

Article

Development of Discrete-Time Waterjet Control Systems Used in Surface Vehicle Thrust Vectoring

Eleftherios K. Loghis ¹ and Nikolaos I. Xiros ^{2,*}¹ School of Electrical and Computer Engineering, National Technical University of Athens, 157 80 Athens, Greece² B. Bollinger School of Naval Architecture and Marine Engineering, The University of New Orleans, New Orleans, LA 70148, USA

* Correspondence: nxiros@uno.edu

Abstract: To create an autonomous surface vehicle, the microcontroller that will be responsible for the vessel's response will have access to both engine thrust controls and steering controls. In the case of a vehicle with waterjet propulsion, the term "controls" refers to engine RPM and nozzle angle. The latter allows the thrust vectoring. Having performed the mathematical modelling of the surface vessel, a method must be found such that the microcontroller actions on the controls create the desired outputs, based on predefined performance objectives. A commonly selected control algorithm considered an industry standard for control applications—at least from the perspective of classical control methods—is the PID control scheme. Many times, a PID control scheme is sufficient for achieving the required performance. There are also many methods that extend the functionality of the PID to robust control schemes, to accomplish more specific targets, such as noise rejection. The vast majority of all these implementations, will have to be implemented using a digital microcontroller or an FPGA. This paper assists theoretically in implementing PID controllers digitally, by elaborating on the discrete-time perspective of PID control. It also provides a test method for evaluating the algorithmic implementation.



Citation: Loghis, E.K.; Xiros, N.I. Development of Discrete-Time Waterjet Control Systems Used in Surface Vehicle Thrust Vectoring. *J. Mar. Sci. Eng.* **2022**, *10*, 1844. <https://doi.org/10.3390/jmse10121844>

Academic Editor: Rosemary Norman

Received: 22 October 2022

Accepted: 23 November 2022

Published: 1 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: PID; discrete-time; digital control

1. Introduction

This study mainly aims to assist in digital implementation of PID controllers, with references to the case the authors needed it for, i.e., for a surface vessel autopilot—course controller. The implementation method described is supported by a simple testbed, which serves as an evaluation tool for the verification of each user's implementation. The mathematical tools needed for this purpose are presented in a step-by-step approach, where each step uses the previous as a background. It must be noted that the implementation presented does not intend to be optimal in any way, meaning that it does not guarantee that provides the most efficient way of digital resources usage. If reduction in memory cells or multipliers is needed (for use in FPGA for example), you may have to refer to Ogata and Kuo [1,2].

As an initiative for the development of this work, has been the need for controlling our ship model velocity and heading, by producing the correct stimulus to our waterjet propulsion. Due to the nature of the waterjet, our target is translated to controlling the magnitude and angle of the force the waterjets create at the stern of the vessel.

Quickly introducing the type of waterjets used, refer to Figure 1a which is a side-cut of a similar waterjet [3]. A PMDC motor attached to the right end of the shaft, turns the pump (No2 at Figure 1a), that moves water from the intake (No1 at Figure 1a) to the output of the nozzle (No4 at Figure 1a). The water mass displacement creates momentum variation, which in turn means force (vector F in Figure 1c,d). Figure 1b is a graphical representation

of the waterjet function, where the velocities of water masses are shown. The force created, pushes the vessel stern as shown schematically in Figure 1c,d.

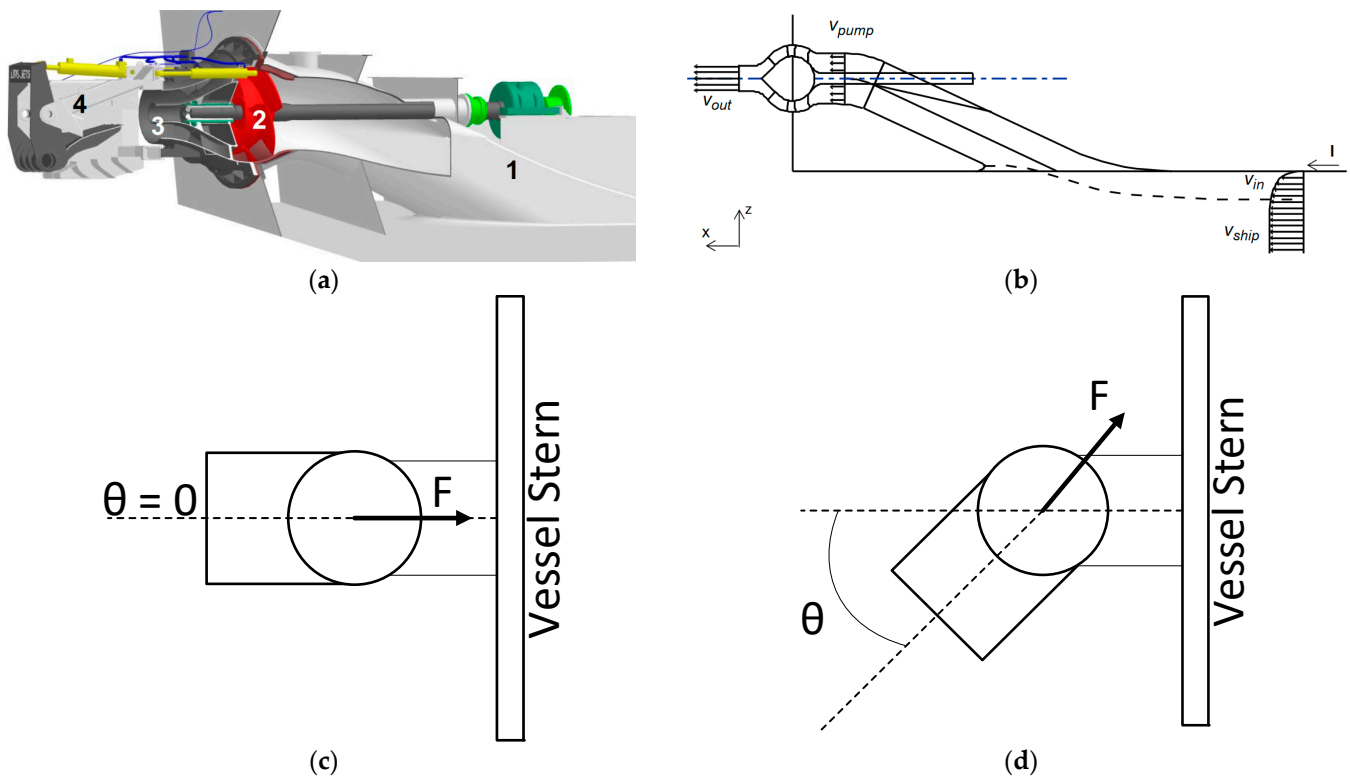


Figure 1. Waterjet schematic views. (a) Side-cut view; (b) functionality schematic; (c) top schematic view at zero turn; (d) top schematic view at θ turn.

What Figure 1d adds as information, is the use of the ball-joint at the attachment point of the nozzle to the main body of the waterjet. By changing the angle θ of the nozzle, the force applied to the vessel changes direction creating turning movement. The interesting part regarding waterjets is that the force from water displacement creates force even at zero initial velocity, thus maneuvering ability is present in all possible velocities range.

2. Materials and Methods

Focusing again on the propulsion system, we used a pair of joined waterjets to create the thrust for the vessel movement. The commands for the propulsion system are generated by two control loops shown in Figure 2, and will be discussed later in this section. A view of the implementation can be seen at Figure 3c,d.

As mentioned, both waterjets were joined. The join was at both turn angle and motor rpm as can be seen schematically at Figure 4. The “trained” readers in RC models will also recognize the nozzle turning joint at Figure 3d numbered item 5. These two types of join creates the need for two control loops for the vessel: one for velocity control and one for heading control. A higher application layer decides on the velocity and heading setpoints, driven by guidance goals that, for example, stem from maintaining a formation in a swarm. For these setpoints the control loops produce the appropriate actions, based on a digitally implemented PID algorithm.

To get an insight on these loops, Figure 2 presents a high-level approach of each one. Figure 2a describes the loop for velocity control. The velocity setpoint is selected by the positioning algorithm. The setpoint is compared to the current vessel velocity, producing an error output. The error output is fed to PID controller, which produces a control signal for the motor drive. The motor drive induces a current by the control signal, which in turn changes the motor torque. The motor torque impacts the waterjet impeller rpm, changing

the waterjet thrust. The thrust applied to the boat produces a new velocity which is then fed back and the loop closes.

The heading control loop shown in Figure 2b functions in a similar way by controlling the waterjet nozzle, which creates a yaw moment that applied to the boat creates a new heading. The heading is fed back closing the heading control loop.

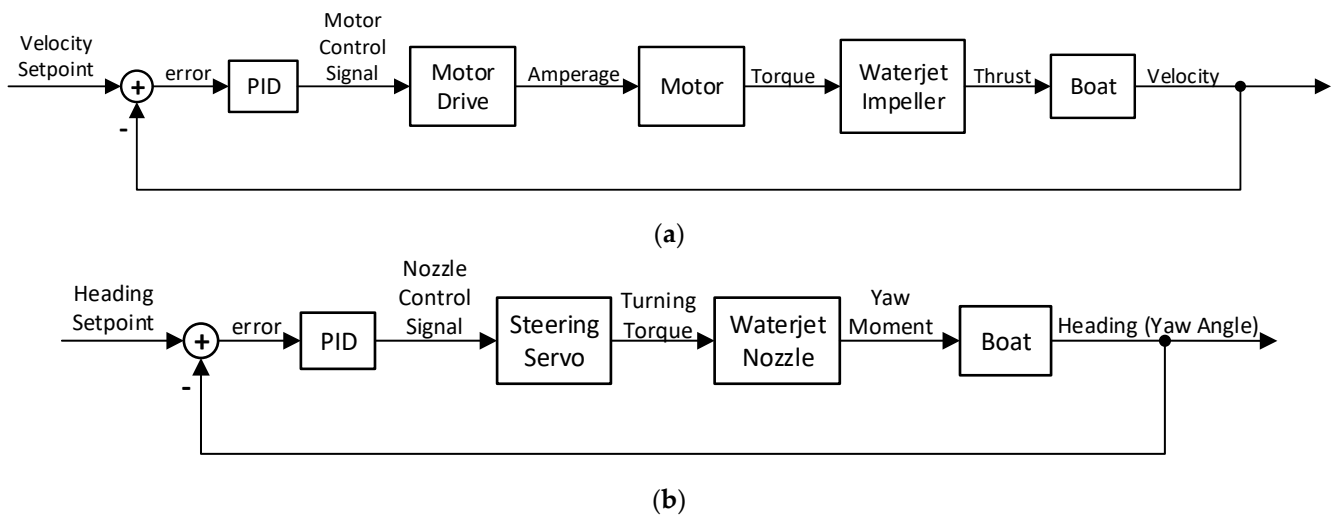
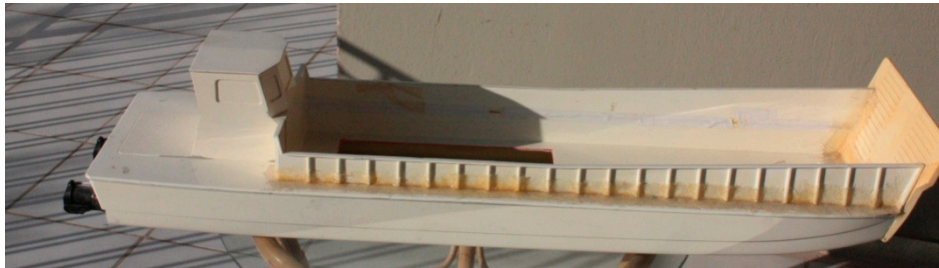
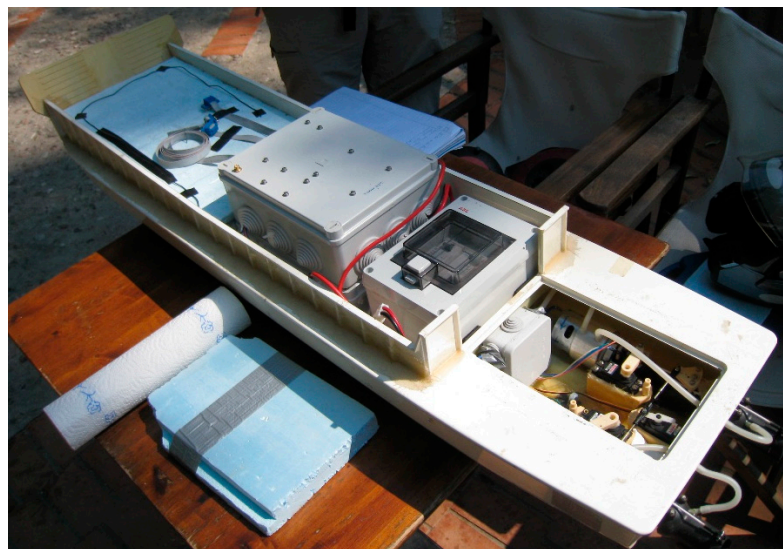


Figure 2. Control loops formed for the vessel. (a) Loop for velocity control; (b) Loop for heading control.



(a)



(b)

Figure 3. Cont.

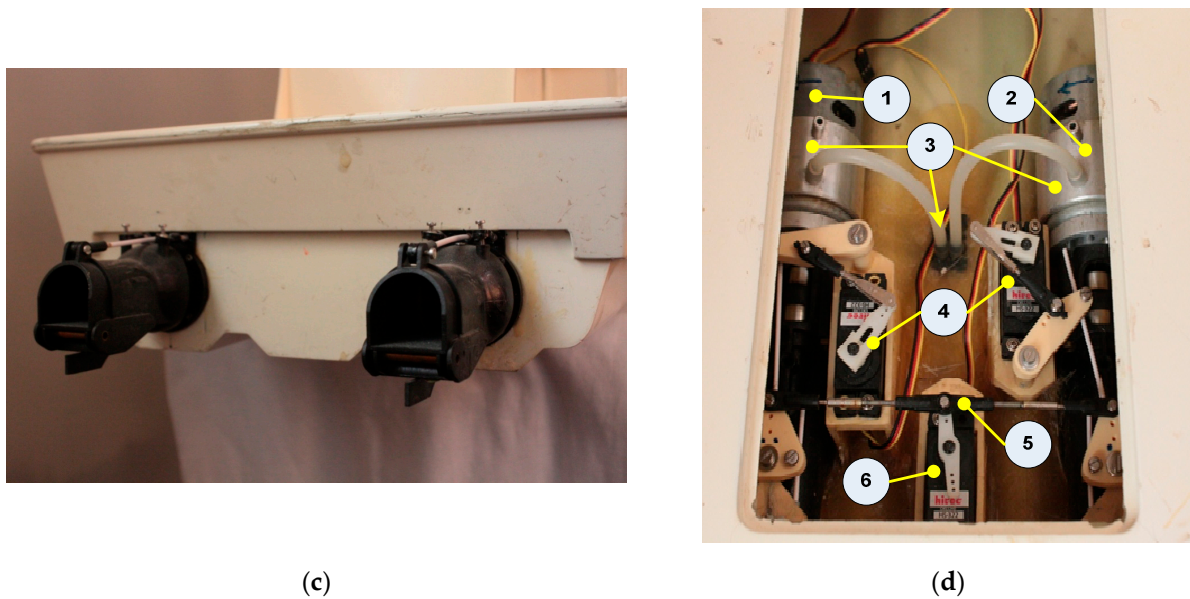


Figure 3. Development platform. (a) Initial phase of construction; (b) Vessel ready for measurements; (c) Waterjet nozzles; (d) Propulsion motors and control servos.

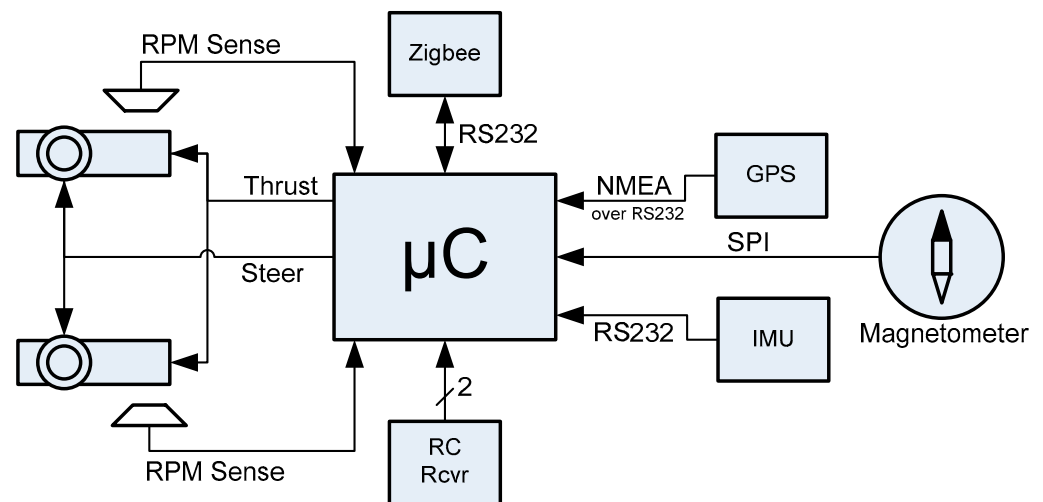


Figure 4. Control system structure.

From the mathematical toolset point of view, the work is largely based on Z-Transform, difference equations and Laplace transform.

All algorithms were built for our development platform, for which we will devote the next section.

The watercraft propulsion powertrain for which the discrete-time controller architectures are proposed in this text is intended for a small boat modeled after landing craft.

Details of the powertrain configuration, consisting of a waterjet propulsor driven by a Permanent-Magnet Direct Motor with Pulse Width Modulation control, are given in Appendix A “Electric Propulsion System for Unmanned Craft”. The boat is intended to complement a low-cost unmanned heterogeneous swarm capable of operating in the air, on sea surface and underwater. The landing craft architecture allows for a unit offering extra deck space that can be used for payload or launching and retrieving rotorcraft or smaller amphibious or ground vehicles, extra energy storage (e.g., fuel tanks, batteries), etc. The unit is required to operate autonomously and this is why effective operational

controllers for all the unit subsystems (of which propulsion is one of the most important ones) were developed.

3. Development Platform

As a development platform for the ideas described, an RC model of a flat hull, ferry boat style vehicle was used, for which more information can be found at Xiros and Loghis [4,5]. The vessel is equipped with two waterjets for propulsion. Figure 3a shows the vessel at the initial steps of its construction. Figure 3b shows the vessel fully equipped with electronics, ready for measurements and control applications.

Figure 3c,d depict details on the propulsion system. Figure 3a focuses on the waterjet nozzles. The ball-shaped end of the nozzles allows changing thrust direction, allowing heading control even at zero translational velocity.

Figure 3b gives an overview of the propulsion motors and control servos. Referring to numbered items:

1. Electric motors. These motors provide the thrust for waterjet propellers.
2. External cooling ring that encapsulates the motors. As the vessel surge velocity increases, an amount of water is forced into the silicone tubes shown by number (3). The water circulates at the outer ring cooling the motors and is rejected from another silicone tube not yet attached at the time this photo was taken.
3. Silicone tubes and input of water for motor cooling.
4. Servos controlling the backward fin that changes the direction of propulsion (forward—backward movement).
5. Shaft that joins the two waterjets for simultaneous turning (left right).
6. The servo motor that is used for turning the waterjets.

All these elements along with appropriate sensors as RPM sensor, magnetometer, GPS and IMU, form the development platform for our algorithms.

Figure 4 provides a block diagram of system interconnections from the electronics perspective. It is apparent that the microcontroller (marked as μC) is the heart of the system since every peripheral refers to it.

4. Introductory Concepts

Before beginning the analysis, it must be noted that digital systems operate in discrete time instants, that are defined by the clock used for the device under consideration timing [1]. The clocks that are important for the mathematical analysis are the clocks of data converters present in the system (analog-to-digital signal converters or ADCs and digital-to-analog signal converters or DACs). These clocks are considered the same regarding frequency and phase for this work. The frequency of these clocks will be denoted as f_s but most importantly, the period of these clocks using the symbol T_s .

Another important notion is the way used to approximate continuous-time differentiation and integration. For the derivative the first-order backward differentiation formula is going to be used [2,6]. For integral approximation a summation approximation is utilized, that will be analyzed in the following section [2].

5. Summation Approximation to Integral

One way of approximating integrals is by using pure summation [2]. This is very much like the approach used by the zero-order hold (ZoH) as we will see in the next sections.

Supposing that we have a function $f(t)$ that describes the values of a signal as a function of time. To approximate the integral of $f(t)$ by summation, the method depicted in Figure 5 is followed. As a reminder, with T_s the sampling period is implied.

As can be seen in Figure 5, the integral of the arbitrary signal is approximated by the area of parallelograms with height equal to the previous instant sample and width equal to T_s . Such parallelograms give an area of $f(k) \times T_s$ for an arbitrary $k = 0, 1, 2, \dots$

For time instant T_s :

$$\int_0^{T_s} f(t)dt \approx f(0) \cdot T_s \quad (1)$$

For time instant $2T_s$:

$$\int_0^{2T_s} f(t)dt \approx f(0) \cdot T_s + f(1) \cdot T_s \quad (2)$$

For time instant nT_s :

$$\int_0^{nT_s} f(t)dt \approx f(0) \cdot T_s + f(1) \cdot T_s + \dots + f(n-1) \cdot T_s \quad (3)$$

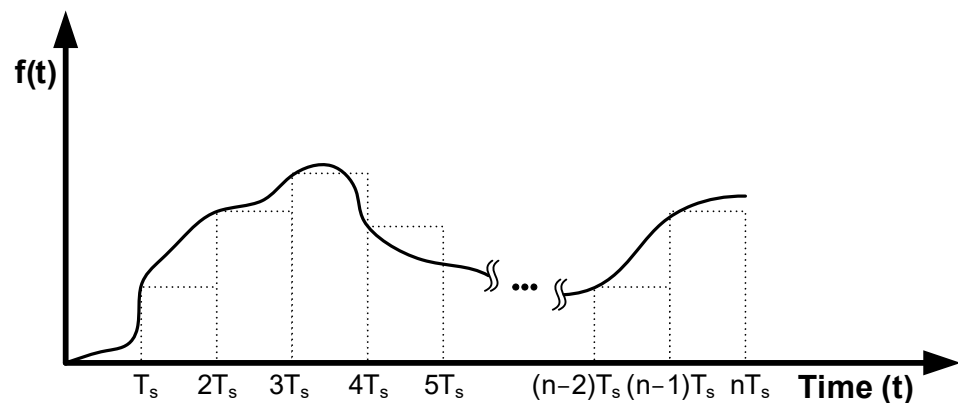


Figure 5. Approximating an integral by summation.

If we use $y[nT_s]$ as the symbol for the approximation of an integral at nT_s time instant, then from the above equations we have:

$$\int_0^{nT_s} f(t)dt \approx y[nT_s] = f(0) \cdot T_s + f(1) \cdot T_s + \dots + f(n-1) \cdot T_s \Rightarrow \quad (4)$$

$$y[nT_s] = \sum_{k=1}^n f[(k-1)T_s] \cdot T_s \quad (5)$$

Since all the signals we are dealing with are causal, their negative time instant values are zero; for this reason (since $f[(-1)T_s] = 0$) Equation (5) becomes:

$$y[nT_s] = \sum_{k=0}^n f[(k-1)T_s] \cdot T_s \quad (6)$$

From Equation (6) we also notice that:

$$y[T_s] = f[0 \cdot T_s] \cdot T_s = f(0) \cdot T_s \quad (7)$$

$$y[2T_s] = f(0) \cdot T + f(T_s) \cdot T_s = y[T_s] + f(T_s) \cdot T_s \quad (8)$$

which means that for the general case

$$y[nT_s] = y[(n-1)T_s] + f[(n-1)T_s] \cdot T_s \quad (9)$$

By taking the Z-transform of Equation (9) we have that

$$Y(z) = z^{-1}Y(z) + z^{-1}T_sF(z) \Leftrightarrow \quad (10)$$

$$Y(z) = \frac{z^{-1}T_s}{1 - z^{-1}}F(z) \quad (11)$$

If we want to depict Equation (11) in the form of a block diagram, then we get

A block diagram that is more efficient regarding memory element usage (very critical for FPGA implementations) can be derived if we simplify the above diagram by block diagram manipulation or by considering that:

$$Y(z) = z^{-1}Y(z) + z^{-1}T_sF(z) = z^{-1}[Y(z) + T_sF(z)] \quad (12)$$

Which means, that the diagram of Figure 6 can be considered equivalent to

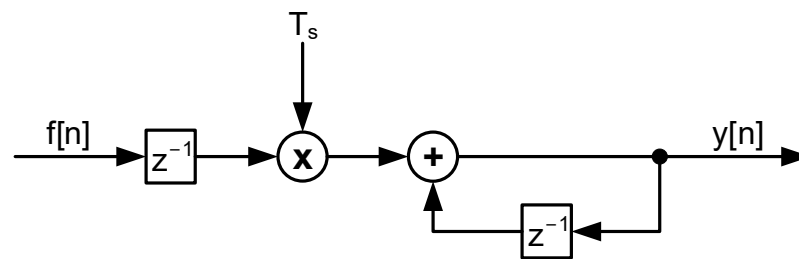


Figure 6. Block diagram form of integration approximation.

Figure 7 improves the realization of Figure 6 by reducing one memory element.

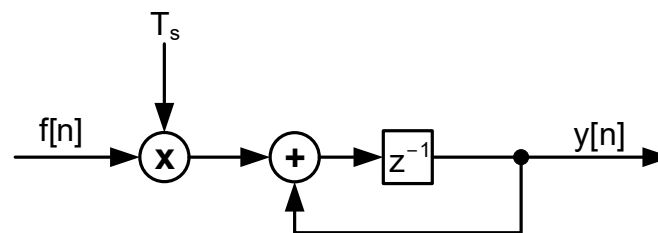


Figure 7. Improved block diagram of integration approximation.

6. Digitizing the Analog PID Functionality

Figure 8 shows the structure of an analog PID controller [7–10]. To convert the analog PID controller to digital [11], we examine each term separately, something that is correct, since linear systems have the property of superposition.

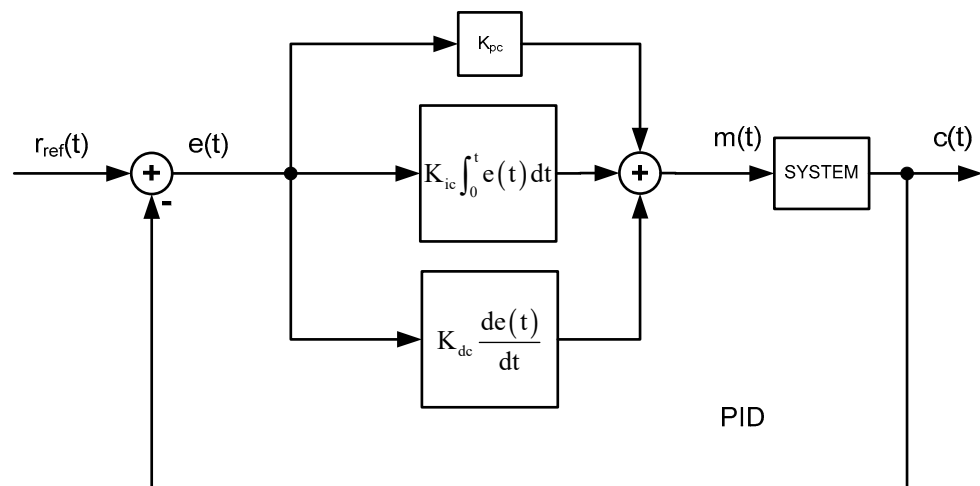


Figure 8. Analog PID.

6.1. P-Term

The P-term is the simplest of all. It amplifies any incoming signal by a gain of K_{pc} , where the c stands for “continuous”, implying the continuous system gain; this was done to avoid mixing these constants with K_{pc} that is derived for digital controllers after manipulations. The c-marking is followed for all PID term gains.

The input to digital PID P-term is the sampled error signal $e[nT_s]$. The output of P-term is:

$$m_p[nT_s] = K_{pc}e[nT_s] \quad (13)$$

or in terms of Z-transform

$$M_p(z) = K_{pc}E(z) \quad (14)$$

The block diagram of P-term follows at Figure 9.

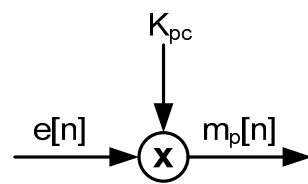


Figure 9. Digital P-term block diagram.

As a note for Figure 9, also applicable to the rest of the document for discrete-time signals, is that by $e[n]$ we imply $e[nT_s]$ and we will use both notations interchangeably throughout this text.

6.2. I-Term

For the I-term we use the approximation analyzed in section “Summation Approximation to Integral”. This results in the following equations

$$K_{ic} \int_0^t e(\tau) d\tau = K_{ic} \int_0^{nT_s} e(\tau) d\tau \approx K_{ic} \sum_{k=0}^n e[(k-1)T_s] \cdot T_s \quad (15)$$

$$m_i[nT_s] = K_{ic} T_s \sum_{k=0}^n e[(k-1)T_s] \quad (16)$$

or in terms of Z-Transform

$$Z\{m_i[nT_s]\} = K_{ic} T_s Z \left\{ \sum_{k=0}^n e[(k-1)T_s] \right\} \Rightarrow \quad (17)$$

$$M_i(z) = K_{ic} \frac{z^{-1} T_s}{1 - z^{-1}} E(z) \quad (18)$$

For the block diagram of I-term we can manipulate Equation (18) to get

$$M_i(z) = z^{-1} [M_i(z) + K_{ic} T_s E(z)] \quad (19)$$

which is implemented as a block diagram in Figure 10.

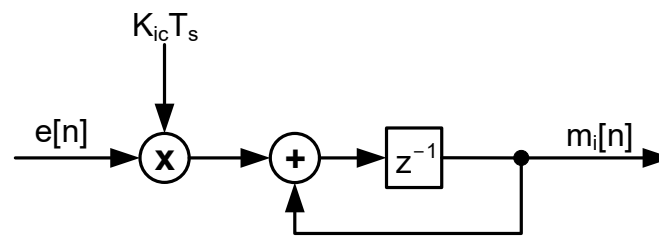


Figure 10. Digital I-term block diagram.

6.3. D-Term

For the D-term we approximate the derivative by differences as follows:

$$m_d(t) = K_{dc} \frac{de(t)}{dt} \approx K_{dc} \frac{\Delta e(t)}{\Delta t} \Rightarrow \quad (20)$$

$$m_d[nT_s] = K_{dc} \frac{e[nT_s] - e[(n-1)T_s]}{T_s} \quad (21)$$

or in terms of Z-transform

$$M_d(z) = \frac{K_{dc}}{T_s} (1 - z^{-1}) E(z) \quad (22)$$

The block diagram of Equation (22) is shown in Figure 11.

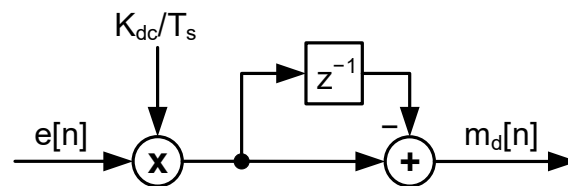


Figure 11. Digital D-term block diagram.

6.4. Combining All Terms

After analyzing each term separately, we can now provide the equation for the total output of digital PID controller:

$$m[nT_s] = m_p[nT_s] + m_i[nT_s] + m_d[nT_s] \Rightarrow \quad (23)$$

$$m[nT_s] = K_{pc}e[nT_s] + K_{ic}T_s \sum_{k=0}^n e[(k-1)T_s] + K_{dc} \frac{e[nT_s] - e[(n-1)T_s]}{T_s} \quad (24)$$

or in terms of Z-transform:

$$M(z) = M_p(z) + M_i(z) + M_d(z) \Rightarrow \quad (25)$$

$$M(z) = K_{pc}E(z) + K_{ic} \frac{z^{-1}T_s}{1 - z^{-1}} E(z) + \frac{K_{dc}}{T_s} (1 - z^{-1}) E(z) \Leftrightarrow \quad (26)$$

$$M(z) = \left[K_{pc} + K_{ic} \frac{z^{-1}T_s}{1 - z^{-1}} + \frac{K_{dc}}{T_s} (1 - z^{-1}) \right] E(z) \quad (27)$$

What should be pointed out is that the digitization of the analog PID introduces T_s as a multiplier in I and D terms. This reveals the utmost importance of sampling frequency selection, because improper selection of T_s can lead to unstable digital implementation of an otherwise stable analog PID controlled system. As a last note to this paragraph, Equation (27) is named as “Positional Form” of PID controller. There is also the “Velocity Form” [1] that will be shown later.

7. Block Diagrams for Digital PID

Equation (27) can also be shown as a block diagram after some mathematical manipulations. We eliminate the $(1 - z^{-1})$ factor from the denominator, which gives

$$(1 - z^{-1})M(z) = \left[(1 - z^{-1})K_{pc} + K_{ic}T_s z^{-1} + \frac{K_{dc}}{T_s} (1 - z^{-1})^2 \right] E(z) \Leftrightarrow \quad (28)$$

$$M(z) = z^{-1}M(z) + \left[\left(K_{pc} + \frac{K_{dc}}{T_s} \right) + \left(K_{ic}T_s - K_{pc} - \frac{2K_{dc}}{T_s} \right) z^{-1} + \frac{K_{dc}}{T_s} z^{-2} \right] E(z) \quad (29)$$

Moving back to difference equations using inverse Z-Transform we have:

$$\begin{aligned} m[nT_s] = & m[(n-1)T_s] + \left(K_{pc} + \frac{K_{dc}}{T_s} \right) e[nT_s] + \\ & + \left(K_{ic}T_s - K_{pc} - \frac{2K_{dc}}{T_s} \right) e[(n-1)T_s] + \\ & + \frac{K_{dc}}{T_s} e[(n-2)T_s] \end{aligned} \quad (30)$$

Equation (29) or Equation (30) give the block diagram of Figure 12.

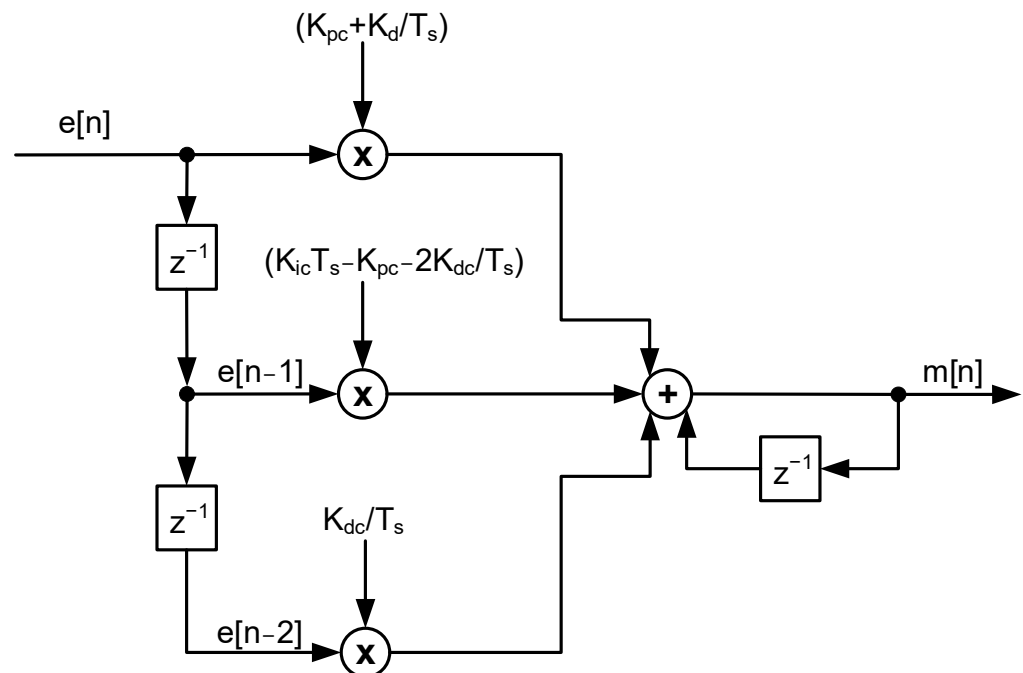


Figure 12. Block diagram for digital PID implementation.

8. Alternative PID Implementation (Velocity Form)

There is also an alternative implementation of Digital PID, which focuses on reducing the impact of sudden changes at input. The mathematical trick that implements the above statement will be shown at the derivation of the equations that follows.

This form can also be used in our application, and relevant results will be given.

To derive the equations for velocity form we begin from Equation (24) of the previous section and elaborate:

$$m[nT_s] = K_{pc}e[nT_s] + K_{ic}T_s \sum_{k=0}^n e[(k-1)T_s] + K_{dc} \frac{e[nT_s] - e[(n-1)T_s]}{T_s} \quad (31)$$

We want to find the difference between successive samples of the controller, which will be denoted as $\nabla m[nT_s]$.

$$\nabla m[nT_s] = m[nT_s] - m[(n-1)T_s] \Rightarrow \quad (32)$$

$$\begin{aligned} \nabla m[nT_s] = & K_{pc}\{e[nT_s] - e[(n-1)T_s]\} + \\ & K_{ic}T_s \left\{ \sum_{k=0}^n e[(k-1)T_s] - \sum_{k=0}^{n-1} e[(k-1)T_s] \right\} + \\ & \frac{K_{dc}}{T_s} \{e[nT_s] - e[(n-1)T_s] - \\ & e[(n-1)T_s] + e[(n-2)T_s]\} \Leftrightarrow \end{aligned} \quad (33)$$

$$\begin{aligned} \nabla m[nT_s] = & \left(K_{pc} + \frac{K_{dc}}{T_s} \right) e[nT_s] + \\ & \left(K_{ic}T_s - K_{pc} - \frac{2K_{dc}}{T_s} \right) e[(n-1)T_s] + \\ & \frac{K_{dc}}{T_s} e[(n-2)T_s] \end{aligned} \quad (34)$$

By replacing $e[nT_s]$ with its equivalent, i.e., $R_{ref}[nT_s] - c[nT_s]$, Equation (34) becomes

$$\begin{aligned} \nabla m[nT_s] = & \left(K_{pc} + \frac{K_{dc}}{T_s} \right) R_{ref}[nT_s] + \\ & \left(K_{ic}T_s - K_{pc} - \frac{2K_{dc}}{T_s} \right) R_{ref}[(n-1)T_s] + \\ & \frac{K_{dc}}{T_s} R_{ref}[(n-2)T_s] - \\ & \left(K_{pc} + \frac{K_{dc}}{T_s} \right) c[nT_s] - \\ & \left(K_{ic}T_s - K_{pc} - \frac{2K_{dc}}{T_s} \right) c[(n-1)T_s] - \\ & \frac{K_{dc}}{T_s} c[(n-2)T_s] \end{aligned} \quad (35)$$

The mathematical trick that is used in derivation of velocity form is to force equality of the last three samples of reference input: $R_{ref}[nT_s] = R_{ref}[(n-1)T_s] = R_{ref}[(n-2)T_s]$.

This was inspired by the factors of R_{ref} , to eliminate the proportional and derivative action on the reference input, thus drastically reducing the effect of a change in the reference input. With this assumption Equation (35) changes to

$$\begin{aligned} \nabla m[nT_s] = & m[nT_s] - m[(n-1)T_s] = \\ & K_{ic}T_s R_{ref}[nT_s] - \\ & \left(K_{pc} + \frac{K_{dc}}{T_s} \right) c[nT_s] - \\ & \left(K_{ic}T_s - K_{pc} - \frac{2K_{dc}}{T_s} \right) c[(n-1)T_s] - \\ & \frac{K_{dc}}{T_s} c[(n-2)T_s] \Leftrightarrow \end{aligned} \quad (36)$$

$$\begin{aligned} m[nT_s] = & m[(n-1)T_s] + \\ & K_{ic}T_s R_{ref}[nT_s] - \\ & \left(K_{pc} + \frac{K_{dc}}{T_s} \right) c[nT_s] - \\ & \left(K_{ic}T_s - K_{pc} - \frac{2K_{dc}}{T_s} \right) c[(n-1)T_s] - \\ & \frac{K_{dc}}{T_s} c[(n-2)T_s] \end{aligned} \quad (37)$$

From Equation (37) we can proceed with Z-transform to provide a formula in z .

$$\begin{aligned} M(z) = & z^{-1}M(z) + K_{ic}T_s R_{ref}(z) - \left(K_{pc} + \frac{K_{dc}}{T_s} \right) C(z) \\ & - \left(K_{ic}T_s - K_{pc} - \frac{2K_{dc}}{T_s} \right) z^{-1}C(z) - z^{-2} \frac{K_{dc}}{T_s} C(z) \Leftrightarrow \end{aligned} \quad (38)$$

$$M(z) = -K_{pc}C(z) + K_{ic}T_s \frac{[R_{ref}(z) - z^{-1}C(z)]}{(1 - z^{-1})} - \frac{K_{dc}}{T_s} (1 - z^{-1})C(z) \quad (39)$$

A block diagram for the implementation of Equation (39) is provided in Figure 13.

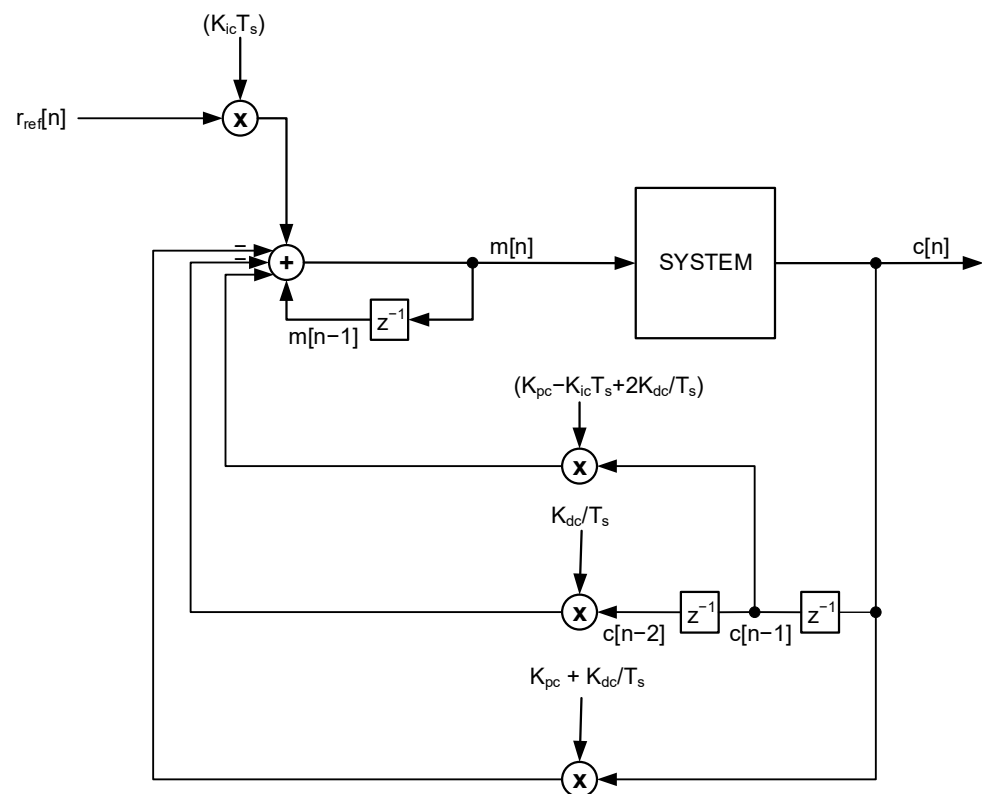


Figure 13. Block diagram of velocity form implementation of PID.

9. Test Bed for PID Control

For testing the PID algorithm, a simple system is used such as an integrator ($1/s$ at Laplace).

The next important thing is to identify which parts of the loop belong to digital domain and which parts belong to the analog domain. This will determine the correct position and number of analog-to-digital (A/Ds or ADCs) and digital-to-analog (D/As or DACs) that are needed.

Considering that the system is continuous time, and that all the calculations for control will be done in the digital domain, we recognize that the parts that belong to the digital domain are all apart from the system itself. This means that the only places where converters to interface the digital to the analog world are needed, are system boundaries.

The block diagram that describes the system interfacing is shown in Figure 14.

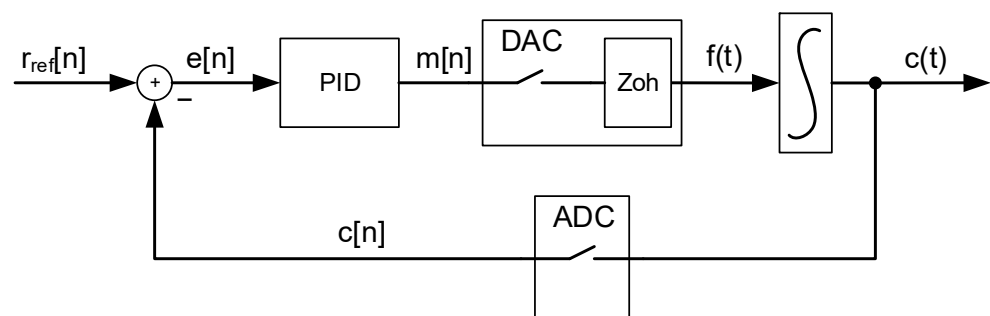


Figure 14. Control loop with PID and data converters.

The important elements that Figure 14 hides, are the reconstruction filter after DAC and the antialiasing filter before ADC. These filters belong to the analog world, are necessary, but have been left out for simplicity. Though for a thorough analysis of the system they must be there, since they have an effect in the total loop response.

10. Mathematical Analysis in Discrete Time

The first thing that must be said, regarding the analysis of the system in discrete time, is that the fictitious impulse samplers representing half DAC (the other half of DAC representation being Zoh) and ADC must be synchronized; this means same frequency and phase. This saves a lot of effort in the mathematical domain (such as sample rate converters) and guarantees that the signals available for processing refer to same sampling instant. Additionally, any internal processing must be aligned with the sampling frequency of both converters and must be carried out at time less than the sampling period.

Having already analyzed the digital PID (refer to equations (27)–(30)) the next step is to analyze Zoh contribution in discrete time.

10.1. Zoh Mathematical Treatment

What zero order hold does is keeping steady the value between sampling instants [12,13]. Mathematically and referring to Figure 15:

$$h(nT_s + t) = x(nT_s), t \in [0, T_s) \quad (40)$$

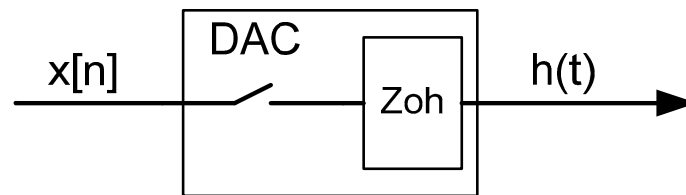


Figure 15. Reference figure for Zoh analysis.

This means that

$$h(t) = x(0)[U(t) - U(t - T_s)] + x(T_s)[U(t - T_s) - U(t - 2T_s)] + \dots + x(nT_s)[U(t - nT_s) - U(t - (n+1)T_s)] \quad (41)$$

By taking the Laplace transform of Equation (42) we have that

$$L\{h(t)\} = x(0)\left(\frac{1}{s} - e^{-sT_s}\frac{1}{s}\right) + x(T_s)\left(e^{-sT_s}\frac{1}{s} - e^{-s2T_s}\frac{1}{s}\right) + \dots + x(nT_s)\left[e^{-snT_s}\frac{1}{s} - e^{-s(n+1)T_s}\frac{1}{s}\right] \Rightarrow \quad (42)$$

$$H(s) = \sum_{n=0}^{\infty} \left[x(nT_s) e^{-snT_s} \frac{(1 - e^{-sT_s})}{s} \right] \Leftrightarrow \quad (43)$$

$$H(s) = \frac{(1 - e^{-sT_s})}{s} \sum_{n=0}^{\infty} x(nT_s) e^{-snT_s} \quad (44)$$

The sum at equation (45) is the starred Laplace transform of signal $x(t)$ (equals Z-transform with $z = e^{sT_s}$), which means that Equation (44) translates to

$$H(s) = G_{Zoh}(s)X^*(s) \quad (45)$$

which in turn means that

$$G_{Zoh}(s) = \frac{(1 - e^{-sT_s})}{s} \quad (46)$$

Since analysis of the system in discrete time involves Z-transform, then it is important to find the Z-transform of a continuous time transfer function when multiplied by the Zoh transfer function [1].

To do so we suppose that an arbitrary signal $Y(s)$ is

$$Y(s) = G_{Zoh}(s)G(s) \Rightarrow \quad (47)$$

$$Y(s) = \frac{1 - e^{-sT_s}}{s} G(s) = \frac{G(s)}{s} - e^{-sT_s} \frac{G(s)}{s} \quad (48)$$

if $\frac{G(s)}{s} = G_1(s)$ then Equation (49) becomes

$$Y(s) = G_1(s) - e^{-sT_s} G_1(s) \Rightarrow \quad (49)$$

$$L^{-1}\{Y(s)\} = L^{-1}\{G_1(s)\} - L^{-1}\{e^{-sT_s} G_1(s)\} \Rightarrow \quad (50)$$

$$y(t) = g_1(t) - \int_0^T \delta(t - T_s - \tau) g_1(\tau) d\tau \Rightarrow \quad (51)$$

$$y(t) = g_1(t) - g_1(t - T_s) \Rightarrow \quad (52)$$

$$Y(z) = G_1(z) - z^{-1} G_1(z) \Leftrightarrow \quad (53)$$

$$Y(z) = (1 - z^{-1}) G_1(z) \quad (54)$$

where

$$G_1(z) = Z\left\{\frac{G(s)}{s}\right\} \quad (55)$$

10.2. Applying Previous Results to Closed Loop

After completing the analysis of positioning the samplers in the loop and Zoh transfer function, we are ready to proceed to full loop analysis.

Starting with the continuous time section, we have (in s):

$$F(s) = M^*(s) \frac{1 - e^{-sT_s}}{s} \quad (56)$$

Note that with the * we imply the starred Laplace transform, i.e., the Laplace transform of an impulse sampled signal [1].

$$C(s) = F(s) \frac{1}{s} \quad (57)$$

Considering $N(s) = 0$ for the moment (superposition step), we have from Equation (58) and Equation (59) that

$$C(s) = M^*(s) \frac{(1 - e^{-sT_s})}{s} \cdot \frac{1}{s} \Rightarrow \quad (58)$$

$$\begin{aligned} C(z) &= Z\left\{M^*(s) \frac{(1 - e^{-sT_s})}{s} \cdot \frac{1}{s}\right\} = M(z) (1 - z^{-1}) Z\left\{\frac{1}{s^2}\right\} \\ &= M(z) (1 - z^{-1}) \frac{T_s z^{-1}}{(1 - z^{-1})^2} \Leftrightarrow \end{aligned} \quad (59)$$

$$C(z) = M(z) \frac{T_s z^{-1}}{(1 - z^{-1})} \quad (60)$$

Taking $M(z)$ from Equation (27), Equation (61) becomes

$$C(z) = \left[K_{pc} + K_{ic} \frac{z^{-1} T_s}{1 - z^{-1}} + \frac{K_{dc}}{T_s} (1 - z^{-1}) \right] \frac{T_s z^{-1}}{(1 - z^{-1})} E(z) \quad (61)$$

Considering that

$$E(z) = R_{ref}(z) - C(z) \quad (62)$$

we have that Equation (63) is

$$C(z) = \left[K_{pc} + K_{ic} \frac{z^{-1} T_s}{1 - z^{-1}} + \frac{K_{dc}}{T_s} (1 - z^{-1}) \right] \frac{T_s z^{-1}}{(1 - z^{-1})} [R_{ref}(z) - C(z)] \Leftrightarrow \quad (63)$$

$$C(z) = \frac{\left[K_{pc} + K_{ic} \frac{z^{-1}T_s}{1-z^{-1}} + \frac{K_{dc}}{T_s} (1-z^{-1}) \right] \frac{T_s z^{-1}}{(1-z^{-1})}}{1 + \left[K_{pc} + K_{ic} \frac{z^{-1}T_s}{1-z^{-1}} + \frac{K_{dc}}{T_s} (1-z^{-1}) \right] \frac{T_s z^{-1}}{(1-z^{-1})}} R_{ref}(z) \quad (64)$$

Equation (65) is the closed loop pulse transfer function from reference input $R_{ref}(z)$ to output $C(z)$.

The block diagram for the implementation of Equation (64) follows at Figure 16.

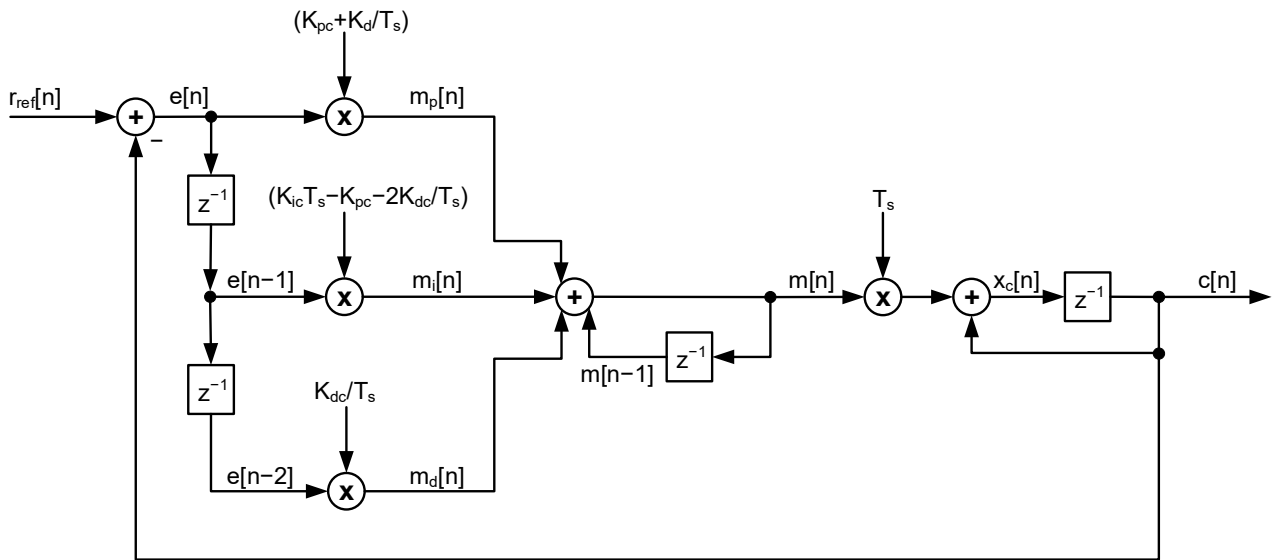


Figure 16. Block diagram for closed loop analysis in discrete time.

Elaborating on Equation (65) we can also find an equation that gives the same result for cross-checking

$$C(z) = \frac{\left[K_{pc} + K_{ic} \frac{z^{-1}T_s}{1-z^{-1}} + \frac{K_{dc}}{T_s} (1-z^{-1}) \right] T_s z^{-1}}{(1-z^{-1}) + \left[K_{pc} + K_{ic} \frac{z^{-1}T_s}{1-z^{-1}} + \frac{K_{dc}}{T_s} (1-z^{-1}) \right] T_s z^{-1}} R_{ref}(z) \Leftrightarrow \quad (65)$$

$$C(z) = \frac{(K_{pc}T_s + K_{dc})z^{-1} + (K_{ic}T_s^2 - K_{pc}T_s - 2K_{dc})z^{-2} + K_{dc}z^{-3}}{1 + (K_{pc}T_s - 2 + K_{dc})z^{-1} + (1 - K_{pc}T_s + K_{ic}T_s^2 - 2K_{dc})z^{-2} + K_{dc}z^{-3}} R_{ref}(z) \Rightarrow \quad (66)$$

$$\begin{aligned} c[n] = & (K_{pc}T_s + K_{dc})R_{ref}[(n-1)T_s] + \\ & + (K_{ic}T_s^2 - K_{pc}T_s - 2K_{dc})R_{ref}[(n-2)T_s] + \\ & + K_{dc}R_{ref}[(n-3)T_s] + \\ & - (K_{pc}T_s - 2 + K_{dc})c[(n-1)T_s] + \\ & - (1 - K_{pc}T_s + K_{ic}T_s^2 - 2K_{dc})c[(n-2)T_s] + \\ & - K_{dc}c[(n-3)T_s] \end{aligned} \quad (67)$$

10.3. Analyzing Closed Loop with Velocity Form

The system diagram using velocity form PID becomes as shown in Figure 17. The diagram also presents the actual positions of the converters in the system, and as discussed in the previous section the analysis takes as a fact that the converters are fully synchronized.

To analyze in Z , we must replace the system box with our approximation of the system, which is a Zoh followed by the $1/s$ function, which gives Equation (60) (see the respective derivation).

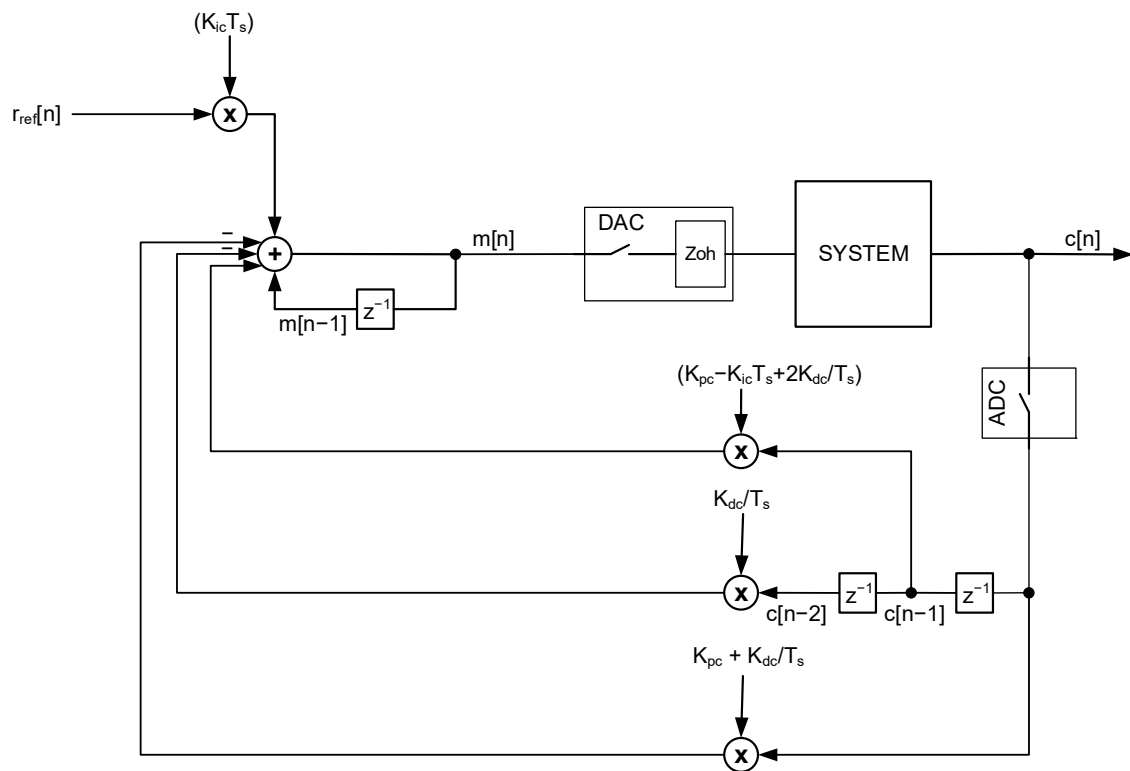


Figure 17. Using Velocity Form to Control the System. Converters are shown.

Repeating Equation (60) here we have:

$$C(z) = M(z) \frac{T_s z^{-1}}{(1 - z^{-1})} \Leftrightarrow C(z) = z^{-1} [C(z) + T_s M(z)] \quad (68)$$

which translates to the diagram of Figure 18.

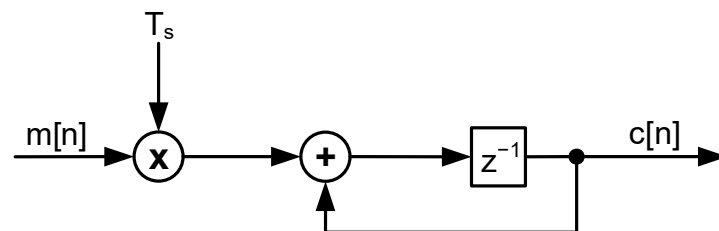


Figure 18. System with Zoh representation.

Replacing the “System” box of Figure 17 with its approximation as shown in Figure 18 we have the result of Figure 19.

This diagram is ready to be analyzed in Z.

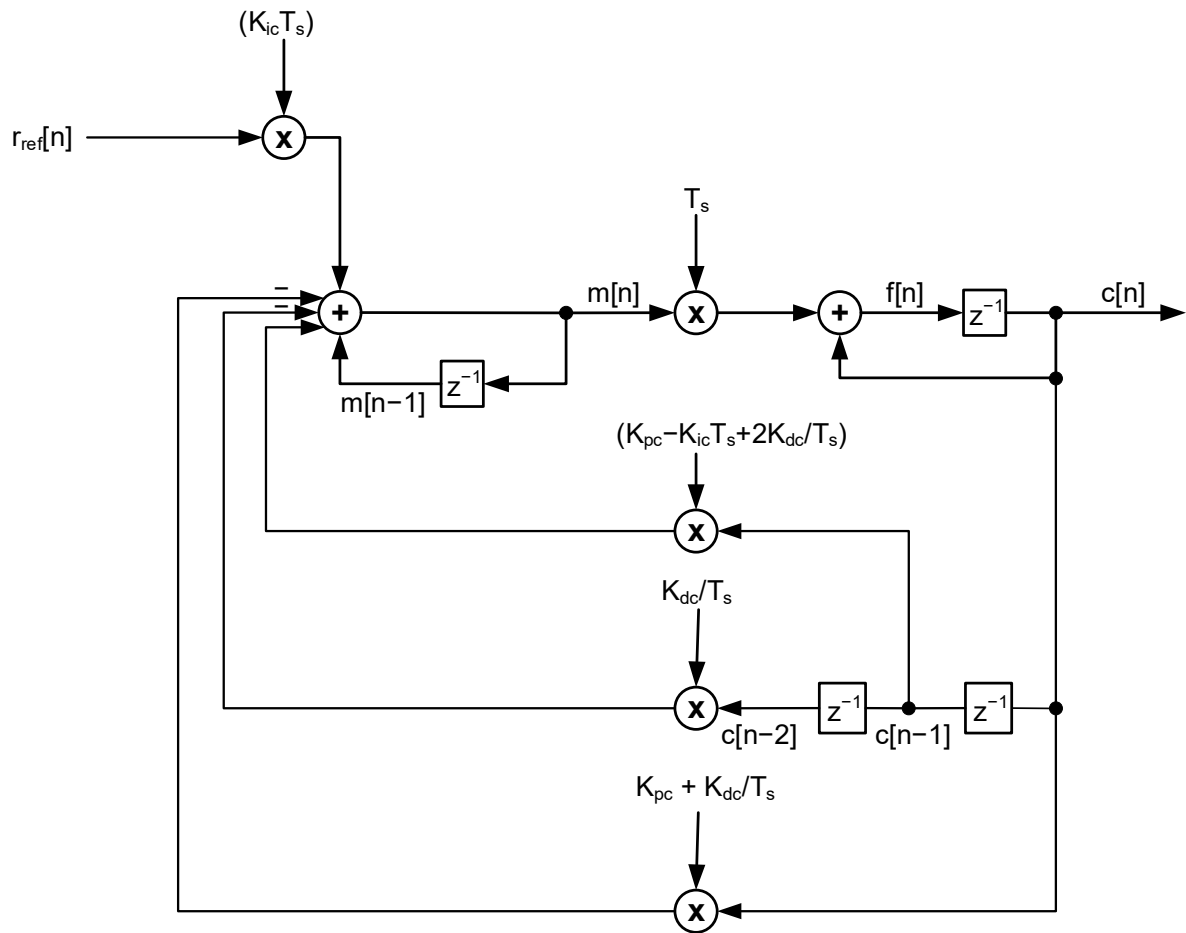


Figure 19. Implementation of system with Velocity Form PID ready for analysis in Z.

Starting with velocity form equation and

$$\left. \begin{aligned} M(z) &= -K_{pc}C(z) + K_{ic}T_s \frac{[R_{ref}(z) - z^{-1}C(z)]}{(1-z^{-1})} - \frac{K_{dc}}{T_s} (1-z^{-1})C(z) \\ C(z) &= M(z) \frac{T_s z^{-1}}{(1-z^{-1})} \end{aligned} \right\} \Rightarrow \quad (69)$$

$$C(z) = \left\{ -K_{pc}C(z) + K_{ic}T_s \frac{[R_{ref}(z) - z^{-1}C(z)]}{(1-z^{-1})} - \frac{K_{dc}}{T_s} (1-z^{-1})C(z) \right\} \frac{T_s z^{-1}}{(1-z^{-1})} \Leftrightarrow \quad (70)$$

$$[1 + (K_{pc}T_s - 2 + K_{dc})z^{-1} + (1 - K_{pc}T_s + K_{ic}T_s^2 - 2K_{dc})z^{-2} + K_{dc}z^{-3}]C(z) = K_{ic}T_s^2 z^{-1} R_{ref}(z) \Rightarrow \quad (71)$$

$$\begin{aligned} c[nT_s] &= K_{ic}T_s^2 R_{ref}[(n-1)T_s] - \\ &\quad (K_{pc}T_s - 2 + K_{dc})c[(n-1)T_s] - \\ &\quad (1 - K_{pc}T_s + K_{ic}T_s^2 - 2K_{dc})c[(n-2)T_s] - \\ &\quad K_{dc}c[(n-3)T_s] \end{aligned} \quad (72)$$

Equation (72) gives the response of the velocity-form controlled system at the sampling instants.

10.4. Fully Analog PID Loop Equations

An important comparison to test the quality of our implementation and provide the proof of concept is to compare it to a fully analog loop, as if our system was controlled by an analog PID controller [14].

For the derivation of the transfer function for the analog PID controlled system of “Figure 8” first note that “SYSTEM” block is replaced by $1/s$, which gives us the system of Figure 20 that will follow. For these systems we have:

$$G_{rc}(s) = \frac{\frac{G_p(s)}{s}}{1 + \frac{G_p(s)}{s}} \quad (73)$$

where $G_{rc}(s)$ is the system transfer function (from input $r(t)$ to output $c(t)$) and $G_p(s)$ is the transfer function of PID block, i.e.,

$$G_p(s) = K_{pc} + K_{ic} \frac{1}{s} + sK_{dc} \quad (74)$$

$$G_{rc}(s) = \frac{K_{pc} + K_{ic} \frac{1}{s} + sK_{dc}}{s + K_{pc} + K_{ic} \frac{1}{s} + sK_{dc}} = \frac{K_{pc}s + K_{ic} + K_{dc}s^2}{(1 + K_{dc})s^2 + K_{pc}s + K_{ic}} \Leftrightarrow \quad (75)$$

$$G_{rc}(s) = \frac{\frac{K_{dc}}{(1+K_{dc})}s^2 + \frac{K_{pc}}{(1+K_{dc})}s + \frac{K_{ic}}{(1+K_{dc})}}{s^2 + \frac{K_{pc}}{(1+K_{dc})}s + \frac{K_{ic}}{(1+K_{dc})}} \quad (76)$$

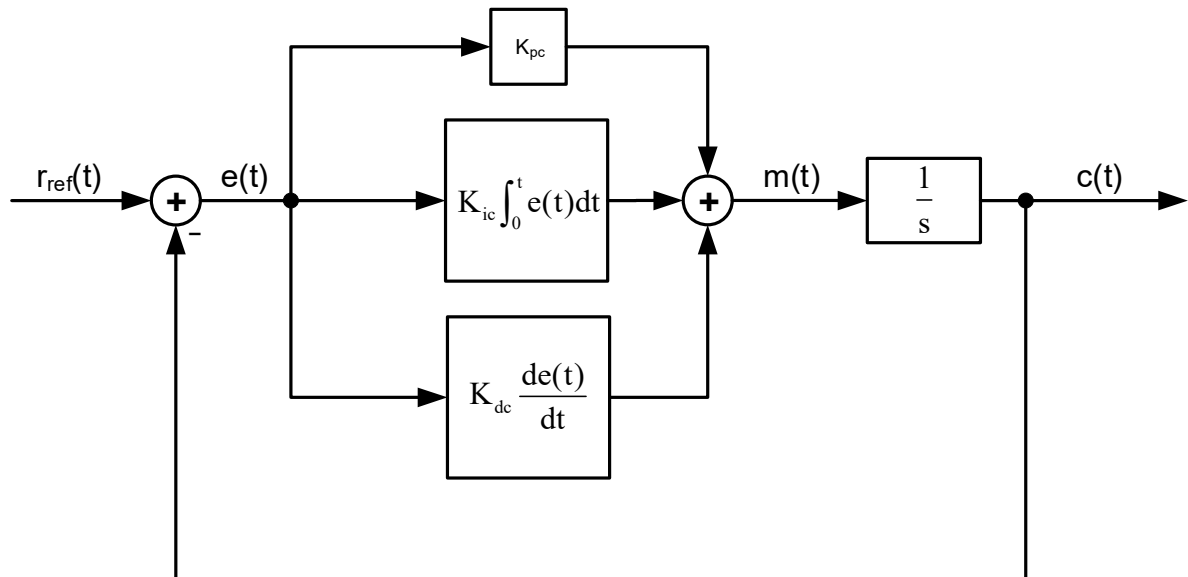


Figure 20. Continuous-time analogous to our control system core.

This transfer function refers to a system with the structure shown in Figure 20.

Comparing denominator of Equation (76) to the denominator of standard 2nd order system, i.e.,

$$s^2 + 2\zeta\omega_n s + \omega_n^2 \quad (77)$$

then to get a ζ and ω_n close to 1 we select $K_{pc} = 2$, $K_{ic} = 1$. Since we want to avoid the derivative term, if possible (to decrease the noise it can create), for our example we will use $K_{dc} = 0$. By applying the above gain choices to Equation (77), we get:

$$G_{rc}(s) = \frac{2s + 1}{s^2 + 2s + 1} \quad (78)$$

The step response of the transfer function of Equation (78) is our reference for comparison. The digital implementations will use the same gains, since the design process begins by selecting PID gains at the analog domain and applying them to the digital implementations described.

11. Results

11.1. Continuous-Time System Response

The result of simulating the analog system (for step response) is presented in Figure 21.

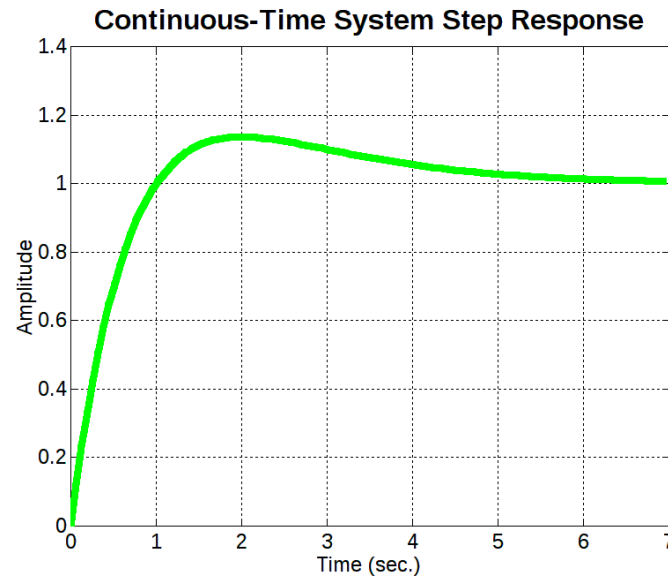


Figure 21. Step response of the analog system described by Equation (77).

As can be seen from Figure 21 the analog system is stable with an overshoot reaching the value of 1.1353.

11.2. Discrete-Time System Response with Positional Form PID

All results for the digital implementations that follow, have been evaluated with a sampling period $T_s = 30$ ms.

The result of block implementation of the digital system shown in Figure 16, evaluated every T_s is shown in figure “Figure 22. Comparison of continuous-time system to digital implementation”. The response matches the respective of the continuous time system, following the continuous-time overshoot. The peak value of the overshoot for the system with digital positional form PID implementation is 1.1395. Figure 22 has also embedded the response of the continuous-time system in green color, to be used as reference.

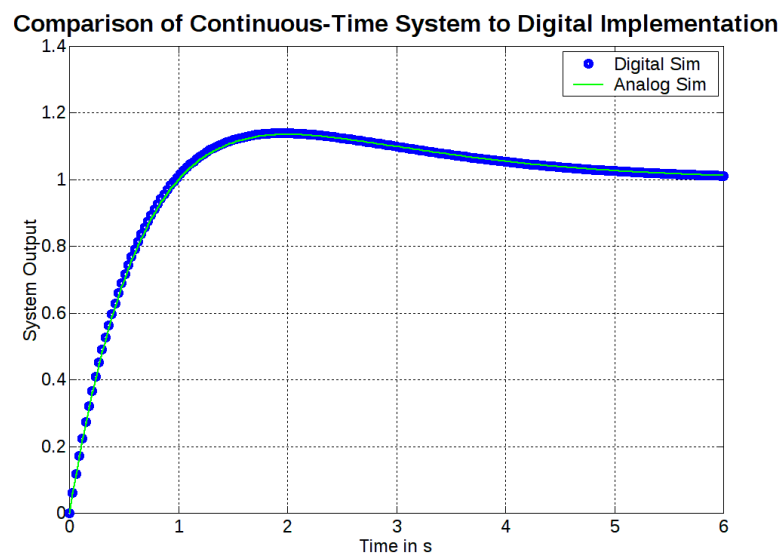


Figure 22. Comparison of continuous-time system to digital implementation.

“Figure 23. Comparison of equation implementation to block implementation” compares the step response that resulted by the calculation of every time instant from Equation (75), compared to calculating every time instant (every T_s) the state of the elements that diagram of Figure 16 is comprised of. The results are plotted both, to allow for comparison.

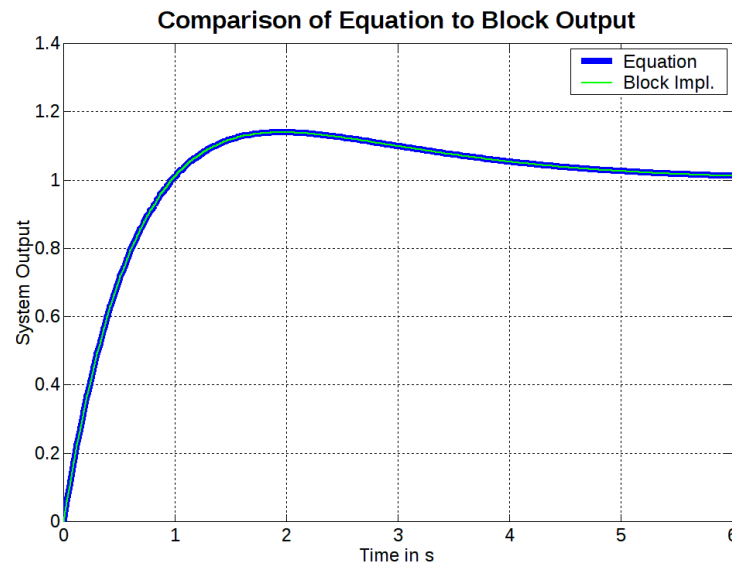


Figure 23. Comparison of equation implementation to block implementation.

11.3. Discrete-Time System Response with Velocity Form PID

Using the same PID gains with the previous subsection, the analysis can be repeated using the velocity form PID control.

Velocity form implementation is expected to have a more sluggish response, since there has been a mathematical manipulation that reduces the effect of input change.

The results of analyzing the system based on the block diagram of Figure 19 are presented in the figure below (Figure 24).

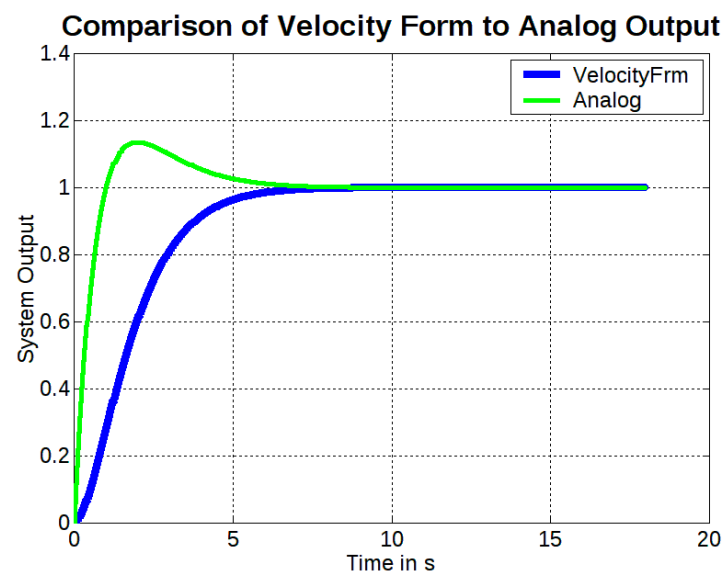


Figure 24. Velocity form PID compared to Analog PID.

As expected, the response using velocity form is more sluggish compared to analog controller, though, in this case, suits the application better since it smooths out the overshoot.

To verify the implementation of velocity form controller, we also verified the block diagram output by comparing to the values returned using Equation (71).

The result of the comparison is shown in Figure 25. As apparent from the figure, the results are an absolute match, which proves the equivalence of the two methods of result extraction.

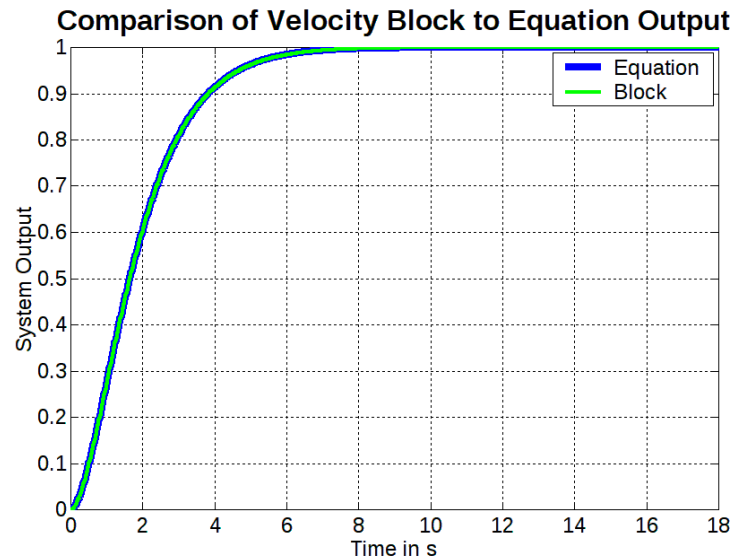


Figure 25. Verifying Velocity Form Control Implementation.

12. Analysis of Results and Discussion

To begin the section and to revise this work, it is noted that our approach for the digital implementation of a PID controller starts at the continuous-time case. After selection of controller gains securing acceptable response for the continuous-time system, two discrete-time implementations were investigated (positional and velocity form implementations), that used the gains calculated for the continuous-time case. For each of the two approaches the block diagram structure used to apply them to a system is presented first. These block diagrams translate directly to software or hardware, since they determine unambiguously and accurately the connections between memory elements (z^{-1}), multipliers and adders that are used. To perform alpha verification of both implementations, a simple testbed is presented consisting of the controller along with an integrator, which stands for the system to be controlled (open-loop plant). As a beta verification for the testbed, difference Equations (66) and (71) are employed to validate overall the testbed results.

Using the above as a basis, the results are analyzed starting from Figure 22. The output of the positional form PID testbed is practically an identical match to the continuous-time system response. This fact is also evident from the peak values of both responses that are very close (1.1353 for continuous-time, 1.1395 for the positional form). This result was also verified twice, using equation Equation (66). Figure 23 shows that the result is an exact match, as it ought to be, since Equation (66) describes exactly the same behavior that the block implementation does.

In contrast to positional form implementation, the response of the velocity-form, discrete-time implementation of the controller is not that close to that of the continuous-time system. As can be clearly seen from Figure 24, the response of such a controller implementation functions as a filter that smoothens the response. Although this may seem like an undesired behavior, in practice it provides more degrees of freedom for the control system designer. This means that by choosing between positional and velocity form implementations, we can expand the possibilities for the same range of gain selections.

Following the verification methodology for the positional form testbed outcome, the results of the velocity form testbed were also verified twice, by the use of Equation (71).

Figure 25 verifies the identity of the two results (block element evaluation versus verification equation).

13. Conclusions

In this work the discrete-time implementation of the controller for the electric propulsion powertrain of a small boat intended to be an integral part of a heterogeneous swarm consisting of autonomous surface watercraft as well as underwater vehicles and aircraft.

Per the analysis and investigations conducted, the conclusion is that the discrete-time implementation of a continuous-time PID controller can be a very good approximation especially if the positional form implementation is used. The velocity form implementation on the other side provides more conservative approximations, resulting in a more sluggish response.

For each of the discrete-time implementations, simple yet effective tools were procured to verify the developed methods and techniques for debugging purposes. A testbed with an integrator as the system to be controlled and an equation to verify mathematically the testbed response were also generated.

As a final note, it is pointed out that one should always check the relative positions of the poles to the unit circle on the z plane, in order to get an insight of the expected response. As a reminder, all the poles of the digital implementation must lie within the unit circle, which corresponds to the left half of the s -plane in the continuous-time case.

Author Contributions: Conceptualization, N.I.X. and E.K.L.; Formal analysis, E.K.L.; Investigation, N.I.X. and E.K.L.; Methodology, N.I.X. and E.K.L.; Supervision, N.I.X.; Validation, N.I.X. and E.K.L.; Writing—original draft, N.I.X. and E.K.L.; Writing—review and editing, N.I.X. and E.K.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to express their gratitude to the National Science Foundation (NSF) and, in particular, the Energy, Power, Control and Networks (EPCN) program for their continued support of this research under grant ECCS-1809182, titled “Collaborative Research: Design and Control of Networked Offshore Hydrokinetic Power-Plants with Energy Storage”.

Conflicts of Interest: Not applicable.

Appendix A. Electric Propulsion System for Unmanned Craft

The controller systems developed in this work are intended for distributed, decentralized control of the powertrain of small, low-cost boats to be used as part of autonomous watercraft swarms. In the case examined in this text, investigation of an electric propulsion control system is presented. The system at hand consists of a PMDC (Permanent Magnet Direct Current) motor [15–17], driving a waterjet. This, in turn, can be the propulsion device for a variety of small, unmanned boats, including one modeled after landing craft. The final objective is to build a number of unmanned, small-scale, low-cost boats, which can be used in the development of actual operational scenarios involving various command and control strategies, including remotely controlled operation, semi-autonomy and full autonomous deployment, task execution and recovery. The landing craft paradigm is useful since such watercraft comes with plenty space and cargo capacity for payload which may include extra energy storage, sensors, rotorcraft, etc. Using the electric motors for powering and the waterjet for propulsion offers significant advantages in maneuverability at low speeds and thrust vectoring capabilities. To best employ these capabilities though accurate and efficient real-time control is needed which is nowadays generally implemented

using microcontroller/microprocessor-based, embedded, digital hardware onboard which discrete-time control laws are executing.

For the system at hand, experimental investigation of the propulsion system was based on a specifically built testbed, as shown in Figure A1a, that allowed for identification of all system components. The testbed consisted of a waterproof, wheeled wagon (carriage) where the motor and the propulsion device, i.e., the waterjet, were mounted. Their arrangement was such that thrust could be developed to the wagon by the waterjet. The wagon was placed in a tank with appropriate dimensions. At the bottom of the tank rail guides were attached. These rails served a dual purpose. They supported the wagon when the tank is empty. Furthermore, they maintained the heading of the wagon when lateral disturbance forces were present. In order to eliminate static friction, however, special care was taken that the displacement of the wagon, when the tank was filled with water up to the operational level, was adequate to develop a lift force equal to the wagon weight. Therefore, in the case that lateral disturbances did arise, the rails maintained heading whilst static friction was suppressed due to elimination of vertical forces. At one end of the tank, a structure was built that could absorb the thrust of the propulsion device; therefore, the wagon could be kept still, as the propulsion system was ran at full load conditions. Then, thrust absorption was performed by a load cell intermitted between the structure's receiving point and the wagon's face of attack, the thrust of the propulsion system was measured when the hull (wagon in the testbed) was still; i.e., the static thrust of the system.

An analysis of the waterjet operation is presented next. In order to determine the thrust-rpm curve of the propulsor which in this case is a waterjet. With reference to the schematic in Figure A1a, the thrust, Ψ , developed by the waterjet in the testbed setup is given by

$$\Psi = \frac{d}{dt}(m \cdot V) \Rightarrow \Psi = \Delta M_{1 \sim 7} = \dot{m} \cdot (V_7 - V_1) \quad (A1)$$

In the above, M_j is the water momentum at the j -th point with respect to Figure A1a, V_j is the mean water velocity on the j -th cross section and m is the mass flow rate through the waterjet's duct. Since the wagon (carriage) is stationary the water velocity at the inlet (point 1) can be assumed to be zero. Hence, the developed thrust can be expressed as follows.

$$\Psi = \dot{m} \cdot V_7 = \rho \cdot Q \cdot V_7 = \rho \cdot Q \cdot \frac{Q}{A_7} = \rho \cdot \frac{Q^2}{A_7} \Rightarrow Q = \sqrt{\frac{A_7 \cdot \Psi}{\rho}} \quad (A2)$$

In the above, Q is the volumetric flow rate through the waterjet, A_7 is the cross-section of the waterjet at point 7 and ρ the water density which is set to 1000 kg/m^3 since fresh water is used for the experiments.

The per-unit-mass energy, E_j , at the j -th point with respect to Figure A1a, is given by the Bernoulli equation.

$$E_j = \frac{1}{2} \cdot V_j^2 + \frac{p_j}{\rho} + g \cdot h_j \Rightarrow \begin{cases} E_{\text{out}} = E_7 = \frac{1}{2} \cdot V_7^2 + \frac{p_7}{\rho} + g \cdot h_7 \\ E_{\text{in}} = E_1 = \frac{p_1}{\rho} + g \cdot h_1 \end{cases} \quad (A3)$$

In the above the fact that V_1 is approx. zero has been taken into account. Additionally, p_j stands for the static pressure at point j of the waterjet. Taking into account the proximity of points 1 and 7 and the fact that point 7 is located slightly higher than 1, one can easily conclude the following for the per-unit-mass energy increase.

$$\frac{p_7}{\rho} + g \cdot h_7 \simeq \frac{p_1}{\rho} + g \cdot h_1 \Rightarrow \Delta E = E_{\text{out}} - E_{\text{in}} = \frac{1}{2} \cdot V_7^2 \quad (A4)$$

The outlet water velocity, V_7 , is proportional to the volumetric flow rate, Q , through the waterjet.

$$V_7 = \frac{Q}{A_7} \Rightarrow \Delta E = \frac{1}{2} \cdot \left(\frac{Q}{A_7} \right)^2 \quad (A5)$$

The per-unit-mass energy increase of the water is also connected to the mechanical brake power, P_{brake} , delivered to the waterjet's rotor by the propulsion powertrain as shown below.

$$P_{\text{brake}} = \dot{m} \cdot \Delta E = \frac{\rho Q^3}{2A_7^2} \quad (A6)$$

Taking into account the above, the following relation between waterjet thrust and mechanical power is finally derived.

$$P_{\text{brake}} = \frac{\rho}{2A_7^2} \cdot \left[\sqrt{\frac{A_7 \cdot \Psi}{\rho}} \right]^3 = \frac{\Psi^{\frac{3}{2}}}{2\sqrt{\rho A_7}} \Leftrightarrow \quad (A7)$$

$$\Psi = \sqrt[3]{4\rho A_7 P_{\text{brake}}^2} \quad (A8)$$

Since the testbed is equipped with a load cell as shown in Figure A1a, that allows measuring the static thrust developed by the waterjet, the first of the above relationships can be used in order to validate the powertrain model used in this work. Specifically, the mechanical brake power absorbed by the waterjet is determined by shaft rpm and the load torque given in equation.

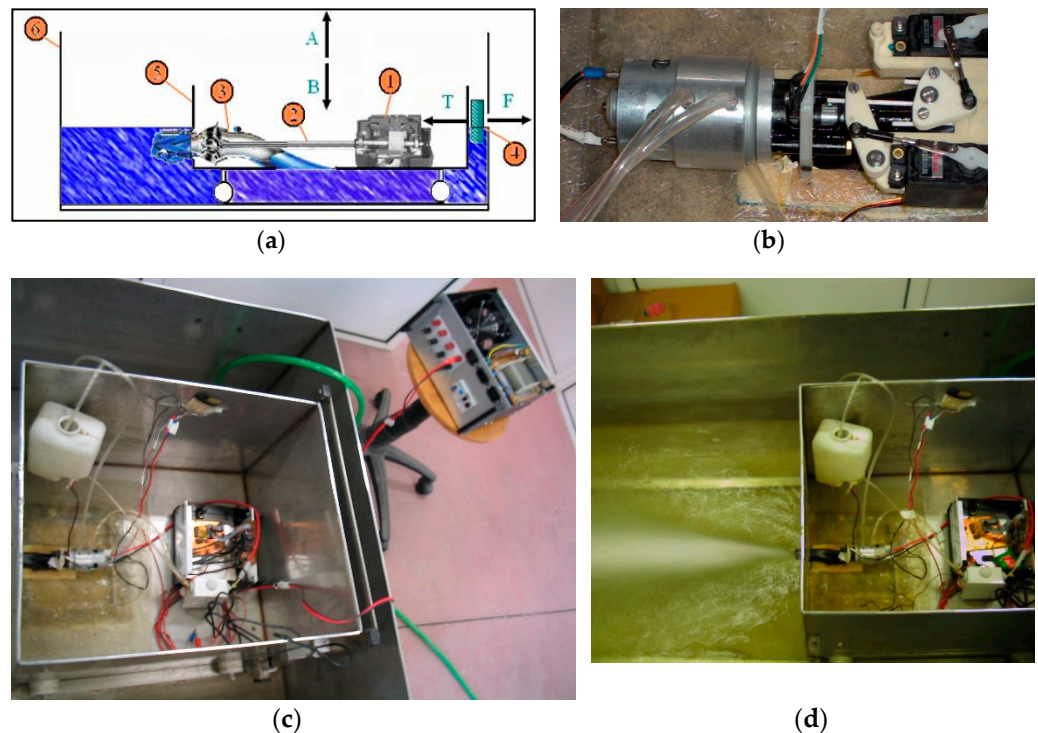


Figure A1. The boat electric propulsion system testbed: principle of operation (a), powertrain with speed sensor (b), testbed in action in pictures (c,d).

On the other hand, the experiments were performed under steady-state conditions, minimizing any disturbance to the system. To verify this hypothesis two series of experiments were performed; an open-loop one, where the control signal was manipulated directly and one with PI speed regulation applied where the rpm setpoint was provided instead. In both cases, the broadest possible operational range was spanned. In both cases the thrust value obtained by the load cell at each operating point was obtained. Based on

the thrust value a brake power estimate was calculated on the basis of equation previously. The brake power values calculated from of the thrust measurements were then compared to the brake power values calculated on the basis the propulsion system model. In specific,

$$\left. \begin{aligned} P_{\text{brake}} &= T_L \cdot \omega \\ \left. \frac{d\omega}{dT} \right|_{d=0} &= 0 \Rightarrow T_L = T_M(\omega, v) - T_f(\omega) \end{aligned} \right\} \Rightarrow \quad (\text{A9})$$

$$P_{\text{brake}} = [T_M(\omega, v) - T_f(\omega)] \cdot \omega \quad (\text{A10})$$

In the above, $T_M(\omega, v)$ stands for motor torque and $T_f(\omega)$ for friction torque.

The experimental results obtained in the two cases are shown in Figure A2 for the open-loop experiment and in Figure A3 for the closed-loop one. Note that the results for thrust are given in kgf (i.e., kp). The typical quadratic and cubic dependence of propulsor thrust and propulsion power on rpm, respectively, is verified satisfactorily in both cases.

Another important observation during the experiments was the effect of the relatively short length of the tank. At high rpm and load values the length was evidently not adequate to attenuate the perturbation introduced to the free surface by the operation of the waterjet. In effect, a stationary wave regime was established. This is manifested by the fact that at high load/rpm values in both experiments the thrust and power recordings were rather a cloud of points around the nominal value with non-negligible variance. This effect is specifically prominent at rpm values above 15,000 rpm.

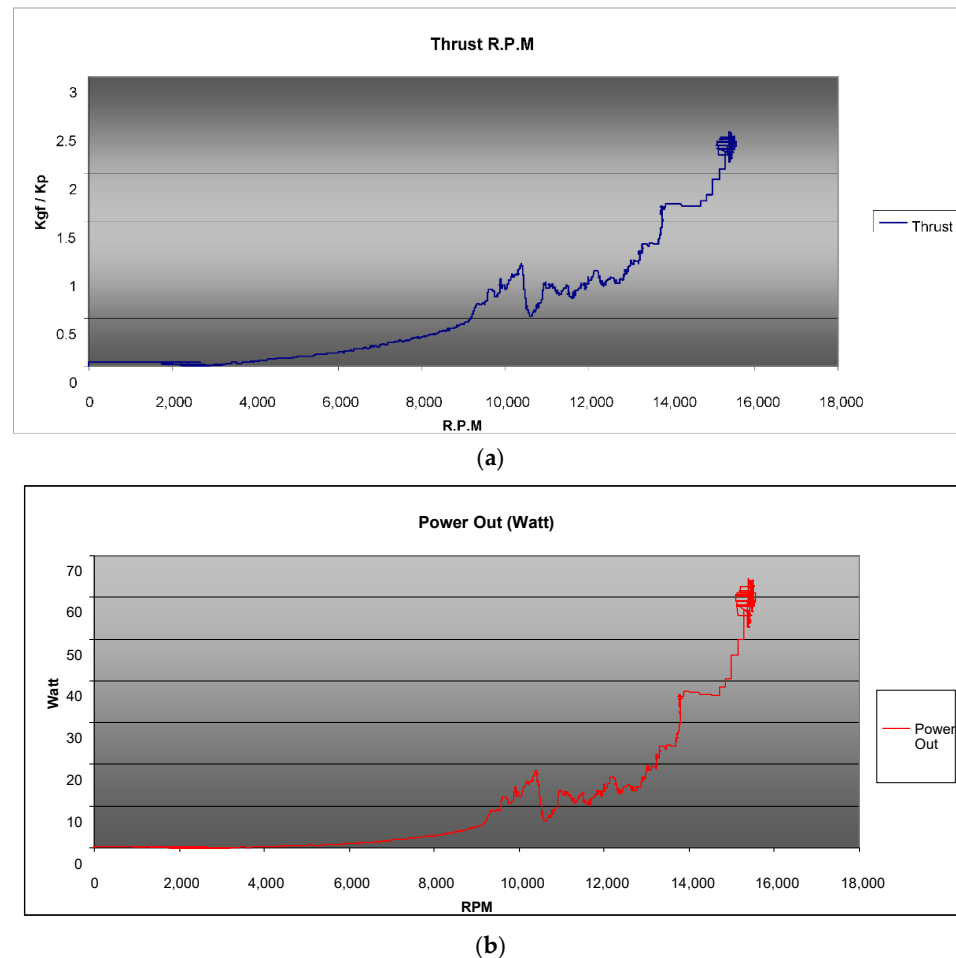


Figure A2. Open-loop thrust (a) and brake power (b) experimental results.

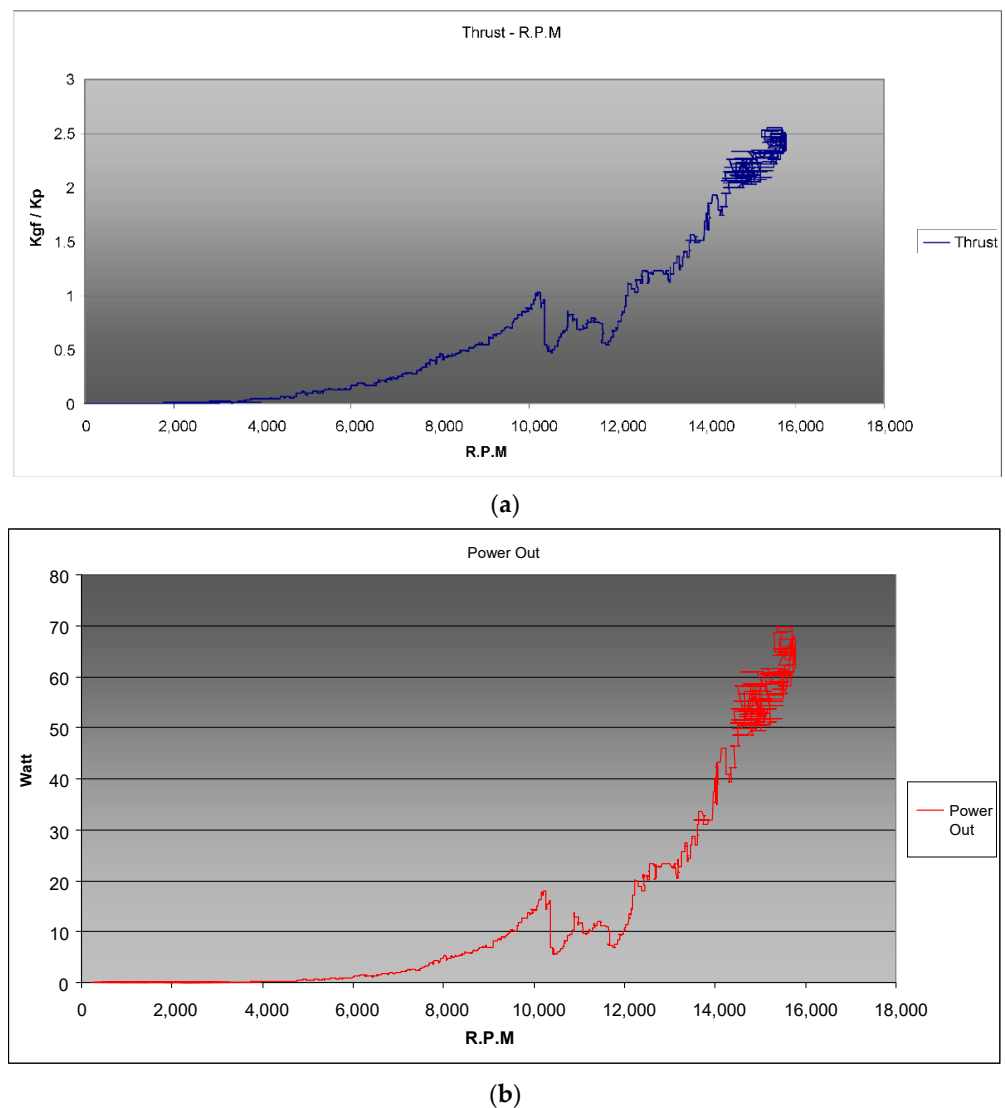


Figure A3. Closed-loop thrust (a) and brake power (b) experimental results.

Finally, it was found that the brake power values calculated on the basis of waterjet thrust measurements were in satisfactory agreement. Consequently, the propulsion power-train model developed was safely used for further design, control and integration studies.

References

1. Ogata, K. *Discrete-Time Control Systems*, 2nd ed.; Prentice-Hall Inc.: Englewood Cliffs, NJ, USA, 1995; ISBN 0-13-328642-8.
2. Kuo, B.C. *Digital Control Systems*, 1st ed.; CBS Publishing Asia LTD: Hong Kong, 1987; ISBN 4-8337-0013-1.
3. Bulten, N.W.H. Numerical Analysis of a Waterjet Propulsion System. Ph.D. Thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 2006.
4. Xiros, N.; Loghis, E. System Identification Using Neural Nets for Dynamic Modeling of a Surface Marine Vehicle. In Proceedings of the ASME 2014 International Mechanical Engineering Congress and Exposition, Montreal, QC, Canada, 14–20 November 2014; p. V010T13A015. Available online: https://www.researchgate.net/publication/304221905_System_Identification_Using_Neural_Nets_for_Dynamic_Modeling_of_a_Surface_Marine_Vehicle/citation/download (accessed on 30 August 2022).
5. Xiros, N.I.; Loghis, E.K. Continuous and Discrete-Time Models of Surface Watercraft Nonlinear Dynamics. *Acta Sci. Comput. Sci.* **2022**, *4*, 15–27.
6. Papoulis, A. *Signal Analysis*, 1st ed.; McGraw-Hill: New York, NY, USA, 1984; ISBN 0-07-066468-4.
7. Nagrath, I.J.; Gopal, M. *Control Systems Engineering*, 2nd ed.; John Wiley & Sons: Singapore, 1986; ISBN 9971-5-1056-1.
8. Golnaraghi, F.; Kuo, B.C. *Automatic Control Systems*, 10th ed.; Wiley: Hoboken, NJ, USA, 2010; ISBN 978-0-470-04896-2.
9. Xiros, N.I. *Robust Control of Diesel Ship Propulsion*, 1st ed.; Springer London: London, UK, 2002; ISBN 978-1-4471-0191-8.

10. Xiros, N.I.; An, P.-C.E. Control Theory and Applications. In *Springer Handbook of Ocean Engineering*; Dhanak, M.R., Xiros, N.I., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 227–276. ISBN 978-3-319-16649-0.
11. Xiros, N.I. Digital Signal Processing. In *Springer Handbook of Ocean Engineering*; Dhanak, M.R., Xiros, N.I., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 197–226. ISBN 978-3-319-16649-0.
12. Oppenheim, A.V.; Schafer, R.W. *Discrete-Time Signal Processing*, 1st ed.; Prentice-Hall Inc.: Hoboken, NJ, USA, 1989; ISBN 0-13-216771-9.
13. Soliman, S.S.; Srinath, M.D. *Continuous and Discrete Signals and Systems*, 2nd ed.; Prentice-Hall Inc.: Hoboken, NJ, USA, 1998; ISBN 0-13-569112-5.
14. Fossen, T. *Guidance and Control of Ocean Vehicles*, 1st ed.; Wiley: Chichester, UK, 1994; ISBN 0-471-94113-1.
15. Alciatore, D.G.; Hstand, M.B. *Introduction to Mechatronics and Measurement Systems*, 4th ed.; McGraw-Hill: New York, NY, USA, 2012; ISBN 978-0-07-338023-0.
16. Kilian, C.T. *Modern Control Technology: Components and Systems*, 3rd ed.; CENGAGE Delmar Learning: Clifton Park, NY, USA, 2005; ISBN 978-1-4018-5806-3.
17. Auslander, D.M.; Kempf, C.J. *Mechatronics: Mechanical System Interfacing*, 1st ed.; Prentice Hall: Hoboken, NJ, USA, 1995; ISBN 978-0-13-120338-9.