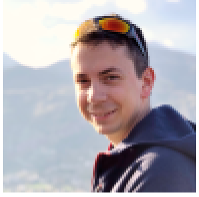
Alpha-Beta-Zero to abc

SD032 | Posted on June 24, 2021 | Updated on May 27, 2025



Julien ORSINGER
Power Applications Specialist
imperix • in

Table of Contents

The “Alpha-Beta-Zero to abc” block computes a three-phase (*abc*) signal from a space vector in a stationary reference frame (*αβ0*).

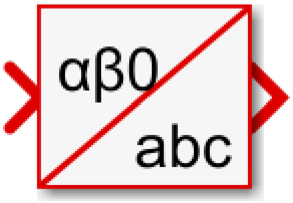
The transformation is performed using the following equation:

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ -1/2 & \sqrt{3}/2 & 1 \\ -1/2 & -\sqrt{3}/2 & 1 \end{bmatrix} \cdot \begin{bmatrix} V_\alpha \\ V_\beta \\ V_0 \end{bmatrix}$$

Simulink block

Signal specification

- The input is a vector of dimension 3, containing the *αβ0* components of the three-phase signal in the stationary reference frame.
- 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 SRF->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 *$\alpha\beta 0$ to abc* function is the following:

```
void ABG2abc(TimeDomain *physical, const SpaceVector *fixed);
```

Code language: C++ (cpp)

Parameters

- `physical`: pointer on the time domain *abc* data that will be updated. The `TimeDomain` structure is defined below.
- `fixed`: pointer on the *$\alpha\beta 0$* space vector that will be transformed. The `SpaceVector` structure is defined below.

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

typedef struct{
        float A;
        float B;
        float C;
} TimeDomain;
```

Code language: C++ (cpp)