





Article

On Accurate Discrete-Time Dynamic Models of an Induction Machine

Ramón Herrera Hernández ¹, Carlos Reusser ^{1,*}, María Coronel ² and Rodrigo Carvajal ¹

¹ School of Electrical Engineering, Pontificia Universidad Católica de Valparaíso (PUCV), Av. Brasil 2147, Valparaíso 2362804, Chile; ramon.herrera.h@pucv.cl (R.H.H.); rodrigo.carvajal@pucv.cl (R.C.)

² Department of Electricity, Universidad Tecnológica Metropolitana (UTEM), Av. Jose Pedro Alessandri 1242, Santiago 7800002, Chile; macoronel@utem.cl

* Correspondence: carlos.reusser@pucv.cl

Abstract: Induction machines have become the standard for highly demanding industrial applications. This has led to the utilization of modern discrete-time control techniques (such as model predictive control) that require the estimation of internal variables that are not subject to measurement (such as the rotational velocity in sensorless applications). From this point of view, it is fundamental to have accurate discrete-time models of induction machines, particularly given their nonlinear nature, so that control techniques perform according to design requirements. In spite of the above, the modeling of induction machines has not received much attention in the literature, even though more powerful machines and faster microcontrollers are currently being used. To better understand induction machine models for control, in this paper, we develop and compare various discrete-time models of induction machines based on Euler, Taylor, and Runge–Kutta methods. In addition, we compare the Extended Kalman Filter and Unscented Kalman Filter for state estimation in terms of accuracy and computational burden. The models are derived and compared through extensive Monte Carlo simulations and the state estimation techniques are compared in terms of root mean squared error, execution time, and maximum absolute error. Our simulations show that, in general, the Taylor method yields more accurate models than both the Runge–Kutta and Euler methods. In particular, the Taylor method results in a root mean square error that is one order of magnitude smaller than the Euler method for stator current and rotor flux linkages. For rotor angular speed, the Runge–Kutta methods are more accurate than both the Taylor and Euler methods, resulting in a root mean square error that is two orders of magnitude smaller than the Euler method. On the other hand, the Extended Kalman Filter results in smaller execution time than the Unscented Kalman Filter, up to two orders of magnitude. In terms of root mean squared error and maximum absolute error, both filtering algorithms perform similarly.

Keywords: induction machine; Bayesian filtering; discretization

MSC: 37M10; 93C55; 93C57; 60G35



Citation: Herrera Hernández, R.; Reusser, C.; Coronel, M.; Carvajal, R. On Accurate Discrete-Time Dynamic Models of an Induction Machine. *Mathematics* **2024**, *12*, 373. <https://doi.org/10.3390/math12030373>

Academic Editor: Ivo Petráš

Received: 6 December 2023

Revised: 13 January 2024

Accepted: 15 January 2024

Published: 24 January 2024



Copyright: © 2024 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/>).

1. Introduction

Induction machines (IM) are basically an alternating current (AC) polyphase machine connected to an AC power source, either in the stator or in the rotor. The winding arrangement on the stator produces a rotating magnetic field in the machine airgap, which induces circulating currents that result in an induced rotating magnetic field in the rotor [1].

Induction machines have become the standard for highly demanding industrial applications. This is due to the simplicity in their construction and working principle, robustness, and wide speed range operation [2]. On the other hand, control strategies such as Field Oriented Control or Direct Torque Control are today's industrial standard control schemes, since they allow for high energy efficiency, i.e., they develop high torque per Ampere ratio [1,2]. Among modern control methods, Model Predictive Control (MPC) has been

widely studied in the literature [3,4]. Moreover, technological advances have allowed for the implementation of MPC strategies, see, e.g., [5,6]. It is important to note that power converters play a key role in driving the machine, delivering the necessary input voltage according to the control requirements. In general, one of the practical limitations in machine drives is the switching frequency of the power converter, which in turn limits the maximum sampling rate. This limitation arises from the fact that the higher the voltage the power converter handles, the lower the switching frequency it can withstand [7]. This sampling rate restriction may limit the performance of the controller since a low sampling rate could result in an inaccurate discrete-time model.

In variable-speed drives, speed is conventionally measured using a sensor, which is susceptible to electromagnetic noise; it requires maintenance and it is unsuitable for abrasive environments [8,9]. For these reasons, speed sensors are not used today. Instead, the rotor speed is estimated using an observer. In addition, the benefits of speed estimation encompass diminished hardware complexity and costs, reduced physical space requirements for the drive, and decreased maintenance demands, to name a few [8,9]. On the other hand, modern control techniques typically require knowledge of the rotor magnetic flux position. Given the cost implications and challenges associated with integrating electromagnetic flux sensors directly into the machine, the flux position is commonly estimated. Therefore, only the stator currents are measured whilst rotor speed and magnetic fluxes are estimated. This kind of drive is known as sensorless [8].

To estimate the rotor speed and rotor flux linkages, model-based methods are typically used, such as model adaptive systems [10,11], full-order observers [12], Luenberger observers [13], and sliding mode observers [14,15], among others. The methods previously mentioned are based on a deterministic formulation of the system. However, deterministic models do not take into account parameter offset from the nominal value, friction, lubricant ageing, or the effects of increased temperature (under normal operation) on the windings, all of which can be understood as model uncertainties. These uncertainties are typically modeled using random variables [16–19]. In this framework, the unknown variables are estimated using Bayesian filtering. In linear systems, the Kalman Filter (KF) is used, whereas for addressing nonlinear state estimation, the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF) are typically used [20–24].

The well-known Kalman Filter has been used in state estimation of induction machines in several works under the assumption of constant speed, see, e.g., [25–27]. This assumption results in a linear system, and since the Kalman Filter is the optimal linear filter [28,29], it is the best option in this case. However, in our problem of interest, the speed is not constant, particularly in the numerical examples where we have considered direct start and load application. This non-constant speed results in a nonlinear system, which in turn requires the utilization of nonlinear state estimation techniques. To the best of our knowledge, the Extended Kalman Filter is the most widely used technique for state estimation in induction machines, see, e.g., [30–34]. Additionally, the Euler method is typically used for the attainment of discrete-time induction machine models, and whilst at high sampling rates it can yield adequate models, our work suggests that higher-order methods yield more accurate models that result in more accurate state estimation in terms of root mean square error.

On the other hand, the state estimation using UKF for an induction machine is presented in [35–37], where it is compared with EKF. In [35,37], it is determined that UKF has a slightly better performance than EKF in state estimation for the induction machine. Conversely, in [36] it is reported that the EKF yields a slightly better performance than UKF. Moreover, UKF has been used in sensorless control [37–40]. In [38], in addition to estimating the states, the rotor resistance and the load torque are also estimated. In [40], UKF is compared with EKF, and UKF provides slightly better estimation. However, in terms of execution time, the UKF algorithm takes six times longer than the EKF algorithm.

Currently, due to the digital nature of controllers and processors, it has become necessary to discretize continuous-time models for their implementation. Historically,

in machine drives, the Euler method was widely adopted in the literature to obtain a discrete-time model of the machine due to its simplicity, see, e.g., [41–43]. In fact, in Bayesian filtering applications on IM drives using EKF [27,32–34] and UKF [35,37–40], Euler discretization methods are typically considered. Higher-order methods like the Taylor or Runge–Kutta, which involve greater computational costs and usually yield more complex systems, are not commonly utilized. The implementation of Taylor discretization, for instance, requires explicit calculations of higher-order derivatives. In contrast, Runge–Kutta employs estimated intermediate points between each sample to construct the model’s evolution [44]. Notably, Runge–Kutta methods are chosen to solve the nonlinear differential equations that define the machine behavior, resulting in “approximate” continuous-time currents, magnetic fluxes, and rotor speed. Moreover, Runge–Kutta methods are extensively utilized in simulation software such as Simulink and PLECS [45,46].

The contribution of this article is twofold: (i) a formulation and comparison of discrete-time models for induction machines using Euler, Taylor, and Runge–Kutta methods are presented, and (ii) a thorough comparison of EKF and UKF applied to the induction machine using the different discrete-time models is presented. Our analysis shows that the Euler method accurately represents the machine’s dynamics, but at very high sampling rates. On the other hand, at lower sampling rates, various discrete-time models based on the Taylor and Runge–Kutta methods can accurately represent the machine model. Finally, for the different discrete-time models, EKF and UKF algorithms are compared through extensive Monte Carlo simulations, in terms of root mean squared error (RMSE), execution time, and maximum absolute error.

The remainder of this article is organized as follows. In Section 2, the continuous-time model of the induction machine is described. In Section 3, the Euler, Taylor, and Runge–Kutta discrete-time models of the induction machine are formulated and compared with the continuous-time model in terms of RMSE. In Section 4, EKF and UKF algorithms are presented based on a stochastic model of the machine. In Section 5, EKF and UKF algorithms using different discrete-time models are compared in terms of execution time, RMSE, and maximum absolute error. Finally, we present our conclusions in Section 6.

2. Continuous-Time Induction Machine Model

The main parts of the induction machine are the stator and the rotor. The stator is made up of three symmetrically distributed coils. On the other hand, the rotor consists of aluminum bars shorted at the ends forming a cylinder. The three-phase coils are represented on a complex space, whose axes are named α - β . The stator space vectors are

$$v_s^{\alpha\beta} = v_s^\alpha + j v_s^\beta = \frac{2}{3} (a^0 v_a + a v_b + a^2 v_c) \tag{1}$$

$$i_s^{\alpha\beta} = i_s^\alpha + j i_s^\beta = \frac{2}{3} (a^0 i_a + a i_b + a^2 i_c) \tag{2}$$

$$\psi_s^{\alpha\beta} = \psi_s^\alpha + j \psi_s^\beta = \frac{2}{3} (a^0 \psi_a + a \psi_b + a^2 \psi_c) \tag{3}$$

where $v_s^{\alpha\beta}$ is the stator voltage space vector on the stationary reference frame α - β ; in the same way, $i_s^{\alpha\beta}$ and $\psi_s^{\alpha\beta}$ are the stator current and flux linkage, respectively, on the α - β reference frame. The unit space vector $a = e^{j\frac{2\pi}{3}}$ denotes the position of the coil on the space. The subscript a, b, c denotes the three-phase component.

The stator electrical equation on the α - β reference frame is

$$v_s^{\alpha\beta} = R_s i_s^{\alpha\beta} + \frac{d}{dt} \psi_s^{\alpha\beta} \tag{4}$$

where R_s is the stator resistance.

Similarly, the rotor electrical equation is

$$0 = R_r i_r^{\alpha\beta} + \frac{d}{dt} \psi_r^{\alpha\beta} + j p \omega_r \psi_r^{\alpha\beta} \tag{5}$$

where $i_r^{\alpha\beta}$ and $\psi_r^{\alpha\beta}$ are the rotor current and flux linkage, respectively, on the α - β reference frame; R_r is the rotor resistance; p is the number of pole pairs; and ω_r is the rotor shaft angular speed.

The flux linkages of the stator and rotor are

$$\psi_s^{\alpha\beta} = L_s i_s^{\alpha\beta} + L_m i_r^{\alpha\beta} \tag{6}$$

$$\psi_r^{\alpha\beta} = L_r i_r^{\alpha\beta} + L_m i_s^{\alpha\beta} \tag{7}$$

where L_s and L_r are the stator and rotor self-inductances, respectively, while L_m is the mutual inductance between stator and rotor.

On the other hand, the mechanical equation at the rotor shaft is

$$J \frac{d}{dt} \omega_r = \mathcal{T}_d - \mathcal{T}_l \tag{8}$$

where J is the inertia; \mathcal{T}_d is the machine developed torque; and \mathcal{T}_l represents the load torque. The machine developed torque is

$$\mathcal{T}_d = \frac{3}{2} p \Re\{j \psi_r^{\alpha\beta} i_s^{(\alpha\beta)*}\} \tag{9}$$

where the superscript * denotes the complex conjugate and $\Re\{\cdot\}$ denotes the real part of a complex number.

State-Space Representation

A state-space representation is used to analyze the IM model given in (2)–(9). For convenience, the selected state variables are the stator current $i_s^{\alpha\beta}$, the rotor flux linkage $\psi_r^{\alpha\beta}$, and the rotor shaft speed ω_r . By manipulating these variables, torque and speed control is achievable [2]. Furthermore, in sensorless drives only the stator current is measured.

Typically, the state vector is expanded and the load torque dynamics is modeled as well [32,39,40]. Including the load torque as a state variable, the IM dynamic equations are

$$\frac{d}{dt} i_s^\alpha = -a_1 i_s^\alpha + a_2 \psi_r^\alpha + a_3 \omega_r \psi_r^\beta + b_1 v_s^\alpha \tag{10}$$

$$\frac{d}{dt} i_s^\beta = -a_1 i_s^\beta + a_2 \psi_r^\beta - a_3 \omega_r \psi_r^\alpha + b_1 v_s^\beta \tag{11}$$

$$\frac{d}{dt} \psi_r^\alpha = a_4 i_s^\alpha - a_5 \psi_r^\alpha - a_6 \omega_r \psi_r^\beta \tag{12}$$

$$\frac{d}{dt} \psi_r^\beta = a_4 i_s^\beta - a_5 \psi_r^\beta + a_6 \omega_r \psi_r^\alpha \tag{13}$$

$$\frac{d}{dt} \omega_r = a_7 [\psi_r^\alpha i_s^\beta - \psi_r^\beta i_s^\alpha] - a_8 \mathcal{T}_l \tag{14}$$

$$\frac{d}{dt} \mathcal{T}_l = g(\mathcal{T}_l, \omega_r) \tag{15}$$

where a and b coefficients are

$$\begin{aligned} a_1 &= \frac{R_\sigma}{\sigma L_s} & a_2 &= \frac{L_m}{\sigma L_s \tau_r L_r} & a_3 &= \frac{L_m}{\sigma L_s L_r} p \\ a_4 &= \frac{L_m}{\tau_r} & a_5 &= \frac{1}{\tau_r} & a_6 &= p \\ a_7 &= \frac{3}{2} \frac{1}{J} p \frac{L_m}{L_r} & a_8 &= \frac{1}{J} & b_1 &= \frac{1}{\sigma L_s} \end{aligned}$$

The above coefficients are in terms of already-named machine parameters and additionally by

$$R_\sigma = R_s + R_r \frac{L_m^2}{L_r^2} \qquad \sigma = 1 - \frac{L_m^2}{L_s L_r} \qquad \tau_r = \frac{L_r}{R_r}$$

where R_σ is known as the stator dynamic transient resistance; σ as the leakage coefficient; and τ_r as the rotor time constant. The function $g(\mathcal{T}_l, \omega_r)$ represents any load torque function. The load torques that are generally found in electric drives are given in Table 1, where K is a proportional constant.

The dynamic equations are depicted in the block diagrams shown in Figure 1, where the α - β components of the space vectors are jointly represented. The states are obtained by integrating the right-hand side of Equations (10)–(15). Specifically, Figure 1a illustrates the joint behavior of the α - β stator current, while Figure 1b captures the joint dynamic behavior of the α - β rotor flux linkages. Figure 1c represents the dynamic behavior of the rotor shaft angular speed, with \otimes representing the cross product of the rotor flux linkages and stator currents ($\psi_r^{\alpha\beta} \times i_s^{\alpha\beta}$); finally, Figure 1d illustrates the dynamic behavior of the load torque.

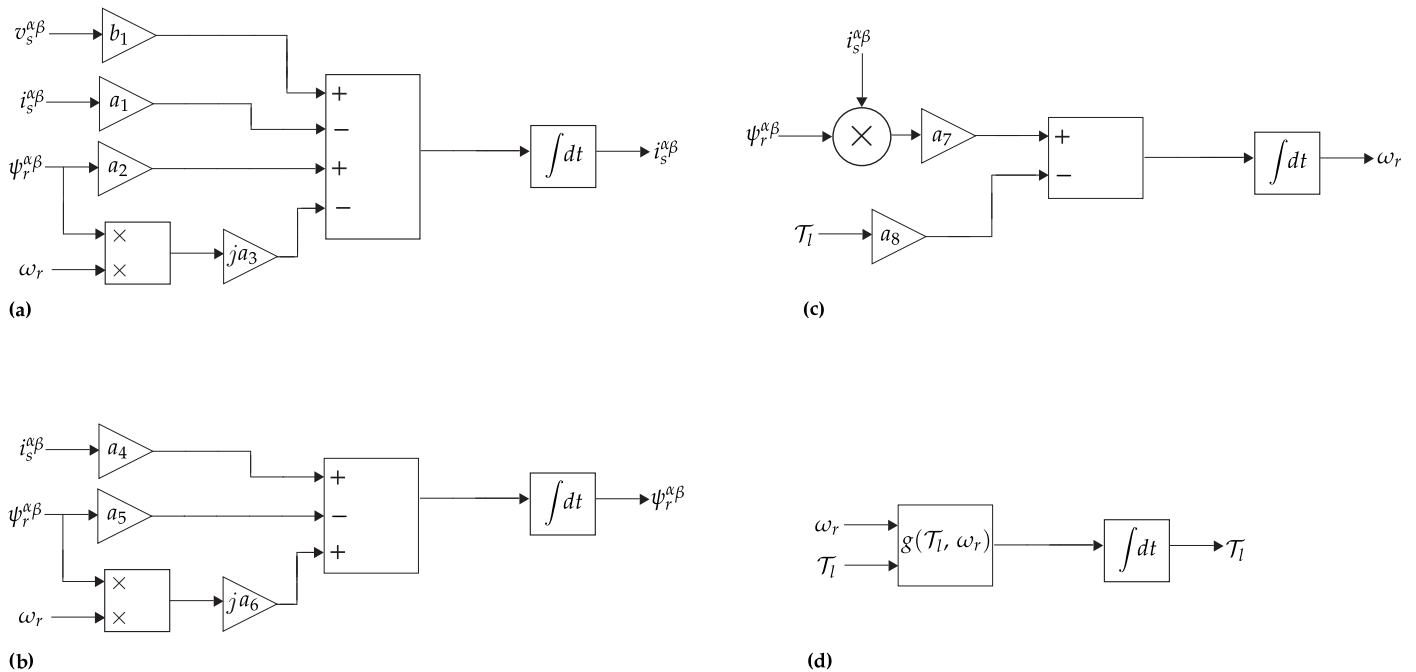


Figure 1. Block diagrams of IM continuous-time model. (a) α - β Stator currents dynamic. (b) α - β Rotor flux linkages dynamic. (c) Rotor shaft angular speed dynamic. (d) Load torque dynamic.

Table 1. Typical load torque functions [47].

Load Torque	Model
Constant	$\mathcal{T}_l = \mathcal{T}_{l0}$
Linear	$\mathcal{T}_l = K \omega_r$
Quadratic	$\mathcal{T}_l = K \omega_r^2$
Inverse	$\mathcal{T}_l = K \omega_r^{-1}$

In compact form, the representation of the IM model in state-space is

$$\frac{d}{dt} \mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{16}$$

$$\mathbf{y} = \mathbf{H} \mathbf{x} \tag{17}$$

Equation (16) is given by (10)–(15); \mathbf{x} is the state vector and \mathbf{u} is the input vector, in this case $\mathbf{v}_s^{\alpha\beta}$. The output (17) is related to the state vector by the matrix \mathbf{H} . Since it is a sensorless drive, only the stator current $i_s^{\alpha\beta}$ is measured, hence

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Note that the state variables depend implicitly on time t (i.e., $\mathbf{x} = \mathbf{x}(t)$). Also, note that the IM state Equation (16) exhibits a nonlinear behavior.

3. Discrete-Time Induction Machine Model

Due to the discrete nature of controllers, mostly discrete signals are processed today. In order to achieve a precise and accurate discretization, the discrete-time model must be as close as possible to the continuous-time model; hence, the error between continuous-time and discrete-time models must be close to zero. The discrete-time model is described in the form

$$\mathbf{x}_{k+1} = \mathbf{f}_d(\mathbf{x}_k, \mathbf{u}_k) \tag{18}$$

$$\mathbf{y}_k = \mathbf{H} \mathbf{x}_k \tag{19}$$

where $\mathbf{x}_k = \mathbf{x}(k T_s)$ and $\mathbf{x}_{k+1} = \mathbf{x}(k T_s + T_s)$; T_s is the time interval between samples. The input \mathbf{u}_k is considered as a zero-order hold (ZOH) sinusoidal wave.

The goal is to approximate numerically the IM dynamic model. There are several methods to numerically approximate differential equations. In this section, three methods are discussed: Euler, Taylor, and Runge–Kutta.

3.1. Euler Method

The Euler method is intuitive and straightforward, and is the basis of the two other methods. It approximates the solution to the differential equation as follows

$$\mathbf{x}_{k+1} \approx \mathbf{x}_k + T_s \left. \frac{d}{dt} \mathbf{x} \right|_{\mathbf{x}_k} \tag{20}$$

The IM model (16) approximated by the Euler Method is

$$i_{s,k+1}^\alpha = i_{s,k}^\alpha + T_s \left(-a_1 i_{s,k}^\alpha + a_2 \psi_{r,k}^\alpha + a_3 \omega_{r,k} \psi_{r,k}^\beta + b_1 v_{s,k}^\alpha \right) \tag{21}$$

$$i_{s,k+1}^\beta = i_{s,k}^\beta + T_s \left(-a_1 i_{s,k}^\beta + a_2 \psi_{r,k}^\beta - a_3 \omega_{r,k} \psi_{r,k}^\alpha + b_1 v_{s,k}^\beta \right) \tag{22}$$

$$\psi_{r,k+1}^\alpha = \psi_{r,k}^\alpha + T_s \left(a_4 i_{s,k}^\alpha - a_5 \psi_{r,k}^\alpha - a_6 \omega_{r,k} \psi_{r,k}^\beta \right) \tag{23}$$

$$\psi_{r,k+1}^\beta = \psi_{r,k}^\beta + T_s \left(a_4 i_{s,k}^\beta - a_5 \psi_{r,k}^\beta + a_6 \omega_{r,k} \psi_{r,k}^\alpha \right) \tag{24}$$

$$\omega_{r,k+1} = \omega_{r,k} + T_s \left(a_7 \left[\psi_{r,k}^\alpha i_{s,k}^\beta - \psi_{r,k}^\beta i_{s,k}^\alpha \right] - a_8 \mathcal{T}_{l,k} \right) \tag{25}$$

$$\mathcal{T}_{l,k+1} = \mathcal{T}_{l,k} + T_s g(\mathcal{T}_{l,k}, \omega_{r,k}) \tag{26}$$

For the Euler method, the local truncation error (i.e., the error committed at each time step) is of the order $\mathcal{O}(T_s^2)$.

3.2. Taylor Method

The derivative of a function is approximated using Taylor series expansions

$$\mathbf{x}_{k+1} \approx \mathbf{x}_k + T_s \left. \frac{d}{dt} \mathbf{x} \right|_{\mathbf{x}_k} + \frac{T_s^2}{2} \left. \frac{d^2}{dt^2} \mathbf{x} \right|_{\mathbf{x}_k} + \frac{T_s^3}{3!} \left. \frac{d^3}{dt^3} \mathbf{x} \right|_{\mathbf{x}_k} + \dots \tag{27}$$

The more terms in the series considered for the approximation, the more accurate the model. According to [48], in order to capture completely the effect of a ZOH-input at the next sampling instant, the input must appear explicitly in the state equations, that is, a discrete-time relative degree ≥ 1 . As simply as possible, the discretization of relative degree = 1 is achieved by expanding by second-order Taylor series the rotor flux, speed, and load torque given in (12)–(15). The discrete-time IM model with relative degree = 1 is given by

$$i_{s,k+1}^\alpha = i_{s,k}^\alpha + T_s \left(-a_1 i_{s,k}^\alpha + a_2 \psi_{r,k}^\alpha + a_3 \omega_{r,k} \psi_{r,k}^\beta + b_1 v_{s,k}^\alpha \right) \tag{28}$$

$$i_{s,k+1}^\beta = i_{s,k}^\beta + T_s \left(-a_1 i_{s,k}^\beta + a_2 \psi_{r,k}^\beta - a_3 \omega_{r,k} \psi_{r,k}^\alpha + b_1 v_{s,k}^\beta \right) \tag{29}$$

$$\begin{aligned} \psi_{r,k+1}^\alpha &= \psi_{r,k}^\alpha + T_s \left(a_4 i_{s,k}^\alpha - a_5 \psi_{r,k}^\alpha - a_6 \omega_{r,k} \psi_{r,k}^\beta \right) + \frac{T_s^2}{2} \left[a_4 \left(-a_1 i_{s,k}^\alpha + a_2 \psi_{r,k}^\alpha + a_3 \omega_{r,k} \psi_{r,k}^\beta + b_1 v_{s,k}^\alpha \right) \right] \\ &+ \frac{T_s^2}{2} \left\{ -a_6 \left[\omega_{r,k} \left(a_4 i_{s,k}^\beta - a_5 \psi_{r,k}^\beta + a_6 \omega_{r,k} \psi_{r,k}^\alpha \right) + \psi_{r,k}^\beta \left(a_7 \left(\psi_{r,k}^\alpha i_{s,k}^\beta - \psi_{r,k}^\beta i_{s,k}^\alpha \right) - a_8 \mathcal{T}_{l,k} \right) \right] \right\} \\ &- \frac{T_s^2}{2} \left[a_5 \left(a_4 i_{s,k}^\alpha - a_5 \psi_{r,k}^\alpha - a_6 \omega_{r,k} \psi_{r,k}^\beta \right) \right] \end{aligned} \tag{30}$$

$$\begin{aligned} \psi_{r,k+1}^\beta &= \psi_{r,k}^\beta + T_s \left(a_4 i_{s,k}^\beta - a_5 \psi_{r,k}^\beta + a_6 \omega_{r,k} \psi_{r,k}^\alpha \right) + \frac{T_s^2}{2} \left[a_4 \left(-a_1 i_{s,k}^\beta + a_2 \psi_{r,k}^\beta - a_3 \omega_{r,k} \psi_{r,k}^\alpha + b_1 v_{s,k}^\beta \right) \right] \\ &+ \frac{T_s^2}{2} \left\{ a_6 \left[\omega_{r,k} \left(a_4 i_{s,k}^\alpha - a_5 \psi_{r,k}^\alpha - a_6 \omega_{r,k} \psi_{r,k}^\beta \right) + \psi_{r,k}^\alpha \left(a_7 \left(\psi_{r,k}^\alpha i_{s,k}^\beta - \psi_{r,k}^\beta i_{s,k}^\alpha \right) - a_8 \mathcal{T}_{l,k} \right) \right] \right\} \\ &- \frac{T_s^2}{2} \left[a_5 \left(a_4 i_{s,k}^\beta - a_5 \psi_{r,k}^\beta + a_6 \omega_{r,k} \psi_{r,k}^\alpha \right) \right] \end{aligned} \tag{31}$$

$$\begin{aligned} \omega_{r,k+1} &= \omega_{r,k} + T_s \left(a_7 \left[\psi_{r,k}^\alpha i_{s,k}^\beta - \psi_{r,k}^\beta i_{s,k}^\alpha \right] - a_8 \mathcal{T}_{l,k} \right) + \frac{T_s^2}{2} \left[a_7 i_{s,k}^\beta \left(a_4 i_{s,k}^\alpha - a_5 \psi_{r,k}^\alpha - a_6 \omega_{r,k} \psi_{r,k}^\beta \right) \right] \\ &+ \frac{T_s^2}{2} \left[a_7 \psi_{r,k}^\alpha \left(-a_1 i_{s,k}^\beta + a_2 \psi_{r,k}^\beta - a_3 \omega_{r,k} \psi_{r,k}^\alpha + b_1 v_{s,k}^\beta \right) \right] - \frac{T_s^2}{2} \left[a_7 i_{s,k}^\alpha \left(a_4 i_{s,k}^\beta - a_5 \psi_{r,k}^\beta + a_6 \omega_{r,k} \psi_{r,k}^\alpha \right) \right] \\ &- \frac{T_s^2}{2} \left[a_7 \psi_{r,k}^\beta \left(-a_1 i_{s,k}^\alpha + a_2 \psi_{r,k}^\alpha + a_3 \omega_{r,k} \psi_{r,k}^\beta + b_1 v_{s,k}^\alpha \right) \right] - \frac{T_s^2}{2} a_8 g(\mathcal{T}_{l,k}, \omega_{r,k}) \end{aligned} \tag{32}$$

$$\mathcal{T}_{l,k+1} = \mathcal{T}_{l,k} + T_s g(\mathcal{T}_{l,k}, \omega_{r,k}) + \frac{T_s^2}{2} \dot{g}(\mathcal{T}_{l,k}, \omega_{r,k}) \tag{33}$$

For a second-order Taylor series expansion, the local truncation error is $\mathcal{O}(T_s^3)$.

3.3. Runge–Kutta Method

The Runge–Kutta (RK) method approximates the solution of the differential equation based on estimates of the solution at different points. In this subsection, two RK methods are taken into account: RK 2 and RK 4. The RK 2 approximation considers two estimates, whereas RK 4 considers four. The advantage that Runge–Kutta has over Taylor is that it is not necessary to explicitly calculate higher-order derivatives of $f(x, u)$.

3.3.1. Runge–Kutta 2

Runge–Kutta 2 is a second-order approximation. This method requires two evaluations of $f(x, u)$ at each time step. The IM model (16) is approximated by

$$x_{k+1} = x_k + \frac{T_s}{2} (r_1 + r_2) \tag{34}$$

where

$$\mathbf{r}_1 = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \quad (35)$$

$$\mathbf{r}_2 = \mathbf{f}(\mathbf{x}_k + T_s \mathbf{r}_1, \mathbf{u}(t_k + T_s)) \quad (36)$$

and $t_k = k T_s$. Note that \mathbf{r}_2 estimation requires future input values. Since the system is assumed to be causal and future inputs are unknown, \mathbf{u}_k is used instead. For the IM, \mathbf{r}_1 and \mathbf{r}_2 are found in Appendix A.1.

3.3.2. Runge–Kutta 4

Runge–Kutta 4 is a fourth-order method that requires four evaluations of $\mathbf{f}(\mathbf{x}, \mathbf{u})$ in one sample period. The approximation of IM model (16) by RK 4 is

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{T_s}{6}(\mathbf{r}_1 + 2\mathbf{r}_2 + 2\mathbf{r}_3 + \mathbf{r}_4) \quad (37)$$

where

$$\mathbf{r}_1 = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \quad (38)$$

$$\mathbf{r}_2 = \mathbf{f}\left(\mathbf{x}_k + \frac{T_s}{2}\mathbf{r}_1, \mathbf{u}\left(t_k + \frac{T_s}{2}\right)\right) \quad (39)$$

$$\mathbf{r}_3 = \mathbf{f}\left(\mathbf{x}_k + \frac{T_s}{2}\mathbf{r}_2, \mathbf{u}\left(t_k + \frac{T_s}{2}\right)\right) \quad (40)$$

$$\mathbf{r}_4 = \mathbf{f}(\mathbf{x}_k + T_s \mathbf{r}_3, \mathbf{u}(t_k + T_s)) \quad (41)$$

Since the input is ZOH and no future input values are taken into account, \mathbf{u}_k is used in \mathbf{r}_2 , \mathbf{r}_3 , and \mathbf{r}_4 . For the IM, the estimate points \mathbf{r}_1 , \mathbf{r}_2 , \mathbf{r}_3 , and \mathbf{r}_4 are found in Appendix A.2.

3.4. Comparison between Models

The previously mentioned discrete-time IM models were contrasted with each other and with a continuous-time model. The simulations were carried out in Matlab. To compare the discrete-time models with the continuous-time one, the latter was obtained using a fifth-order Dormand–Prince (DOPRI 5) method with the same discretization time as the discrete-time models. More information about this method is given in Appendix B. For the simulations, $T_s = 200 \mu\text{s}$ and a 4 kW IM was used; its nominal parameters are $V = 380 \text{ V}$; $I = 8.6 \text{ A}$; $f = 50 \text{ Hz}$; two pole pairs; $R_s = 1.32 \Omega$; $R_r = 2.63 \Omega$; $L_m = 0.1889 \text{ H}$; $L_s = 0.1972 \text{ H}$; $L_r = 0.2012 \text{ H}$; and $J = 0.528 \text{ kg}\cdot\text{m}^2$.

To illustrate the differences between discrete-time and continuous-time models, two scenarios are taken into account. First, the IM is directly started from the grid. Second, a 15 Nm step-type load impact occurs suddenly at 4 s. The total simulation time was 6 s.

Table 2 contrasts the discrete-time models with the continuous-time model by presenting the computed root mean squared error (RMSE) between them, calculated as follows

$$RMSE(\mathbf{x}) = \sqrt{\frac{1}{N} \sum_{i=1}^N (\mathbf{x}_{i, cont} - \mathbf{x}_{i, disc})^2} \quad (42)$$

To begin with, in Table 2, the best approximation for the stator current and rotor flux linkage is obtained by Taylor. The RK 2 and RK 4 models are not as accurate as Taylor, due to the fact that RK 2 and RK 4 do not take into account future input estimation for the approximation. Conversely, for the rotor speed RK 4 and RK 2 provide the best approximation. Finally, Euler results in the worst approximation as expected.

Table 2. Models' root mean squared error.

State	Euler	Taylor	RK 2	RK 4
i_s^α	2.3288	0.3743	0.5830	0.4188
i_s^β	2.3286	0.3723	0.5985	0.4177
ψ_r^α	0.0567	0.0091	0.0245	0.0191
ψ_r^β	0.0567	0.0089	0.0286	0.0190
ω_r	21.6914	11.3117	1.9997	0.1401
\mathcal{T}_l	0.0046	0.0046	7.0356×10^{-5}	7.0171×10^{-9}

Figure 2 shows the α - β stator currents and rotor flux linkages for the discrete-time and continuous-time models with no load condition at steady state. Figure 2a shows that the Taylor and RK 4 methods yield discrete-time values that are almost identical to the continuous-time values. In the same figure, we can observe that the RK 2 method yields a larger difference since the amplitude of the currents is smaller than the continuous-time currents, whilst the Euler method results in even larger errors since the resulting currents have a much smaller amplitude. In Figure 2b, the rotor flux linkages for each method are shown. The Taylor, RK 2, and RK 4 signals resemble the continuous-time signals. However, the Euler method results in a slightly larger difference from the continuous-time flux than the Taylor and RK methods. On the other hand, Figure 3 illustrates the rotor angular speed and zero load torque for all models described. As shown, both the Taylor and Euler models are less accurate under no-load conditions than the RK methods. In particular, RK 4 provides the most accurate speed approximation, followed by RK 2.

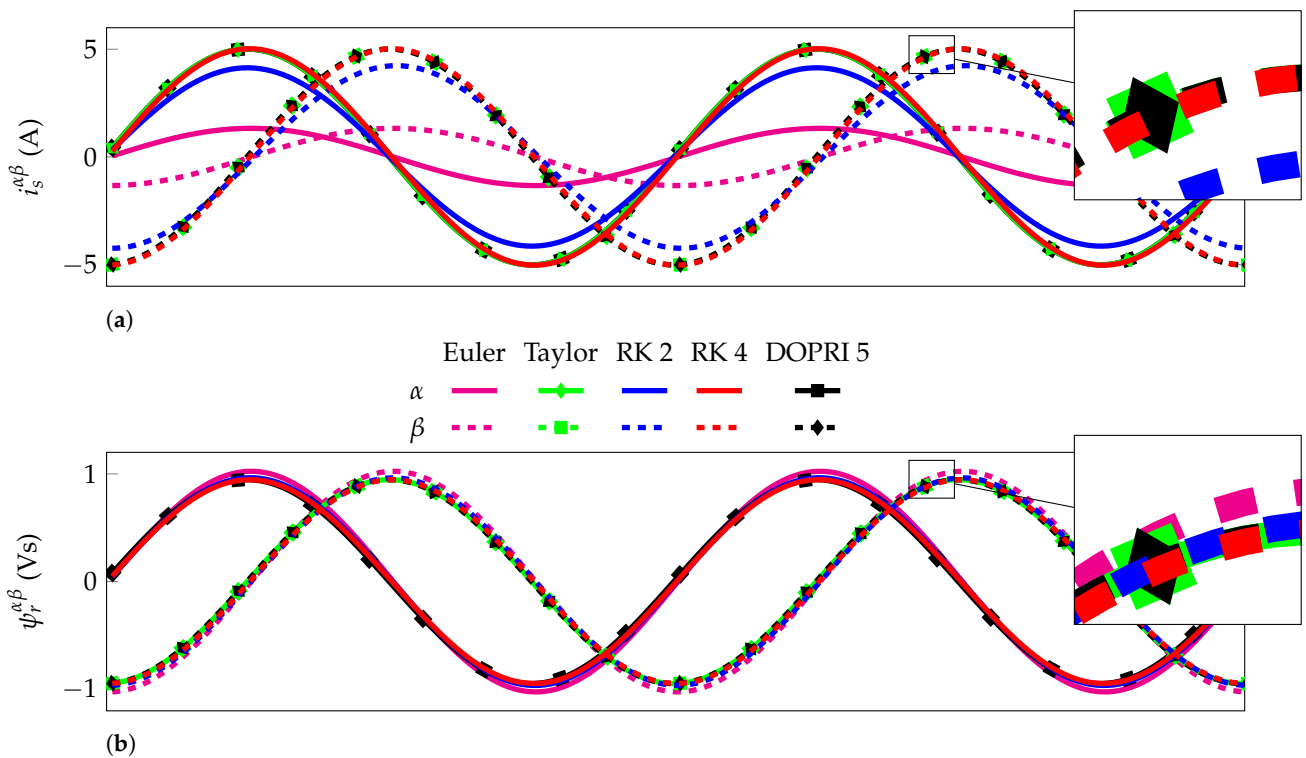


Figure 2. No-load machine. (a) Stator α - β current. (b) Rotor α - β flux linkage.

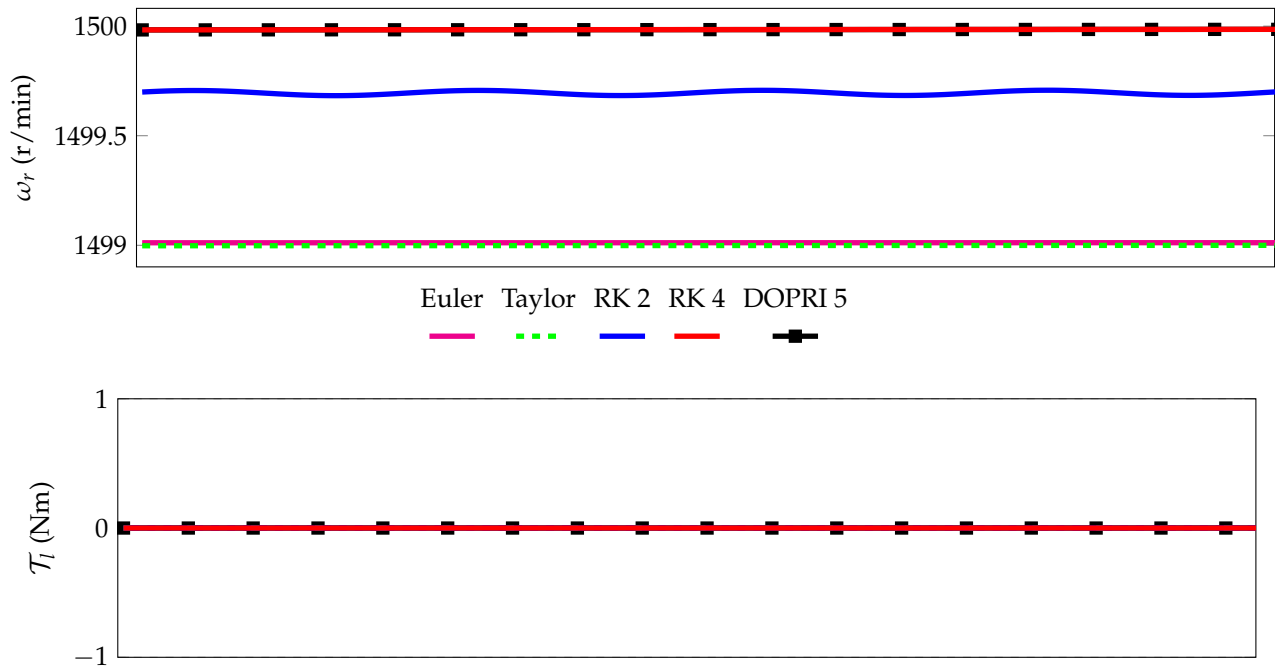


Figure 3. No-load machine. (a) Rotor shaft speed. (b) Load torque.

On the other hand, notice that after the sudden step-type load torque impact occurs, stator current increases while rotor speed decreases. Figure 4 shows the stator currents and rotor flux linkages for the discrete-time and continuous-time models after the torque impact occurs. In this case, the Euler method yields a bad approximation, since the resulting currents not only exhibit a smaller amplitude but also a large phase error. In contrast, the Taylor, RK 2, and RK 4 models resemble the continuous-time model. Figure 5 illustrates the rotor angular speed and the load torque impact for all models. We can observe that, once the speed reaches the steady state, RK 4 and Taylor result in a very similar curve to the continuous-time model, followed by RK 2 and finally Euler. In addition, since load torque does not depend on rotor angular speed or any other state variable, the differences between the discrete-time models and the continuous-time model are, in general, small, as shown in Figure 5b.

In general, accurate models are desired since, in modern industrial systems, modern control techniques are applied. If the model is not accurate, model-based observers may fail to represent the actual behavior of the states, providing inaccurate feedback to the controller, hence resulting in poor control performance. Since with the Euler method the model error is reflected not only in the magnitude but also in the phase of the signals of interest, the overall control loop's performance decreases greatly. This situation is naturally avoided in order to satisfy control system performance and requirements.

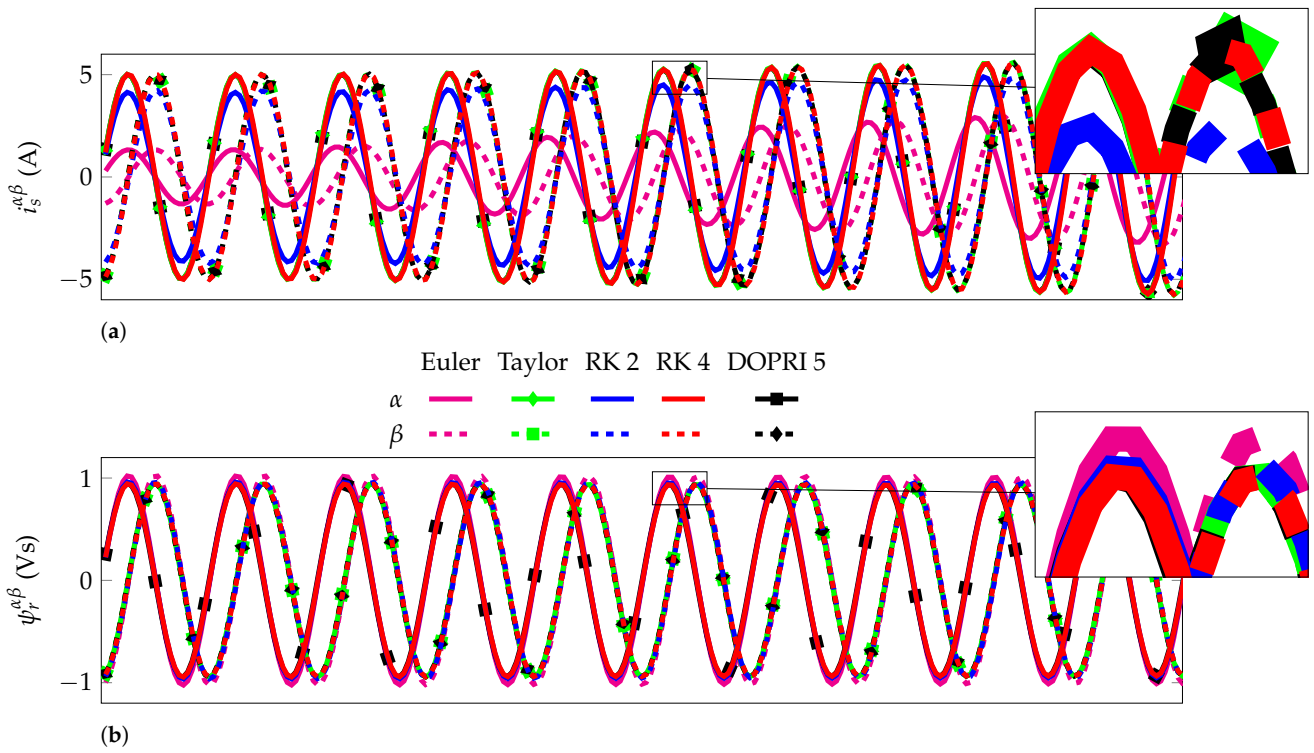


Figure 4. Torque impact. (a) Stator α - β current. (b) Rotor α - β flux linkage.

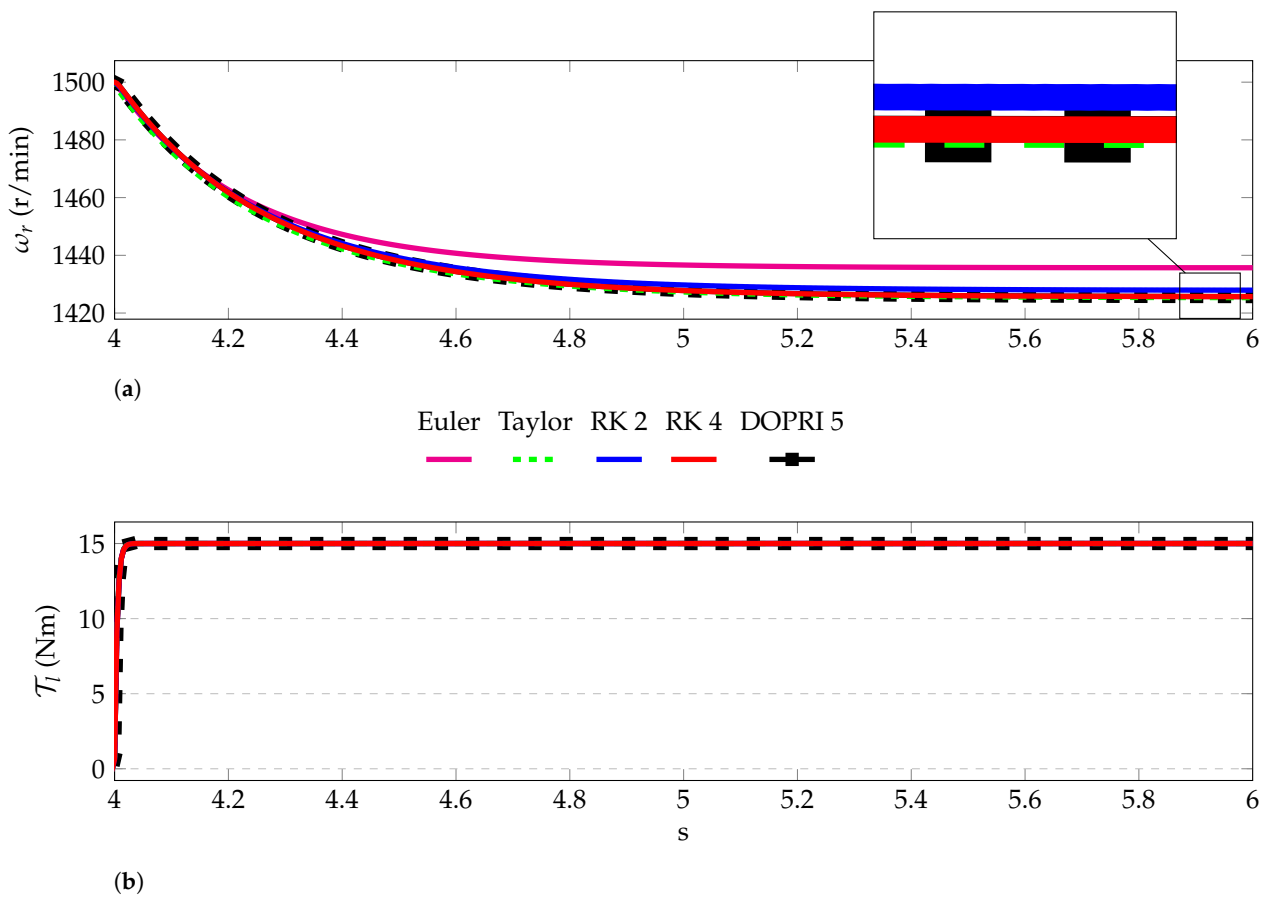


Figure 5. Torque impact. (a) Rotor shaft speed. (b) Load torque.

Changes in Sample Time

In this section, the behavior of the RMSE between the discrete-time models and the continuous-time model is observed as the discretization time varies. The discretization time is adjusted until the error of the model discretized by Euler is nearly zero. Two scenarios are considered; first, the IM is directly started, and then a sudden load impact occurs at 4 s. This follows the same approach as the previous section.

Figure 6 illustrates how RMSE varies according to T_s for α - β stator currents, α - β rotor flux linkages, and rotor shaft speed. Since the error between the α and β components of the stator current and rotor flux linkage components is similar, the average value is used in this figure. As expected, as the sampling time decreases, error decreases as well. In particular, the Taylor discrete-time model is better than RK 2 and RK 4 in terms of stator current and rotor flux linkage RMSE. Conversely, RK 2 and RK 4 are more accurate for speed approximation. Regarding the load torque approximation, RK 2 and RK 4 are better; nevertheless, Euler and Taylor are near zero. On the other hand, Euler RMSE rises dramatically as T_s increases. Thus, using Euler for approximating the IM model is not advisable when the T_s exceeds 50 μ s. To conclude, Taylor is the best approximation, prioritizing stator currents and rotor flux linkage approximations.

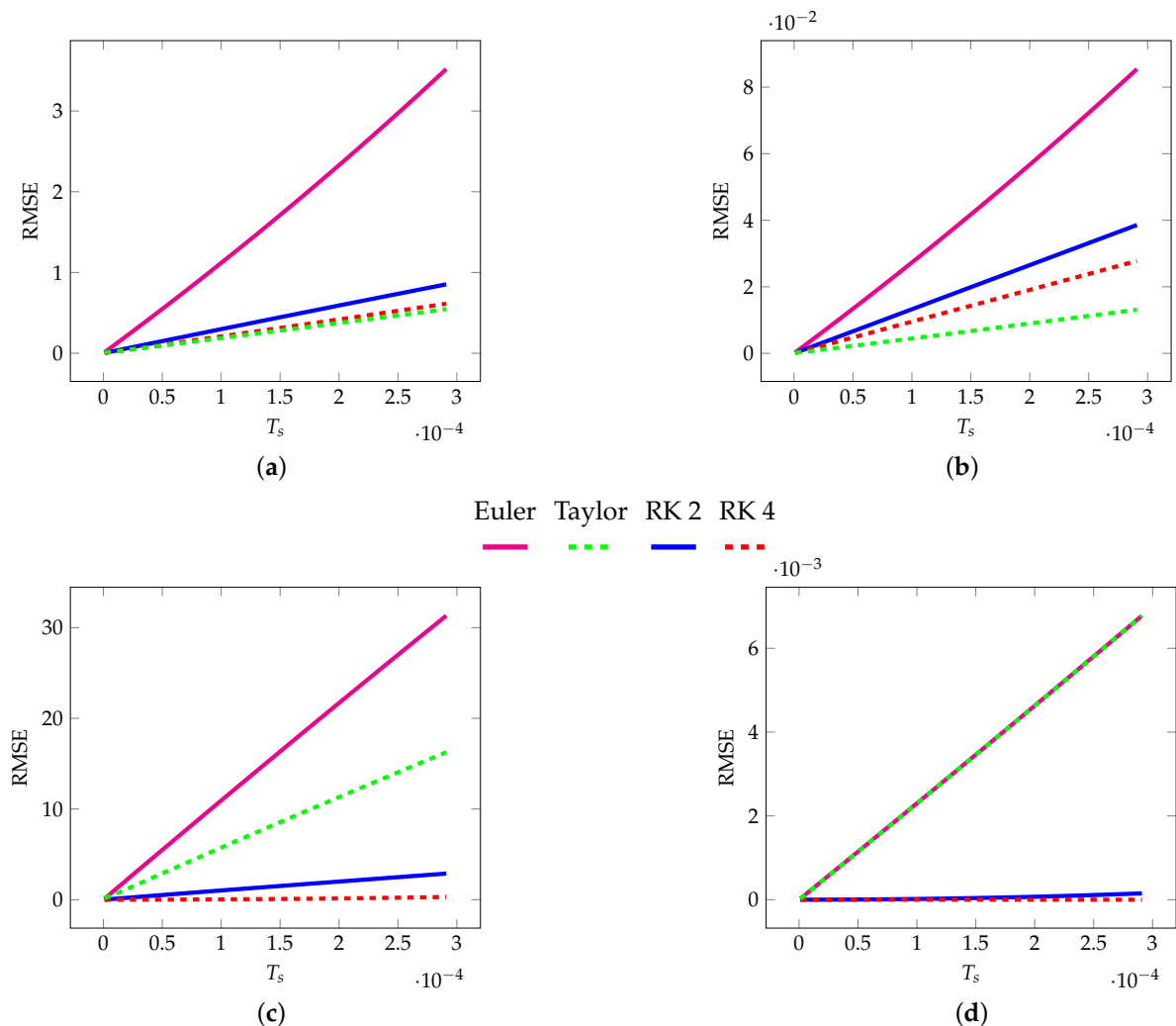


Figure 6. Variation of RMSE as a function of sampling time. (a) α - β Stator current RMSE. (b) α - β Rotor flux linkage RMSE. (c) Rotor shaft speed RMSE. (d) Load torque RMSE.

4. Filtering

The continuous-time deterministic model of the IM was described in (16) and (17). Even so, those equations do not include model uncertainties or unknown disturbances, such as winding temperature rises, mechanical vibrations, switching noise, etc. All these uncertain effects can be represented by adding Gaussian noise in the state Equation (16) and output Equation (17) [16–19]. Also, considering the discrete nature of the measurement and control, a discrete-time representation of the model is used. Therefore, the discrete-time stochastic model of the IM is as follows

$$x_{k+1} = f_d(x_k, u_k) + w_k \tag{43}$$

$$y_k = H x_k + v_k \tag{44}$$

where w_k and v_k are Gaussian noise with zero mean and covariance matrix Q and R , respectively. The term w_k is associated with model uncertainties, while v_k represents measurement noise.

Under a stochastic point of view, the state estimation is obtained from noisy measurements. In Bayesian inference, the estimation corresponds to calculating the joint probability density function (pdf) of the states given the measurements $P(x_{0:T} | y_{1:T})$, which incorporates all the statistical information about the state $P(x_{0:T})$ that is contained in the available observation $P(y_{1:T} | x_{0:T})$ and the initial condition $P(x_{0:T})$ [19,21]. Using the Bayes theorem, the joint pdf is

$$P(x_{0:T} | y_{1:T}) = \frac{P(y_{1:T} | x_{0:T}) P(x_{0:T})}{P(y_{1:T})} \tag{45}$$

where the denominator $P(y_{1:T})$ is a normalization constant. It is necessary to mention that $x_{0:T}$ and $y_{1:T}$ are the time series of the state vector $\{x_0, \dots, x_T\}$ and of the measurements $\{y_1, \dots, y_T\}$ vector, respectively. For real-time applications, it is unnecessary and computationally inefficient to compute the full joint distribution. For this reason, the marginal distribution of the current state given all the measurements $P(x_k | y_{1:k})$, also known as posterior, is computed instead. Additionally, assuming the IM model to be Markovian, it is possible to compute the posterior recursively [19,21]. The recursion is based on calculating the posterior distribution in two steps, starting from an initial prior distribution $P(x_0)$.

- First: predict. The prediction of the distribution of the states at the current time k is computed from the dynamic model by means of the Chapman–Kolmogorov equation

$$P(x_k | y_{1:k-1}) = \int P(x_k | x_{k-1}) P(x_{k-1} | y_{1:k-1}) dx_{k-1} \tag{46}$$

- Second: update. Given the measurement y_k at time k , the posterior distribution is calculated by solving

$$P(x_k | y_{1:k}) = \frac{1}{Z_k} P(y_k | x_k) P(x_k | y_{1:k-1}) \tag{47}$$

where Z_k is a normalization constant.

Unfortunately, since the IM model is nonlinear, it is not possible to solve the filtering Equations (46) and (47) in closed form. Nevertheless, there exist state estimation methods for nonlinear models; two of the classic methods are based on the Kalman Filter (KF): the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF). Both are addressed in this section.

4.1. Extended Kalman Filter

The most well-known nonlinear filtering method is the Extended Kalman Filter, which propagates the mean and covariance through a linearization of the system resulting in a

Gaussian posterior pdf. However, EKF does not ensure the convergence of the estimation, since it depends on the linearization of the model around an operating point that is also estimated [20–24]. The implementation steps of EKF for N samples are summarized in Algorithm 1, where \hat{x} is the state estimate and P is its covariance matrix. The subscript k denotes the posterior and $k|k - 1$ the prior. K is the Kalman gain; I is the identity matrix; and $F_{k|k-1}$ is the Jacobian matrix of $f_d(x, u)$ evaluated at $k - 1$.

Algorithm 1: EKF Algorithm

1 **Input:** Prior distribution mean and covariance $x(0), P_0$.

2 **for** $k = 1$ to N **do**

Prediction Step:

3 Compute the prior estimate using the discrete-time model

$$\hat{x}_{k|k-1} = f_d(\hat{x}_{k-1|k-1}, u_k) \tag{48}$$

$$P_{k|k-1} = F_{k|k-1} P_{k-1|k-1} F_{k|k-1}^T + Q \tag{49}$$

Update Step :

4 Compute the Kalman gain

$$K = P_{k|k-1} H^T [H P_{k|k-1} H^T + R]^{-1} \tag{50}$$

5 Compute the state estimation and covariance

$$\hat{x}_k = \hat{x}_{k|k-1} + K (y_k - H \hat{x}_{k|k-1}) \tag{51}$$

$$P_k = (I - K H) P_{k|k-1} \tag{52}$$

6 **end**

7 **Output:** The state estimate \hat{x}_k and the covariance matrix P_k , for $k = 1, \dots, N$

4.2. Unscented Kalman Filter

On the other hand, there is the Unscented Kalman Filter, which by means of the Unscented Transform (UT) propagates representative points of the pdf through the non-linearity of the system to estimate the state. The advantage that UKF has over EKF is that the representative points are adjustable by the user, attaining in general a better estimation [20–24]. These representative points, named Sigma Points (SPs), ensure the propagation of the mean and the covariance [49]. A total of $2n + 1$ SP are required, with $n = \dim\{y\}$. The SPs are computed as follows

$$\mathcal{X}^{(0)} = x \tag{53}$$

$$\mathcal{X}^{(i)} = x + \sqrt{(n + \lambda)} (\sqrt{P})_i \tag{54} \quad i = 1, \dots, n$$

$$\mathcal{X}^{(i)} = x - \sqrt{(n + \lambda)} (\sqrt{P})_i \tag{55} \quad i = n + 1, \dots, n$$

where λ is a scale factor that modifies the dispersion of the SP around the mean. The term \sqrt{P}_i is computed using Cholesky factorization. Also, weight factors $W_i^{(m)}$ and $W_i^{(c)}$ are associated to the mean and covariance, defined as

$$W_0^{(m)} = \frac{\lambda}{n + \lambda} \qquad W_0^{(c)} = \frac{\lambda}{n + \lambda} + (1 - \alpha^2 + \beta)$$

$$W_{i \neq 0}^{(m)} = \frac{\lambda}{2(n + \lambda)} \qquad W_{i \neq 0}^{(c)} = \frac{\lambda}{2(n + \lambda)}$$

where typically, $\beta = 2$ for Gaussian distributions; $0 \leq \alpha \leq 1$; $\lambda = \alpha^2(n - \kappa) - n$ with $\kappa = 3 - n$.

The steps to implement the UKF are summarized in Algorithm 2. S_k is the covariance matrix of the measurement; C_k is the cross-covariance between the state and the measurement; and K_k is the filter gain.

Algorithm 2: UKF algorithm

1 **Input:** Prior distribution mean and covariance $x(0), P_0$. The constants α, β , and κ
 2 **for** $k = 1$ to N **do**

Prediction Step:

3 Compute the Sigma Points \mathcal{X}_{k-1} using Equations (53)–(55).
 4 Propagate the Sigma Points through the discrete-time model and compute its mean and covariance

$$\mathcal{X}_{k|k-1} = f_d(\mathcal{X}_{k-1}, \mathbf{u}_k) \tag{56}$$

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2n} W_i^{(m)} \mathcal{X}_{k|k-1}^{(i)} \tag{57}$$

$$P_{k|k-1} = \sum_{i=0}^{2n} W_i^{(c)} \left(\mathcal{X}_{k|k-1}^{(i)} - \hat{\mathbf{x}}_{k|k-1} \right) \left(\mathcal{X}_{k|k-1}^{(i)} - \hat{\mathbf{x}}_{k|k-1} \right)^\top + Q \tag{58}$$

Update Step :

5 Propagate the Sigma Points through the measurement model and compute its mean and covariance

$$\mathcal{Y}_{k|k-1} = H \mathcal{X}_{k|k-1} \tag{59}$$

$$\hat{\mathbf{y}}_{k|k-1} = \sum_{i=0}^{2n} W_i^{(m)} \mathcal{Y}_{k|k-1}^{(i)} \tag{60}$$

$$S_k = \sum_{i=0}^{2n} W_i^{(c)} \left(\mathcal{Y}_{k|k-1}^{(i)} - \hat{\mathbf{y}}_{k|k-1} \right) \left(\mathcal{Y}_{k|k-1}^{(i)} - \hat{\mathbf{y}}_{k|k-1} \right)^\top + R \tag{61}$$

6 Compute the cross-covariance and the filter gain

$$C_k = \sum_{i=0}^{2n} W_i^{(c)} \left(\mathcal{X}_{k|k-1}^{(i)} - \hat{\mathbf{x}}_{k|k-1} \right) \left(\mathcal{Y}_{k|k-1}^{(i)} - \hat{\mathbf{y}}_{k|k-1} \right)^\top \tag{62}$$

$$K_k = C_k S_k^{-1} \tag{63}$$

7 Compute the state estimate and the covariance

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + K_k \left[\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1} \right] \tag{64}$$

$$P_k = P_{k|k-1} - K_k S_k K_k^\top. \tag{65}$$

8 **end**

9 **Output:** The state estimate $\hat{\mathbf{x}}_k$ and the covariance matrix P_k , for $k = 1, \dots, N$

5. Numerical Examples

In this section, EKF and UKF were implemented for each IM discrete-time model discussed above. The simulations were carried out in Matlab. For the simulations, all the initial conditions are zero; the discretization time is $T_s = 200 \mu\text{s}$; the noise covariance matrices are $\mathbf{Q} = \text{diag}\{2.12 \times 10^{-2}, 2.12 \times 10^{-2}, 10^{-6}, 10^{-6}, 10^{-3}, 9.64 \times 10^{-4}\}$ and $\mathbf{R} = \text{diag}\{1/9, 1/9\}$. The SP coefficients were selected empirically as in [40], where $\alpha = 10^{-1}$, $\beta = 2$, and $\kappa = -3$. The same noise covariance matrices were used for EKF and UKF. To test the IM models in filtering, a thousand Monte Carlo simulations were carried out, which is equivalent to measuring the same event a thousand times, in this case a direct start from a 50 Hz grid. All models in both filtering algorithms were compared in terms of execution time RMSE, where the actual system was simulated using DOPRI 5. Additionally, the maximum absolute error for each model using EKF and UKF was acquired to compare the maximum deviation from the actual value for each state, particularly at startup and after starting.

Tables 3 and 4 show the average RMSE from the Monte Carlo simulations in filtering using EKF and UKF, respectively. Both tables consider the Euler, Taylor, and Runge–Kutta IM models. From the tables, for the same IM model a considerable similarity between EKF and UKF is seen. In regard to IM models in both algorithms, there was a significant improvement in state estimation with the Taylor and Runge–Kutta models as expected. In currents and flux linkages, Taylor has less error than Runge–Kutta, due to the fact that RK models do not take into account future values of the input since they are unknown. Nevertheless, Runge–Kutta speed estimation is better than Taylor. Finally, load torque estimation is nearly the same in all discrete-time models.

On the other hand, Table 5 shows average execution time per sample of the EKF and UKF algorithms for different IM models. This analysis is of particular interest in practical applications since IMs are usually coupled with modern control techniques that rely on state estimation, among other things. Hence, a low execution time allows for carrying out other computing tasks, such as optimization in a Model Predictive Control setup [6]. From Table 5, we can observe that as the complexity of the model increases, so does the execution time when utilizing both EKF and UKF. In particular, when utilizing UKF, apart from Euler, RK 2 has the smallest execution time followed by Taylor and finally RK 4. As for EKF and UKF, clearly the former requires much less execution time than the latter, no matter which discretization technique is used. In fact, our results show that UKF requires more than half of the sampling interval during the execution of the algorithm. This could leave not enough time for other tasks. However, more research is needed in this area in a practical setup since filtering can be implemented in different computing platforms with different architectures, such as a microcontroller or a field programmable gate array.

Table 3. EKF average root mean squared error.

State	EKF			
	Euler	Taylor	RK 2	RK 4
i_s^α	0.3612	0.1977	0.2029	0.2026
i_s^β	0.3577	0.1967	0.2017	0.2013
ψ_r^α	0.0777	0.0377	0.0433	0.0433
ψ_r^β	0.0784	0.0379	0.0456	0.0456
ω_r	28.4063	27.2101	24.2762	24.5003
\mathcal{T}_l	0.1038	0.1038	0.1042	0.1042

Table 4. UKF average root mean squared error.

State	UKF			
	Euler	Taylor	RK 2	RK 4
i_s^α	0.3611	0.1978	0.2029	0.2026
i_s^β	0.3575	0.1966	0.2016	0.2012
ψ_r^α	0.0777	0.0412	0.0431	0.0429
ψ_r^β	0.0784	0.0425	0.0441	0.0443
ω_r	28.7982	28.0307	24.6992	24.8631
\mathcal{T}_l	0.1038	0.1038	0.1042	0.1042

Table 5. EKF and UKF average execution times per time step.

Method	EKF Execution Time	UKF Execution Time
Euler	6.9456×10^{-6} s	9.9939×10^{-5} s
Taylor	7.4185×10^{-6} s	1.3481×10^{-4} s
RK 2	8.0369×10^{-6} s	1.1440×10^{-4} s
RK 4	9.7486×10^{-6} s	1.4439×10^{-4} s

Tables 6 and 7 show the maximum absolute error at the startup of each state for each model using EKF and UKF algorithms, respectively, at startup. The maximum error is not symmetric, especially in the α - β components of stator currents and rotor flux linkages. This asymmetry is primarily due to the high uncertainty of the rotor angular speed. At the start, all initial conditions are zero; therefore, obtaining an accurate approximation is challenging. Notice that the maximum absolute errors are similar for all the discretization methods and both EKF and UKF. By themselves, these values only show what the maximum deviation is but not the overall behavior of each technique. However, they show what to expect in terms of the worst estimation in a single experiment. Moreover, if we also consider the average root mean squared error from Tables 3 and 4, we can conclude that the Taylor and RK methods exhibit a reduced spread around the mean when compared to the Euler method. On the other hand, Tables 8 and 9 illustrate the maximum absolute error after the startup of each state for each model using EKF and UKF algorithms, respectively. For the Euler, Taylor, and RK 2 models, EKF and UKF’s maximum absolute error are similar; in particular, UKF RK 4 is slightly less than EKF RK 4. Once the rotor speed reaches the steady state, the estimation shows better performance compared to the startup.

Table 6. Max absolute error EKF at startup.

State	EKF			
	Euler	Taylor	RK 2	RK 4
i_s^α	3.1673	3.1673	3.1673	3.1673
i_s^β	1.3279	1.3279	1.3280	1.3280
ψ_r^α	6.3224	6.3765	6.6773	6.6699
ψ_r^β	5.8992	5.8999	5.9002	5.9041
ω_r	103.6	104.44	97.951	98.636
\mathcal{T}_l	0.3806	0.3804	0.3805	0.3805

Table 7. Max absolute error UKF at startup.

State	UKF			
	Euler	Taylor	RK 2	RK 4
i_s^α	3.1673	3.1673	3.1673	3.1673
i_s^β	1.3279	1.3279	1.3280	1.3239
ψ_r^α	6.3166	6.3211	6.6758	6.4512
ψ_r^β	5.8992	5.9000	5.9001	5.8696
ω_r	103.92	105.17	98.554	99.191
\mathcal{T}_l	0.3806	0.3804	0.3805	0.3805

Table 8. Max absolute error EKF after starting.

State	EKF			
	Euler	Taylor	RK 2	RK 4
i_s^α	1.4441	1.0330	1.0395	1.0388
i_s^β	1.4868	1.0503	1.0483	1.0443
ψ_r^α	0.1287	0.0413	0.0682	0.0683
ψ_r^β	0.1297	0.0377	0.0620	0.0621
ω_r	23.818	11.507	14.465	13.498
\mathcal{T}_l	0.5516	0.5519	0.5518	0.5518

Table 9. Max absolute error UKF after starting.

State	UKF			
	Euler	Taylor	RK 2	RK 4
i_s^α	1.4440	1.0356	1.0395	0.7169
i_s^β	1.4866	1.0514	1.0484	0.5494
ψ_r^α	0.1286	0.0406	0.0682	0.0344
ψ_r^β	0.1296	0.0357	0.0620	0.0049
ω_r	23.812	11.784	14.459	3.116
\mathcal{T}_l	0.5516	0.5519	0.5518	0.0867

6. Conclusions

In this article, we formulated and compared different discretization techniques, namely, Euler. The results show that the Taylor method exhibits slightly lower RMSE compared to RK methods for stator currents and rotor flux linkages. This is mainly because, in the RK methods, it was assumed that the future values of the input were equal to the input at the current time instant. Since a direct start from the grid was simulated, the retention of the input does not fully reflect the continuous-time sinusoidal input behavior. However, when implementing these methods with an input provided by an inverter, input retention is fulfilled at every time instant. Therefore, a more accurate representation of the system is expected in practical setups.

Overall, Taylor exhibits the smallest α - β stator current and rotor flux linkage RMSE, followed by RK 4 and then RK 2. Conversely, the RK 4 model demonstrates the smallest rotor shaft speed error, followed by the RK 2 and Taylor models. The Taylor method resulted in a similar rotor shaft speed error to the Euler method under no-load conditions; however, when a load torque was applied, the Taylor method exhibited a performance similar to RK models and signals that resembled the continuous-time signals. Regarding

the load torque, all models performed similarly. These behaviors persisted as T_s varied. Also, it was demonstrated that the Euler method can only accurately represent the machine at very high sampling rates.

We also compared state estimation techniques, namely, EKF and UKF, utilizing the discrete-time models from the Euler, Taylor, and RK methods. EKF and UKF show similar RMSE; however, in terms of execution time, EKF required fewer computations. In most cases, as the complexity of the model increases, the execution time increases as well. Nonetheless, for the UKF algorithm, the RK 2 model is faster than the Taylor model. The execution time required for one iteration of the EKF and UKF utilizing the Taylor and RK methods is less than the sampling period. Therefore, these techniques are suitable for implementation in modern control techniques. In contrast, when utilizing the Euler method at high sampling rates, both filtering techniques result in an execution time per iteration that exceeds the sampling period. Finally, in terms of the maximum error, EKF and UKF are similar; therefore, both algorithms are similar in the maximum deviation from the actual state. Particularly, UKF RK 4 is slightly better than EKF RK 4.

Author Contributions: Each author played an essential role in the development of this research. Conceptualization, R.H.H., C.R. and R.C.; formal analysis, R.H.H., C.R., M.C. and R.C.; funding acquisition, M.C. and R.C.; investigation, R.H.H., C.R., M.C. and R.C.; methodology, R.H.H., C.R., M.C. and R.C.; resources, C.R. and R.C.; software, R.H.H.; validation, R.H.H., C.R., M.C. and R.C.; visualization, R.H.H., C.R., M.C. and R.C.; writing—original draft preparation, R.H.H., C.R., M.C. and R.C.; writing—review and editing, R.H.H., C.R., M.C. and R.C. All authors have read and agreed to the published version of the manuscript.

Funding: DI Regular 039.309/2023, Pontificia Universidad Católica de Valparaíso; Pontificia Universidad Católica de Valparaíso Postgraduate Research Scholarship; ANID-FONDECYT grant number: 3230398, 11201187, and 1211676; Advanced Center for Electrical and Electronic Engineering, AC3E, Basal Project FB0008, ANID, Chile.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

IM	Induction Machine
KF	Kalman Filter
EKF	Extended Kalman Filter
UKF	Unscented Kalman Filter
UT	Unscented Transform
MPC	Model Predictive Control
RMSE	Root Mean Squared Error
ZOH	Zero-Order Hold
RK	Runge–Kutta
DOPRI	Dormand–Prince
pdf	Probability Density Function
SP	Sigma Points

Appendix A. Runge–Kutta Estimate Points

In this appendix section, Runge–Kutta estimate points of the evaluation of the IM model are presented. As discussed, future estimation points of the input are not possible; therefore, input values at the actual sample instant k are taken into account instead.

Appendix A.1. RK 2

For Runge–Kutta 2, the two estimate points are

$$\mathbf{r}_1 = \begin{bmatrix} -a_1 i_{s,k}^\alpha + a_2 \psi_{r,k}^\alpha + a_3 \omega_{r,k} \psi_{r,k}^\beta + b_1 v_{s,k}^\alpha \\ -a_1 i_{s,k}^\beta + a_2 \psi_{r,k}^\beta - a_3 \omega_{r,k} \psi_{r,k}^\alpha + b_1 v_{s,k}^\beta \\ a_4 i_{s,k}^\alpha - a_5 \psi_{r,k}^\alpha - a_6 \omega_{r,k} \psi_{r,k}^\beta \\ a_4 i_{s,k}^\beta - a_5 \psi_{r,k}^\beta + a_6 \omega_{r,k} \psi_{r,k}^\alpha \\ a_7 [\psi_{r,k}^\alpha i_{s,k}^\beta - \psi_{r,k}^\beta i_{s,k}^\alpha] - a_8 \mathcal{T}_{l,k} \\ g(\mathcal{T}_{l,k}, \omega_{r,k}) \end{bmatrix} \tag{A1}$$

$$\mathbf{r}_2 = \begin{bmatrix} -a_1 (i_{s,k}^\alpha + T_s \mathbf{r}_1(1)) + a_2 (\psi_{r,k}^\alpha + T_s \mathbf{r}_1(3)) + a_3 (\omega_{r,k} + T_s \mathbf{r}_1(5)) (\psi_{r,k}^\beta + T_s \mathbf{r}_1(4)) + b_1 v_{s,k}^\alpha \\ -a_1 (i_{s,k}^\beta + T_s \mathbf{r}_1(2)) + a_2 (\psi_{r,k}^\beta + T_s \mathbf{r}_1(4)) - a_3 (\omega_{r,k} + T_s \mathbf{r}_1(5)) (\psi_{r,k}^\alpha + T_s \mathbf{r}_1(3)) + b_1 v_{s,k}^\beta \\ a_4 (i_{s,k}^\alpha + T_s \mathbf{r}_1(1)) - a_5 (\psi_{r,k}^\alpha + T_s \mathbf{r}_1(3)) - a_6 (\omega_{r,k} + T_s \mathbf{r}_1(5)) (\psi_{r,k}^\beta + T_s \mathbf{r}_1(4)) \\ a_4 (i_{s,k}^\beta + T_s \mathbf{r}_1(2)) - a_5 (\psi_{r,k}^\beta + T_s \mathbf{r}_1(4)) + a_6 (\omega_{r,k} + T_s \mathbf{r}_1(5)) (\psi_{r,k}^\alpha + T_s \mathbf{r}_1(3)) \\ a_7 [(\psi_{r,k}^\alpha + T_s \mathbf{r}_1(3)) (i_{s,k}^\beta + T_s \mathbf{r}_1(2)) - (\psi_{r,k}^\beta + T_s \mathbf{r}_1(4)) (i_{s,k}^\alpha + T_s \mathbf{r}_1(1))] - a_8 \mathcal{T}_{l,k} \\ g(\mathcal{T}_{l,k} + T_s \mathbf{r}_1(6), \omega_{r,k} + T_s \mathbf{r}_1(6)) \end{bmatrix} \tag{A2}$$

where $\mathbf{r}(i)$ is the i -th row of the vector \mathbf{r} .

Appendix A.2. RK 4

For Runge–Kutta 4, the first estimate point \mathbf{r}_1 is equal to (A1); the other three are

$$\mathbf{r}_2 = \begin{bmatrix} -a_1 \left(i_{s,k}^\alpha + \frac{T_s}{2} \mathbf{r}_1(1) \right) + a_2 \left(\psi_{r,k}^\alpha + \frac{T_s}{2} \mathbf{r}_1(3) \right) + a_3 \left(\omega_{r,k} + \frac{T_s}{2} \mathbf{r}_1(5) \right) \left(\psi_{r,k}^\beta + \frac{T_s}{2} \mathbf{r}_1(4) \right) + b_1 v_{s,k}^\alpha \\ -a_1 \left(i_{s,k}^\beta + \frac{T_s}{2} \mathbf{r}_1(2) \right) + a_2 \left(\psi_{r,k}^\beta + \frac{T_s}{2} \mathbf{r}_1(4) \right) - a_3 \left(\omega_{r,k} + \frac{T_s}{2} \mathbf{r}_1(5) \right) \left(\psi_{r,k}^\alpha + \frac{T_s}{2} \mathbf{r}_1(3) \right) + b_1 v_{s,k}^\beta \\ a_4 \left(i_{s,k}^\alpha + \frac{T_s}{2} \mathbf{r}_1(1) \right) - a_5 \left(\psi_{r,k}^\alpha + \frac{T_s}{2} \mathbf{r}_1(3) \right) - a_6 \left(\omega_{r,k} + \frac{T_s}{2} \mathbf{r}_1(5) \right) \left(\psi_{r,k}^\beta + \frac{T_s}{2} \mathbf{r}_1(4) \right) \\ a_4 \left(i_{s,k}^\beta + \frac{T_s}{2} \mathbf{r}_1(2) \right) - a_5 \left(\psi_{r,k}^\beta + \frac{T_s}{2} \mathbf{r}_1(4) \right) + a_6 \left(\omega_{r,k} + \frac{T_s}{2} \mathbf{r}_1(5) \right) \left(\psi_{r,k}^\alpha + \frac{T_s}{2} \mathbf{r}_1(3) \right) \\ a_7 \left[\left(\psi_{r,k}^\alpha + \frac{T_s}{2} \mathbf{r}_1(3) \right) \left(i_{s,k}^\beta + \frac{T_s}{2} \mathbf{r}_1(2) \right) - \left(\psi_{r,k}^\beta + \frac{T_s}{2} \mathbf{r}_1(4) \right) \left(i_{s,k}^\alpha + \frac{T_s}{2} \mathbf{r}_1(1) \right) \right] - a_8 \mathcal{T}_{l,k} \\ g \left(\mathcal{T}_{l,k} + \frac{T_s}{2} \mathbf{r}_1(6), \omega_{r,k} + \frac{T_s}{2} \mathbf{r}_1(6) \right) \end{bmatrix} \tag{A3}$$

$$\mathbf{r}_3 = \begin{bmatrix} -a_1 \left(i_{s,k}^\alpha + \frac{T_s}{2} \mathbf{r}_2(1) \right) + a_2 \left(\psi_{r,k}^\alpha + \frac{T_s}{2} \mathbf{r}_2(3) \right) + a_3 \left(\omega_{r,k} + \frac{T_s}{2} \mathbf{r}_2(5) \right) \left(\psi_{r,k}^\beta + \frac{T_s}{2} \mathbf{r}_2(4) \right) + b_1 v_{s,k}^\alpha \\ -a_1 \left(i_{s,k}^\beta + \frac{T_s}{2} \mathbf{r}_2(2) \right) + a_2 \left(\psi_{r,k}^\beta + \frac{T_s}{2} \mathbf{r}_2(4) \right) - a_3 \left(\omega_{r,k} + \frac{T_s}{2} \mathbf{r}_2(5) \right) \left(\psi_{r,k}^\alpha + \frac{T_s}{2} \mathbf{r}_2(3) \right) + b_1 v_{s,k}^\beta \\ a_4 \left(i_{s,k}^\alpha + \frac{T_s}{2} \mathbf{r}_2(1) \right) - a_5 \left(\psi_{r,k}^\alpha + \frac{T_s}{2} \mathbf{r}_2(3) \right) - a_6 \left(\omega_{r,k} + \frac{T_s}{2} \mathbf{r}_2(5) \right) \left(\psi_{r,k}^\beta + \frac{T_s}{2} \mathbf{r}_2(4) \right) \\ a_4 \left(i_{s,k}^\beta + \frac{T_s}{2} \mathbf{r}_2(2) \right) - a_5 \left(\psi_{r,k}^\beta + \frac{T_s}{2} \mathbf{r}_2(4) \right) + a_6 \left(\omega_{r,k} + \frac{T_s}{2} \mathbf{r}_2(5) \right) \left(\psi_{r,k}^\alpha + \frac{T_s}{2} \mathbf{r}_2(3) \right) \\ a_7 \left[\left(\psi_{r,k}^\alpha + \frac{T_s}{2} \mathbf{r}_2(3) \right) \left(i_{s,k}^\beta + \frac{T_s}{2} \mathbf{r}_2(2) \right) - \left(\psi_{r,k}^\beta + \frac{T_s}{2} \mathbf{r}_2(4) \right) \left(i_{s,k}^\alpha + \frac{T_s}{2} \mathbf{r}_2(1) \right) \right] - a_8 \mathcal{T}_{l,k} \\ g \left(\mathcal{T}_{l,k} + \frac{T_s}{2} \mathbf{r}_2(6), \omega_{r,k} + \frac{T_s}{2} \mathbf{r}_2(6) \right) \end{bmatrix} \tag{A4}$$

$$\mathbf{r}_4 = \begin{bmatrix} -a_1 \left(i_{s,k}^\alpha + T_s \mathbf{r}_3(1) \right) + a_2 \left(\psi_{r,k}^\alpha + T_s \mathbf{r}_3(3) \right) + a_3 \left(\omega_{r,k} + T_s \mathbf{r}_3(5) \right) \left(\psi_{r,k}^\beta + T_s \mathbf{r}_3(4) \right) + b_1 v_{s,k}^\alpha \\ -a_1 \left(i_{s,k}^\beta + T_s \mathbf{r}_3(2) \right) + a_2 \left(\psi_{r,k}^\beta + T_s \mathbf{r}_3(4) \right) - a_3 \left(\omega_{r,k} + T_s \mathbf{r}_3(5) \right) \left(\psi_{r,k}^\alpha + T_s \mathbf{r}_3(3) \right) + b_1 v_{s,k}^\beta \\ a_4 \left(i_{s,k}^\alpha + T_s \mathbf{r}_3(1) \right) - a_5 \left(\psi_{r,k}^\alpha + T_s \mathbf{r}_3(3) \right) - a_6 \left(\omega_{r,k} + T_s \mathbf{r}_3(5) \right) \left(\psi_{r,k}^\beta + T_s \mathbf{r}_3(4) \right) \\ a_4 \left(i_{s,k}^\beta + T_s \mathbf{r}_3(2) \right) - a_5 \left(\psi_{r,k}^\beta + T_s \mathbf{r}_3(4) \right) + a_6 \left(\omega_{r,k} + T_s \mathbf{r}_3(5) \right) \left(\psi_{r,k}^\alpha + T_s \mathbf{r}_3(3) \right) \\ a_7 \left[\left(\psi_{r,k}^\alpha + T_s \mathbf{r}_3(3) \right) \left(i_{s,k}^\beta + T_s \mathbf{r}_3(2) \right) - \left(\psi_{r,k}^\beta + T_s \mathbf{r}_3(4) \right) \left(i_{s,k}^\alpha + T_s \mathbf{r}_3(1) \right) \right] - a_8 \mathcal{T}_{l,k} \\ g \left(\mathcal{T}_{l,k} + T_s \mathbf{r}_3(6), \omega_{r,k} + T_s \mathbf{r}_3(6) \right) \end{bmatrix} \tag{A5}$$

where $\mathbf{r}(i)$ is the i -th row of the vector \mathbf{r} .

Appendix B. DOPRI 5

Dormand–Prince (DOPRI) belongs to the family of Runge–Kutta methods; however, it is generally employed as an adaptive method, adjusting the sampling time by comparing the error between a fourth- and fifth-order approximation, with the latter being deemed more accurate. In this particular case, given the fixed sampling time, only the fifth-order approximation is utilized.

DOPRI requires evaluating the function six times, resulting in a longer execution time compared to the methods presented earlier. However, DOPRI aims to emulate the continuous-time model; hence, the execution time is not taken into account.

The DOPRI equations are

$$x_{k+1} = x_k + T_s \left(\frac{35}{384} r_1 + \frac{500}{1113} r_3 + \frac{125}{192} r_4 - \frac{2187}{6784} r_5 + \frac{11}{84} r_6 \right) \quad (\text{A6})$$

where

$$r_1 = f(t, x) \quad (\text{A7})$$

$$r_2 = f\left(t + \frac{1}{5} T_s, x + T_s \frac{1}{5} r_1\right) \quad (\text{A8})$$

$$r_3 = f\left(t + \frac{3}{10} T_s, x + T_s \left(\frac{3}{40} r_1 + \frac{9}{40} r_2\right)\right) \quad (\text{A9})$$

$$r_4 = f\left(t + \frac{4}{5} T_s, x + T_s \left(\frac{44}{45} r_1 - \frac{56}{15} r_2 + \frac{32}{9} r_3\right)\right) \quad (\text{A10})$$

$$r_5 = f\left(t + \frac{8}{9} T_s, x + T_s \left(\frac{19372}{6561} r_1 - \frac{25360}{2187} r_2 + \frac{64448}{6561} r_3 - \frac{212}{729} r_4\right)\right) \quad (\text{A11})$$

$$r_6 = f\left(t + T_s, x + T_s \left(\frac{9017}{3168} r_1 - \frac{355}{33} r_2 + \frac{46732}{5247} r_3 + \frac{49}{176} r_4 - \frac{5103}{18656} r_5\right)\right) \quad (\text{A12})$$

References

- Novotny, D.W.; Lipo, T.A. *Vector Control and Dynamics of AC Drives*; Monographs in Electrical and Electronic Engineering; Clarendon Press: Oxford, UK, 1996.
- Sul, S. *Control of Electric Machine Drive Systems*; Wiley: Hoboken, NJ, USA, 2010. [CrossRef]
- Rawlings, J.B. *Model Predictive Control Theory and Design*; Nob Hill Pub.: Madison, WI, USA, 2009; 533p.
- Wang, L. *Model Predictive Control System Design and Implementation Using MATLAB®*; Springer: London, UK, 2009.
- Cortes, P.; Kazmierkowski, M.; Kennel, R.; Quevedo, D.; Rodriguez, J. Predictive Control in Power Electronics and Drives. *IEEE Trans. Ind. Electron.* **2008**, *55*, 4312–4324. [CrossRef]
- Pérez, J.R. *Predictive Control of Power Converters and Electrical Drives*; Wiley: Hoboken, NJ, USA, 2012.
- ONSEMI. IGBT Technologies and Applications Overview: How and When to Use an IGBT. White Paper, 2023. Document ID: TND6235/D. Available online: <https://www.onsemi.com/design/resources/technical-documentation#ZHQ9V2hpdGUGUGFwZXJzO3A9MTs=> (accessed on 12 December 2023).
- Holtz, J. Sensorless control of induction motor drives. *Proc. IEEE* **2002**, *90*, 1359–1394. [CrossRef]
- Wang, G.; Yu, Y.; Zhang, G.; Wang, B.; Yang, M.; Xu, D. Sensorless Control of Motor Drives. In *Control of Power Electronic Converters and Systems*; Elsevier: Amsterdam, The Netherlands, 2018; pp. 331–369. [CrossRef]
- Smith, A.N.; Gadoue, S.M.; Finch, J.W. Improved Rotor Flux Estimation at Low Speeds for Torque MRAS-Based Sensorless Induction Motor Drives. *IEEE Trans. Energy Convers.* **2016**, *31*, 270–282. [CrossRef]
- Pal, A.; Das, S.; Chattopadhyay, A.K. An Improved Rotor Flux Space Vector Based MRAS for Field-Oriented Control of Induction Motor Drives. *IEEE Trans. Power Electron.* **2018**, *33*, 5131–5141. [CrossRef]
- Qu, Z.; Hinkkanen, M.; Harnfors, L. Gain Scheduling of a Full-Order Observer for Sensorless Induction Motor Drives. *IEEE Trans. Ind. Appl.* **2014**, *50*, 3834–3845. [CrossRef]
- Zorgani, Y.A.; Koubaa, Y.; Boussak, M. MRAS state estimator for speed sensorless ISFOC induction motor drives with Luenberger load torque estimation. *ISA Trans.* **2016**, *61*, 308–317. [CrossRef] [PubMed]
- Zhao, L.; Huang, J.; Liu, H.; Li, B.; Kong, W. Second-Order Sliding-Mode Observer With Online Parameter Identification for Sensorless Induction Motor Drives. *IEEE Trans. Ind. Electron.* **2014**, *61*, 5280–5289. [CrossRef]
- Comanescu, M. Design and Implementation of a Highly Robust Sensorless Sliding Mode Observer for the Flux Magnitude of the Induction Motor. *IEEE Trans. Energy Convers.* **2016**, *31*, 649–657. [CrossRef]
- Söderström, T. *Discrete-Time Stochastic Systems Estimation and Control*; Springer: London, UK, 2002; 375p.
- Bertsekas, D.P.; Tsitsiklis, J.N. *Introduction to Probability*, 2nd ed.; Athena Scientific: Nashua, NH, USA, 2008; 416p.
- Stark, H.; Woods, J.W. *Probability and Random Processes with Applications to Signal Processing*; Pearson Education, Limited: London, UK, 2011; 704p.
- Jazwinski, A.H. *Stochastic Processes and Filtering Theory*; Elsevier Science and Technology Books: Amsterdam, The Netherlands, 1970; 376p.
- Haykin, S. *Kalman Filtering and Neural Networks*; Wiley-Interscience: Hoboken, NJ, USA, 2001; 304p.
- Särkkä, S. *Bayesian Filtering and Smoothing*; Cambridge University Press: Cambridge, UK, 2013; 254p.

22. Anderson, B.D.O. *Optimal Filtering*; Dover Publications: Mineola, NY, USA, 2004.
23. Simon, D. *Optimal State Estimation*; Wiley-Interscience: Hoboken, NJ, USA, 2006; 552p.
24. Candy, J.V. *Bayesian Signal Processing Classical, Modern, and Particle Filtering Methods*; John Wiley & Sons: Hoboken, NJ, USA, 2016; 640p.
25. Chen, F.; Dunnigan, M. Comparative study of a sliding-mode observer and Kalman filters for full state estimation in an induction machine. *IEE Proc.-Electr. Power Appl.* **2002**, *149*, 53. [[CrossRef](#)]
26. Zahraoui, Y.; Akherraz, M. Kalman Filtering Applied to Induction Motor State Estimation. In *Dynamic Data Assimilation-Beating the Uncertainties*; IntechOpen: London, UK; 2020. [[CrossRef](#)]
27. Singh, K.; Singh, M. Design of Kalman Filter for induction motor drive. In Proceedings of the 2013 Students Conference on Engineering and Systems (SCES), Allahabad, India, 12–14 April 2013; pp. 1–6. [[CrossRef](#)]
28. Anderson, B.; Moore, J. *Optimal Filtering*; Dover Books on Electrical Engineering; Dover Publications: New York, NY, USA, 2012.
29. Kailath, T. *Lectures on Wiener and Kalman Filtering*; CISM International Centre for Mechanical Sciences Series; Springer: Berlin/Heidelberg, Germany, 1981.
30. Kim, Y.R.; Sul, S.K.; Park, M.H. Speed sensorless vector control of induction motor using extended Kalman filter. *IEEE Trans. Ind. Appl.* **1994**, *30*, 1225–1233. [[CrossRef](#)]
31. Ramachandiran, G.; Veeranan, S.; Reddy, B. Sensorless Control of Induction Motor with Extended Kalman Filter on TMS320F2812 Processor. *Full Pap. Int. J. Recent Trends Eng.* **2009**, *2*, 14.
32. Alonge, F.; D’Ippolito, F. Extended Kalman Filter for sensorless control of induction motors. In Proceedings of the 2010 First Symposium on Sensorless Control for Electrical Drives, Padua, Italy, 9–10 July 2010. [[CrossRef](#)]
33. Basheer, O.; Obaid, M. Flux and Speed Estimation of Induction Motors using Extended Kalman Filter. *Int. J. Comput. Appl.* **2018**, *181*, 27–31. [[CrossRef](#)]
34. Et-Taaj, L.; Boulghasoul, Z.; Elkharki, A.; Kandoussi, Z.; Elbacha, A. Extended Kalman Filter for High performances Sensorless Induction Motor drive. In Proceedings of the 2019 4th World Conference on Complex Systems (WCCS), Ouarzazate, Morocco, 22–25 April 2019. [[CrossRef](#)]
35. Akin, B.; Orguner, U.; Ersak, A. State estimation of induction motor using unscented Kalman filter. In Proceedings of the 2003 IEEE Conference on Control Applications, Istanbul, Turkey, 25 June 2003; Volume 2, pp. 915–919. [[CrossRef](#)]
36. Kumar, S.; Prakash, J.; Kanagasabapathy, P. A critical evaluation and experimental verification of Extended Kalman Filter, Unscented Kalman Filter and Neural State Filter for state estimation of three phase induction motor. *Appl. Soft Comput.* **2011**, *11*, 3199–3208. [[CrossRef](#)]
37. Jafarzadeh, S.; Lascu, C.; Fadali, M.S. State Estimation of Induction Motor Drives Using the Unscented Kalman Filter. *IEEE Trans. Ind. Electron.* **2012**, *59*, 4207–4216. [[CrossRef](#)]
38. Dominguez, P.A.; Silva, C.A.; Yuz, J.I. State and Resistance Estimation in Sensorless FOC Induction Motor Drive Using a Reduced Order Unscented Kalman Filter. In Proceedings of the 2012 VI Andean Region International Conference, Cuenca, Ecuador, 7–9 November 2012. [[CrossRef](#)]
39. Rigatos, G.; Siano, P. Sensorless nonlinear control of induction motors using Unscented Kalman Filtering. In Proceedings of the IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society, Montreal, QC, Canada, 25–28 October 2012. [[CrossRef](#)]
40. Yildiz, R.; Barut, M.; Zerdali, E. A Comprehensive Comparison of Extended and Unscented Kalman Filters for Speed-Sensorless Control Applications of Induction Motors. *IEEE Trans. Ind. Inform.* **2020**, *16*, 6423–6432. [[CrossRef](#)]
41. Yan, L.; Wang, F.; Dou, M.; Zhang, Z.; Kennel, R.; Rodríguez, J. Active Disturbance-Rejection-Based Speed Control in Model Predictive Control for Induction Machines. *IEEE Trans. Ind. Electron.* **2020**, *67*, 2574–2584. [[CrossRef](#)]
42. Yan, L.; Wang, F. Observer-Predictor-Based Predictive Torque Control of Induction Machine for Robustness Improvement. *IEEE Trans. Power Electron.* **2021**, *36*, 9477–9486. [[CrossRef](#)]
43. Liu, S.; Wang, Z.; Chen, Y.; Wei, G. Protocol-Based Unscented Kalman Filtering in the Presence of Stochastic Uncertainties. *IEEE Trans. Autom. Control.* **2020**, *65*, 1303–1309. [[CrossRef](#)]
44. Kharab, A.; Guenther, R. *An Introduction to Numerical Methods*, 4th ed.; CRC Press: Boca Raton, FL, USA, 2018.
45. Mathworks. *Simulink User’s Guide*, 23.2 ed.; The Mathworks Inc.: Natick, MA, USA, 2023. Available online: https://www.mathworks.com/help/pdf_doc/simulink/simulink_ug.pdf (accessed on 12 December 2023).
46. Plexim GmbH. *PLECS—The Simulation Platform for Power Electronic Systems*, 4.7 ed.; Plexim GmbH: Zurich, Switzerland, 2023. Available online: <https://www.plexim.com/files/plecsmanual.pdf> (accessed on 12 December 2023).
47. El-Sharkawi, M. *Fundamentals of Electric Drives (Electrical Engineering)*; Brooks/Cole: San Francisco, CA, USA, 2000; 336p.
48. Silva, C.A.; Yuz, J.I. On sampled-data models for model predictive control. In Proceedings of the IECON 2010-36th Annual Conference on IEEE Industrial Electronics Society, Glendale, AZ, USA, 7–10 November 2010. [[CrossRef](#)]
49. Julier, S.; Uhlmann, J.K.; Durrant-Whyte, H. A new approach for filtering nonlinear systems. In Proceedings of the 1995 American Control Conference-ACC’95, Seattle, WA, USA, 21–23 June 1995; Volume 3, pp. 1628–1632.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.