H(s) = H_{\text{bp}}(s)$ and $G_{\text{robot}}(s) = (H(s) + G(s))$.

1) Example: Position Control of a Mass-Spring-Damper:

For simplicity of discussion we will focus on the SISO LTI case and consider the position $X(s)$ control of a cart of mass m which can exert a force $\tau_u(s)$ while attached to a wall by a spring with stiffness k and damper with dampening coefficient c in which $\frac{X(s)}{\tau_u(s)} = G(s) = \frac{\omega_n^2}{k(s^2 + 2\zeta\omega_n s + \omega_n^2)}$, $\omega_n = \sqrt{\frac{k}{m}}$ and $\zeta = \frac{c}{2\sqrt{km}}$. It is well known that the force to velocity mapping of this mass-spring-damper system is passive; however, due to the stiffness of the spring, even our PD-controller with integral action will be unable to achieve near-perfect tracking when trying to close the loop using only velocity feedback. Therefore in order to achieve near-perfect tracking we need to close the loop directly using position feedback. As described in Section V-B, we shall recover a strictly-output mapping by augmenting the position output with a band-pass-filtered command output such that $\dot{\Theta}(s) = X(s) + H_{\text{bp}}(s)\tau_u(s) = (G(s) + H_{\text{bp}}(s))\tau_u(s) = G_{\text{robot}}(s)\tau_u(s)$. The band-pass filter will be of the form $H_{\text{bp}}(s) = k_{\text{bp}} \frac{\omega_{\text{bp}}^2 s}{s^2 + 2\zeta_{\text{bp}} \omega_{\text{bp}} s + \omega_{\text{bp}}^2}$.

We close the loop on $\dot{\Theta}(s)$ with our digital PD-controller in order to get near-perfect tracking. For the case when $m = 1$ kg, $k = 10$ N/m, $c = 1$ (N-s)/m we have that $\zeta = .16$ and $\omega_n = 3.16$ in order to make the system strictly-output passive we choose $k_{\text{bp}} = \frac{41}{k}$, $\zeta_{\text{bp}} = 0.5$ and $\omega_{\text{bp}} = 1.6\omega_n$. The resulting passivity indexes are plotted in Fig. 3 and respective bode-plots are depicted in Fig. 4 which indicate that $G_{\text{robot}}(s)$ is passive and asymptotically stable. Solving the LMIs given in Corollary 2 we can verify that $G_{\text{robot}}(s)$ is strictly-output passive in which $\epsilon = 0.1977$. Finally we evaluate our control framework when $b = 1$, $T_s = .05$ s, $\epsilon_c = 1.0e - 5$ and $\epsilon = 0$ as the output $x(t)$ tracks the desired reference $x_r(t)$ depicted in Fig. 5. We compare our response $x(t)$ to the response of a continuous-time controller implementation when $G_c(s) = \frac{k_p + k_d s}{s}$ and $H_{\text{bp}}(s) = 0$ which is denoted $x_{no-H}(t)$. It is clear from Fig. 5 that a better performing system can be achieved when adding $H_{\text{bp}}(s)$ in parallel in order to achieve a strictly-output passive system. Although not depicted, the system response for the ideal continuous-time controller implementation with $H_{\text{bp}}(s)$ included is nearly identical to our digital controller implementation except for the

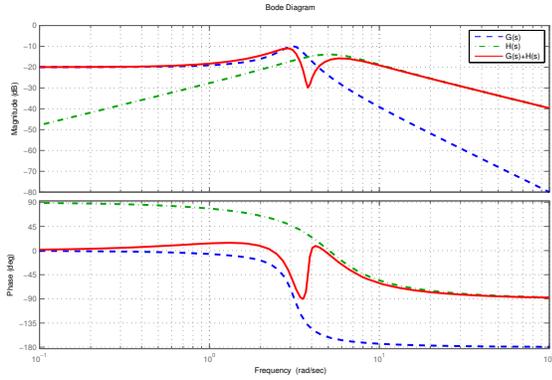


Fig. 4. Bode-plot for $G(s)$, $H(s) = H_{bp}(s)$ and $G_{robot}(s) = (H(s) + G(s))$.

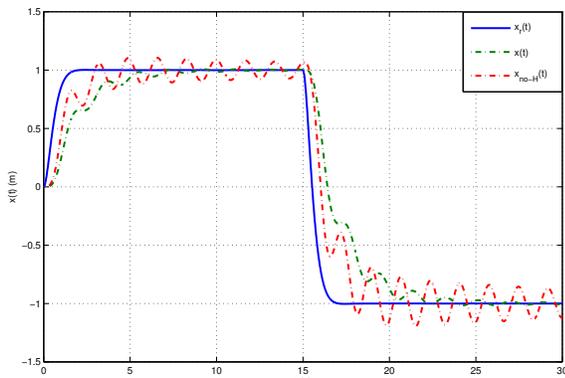


Fig. 5. System response $x(t)$ as it tracks $x_r(t)$ for mass-spring-damper system.

T_s sample delay in the output of $x(t)$. Therefore with little loss in performance, the output of a class of asymptotically stable minimum-phase systems can be augmented with low-complexity analog filters $H_*(s)$ in order to create a strictly output passive system which can be integrated in to our proposed digital control framework. L_2^m -stability is then independent of the sampling rate T_s chosen and communication delays incurred between the digital controller and the PS and PH interfaces.

VI. EXPERIMENTAL EVALUATION

This section presents experimental results for an NCS consisting of an asynchronous passive controller and a soft real-time simulated passive plant representing a robotic arm using an actual 802.11b wireless network. The controller is implemented in an asynchronous manner so that the reference input $\hat{\Theta}_{sr}[i]$ is buffered and processed as measurements from the plant $u_{pd}[i]$ arrive over the wireless network. The plant, a Simulink-based model which requires a variable time step solver can not be simulated in a hard-real-time manner. However, we are able to pace the simulation in a soft real-time manner (the simulation can be paced such that the simulation-time proceeds closely to the operating systems clock) such

that the experienced network delays correspond to delays an actual networked controlled plant would be subjected to.²

A. Robotic System Simulation

We consider the Pioneer 3 (P3) arm which is a robotic manipulator built for the P3-DX and P3-AT ActivMedia mobile robots. The P3 Arm has two main segments, the manipulator and the gripper. The manipulator has five degrees of freedom and the gripper has an additional one. Fig. 6 shows the home position of the P3 arm including the locations for the centers of gravity using the point mass assumption. The dynamic model

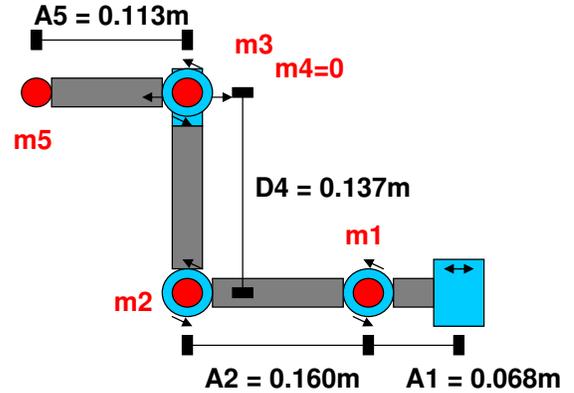


Fig. 6. Pioneer 3 Arm

of the robotic arm is described by Equation (3) and is derived using the Lagrangian approach for computing the elements of the mass matrix, Coriolis and centrifugal vector, and gravity vector [48]. The model is implemented as a Simulink block using the “Robotics Toolbox for Matlab” [49] and includes gravity compensation and velocity damping as described in Section IV.

B. Passive Control Architecture

In order to choose an appropriate set of continuous time gains k_p and k_d we focus our attention on joint 1 which is subject to the largest (changes of) inertia J as can be deduced from Fig. 6 such that $G_{pm}(s) = \frac{1}{J_s}$. Similarly we approximate the controller to be of the form $G_c(s) = \frac{k_p + k_d s}{s}$. Next using basic loop shaping techniques we desire the system to have a crossover frequency (ω_c s.t. $20 \log_{10}(|G_{pm}(j\omega_c)G_c(j\omega_c)|) = 0$ dB), in which $\omega_c = \frac{\omega_n}{N}$. $\omega_n = \frac{\pi}{T_s}$ is denoted as the Nyquist frequency. Therefore, the control gains can be computed based on a desired phase margin $0 < \phi \leq 90$ (degrees) as follows: i) $\tau = \frac{(\phi - 40)}{5\omega_c}$, ii) $k_p = \frac{J\omega_c^2}{(\tau\omega_c + 1)}$, iii) $k_d = k_p\tau$. Although the phase margin will never exceed 90 degrees, you can still calculate appropriate gains for k_p and k_d for $\phi > 90$ using the above straight line approximation. All simulations given are for $\phi = 80$ degrees, $N = 2$, and $J = .293$ kg-m². The remaining system parameters are as follows: $T_s = 0.1$ seconds, $\epsilon_c = 1.0e-6$, $\epsilon = 0.5$, $k_p = 8.02$ & $k_d = 4.1$.

²We selected to simulate the plant because robotic arms such as the Pioneer 3 are controlled using simple servos and they do not provide feedback.

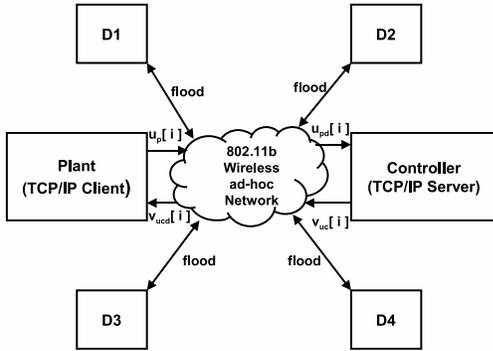


Fig. 7. Experimental Setup

C. NCS Setup

The experimental setup is shown in Fig. 7. The network is a wireless 802.11b ad hoc network with six wireless nodes. One node contains the passive controller written in C, another node contains a Simulink program which simulates the robotic arm. The controller and the plant use the TCP/IP Send and TCP/IP Receive blocks in Simulink to communicate with the controller. The remaining four nodes are used to send disturbance packets onto the network.

In order to evaluate the stability and the robustness to time-varying network delays of the proposed architecture, we record the joint angles of the arm and the round-trip delays observed at the plant. The controller produces a trajectory for the robot to follow. The first stage moves the robotic arm from the zero home position to the position of $[1 \ 0.8 \ 0.6 \ 0.4 \ 0.2 \ 0]$ rad within five seconds. For the second stage, the robot remains in place for five seconds. In the third stage the robot returns to the home position within five seconds.

During a simulation, the controller waits for a connection from the computer containing the passive robotic model. During this time some or all of the disturbance machines send ping floods to the computer containing the passive controller. When the node containing the passive plant is able to send and receive data successfully, the plant model records the packet round-trip time. Specifically, the round-trip delay (Fig. 9) corresponds to the time difference when $u_p[i_{\text{sent}}]$ is sent ($t_{\text{sent}} = i_{\text{sent}}T_s$) and when the corresponding control command arrives back to the plant in the form of a wave variable $v_{uc}[i_{\text{arrived}}]$ ($t_{\text{arrived}} = i_{\text{arrived}}T_s$), in other words $\Delta t_{\text{round trip}} = (i_{\text{arrived}} - i_{\text{sent}})T_s$.

Experiment 1: Nominal Conditions: In experiment 1, the controller and plant operate and communicate with each other without any communication from the disturbance nodes. This experiment shows how the system behaves under nominal conditions. Fig. 8 displays the joint angles of the robotic arm that follow the reference trajectory provided to the controller. The round-trip network delay, as seen in Fig. 9 is minimal and repeatable, and it has very little effect on the stability of

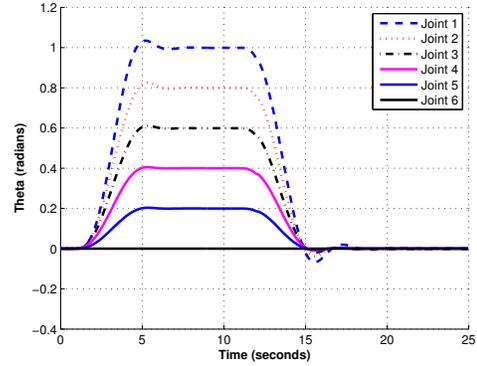


Fig. 8. Robot Performance With No Network Disturbance

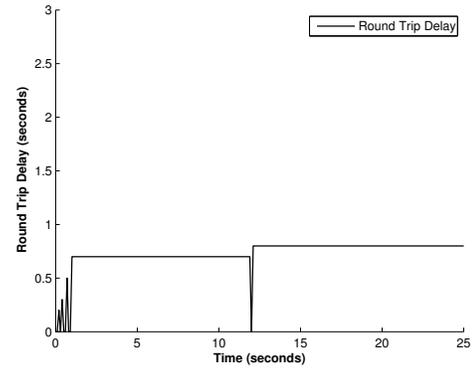


Fig. 9. Packet Round-Trip Delay With No Network Disturbance

the robot model. The delay is a product of internal processing of both the plant and the controller rather than network delay itself.

Experiment 2: Network disturbances: Figs. 10, 12, 14, and 16 show how the robotic model behaves in the face of network disturbance. During the experiment, each disturbance node outputs ping flood packets as fast as they come back or one hundred times per second, whichever is more. When one node or two nodes send out ping floods, the robot behavior is very close to the nominal case. However, when three and four disturbance nodes participate on the network, the controller computer has difficulty receiving messages from and sending messages to the plant computer. This is the case that demonstrates the advantages of the passive control architecture. When the plant is unable to communicate with the controller, the robot simply stops and waits for the next packet from the controller. This can be seen in Figs. 14 and 16. These results show that in the face of crippling network traffic, the robot remains stable.

Experiment 3: CPU Disturbances: In experiment 3, the disturbance nodes are silenced. In this experiment, the controller computer executes two programs simultaneously, the passive control program and a disturbance program. The disturbance program uses the Cygwin/Unix low-level copy program "dd" to continuously write random numbers to a file. This process takes overloads the CPU of the controller node. Both programs have the same priority, and both share the same single core

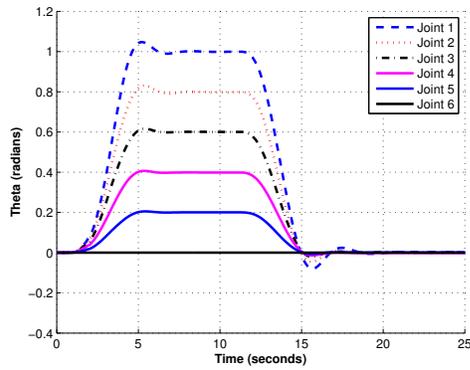


Fig. 10. Robot Performance With One Disturbance Node

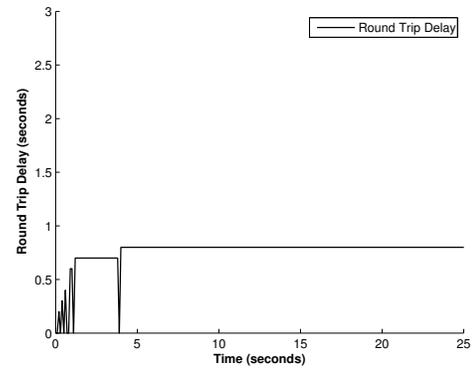


Fig. 13. Packet Round-Trip Delay With Two Disturbance Nodes

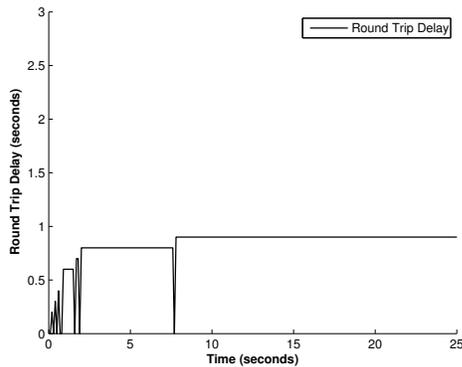


Fig. 11. Packet Round-Trip Delay With One Disturbance Node

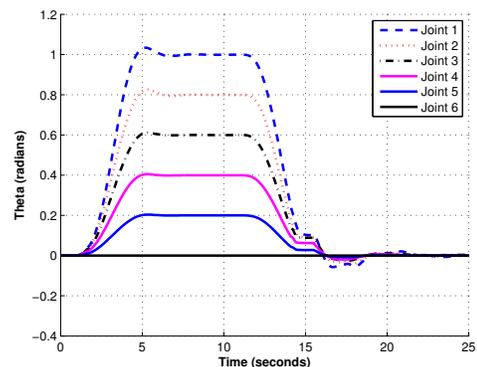


Fig. 14. Robot Performance With Three Disturbance Nodes

processor. Figs. 18 and 19 show how the robotic model behaves when the controlling computer is at 100 percent CPU load. The delay graph shows that the round trip delay is similar to the nominal case in experiment 1, and Fig. 18 also shows a similar performance to the system in experiment 1. CPU load increasing programs running on the controller computer had a negligible effect on degrading system performance.

VII. CONCLUSIONS AND FUTURE WORK

The paper presents a passive control architecture that offers advantages in building CPSs whose stability is guaranteed

independent of networking delay uncertainties if data transmitted over a given network is only processed once at the respective receiving controller or plant nodes. Thus improving orthogonality across the controller design and implementation design layers and empowering model-driven development. We have presented an architecture for a system consisting of a robotic manipulator controlled by a digital controller over a wireless network and we have proved the networked control system to be stable. Finally, we have evaluated the system using experimental results validating the significant advantages of the passivity-based architecture especially in the presence of

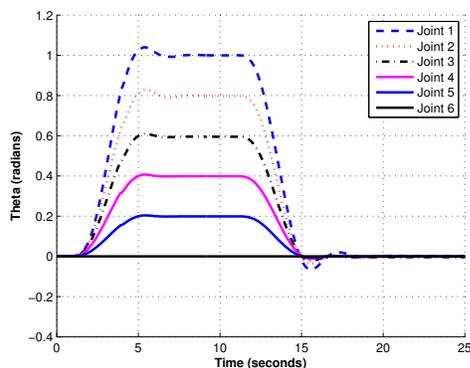


Fig. 12. Robot Performance With Two Disturbance Nodes

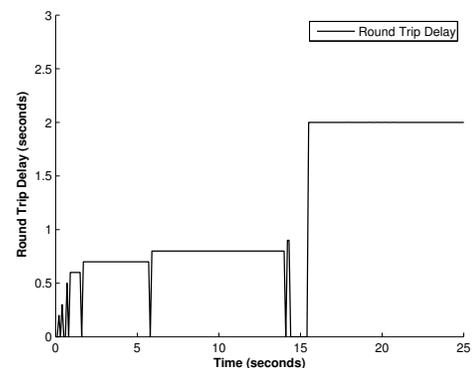


Fig. 15. Packet Round-Trip Delay With Three Disturbance Nodes

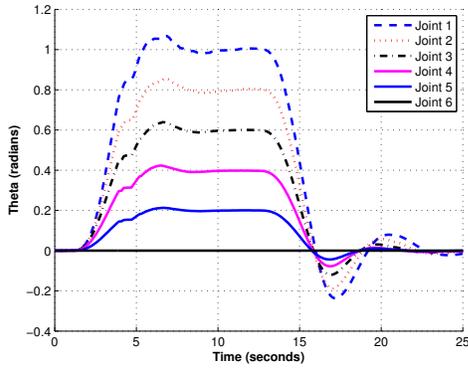


Fig. 16. Robot Performance With Four Disturbance Nodes

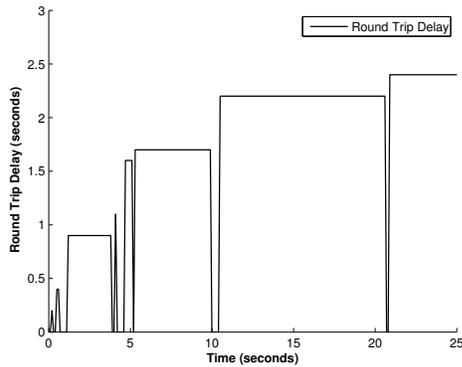


Fig. 17. Packet Round-Trip Delay With Four Disturbance Nodes

time-varying delays. Our current and future work focuses on methods that provide an effective way to interconnect multiple passive systems and controllers as well as an integrated end-to-end tool chain for the model-based design of CPSs based on passivity.

REFERENCES

- [1] J. Baillieul and P. Antsaklis, "Control and communication challenges in networked control systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 9 – 28, 2007.
- [2] I. T. MathWorks, "Matlab," *The Language of Technical Computing, Version 7.6*, 2008.

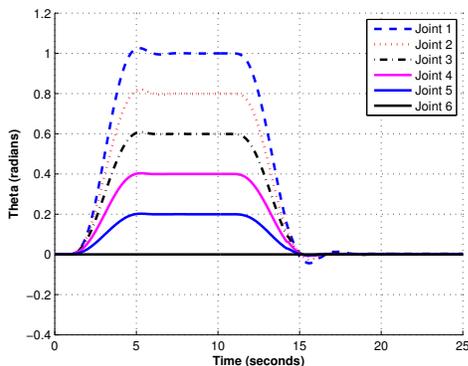


Fig. 18. Robot Performance With 100 Percent CPU Load

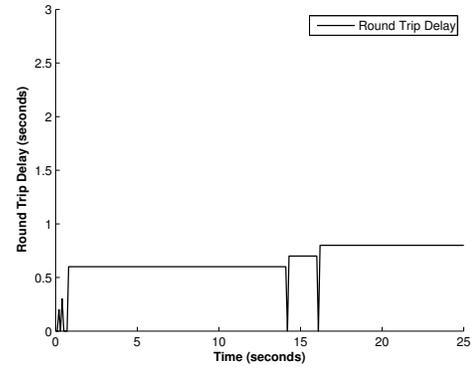


Fig. 19. Packet Round-Trip Delay With 100 Percent CPU Load

- [3] —, "Simulink," *Dynamic System Simulation for MATLAB, Version 7.1*, 2008.
- [4] A. L. Sangiovanni-Vincentelli, "Quo vadis sld: Reasoning about trends and challenges of system-level design," *Proceedings of the IEEE*, vol. 95, no. 3, pp. 467 – 506, 2007.
- [5] A. C. M. Ohlin, D. Henriksson, "TrueTime 1.5 - reference manual," Department of Automatic Control, Lund University, Tech. Rep., 2007.
- [6] H. Kopetz, "The time-triggered architecture," *Proceedings of the IEEE*, vol. 91, no. 1, pp. 92 – 126, 2003.
- [7] J. Skaf and S. Boyd, "Analysis and synthesis of state-feedback controllers with timing jitter," *IEEE Transactions on Automatic Control*, to appear.
- [8] C. Desoer and M. Vidyasagar, *Feedback Systems: Input-Output Properties*. Academic Press, Inc., 1975.
- [9] P. Li and R. Horowitz, "Control of smart machines, part 1: Problem formulation and non-adaptive control," *IEEE/ASME Trans. Mechatron.*, vol. 2, no. 4, pp. 237–247, 1997.
- [10] G. Niemeyer and J.-J. E. Slotine, "Towards force-reflecting teleoperation over the internet," in *IEEE International Conference on Robotics and Automation*, 1998, pp. 1909 – 1915.
- [11] P. Beresteky, N. Chopra, and M. W. Spong, "Discrete time passivity in bilateral teleoperation over the internet," in *IEEE International Conference on Robotics and Automation*, 2004, pp. 4557 – 4564.
- [12] D. Lee and P. Li, "Passive coordination control of nonlinear mechanical teleoperator," *IEEE Trans. Rob. Autom.*, vol. 21, no. 5, pp. 935–951, 2005.
- [13] S. Stramigioli, C. Secchi, A. J. van der Schaft, and C. Fantuzzi, "Sampled data systems passivity and discrete port-hamiltonian systems," *IEEE Transactions on Robotics*, vol. 21, no. 4, pp. 574 – 587, 2005. [Online]. Available: <http://dx.doi.org/10.1109/TRO.2004.842330>
- [14] C. Secchi, S. Stramigioli, and C. Fantuzzi, *Control of interactive robotic interfaces: A port-Hamiltonian approach*. Springer Verlag, 2007.
- [15] S. Hirche and M. Buss, "Transparent Data Reduction in Networked Telepresence and Teleaction Systems. Part II: Time-Delayed Communication," *Presence: Teleoperators and Virtual Environments*, vol. 16, no. 5, pp. 532–542, 2007.
- [16] A. Fettweis, "Wave digital filters: theory and practice," *Proceedings of the IEEE*, vol. 74, no. 2, pp. 270 – 327, 1986.
- [17] N. Kottenstette and P. J. Antsaklis, "Stable digital control networks for continuous passive plants subject to delays and data dropouts," in *Proceedings of the 46th IEEE Conference on Decision and Control*, 2007, pp. 4433 – 4440.
- [18] N. Chopra, P. Beresteky, and M. Spong, "Bilateral teleoperation over unreliable communication networks," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 2, pp. 304–313, 2008.
- [19] S. Hirche, T. Matiakis, and M. Buss, "A distributed controller approach for delay-independent stability of networked control systems," *Automatica*, vol. 45, no. 8, pp. 1828–1836, 2009.
- [20] J. Bao, P. Lee, F. Wang, W. Zhou, and Y. Samyudia, "A new approach to decentralised process control using passivity and sector stability conditions," *Chemical Engineering Communications*, vol. 182, no. 1, pp. 213–237, 2000.
- [21] B. Ydstie, "Passivity based control via the second law," *Computers & Chemical Engineering*, vol. 26, no. 7-8, pp. 1037–1048, 2002.
- [22] K. Jillson and B. Erik Ydstie, "Process networks with decentralized

- inventory and flow control," *Journal of Process Control*, vol. 17, no. 5, pp. 399–413, 2007.
- [23] N. Kottenstette, X. Koutsoukos, J. Hall, P. Antsaklis, and J. Sztinapovits, "Passivity-based design of wireless networked control systems for robustness to time-varying delays," in *29th IEEE Real-Time Systems Symposium (RTSS 2008)*, 2008.
- [24] G. Zames, "On the input-output stability of time-varying nonlinear feedback systems. i. conditions derived using concepts of loop gain, conicity and positivity," *IEEE Transactions on Automatic Control*, vol. AC-11, no. 2, pp. 228 – 238, 1966.
- [25] J. C. Willems, "Dissipative dynamical systems - part i: General theory," *Soviet Journal of Optical Technology (English translation of Optiko-Mekhanicheskaya Promyshlennost)*, vol. 45, no. 5, pp. 321 – 351, 1972.
- [26] D. J. Hill and P. J. Moylan, "Stability results for nonlinear feedback systems," *Automatica*, vol. 13, pp. 377–382, 1977.
- [27] R. J. Anderson and M. W. Spong, "Bilateral control of teleoperators with time delay," *Proceedings of the IEEE Conference on Decision and Control Including The Symposium on Adaptive Processes*, pp. 167 – 173, 1988. [Online]. Available: <http://dx.doi.org/10.1109/CDC.1988.194290>
- [28] G. Niemeyer and J.-J. E. Slotine, "Telemanipulation with time delays," *International Journal of Robotics Research*, vol. 23, no. 9, pp. 873 – 890, 2004. [Online]. Available: <http://dx.doi.org/10.1177/0278364904045563>
- [29] M. Kuschel, P. Kremer, and M. Buss, "Passive haptic data-compression methods with perceptual coding for bilateral presence systems," *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICSPART A: SYSTEMS AND HUMANS*, vol. 39, no. 6, 2009.
- [30] C. Secchi, S. Stramigioli, and C. Fantuzzi, "Digital passive geometric telemanipulation," in *IEEE International Conference on Robotics and Automation*, 2003, pp. 3290 – 3295.
- [31] W. M. Haddad and V. S. Chellaboina, *Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach*. Princeton University Press, 2008.
- [32] N. Chopra, M. W. Spong, S. Hirche, and M. Buss, "Bilateral teleoperation over the internet: the time varying delay problem," *Proceedings of the American Control Conference*, vol. 1, pp. 155 – 160, 6 2003.
- [33] N. Kottenstette, H. LeBlanc, E. Eyisi, and X. Koutsoukos, "Multi-rate networked control of conic systems," Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN, Tech. Rep., 09/2009 2009. [Online]. Available: http://www.isis.vanderbilt.edu/sites/default/files/tr_quantization_revised_4_2010.pdf
- [34] A. van der Schaft, *L2-Gain and Passivity in Nonlinear Control*. Se-caucus, NJ, USA: Springer-Verlag New York, Inc., 1999.
- [35] R. Ortega and M. Spong, "Adaptive motion control of rigid robots: A tutorial," in *27th IEEE Conference on Decision and Control*, 1988, pp. 1575 – 84.
- [36] W. S. Levine, *Control System Applications*. CRC Press, 2000.
- [37] N. Kottenstette, J. Hall, X. Koutsoukos, P. J. Antsaklis, and J. Sztinapovits, "Digital control of multiple discrete passive plants over networks," *International Journal of Systems, Control and Communications (IJSCC)*, no. Special Issue on Progress in Networked Control Systems, 2009, to Appear. [Online]. Available: <http://www.isis.vanderbilt.edu/node/4050>
- [38] N. Kottenstette and P. J. Antsaklis, "Digital control networks for continuous passive plants which maintain stability using cooperative schedulers," *University of Notre Dame, Technical Report isis-2007-002*, pp. 1–16, 2007. [Online]. Available: <http://www.nd.edu/%7Eisis/techreports/isis-2007-002-updated.pdf>
- [39] J.-H. Ryu, Y. S. Kim, and B. Hannaford, "Sampled- and continuous-time passivity and stability of virtual environments," *IEEE Transactions on Robotics*, vol. 20, no. 4, pp. 772 – 6, 2004.
- [40] R. Costa-Castello and E. Fossas, "On preserving passivity in sampled-data linear systems," *2006 American Control Conference (IEEE Cat. No. 06CH37776C)*, pp. 1–6, 2006.
- [41] —, "On preserving passivity in sampled-data linear systems by using state observers," *Proceedings of the European Control Conference*, pp. 5276 – 5281, 2007.
- [42] S. Monaco, D. Normand-Cyrot, and F. Triefensee, "From passivity under sampling to a new discrete-time passivity concept," *47th IEEE Conference on Decision and Control*, pp. 3157–3162, 2008.
- [43] N. Kottenstette and P. J. Antsaklis, "Wireless digital control of continuous passive plants over token ring networks," *International Journal of Robust and Nonlinear Control*, 2008, to appear (accepted September 2008).
- [44] S. Johannessen, "Time synchronization in a local area network," *IEEE Control Systems Magazine*, vol. 24, no. 2, pp. 61–69, 2004.
- [45] H. Kopetz, A. Ademaj, P. Grillinger, and K. Steinhammer, "The time-triggered ethernet (tte) design," *Object-Oriented Real-Time Distributed Computing, IEEE International Symposium on*, vol. 0, pp. 22–33, 2005.
- [46] B. Sundararaman, U. Buy, and A. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 3, no. 3, pp. 281–323, 2005.
- [47] N. Kottenstette and P. Antsaklis, "Relationships between positive real, passive dissipative, & positive systems," *American Control Conference*, pp. 409–416, 2010.
- [48] J. J. Craig, *Introduction to Robotics: Mechanics and Control*. Addison-Wesley, 1989.
- [49] P. I. Corke, "Robotic toolbox for matlab, release 7.1," CSIRO, Tech. Rep., 2002.