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
   1.1 Purpose ..................................................................1
   1.2 Document Overview ............................................1
2 Technical Background .....................................................1
   2.1 Overview ............................................................1
3 Clarke transform ..........................................................2
   3.1 Theory ..................................................................2
   3.2 Application ..........................................................5
4 Park transform ..............................................................6
   4.1 Theory ..................................................................6
   4.2 Function Description .............................................8

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:

\[
T_e = C_T \Phi I. \quad (1)
\]

\[
T_e = C_T \Phi I \cos \phi. \quad (2)
\]

where \(T_e\) is magnetic torque, \(C_T\) is torque modulus, \(I\) is armature current, \(\Phi\) is flux, \(\phi\) is rotor power factor.
As DC motor, indirectly determine and control $\Phi$ 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 $\Phi$, the $\varphi$ is also important. The $\varphi$ will change with the rotor current frequency. The $\Phi$ 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 $i_a, i_b, i_c$ will bring a rotating magnetic field $\Phi$ with the speed $\omega$.

```
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 ($\Delta$phasic=90°) through 2-phase balance AC current $i_\alpha, i_\beta$, also could bring rotating magnetic field. Figure 2 when balanced 3-phase fixed windings and 2-phase symmetry windings bring rotating magnetic field $\Phi$ value and speed are equality, the 3-phase windings equivalent with 2-phase windings.

Clarke transform is converts balanced 3-phase ($i_a, i_b, i_c$) into balanced 2-phase($i_\alpha, i_\beta$). Figure 3

```
Figure 3. 3-phase Transfer to 2-phase
```

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

\[
\begin{align*}
    i_\alpha &= k_1 (i_a - i_b \cos 60° - i_c \cos 60°) = k_1 (i_a - \frac{1}{2} i_b - \frac{1}{2} i_c) \\
    i_\beta &= k_1 (i_b \sin 60° - i_c \sin 60°) = k_1 \left( \frac{\sqrt{3}}{2} i_b - \frac{\sqrt{3}}{2} i_c \right)
\end{align*}
\]

Where $i_a, i_b, i_c$ is the 3-phase current, $i_\alpha, i_\beta$ is 2-phase current, $k_1$ is the balanced coefficient.

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

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

\[
\begin{bmatrix}
  i_a' \\
  i_\beta'
\end{bmatrix} = k_1 \begin{bmatrix}
  1 & -\frac{1}{2} & -\frac{1}{2} \\
  0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2}
\end{bmatrix} \begin{bmatrix}
  i_a \\
  i_b \\
  i_c
\end{bmatrix} = C_T \begin{bmatrix}
  i_a \\
  i_b \\
  i_c
\end{bmatrix}
\]

Where:

\[
C_T = k_1 \begin{bmatrix}
  1 & -\frac{1}{2} & -\frac{1}{2} \\
  0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2}
\end{bmatrix}
\]

If the power fixedness, the formulae transform:

\[
C_T^{-1} = k_1 \begin{bmatrix}
  1 & 0 \\
  -\frac{1}{2} & \frac{\sqrt{3}}{2} \\
  -\frac{1}{2} & -\frac{\sqrt{3}}{2}
\end{bmatrix}
\]

\[
C_T^{-1} C_T = k_1^2 \begin{bmatrix}
  1 & 0 \\
  -\frac{1}{2} & \frac{\sqrt{3}}{2} \\
  -\frac{1}{2} & -\frac{\sqrt{3}}{2}
\end{bmatrix} \begin{bmatrix}
  1 & -\frac{1}{2} & -\frac{1}{2} \\
  0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2}
\end{bmatrix} = \begin{bmatrix}
  1 & 0 & 0 \\
  0 & 1 & 0 \\
  0 & 0 & 1
\end{bmatrix}
\]

Then

\[
k_2 = \frac{1}{\sqrt{2}} \quad k_1 = \frac{1}{\sqrt{3}}
\]

Take \( k_2, k_1 \) into the \( C_T \)

\[
C_T = \frac{1}{\sqrt{3}} \begin{bmatrix}
  1 & -\frac{1}{2} & -\frac{1}{2} \\
  0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2}
\end{bmatrix}
\]

This formula is the Clarke transform matrix.

If take \( k_2, k_1 \) into the \( C_T^{-1} \)

\[
C_T^{-1} = \frac{1}{\sqrt{2}} \begin{bmatrix}
  1 & 0 & \frac{1}{\sqrt{2}} \\
  -\frac{1}{2} & \frac{\sqrt{3}}{2} & \frac{1}{\sqrt{2}}
\end{bmatrix}
\]

\[
-\frac{1}{2} & -\frac{\sqrt{3}}{2} & \frac{1}{\sqrt{2}}
\end{bmatrix}
\]
In motor theory, the balanced 3-phase AC motor current have:

\[
\begin{align*}
    i_a &= I \sin(\omega t) \\
    i_b &= I \sin(\omega t + 2\pi/3) \\
    i_c &= I \sin(\omega t - 2\pi/3)
\end{align*}
\]  \hfill (11)

Take \( k_2 = \frac{1}{\sqrt{2}} \), \( k_1 = \frac{\sqrt{2}}{\sqrt{3}} \) and formula (11) into formula (5)

\[
\begin{bmatrix}
    i_a \\
    i_b \\
    i_c 
\end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix}
    1 & -\frac{1}{2} & -\frac{1}{2} \\
    0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\
    \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}
\end{bmatrix} \begin{bmatrix}
    I \sin(\omega t) \\
    I \sin(\omega t + 2\pi/3) \\
    I \sin(\omega t - 2\pi/3)
\end{bmatrix}
\]

Here, if define \( i_a = i_a \); The formula can transformed to

\[
\begin{align*}
    i_a &= I \sin(\omega t) \\
    i_b &= \sqrt{\frac{2}{3}} i_a + \frac{\sqrt{2}}{3} i_c = I \sin(\omega t + \pi/2)
\end{align*}
\]  \hfill (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 \( \alpha-\beta \) 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**
Coordinate Transform in Motor Control

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

\[
\begin{align*}
    i_a &= i_x \\
    i_b &= (-i_a + \sqrt{3}i_c) / 2 \\
    i_c &= (-i_a - \sqrt{3}i_b) / 2
\end{align*}
\]  

(13)

3.2 Application

3.2.1 Function Description

**Function Name:** ClarkAmplitude

**C file name:** ClarkAmplitude.C, ClarkAmplitude.H

3.2.2 Function interface:

```c
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

<table>
<thead>
<tr>
<th>Item</th>
<th>Name</th>
<th>Description</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inputs</td>
<td>a</td>
<td>phase-a of balanced 3- phase</td>
<td>Q15 VAL32</td>
</tr>
<tr>
<td></td>
<td>b</td>
<td>phase-b of balanced 3- phase</td>
<td>Q15 VAL32</td>
</tr>
<tr>
<td></td>
<td>c</td>
<td>phase-c of balanced 3- phase</td>
<td>Q15 VAL32</td>
</tr>
<tr>
<td>Outputs</td>
<td>alpha</td>
<td>alpha - alpha of fixed 2- phase</td>
<td>Q15 VAL32</td>
</tr>
<tr>
<td></td>
<td>beta</td>
<td>alpha - beta of fixed 2- phase</td>
<td>Q15 VAL32</td>
</tr>
</tbody>
</table>
3.2.3 Module usage

The following code is example for this module.

```c
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 (\(\Delta\) phasic=90°) through 2-phase balance AC current \(i_\alpha, i_\beta\), and keep the windings stop, will bring rotating magnetic EMF \(F_1\) with speed \(\omega\). Figure 4-1.

2-phase symmetry windings (\(\Delta\) phasic=90°) through 2-phase balance DC current \(i_d, i_q\), and rotate windings at speed \(\omega\), will bring rotating magnetic EMF \(F_2\), Figure 4-2.

If EMF \(F_1=F_2\), stationary system 2-phase through AC current equivalent with rotating system 2-phase through DC current.

Figure 7. Current with \(\alpha\beta\) axis

Figure 8. Current with \(d\ q\) axis

The Park transformation convert the stationary 2-phase \((i_\alpha,i_\beta)\) system into rotating 2-phase \((i_d,i_q)\) system. Figure 9

Figure 9. Park Transform

Where the \(i_\alpha,i_\beta\) come from Clarke transform; and the \(\theta\) come from the rotor is displaced from the direct axis of the stator reference frame by the rotor angle \(\theta\). Because it can be seen that the angle between the real axis \((\alpha)\) of the general reference frame and the real axis of the reference frame rotating with the rotor is \(\theta\),
As shown in Figure 4-4,

\[
\begin{align*}
  i_d &= i_q \cos \theta + i_p \sin \theta \\
  i_q &= -i_q \sin \theta + i_p \cos \theta
\end{align*}
\]  

(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:

\[
\begin{bmatrix}
  i_d \\
  i_q \\
\end{bmatrix} = 
\begin{bmatrix}
  \cos \theta & \sin \theta \\
  -\sin \theta & \cos \theta
\end{bmatrix}
\begin{bmatrix}
  i_{α} \\
  i_{β}
\end{bmatrix} = C_P \begin{bmatrix}
  i_{α} \\
  i_{β}
\end{bmatrix}
\]  

(15)

Where:

\[
C_P = 
\begin{bmatrix}
  \cos \theta & \sin \theta \\
  -\sin \theta & \cos \theta
\end{bmatrix}
\]  

(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:

\[
C_P^{-1} = 
\begin{bmatrix}
  \cos \theta & -\sin \theta \\
  \sin \theta & \cos \theta
\end{bmatrix}
\]  

(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 \( \alpha-\beta \) axis. This transformation uses the Inverse Park Transform, Figure 11.

\[
\begin{align*}
\alpha & = i_\alpha \cos \theta - i_\beta \sin \theta \\
\beta & = i_\beta \sin \theta + i_\alpha \cos \theta
\end{align*}
\]  

(20)

Through the Inverted Park matrix \( C_p^{-1} \) and the Figure 4-2, the Inverted Park formula:

4.2 Function Description

4.2.1 Function Name: Park

C file name: Park.C, Park.H

Function interface:

```c
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

<table>
<thead>
<tr>
<th>Item</th>
<th>Name</th>
<th>Description</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inputs</td>
<td>alpha</td>
<td>phase- alpha of fixed 2- phase</td>
<td>Q15_VAL32</td>
</tr>
<tr>
<td></td>
<td>beta</td>
<td>phase- beta of fixed 2- phase</td>
<td>Q15_VAL32</td>
</tr>
<tr>
<td></td>
<td>Theta</td>
<td>Phase angle between stationary and rotating frame</td>
<td>Q15_VAL32</td>
</tr>
<tr>
<td>Outputs</td>
<td>d</td>
<td>alpha - alpha of rotational 2-axis system</td>
<td>Q15_VAL32</td>
</tr>
<tr>
<td></td>
<td>q</td>
<td>alpha - beta of rotational 2-axis system</td>
<td>Q15_VAL32</td>
</tr>
</tbody>
</table>

4.2.2 Module usage
The following code is example for this module.

```c
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
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
```
### Coordinate Transform in Motor Control

<table>
<thead>
<tr>
<th>Item</th>
<th>Name</th>
<th>Description</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inputs</td>
<td>d</td>
<td>alpha - d of rotating 2-phase</td>
<td>Q15_VAL32</td>
</tr>
<tr>
<td></td>
<td>q</td>
<td>alpha - q of rotating 2-phase</td>
<td>Q15_VAL32</td>
</tr>
<tr>
<td></td>
<td>θ</td>
<td>Phase angle between stationary and rotating frame</td>
<td>Q15_VAL32</td>
</tr>
<tr>
<td>Outputs</td>
<td>alpha</td>
<td>alpha - alpha of fixed 2-phase</td>
<td>Q15_VAL32</td>
</tr>
<tr>
<td></td>
<td>beta</td>
<td>alpha - beta of fixed 2-phase</td>
<td>Q15_VAL32</td>
</tr>
</tbody>
</table>

#### 4.2.4 Module usage

The following code is an example for this module.

```c
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

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

All other trademarks or registered trademarks referenced herein are the property of their respective owners.