dq0 to abc

SD030 | Posted on June 24, 2021 | Updated on May 27, 2025

Julien ORSINGER
Power Applications Specialist
imperix

Table of Contents

The “dq0 to abc” block computes a three-phase (*abc*) signal from a space vector in a rotating reference frame (*dq0*). The angle of the rotating reference frame is given by the second input $\theta = \omega t$.

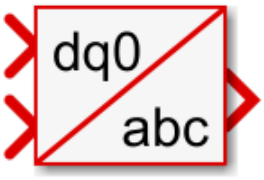
The transformation is performed by applying successively a *dq0 to alpha-beta-0*, and an *alpha-beta-0 to abc* transformation:

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ -1/2 & \sqrt{3}/2 & 1 \\ -1/2 & -\sqrt{3}/2 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} V_d \\ V_q \\ V_0 \end{bmatrix}$$

Simulink block

Signal specification

- The first input is a vector of dimension 3, containing the *dq0* components of the three-phase signal in the rotating reference frame.
- The second input is the angle θ of the rotating reference frame, in radians.
- The output is a vector of dimension 3, containing the *abc* components of the three-phase signal.



Parameters

None.

PLECS block

None. The PLECS block *Transformation RRF->3ph* can be used instead.

C++ functions

The user template located in the installation folder of CPP SDK contains an API folder with implementations of the coordinate transformation functions. The *dq0 to abc* function is the following:

```
void DQ02abc(TimeDomain *physical, const SpaceVector *rotating, const float theta);
```
Code language: C++ (cpp)

Parameters

- `physical`: pointer on the time domain *abc* data that will be updated. The `TimeDomain` structure is defined below.
- `rotating`: pointer on the *dq0* space vector that will be transformed. The `SpaceVector` structure is defined below.
- theta: the angle of the rotating reference frame, in radians.

```
typedef struct{
        float real;       // d-axis component
        float imaginary;  // q-axis component
        float offset;     // homopolar component
} SpaceVector;

typedef struct{
        float A;
        float B;
        float C;
} TimeDomain;
```
Code language: C++ (cpp)