Park, Inverse Park and Clarke, Inverse Clarke Transformations MSS Software Implementation

User Guide

Table of Contents

Introduction

The behavior of three-phase machines is usually described by their voltage and current equations. The coefficients of the differential equations that describe their behavior are time varying (except when the rotor is stationary). The mathematical modeling of such a system tends to be complex since the flux linkages, induced voltages, and currents change continuously as the electric circuit is in relative motion. For such a complex electrical machine analysis, mathematical transformations are often used to decouple variables and to solve equations involving time varying quantities by referring all variables to a common frame of reference.

Among the various transformation methods available, the well known are:

- Clarke Transformation
- Park Transformation

Clarke Transformation

This transformation converts balanced three-phase quantities into balanced two-phase quadrature quantities.

Park Transformation

This transformation converts vectors in balanced two-phase orthogonal stationary system into orthogonal rotating reference frame.

Basically, the three reference frames considered in this implementation are:

1. Three-phase reference frame, in which I_a, I_b, and I_c are co-planar three-phase quantities at an angle of 120 degrees to each other.
2. Orthogonal stationary reference frame, in which I_α (along α axis) and I_β (along β axis) are perpendicular to each other, but in the same plane as the three-phase reference frame.
3. Orthogonal rotating reference frame, in which Id is at an angle θ (rotation angle) to the α axis and I_q is perpendicular to I_d along the q axis.

Figure 1 shows the three reference frames.

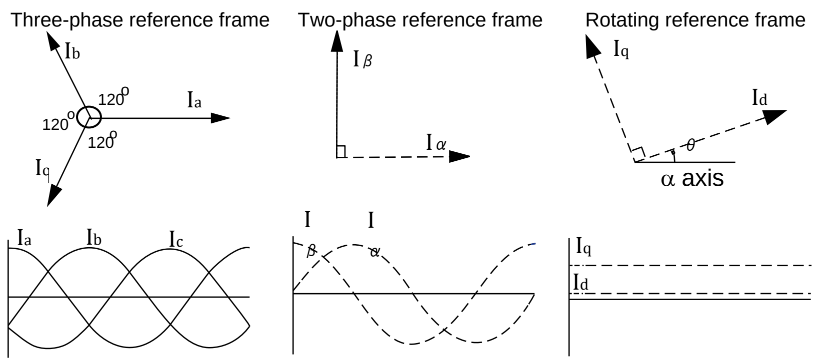

Figure 1 · Reference Frames

The combined representation of the quantities in all reference frames is shown in Figure 2.

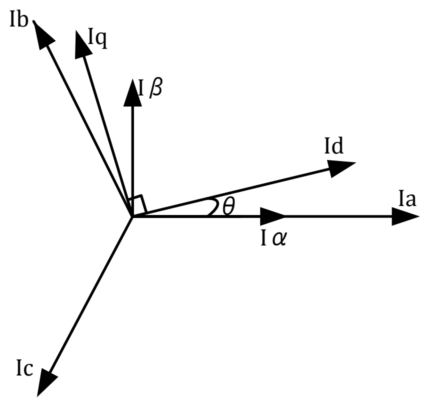

Figure 2 · Combined Vector Representation

Transformations Theory

Introduction

Clarke and Park transformations are mainly used in vector control architectures related to permanent magnet synchronous machines (PMSM) and asynchronous machines. This section explains the Park, Inverse Park and Clarke, Inverse Clarke transformations.

Clarke Transformation

The three-phase quantities are translated from the three-phase reference frame to the two-axis orthogonal stationary reference frame using Clarke transformation as shown in Figure 3.The Clarke transformation is expressed by the following equations:

$$I_\alpha = \frac{2}{3}(I_a) - \frac{1}{3}(I_b - I_c)$$

EQ1

$$I_\beta = \frac{2}{\sqrt{3}}(I_b - I_c)$$

EQ2

where,

I_a, I_b, and I_c are three-phase quantities

I_α and I_β are stationary orthogonal reference frame quantities

When I_α is superposed with Ia and $I_a + I_b + I_c$ is zero, I_a, I_b, and I_c can be transformed to I_α and I_β as:

$$I_\alpha = I_a$$

EQ3

$$I_\beta = \frac{1}{\sqrt{3}}(I_a + 2I_b)$$

EQ4

where $I_a + I_b + I_c = 0$

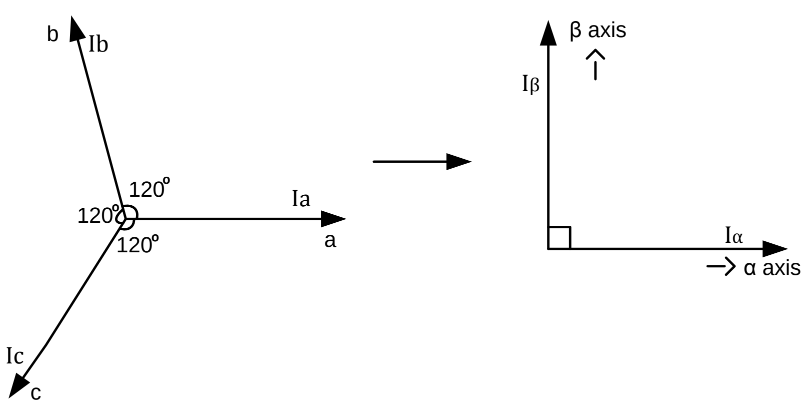

Figure 3 · Clarke Transformation

Inverse Clarke Transformation

The transformation from a two-axis orthogonal stationary reference frame to a three-phase stationary reference frame is accomplished using Inverse Clarke transformation as shown in Figure 4. The Inverse Clarke transformation is expressed by the following equations:

$$V_a = V_\alpha$$

EQ5

$$V_b = \frac{-V_\alpha + \sqrt{3} * V_\beta}{2}$$

EQ6

$$V_c = \frac{-V_\alpha - \sqrt{3} * V_\beta}{2}$$

EQ7

where,

V_a, V_b, V_c are three-phase quantities

V_α, V_β are stationary orthogonal reference frame quantities

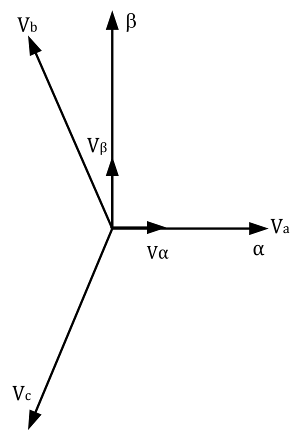

Figure 4 · Inverse Clarke Transformation

Park Transformation

The two-axis orthogonal stationary reference frame quantities are transformed into rotating reference frame quantities using Park transformation as shown in Figure 5. The Park transformation is expressed by the following equations:

$$I_d = I_\alpha * \cos(\theta) + I_\beta * \sin(\theta)$$

EQ8

$$I_q = I_\beta * \cos(\theta) - I_\alpha * \sin(\theta)$$

EQ9

where,

I_d, I_q are rotating reference frame quantities

I_α, I_β are orthogonal stationary reference frame quantities

θ is the rotation angle

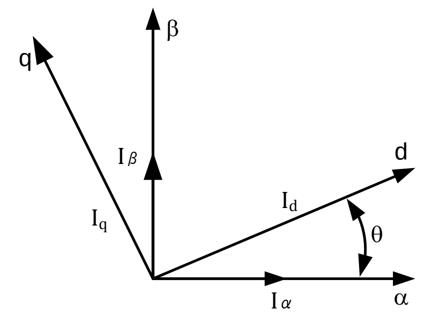

Figure 5 · Park Transformation

Inverse Park Transformation

The quantities in rotating reference frame are transformed to two-axis orthogonal stationary reference frame using Inverse Park transformation as shown in Figure 6. The Inverse Park transformation is expressed by the following equations:

$$V_\alpha = V_d * \cos(\theta) - V_q * \sin(\theta)$$

EQ10

$$V_\beta = V_q * \cos(\theta) + V_d * \sin(\theta)$$

EQ11

where,

V_α, V_β are orthogonal stationary reference frame quantities

V_d, V_q are rotating reference frame quantities

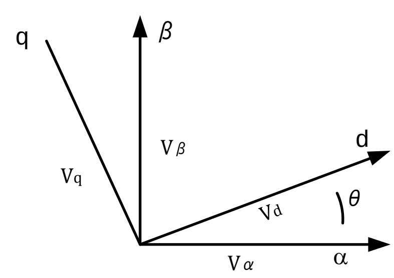

Figure 6 · Inverse Park Transformation

API Type Definitions

This section describes the type definitions used in MSS software implementation of the Park, Inverse Park and Clarke, Inverse Clarke transformation blocks.

The following are the different data types defined for inputs and outputs.

Park, Inverse Park and Clarke, Inverse Clarke Transformation Type Definitions

clarkeinput_type

Table 1 gives the type definition for the inputs of Clarke transform.

Table 1 · clarkeinput_type

Name	clarkeinput_type	
Type	typedef struct { int32_t ia; int32_t ib; int32_t ic; }clarkeinput_type;	
File	Lib.h	
Variable Description	int32_t ia	The phase A current
	int32_t ib	The phase B current
	int32_t ic	The phase C current

clarkeoutput_type

Table 2 gives the type definition for the outputs of Clarke transform.

Table 2 · clarkeoutput_type

Name	clarkeoutput_type	
Type	typedef struct { int32_t ialpha; int32_t ibeta; }clarkeoutput_type;	
File	Lib.h	
Variable Description	int32_t ialpha	The current component in stationary orthogonal reference frame on alpha axis
	int32_t ibeta	The current component in stationary orthogonal reference frame on beta axis

invclarkeinput_type

Table 3 gives the type definition for the inputs of Inverse Clarke transform.

Table 3 · invclarkeinput_type

Name	invclarkeinput_type	
Type	typedef struct { int32_t valpha; int32_t vbeta; }invclarkeinput_type;	
File	Lib.h	
Variable Description	int32_t valpha	The voltage component in stationary orthogonal reference frame (V_{α})
	int32_t vbeta	The voltage component in stationary orthogonal reference frame (V_{β})

invclarkeoutput_type

Table 4 gives the type definition for the outputs of Inverse Clarke transform.

Table 4 · invclarkeoutput_type

Name	invclarkeoutput_type	
Type	typedef struct { int32_t va; int32_t vb; int32_t vc; }invclarkeoutput_type;	
File	Lib.h	
Variable Description	int32_t va,vb,vc	The Phase Voltages in three phase stationary reference frame

parkinput_type

Table 5 gives the type definition for the inputs of Park transform.

Table 5 · parkinput_type

Name	parkinput_type	
Type	typedef struct { int32_t ialpha; int32_t ibeta; int32_t cos; int32_t sin; }parkinput_type;	
File	Lib.h	
Variable Description	int32_t ialpha	The current component in stationary orthogonal reference frame on alpha axis
	int32_t ibeta	The current component in stationary orthogonal reference frame on beta axis

	Int32_t cos	Cosine component of electrical angle
	int32_t sin	Sine component of electrical angle

parkoutput_type

Table 6 gives the type definition for the outputs of Park transform.

Table 6 · parkoutput_type

Name	parkoutput_type	
Type	typedef struct { int32_t id; int32_t iq; }parkoutput_type;	
File	Lib.h	
Variable Description	int32_t id	The direct axis current component in rotor reference frame (I_d)
	int32_t iq	The quadrature axis current component in rotor reference frame (I_q)

invparkinput_type

Table 7 gives the type definition for the inputs of Inverse Park transform.

Table 7 · invparkoutput_type

Name	invparkinput_type	
Type	typedef struct { int32_t vd; int32_t vq; int32_t cos; int32_t sin; }invparkinput_type;	
File	Lib.h	
Variable Description	int32_t vd	The direct axis voltage component in rotor reference frame (V_d)
	int32_t vq	The quadrature axis voltage component in rotor reference frame (V_q)
	Int32_t cos	Cosine component of electrical angle
	int32_t sin	Sine component of electrical angle

invparkoutput_type

Table 8 gives the type definition for the outputs of Inverse Park transform.

Table 8 · invparkoutput_type

Name	invparkoutput_type	
Type	typedef struct { int32_t valpha; int32_t vbeta; }invparkoutput_type;	
File	Lib.h	
Variable Description	int32_t valpha	The voltage component in stationary orthogonal reference frame (V_α)
	int32_t vbeta	The voltage component in stationary orthogonal reference frame (V_β)

API Functions Description

In the current implementation, Clarke transform is used to determine the real (iα) and imaginary (iβ) currents from the three phase currents. Inverse Clarke transform is used to determine the three phase voltages in stationary reference frame. Park transform is used for the transformation of real (iα) and imaginary (iβ) currents from the stationary to the moving reference frame (id, iq). Inverse Park transform determines the stationary orthogonal reference frame voltages (vα, vβ) from the moving reference frame voltages (vd, vq).

Park, Inverse Park and Clarke, Inverse Clarke Transformation

This section gives the description of various functions used in the software implementation.

Clarke_Lib_Do

Table 9 describes the Clarke_Lib_Do API.

Table 9 · Specification of API Clarke_Lib_Do

Syntax	void Clarke_Lib_Do(const clarkeinput_type *threephasecurrent_ptr, clarkeoutput_type *twophaserefcurrent_ptr)
Re-entrancy	Re-entrant
Parameters (Inputs)	threephasecurrent_ptr: Pointer to the input structure
Parameters (output)	twophaserefcurrent_ptr: Pointer to the output structure
Return	None
Algorithm Description	This function computes the following equations: $ialpha = i_a$ $ibeta = \frac{1}{\sqrt{3}}(i_a + 2 * i_b)$

InvClarke_Lib_Do

Table 10 describes the InvClarke_Lib_Do API.

Table 10 · Specification of API InvClarke_Lib_Do

Syntax	void InvClarke_Lib_Do (const invclarkeinput_type *invclarkeinput_ptr, invclarkeoutput_type *invclarkeoutput_ptr)
Re-entrancy	Re-entrant
Parameters (Inputs)	invclarkeinput_ptr: Pointer to the Inverse Clarke input structure
Parameters (output)	invclarkeoutput_ptr: Pointer to the Inverse Clarke output structure
Return	None
Algorithm Description	This function computes the following equations: $v_a = valpha$ $v_b = (-valpha + \sqrt{3} * vbeta)/2$ $v_c = (-valpha - \sqrt{3} * vbeta)/2$

Park_Lib_Do

Table 11 describes the Park_Lib_Do API.

Table 11 · Specification of API Park_Lib_Do

Syntax	void Park_Lib_Do(const parkinput_type *parkinput_ptr, parkoutput_type *parkouput_ptr)
Re-entrancy	Re-entrant
Parameters (Inputs)	parkinput_ptr: Pointer to the park input structure
Parameters (output)	parkouput_ptr: Pointer to the park output structure
Return	None
Algorithm Description	This function computes the following equations: $i_d = ialpha * \cos(\theta) + ibeta * \sin(\theta)$ $i_q = ibeta * \cos(\theta) - ialpha * \sin(\theta)$

InvPark_Lib_Do

Table 12 describes the InvPark_Lib_Do API.

Table 12 · Specification of API InvPark_Lib_Do

Syntax	void InvPark_Lib_Do (const invparkinput_type *invparkinput_ptr, invparkoutput_type *invparkoutput_ptr)
Re-entrancy	Re-entrant
Parameters (Inputs)	invparkinput_ptr: Pointer to the Inverse park input structure
Parameters (output)	invparkoutput_ptr: Pointer to the Inverse park output structure
Return	None
Algorithm Description	This function computes the following equations: $valpha = v_d * \cos(\theta) - v_q * \sin(\theta)$ $vbeta = v_q * \cos(\theta) + v_d * \sin(\theta)$

Product Support

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From North America, call **800.262.1060**
From the rest of the world, call **650.318.4460**
Fax, from anywhere in the world **408.643.6913**

Customer Technical Support Center

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Technical Support

Visit the Microsemi SoC Products Group Customer Support website for more information and support (http://www.microsemi.com/soc/support/search/default.aspx). Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on website.

Website

You can browse a variety of technical and non-technical information on the Microsemi SoC Products Group home page, at http://www.microsemi.com/soc/.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is soc_tech@microsemi.com.

My Cases

Microsemi SoC Products Group customers may submit and track technical cases online by going to My Cases.

Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support via email (soc_tech@microsemi.com) or contact a local sales office. Sales office listings can be found at www.microsemi.com/soc/company/contact/default.aspx.

ITAR Technical Support

For technical support on RH and RT FPGAs that are regulated by International Traffic in Arms Regulations (ITAR), contact us via soc_tech_itar@microsemi.com. Alternatively, within My Cases, select **Yes** in the ITAR drop-down list. For a complete list of ITAR-regulated Microsemi FPGAs, visit the ITAR web page.

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo CA 92656 USA
Within the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

50200359-0/11.13