AN205345

Coordinate Transform in Motor Control

This application note describes the coordinate transform which with the Clarke, Park, Inverse Clarke and Inverse Park transformation and describes the coordinate transform's Theory, Block, Function, Flow, Sample and Parameter in the ARM Inverter Platform.

Contents

1 Introduction

1.1 Purpose

This application note describes the coordinate transform which with the Clarke, Park, Inverse Clarke and Inverse Park transformation.

This application note describes the coordinate transform's Theory, Block, Function, Flow, Sample and Parameter in the ARM Inverter Platform.

1.2 Document Overview

The rest of document is organized as the following:

Chapter 2 explains the technical background.

Chapter 3 explains Clark Transform.

Chapter 4 explains Park Transform.

2 Technical Background

Why need use coordinate transform?

2.1 Overview

In motor control, the motor's start, stop, and speed and so on, all of them need to change the motor magnetic torque, the motor torque with armature current is described in the formulae:

DC motor: $$\mathrm{T_e} = \mathrm{C_T}\Phi\mathrm{I}. \tag{1}$$

AC motor: $$\mathrm{T_e} = \mathrm{C_T}\Phi\mathrm{I}\cos\varphi \tag{2}$$

where $\mathrm{T_e}$ is magnetic torque, $\mathrm{C_T}$ is torque modulus, I is armature current, Φ is flux, φ is rotor power factor.

As DC motor, indirectly determine and control Φ and I to control motor torque. The flux and the current have direct proportion with motor torque. Changes the current will change the motor torque. It's very simply.

But as AC motor, the motor torque control not only need I and Φ, the φ is also important. The φ will change with the rotor current frequency. The Φ come from stator current and rotor current, it will change along with the motor load change, so AC motor in dynamic running, it's controlled more difficult than DC motor.

AC motor should have the parameter relation like the DC motor.

Transvector Control could simulate AC motor into DC motor, and simply the control.

The base theory: Use the 3-phase AC motor rotating magnetic field transform into like DC motor 2-phase rotating magnetic field, then control the 2-phase current to control the torque.

3 Clarke transform

Clarke transform 's theory

3.1 Theory

3.1.1 Clarke transform theory

In motor theory, balanced 3-phase AC motor have 3 fixed windings. Through 3-phase balanced AC current ia,ib,ic will bring a rotating magnetic field Φ with the speed ω.

Figure 1. 3-phase Balanced AC Current

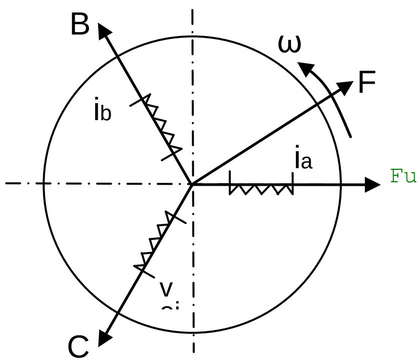


Figure 2. 2-phase Balance AC Current

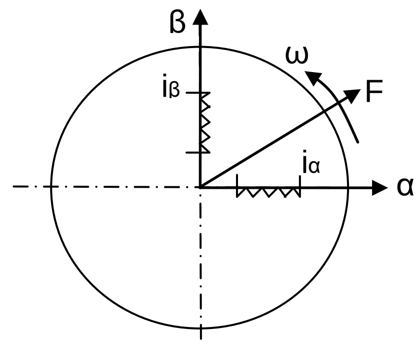
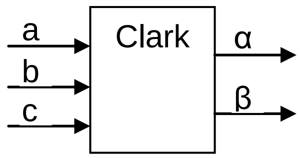


Figure 1 not only balanced 3-phase fixed windings could bring rotating magnetic field, 2-phase symmetry windings (△phasic=90°) through 2-phase balance AC current iα, iβ, also could bring rotating magnetic field. Figure 2 when balanced 3-phase fixed windings and 2-phase symmetry windings bring rotating magnetic field Φ value and speed are equality, the 3-phase windings equivalent with 2-phase windings.

Clarke transform is converts balanced 3-phase (ia,ib,ic) into balanced 2-phase(iα,iβ).Figure3

Figure 3. 3-phase Transfer to 2-phase

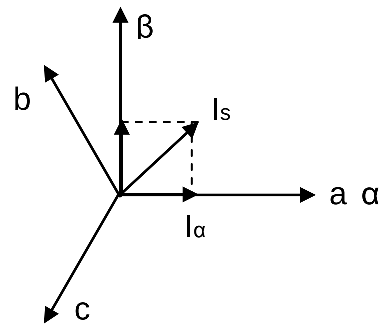


Clarke transform must keep the power fixedness, in Figure 4,

$$\begin{cases} i_\alpha = k_1\left(i_a - i_b\cos 60^\circ - i_c\cos 60^\circ\right) = k_1\left(i_a - \frac{1}{2}i_b - \frac{1}{2}i_c\right) \\ i_\beta = k_1\left(i_b\sin 60^\circ - i_c\sin 60^\circ\right) = k_1\left(\frac{\sqrt{3}}{2}i_b - \frac{\sqrt{3}}{2}i_c\right) \end{cases} \quad (3)$$

Where i_a, i_b, i_c is the 3-phase current i_α, i_β is 2-phase current, k_1 is the balanced coefficient.

Figure 4. Current with 3-phase Transfer to 2-phase



Add a zero-axis value: $i_0 = k_1 k_2 (i_a + i_b + i_c)$ (4)

Change the math formulae into motor application, have the matrix formulae

$$\begin{bmatrix} i_\alpha \\ i_\beta \\ i_0 \end{bmatrix} = k_1 \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ k_2 & k_2 & k_2 \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = C_T \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \tag{5}$$

Where:

$$C_T = k_1 \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ k_2 & k_2 & k_2 \end{bmatrix} \tag{6}$$

If the power fixedness, the formulae transform:

$$C_T{}^{-1} = k_1 \begin{bmatrix} 1 & 0 & k_2 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & k_2 \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & k_2 \end{bmatrix} \tag{7}$$

$$C_T{}^{-1} C_T = k_1^2 \begin{bmatrix} 1 & 0 & k_2 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & k_2 \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & k_2 \end{bmatrix} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ k_2 & k_2 & k_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

Then

$$k_2 = \frac{1}{\sqrt{2}}, \quad k_1 = \sqrt{\frac{2}{3}}$$

Take k_2, k_1 into the C_T

$$C_T = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \tag{9}$$

This formula is the Clarke transform matrix.

If take k_2, k_1 into the $C_T{}^{-1}$

$$C_T{}^{-1} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & 0 & \frac{1}{\sqrt{2}} \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & \frac{1}{\sqrt{2}} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & \frac{1}{\sqrt{2}} \end{bmatrix} \tag{10}$$

In motor theory, the balanced 3-phase AC motor current have:

$$\begin{cases} i_a = I\sin(\omega t) \\ i_b = I\sin(\omega t + 2\pi/3) \\ i_c = I\sin(\omega t - 2\pi/3) \end{cases} \tag{11}$$

Take $\mathrm{k}_2 = \frac{1}{\sqrt{2}}$, $\mathrm{k}_1 = \sqrt{\frac{2}{3}}$ and formula (11) into formula (5)

$$\begin{bmatrix} \mathrm{i}_\alpha \\ \mathrm{i}_\beta \\ \mathrm{i}_0 \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} I\sin(\omega t) \\ I\sin(\omega t + 2\pi/3) \\ I\sin(\omega t - 2\pi/3) \end{bmatrix}$$

Here, if define $\mathrm{i}_\alpha = \mathrm{i}_\mathrm{a}$;

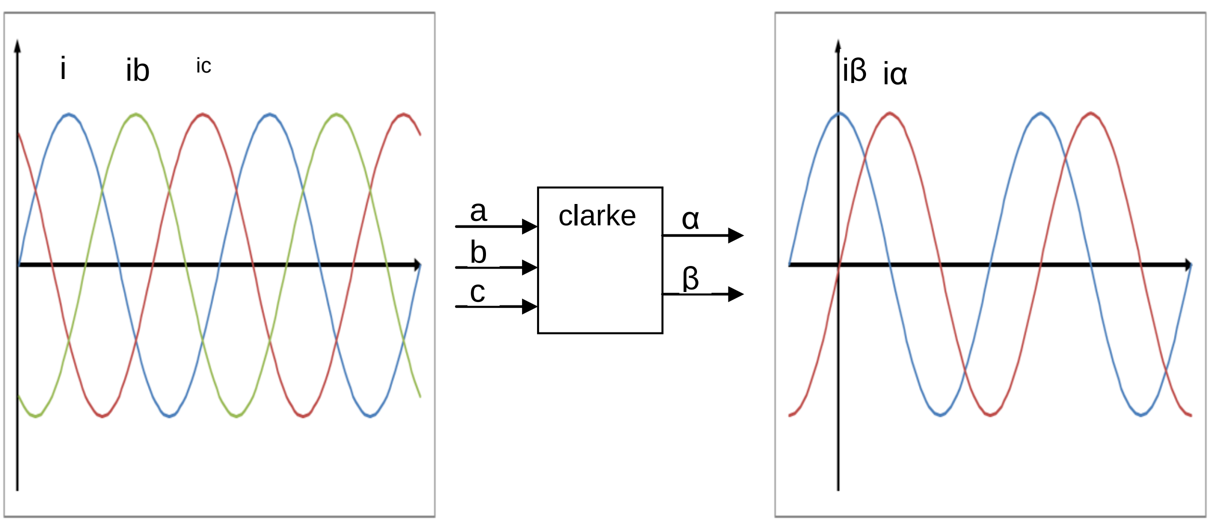
The formula can transformed to

$$\begin{cases} \mathrm{i}_\alpha = \mathrm{i}_\mathrm{a} = I\sin(\omega t) \\ \mathrm{i}_\beta = \frac{\sqrt{3}\mathrm{i}_\mathrm{a}}{3} + \frac{2\sqrt{3}\mathrm{i}_\mathrm{b}}{3} = I\sin(\omega t + \pi/2) \end{cases} \tag{12}$$

This transformation course use wave shown in Figure 5 below:

This formula is the Inverted Clarke transform matrix.

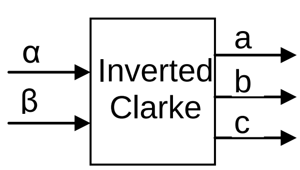
Figure 5. Current Wave with Clark Transformation Course



3.1.2 Inverted Clarke transform theory

In motor theory, when have two current component vectors in the stationary α-β axis, through complementary inverse transforms to get back to the 3-phase stationary A,B,C axis. This transformation uses the Inverse Clarke transformation, Figure 6

Figure 6. 2-phase transfer to 3-phase



Through the Inverted Park matrix C_T^{-1} and the Figure 3-4, the Inverted Clarke transform formula:

$$\begin{cases} i_a = i_\alpha \\ i_b = (-i_\alpha + \sqrt{3} i_\beta)/2 \\ i_c = (-i_\alpha - \sqrt{3} i_\beta)/2 \end{cases} \qquad (13)$$

3.2 Application

3.2.1 Function Description

```
Function Name: ClarkAmplitude
C file name:  ClarkAmplitude.C,  ClarkAmplitude.H
```

3.2.2 Function interface:

```
void ClarkAmplitude(volatile   _stDataInThreeAxis *stDataInThreeAxis,  //Inputs:  3-axis system
                               volatile    _stDataInFixAxis *stDataInFixAxis   //Outputs: fixed 2-axis system )
typedef struct
{
   Q15_VAL32 a_Q15;// phase-a  variable
   Q15_VAL32 b_Q15;// phase-b  variable
   Q15_VAL32 c_Q15;// phase-c  variable
}_stDataInThreeAxis;
_stDataInThreeAxis *stDataInThreeAxis
typedef struct
{
   Q15_VAL32 alpha_Q15;// stationary d-axis variable
   Q15_VAL32 beta_Q15;// stationary d-axis variable
}_stDataInFixAxis;
_stDataInFixAxis *stDataInFixAxis
```

Table 1. Input and Output of the Clark Transform Function

Item	Name	Description	Format
Inputs	a	phase-a of balanced 3- phase	Q15_VAL32
	b	phase-b of balanced 3- phase	Q15_VAL32
	c	phase-c of balanced 3- phase	Q15_VAL32
Outputs	alpha	alpha - alpha of fixed 2- phase	Q15_VAL32
	beta	alpha - beta of fixed 2- phase	Q15_VAL32

3.2.3 Module usage

The following code is example for this module.

```
Main()
{
}
void  example_Clarke()
{
     stDataInThreeAxis.a_Q15=INa;//input phase-a
    stDataInThreeAxis.b_Q15=INb;// input phase-b
    ClarkAmplitude(&stDataInThreeAxis, pCurrentInFixAxis);// calculate  clarke
    OUTa=CurrentInFixAxis. alpha_Q15;//Output alpha
    OUTb=CurrentInFixAxis. beta_Q15;//Output beta
}
```

4 Park transform

Park transform's theory

4.1 Theory

4.1.1 Park Transform theory

2-phase symmetry windings (△phasic=90°) through 2-phase balance **AC** current i_α,i_β, and keep the windings stop, will bring rotating magnetic EMF F1 with speed ω. Figure 4-1.

2-phase symmetry windings (△phasic=90°) through 2-phase balance **DC** current i_d,i_q, and rotate windings at speed ω, will bring rotating magnetic EMF F2, Figure 4-2.

If EMF F1=F2, stationary system 2-phase through AC current equivalent with rotating system 2-phase through DC current.

Figure 7. Current with αβaxis

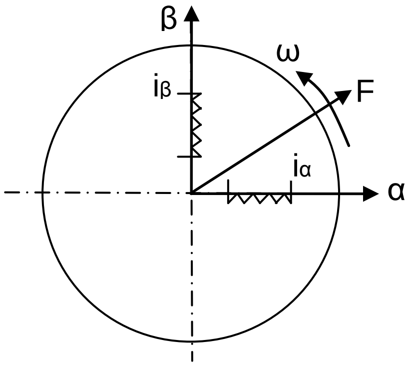
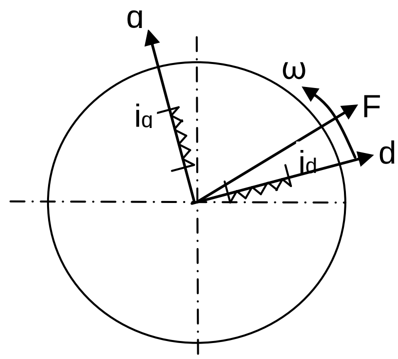
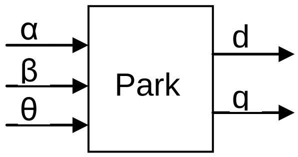


Figure 8. Current with d q axis



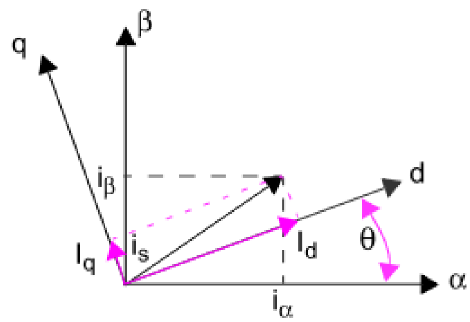
The Park transformation convert the stationary 2-phase (i_α,i_β) system into rotating 2-phase (i_d,i_q) system. Figure 9

Figure 9. Park Transform



Where the i_α,i_β come from Clarke transform; and the θ come from the rotor is displaced from the direct axis of the stator reference frame by the rotor angle θ. Because it can be seen that the angle between the real axis (x) of the general reference frame and the real axis of the reference frame rotating with the rotor is θ,

Figure 10. Current Transform with Park



As shown in Figure 4-4,

$$\begin{cases} i_d = i_\alpha \cos\theta + i_\beta \sin\theta \\ i_q = -i_\alpha \sin\theta + i_\beta \cos\theta \end{cases} \quad (14)$$

Where i_d, i_q is the rotating 2-phase current, i_α, i_β is stationary 2-phase current, θ is the angle between i_α and i_β.

Change the math formulae into motor application, have the matrix formulae:

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = C_P \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \quad (15)$$

Where:

$$C_P = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \quad (16)$$

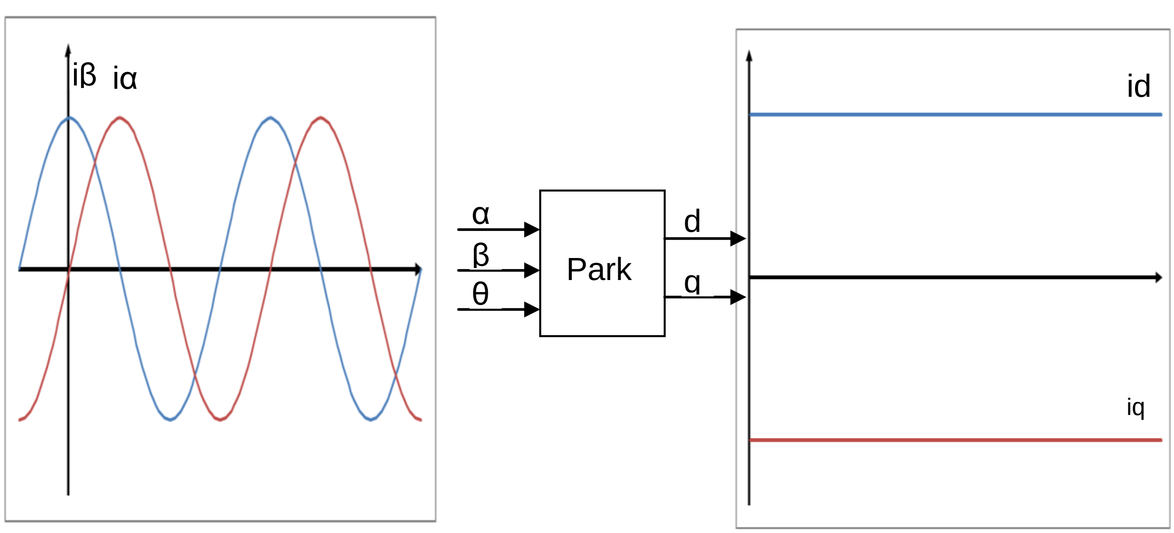
This matrix convert the stationary 2-phase(α, β) system into rotating 2-phase(d, q) system, it's called Park transform.

If the power fixedness, the formulae transform:

$$C_P^{-1} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad (17)$$

This matrix convert rotating 2-phase(d, q) system into stationary 2-phase(α, β) system, it's called Inverted Park transform. This transform course use wave shown in figure 11 below:

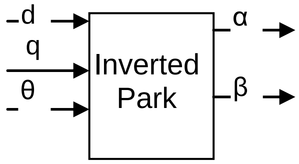
Figure 11. Current Wave with Park Transformation Course



4.1.2 Inverted Park Transform theory

In motor theory, these are two current component vectors in the rotating d-q axis, through the complementary inverse transformation to get back to the 2-phase stationary α-β axis. This transformation uses the Inverse Park Transform, Figure 11.

Figure 12. Inverted Park Transform



Through the Inverted Park matrix C_P^{-1} and the Figure 4-2, the Inverted Park formula:

$$\begin{cases} i_\alpha = i_d\cos\theta - i_q\sin\theta \\ i_\beta = i_d\sin\theta + i_q\cos\theta \end{cases} \quad (20)$$

4.2 Function Description

4.2.1 Function Name: Park`

C file name: Park.C, Park.H

Function interface:

```
void Park(_stDataInFixAxis *stDataInFixAxis,  //fixed 2-axis system
          _stDataInRotAxis *stDataInRotAxis //rotational 2-axis system)
typedef struct
{
   Q15_VAL32 alpha_Q15;
   Q15_VAL32 beta_Q15;
}_stDataInFixAxis;
_stDataInFixAxis *stDataInFixAxis
typedef struct
{
   Q15_VAL32 d_Q15;
   Q15_VAL32 q_Q15;
   Q15_VAL32 theta_Q15;
}_stDataInRotAxis;
_stDataInRotAxis *stDataInRotAxis
```

Table 2. Input and Output of the Park Transform Function

Item	Name	Description	Format
Inputs	alpha	phase- alpha of fixed 2- phase	Q15_VAL32
	beta	phase- beta of fixed 2- phase	Q15_VAL32
	Theta	Phase angle between stationary and rotating frame	Q15_VAL32
Outputs	d	alpha - alpha of rotational 2-axis system	Q15_VAL32
	q	alpha - beta of rotational 2-axis system	Q15_VAL32

4.2.2 Module usage

The following code is example for this module.

```
Main()
{
}
void  example_Park()
{
    CurrentInFixAxis. alpha_Q15= INa;//Input alpha
    CurrentInFixAxis. beta_Q15= INb;//Input beta
   stDataInRotAxis. theta_Q15=Intheta;//input Phase  angle
    Park(&CurrentInFixAxis, stDataInRotAxis);
    OUTa= stDataInRotAxis. d_Q15;
    OUTb= stDataInRotAxis. q_Q15;
}
```

4.2.3 Function Name: ParkInv

```
C file name: ParkInv.C, Park.H
Function interface:
void InvPark(_stDataInRotAxis *stDataInRotAxis, // rotating 2- phase system
       _stDataInFixAxis *stDataInFixAxis  // fixed 2- phase system)
typedef struct
{
    Q15_VAL32 d_Q15;
    Q15_VAL32 q_Q15;
    Q15_VAL32 theta_Q15;
}_stDataInRotAxis;
_ stDataInRotAxis * stDataInRotAxis
typedef struct
{
    Q15_VAL32 alpha_Q15;
    Q15_VAL32 beta_Q15;
}_stDataInFixAxis;
_stDataInFixAxis *stDataInFixAxis
```

Item	Name	Description	Format
Inputs	d	alpha - d of rotating 2- phase	Q15_VAL32
	q	alpha - q of rotating 2- phase	Q15_VAL32
	θ	Phase angle between stationary and rotating frame	Q15_VAL32
Outputs	alpha	alpha - alpha of fixed 2- phase	Q15_VAL32
	beta	alpha - beta of fixed 2- phase	Q15_VAL32

4.2.4 Module usage

The following code is example for this module.

```
Main()
{
}
void example_ParkInv()
{
     stDataInRotAxis.d_Q15=INd;//input phase-d
     stDataInRotAxis.q_Q15=INq;// input phase-q
      stDataInRotAxis.theta_Q15=Intheta;//input Phase angle
     InvPark (&stDataInRotAxis, pCurrentInFixAxis);// calculate Inverte Park
    OUTa=CurrentInFixAxis. alpha_Q15;//Output alpha
    OUTb=CurrentInFixAxis. beta_Q15;//Output beta
}
```

Document History

Document Title: AN205345 - Coordinate Transform in Motor Control

Document Number: 002-05345

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	CCTA	03/24/2011 04/07/2011 06/13/2011	Initial Release Correct some grammar error Modified the format
*A	5045216	CCTA	01/06/2016	Migrated Spansion Application Note from MCU-AN-510106-E-12 to Cypress format
*B	5709615	AESATP12	04/26/2017	Updated logo and copyright.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709