

abc to Alpha-Beta-Zero

SD031 | Posted on June 24, 2021 | Updated on May 27, 2025



Julien ORSINGER

Power Applications Specialist

imperix • in

Table of Contents

- [Simulink block](#)
 - [Signal specification](#)
 - [Parameters](#)
- [PLECS block](#)
- [C++ functions](#)

The “abc to Alpha-Beta-Zero” block computes the coordinates of a three-phase (abc) signal in a stationary reference frame ($\alpha\beta 0$).

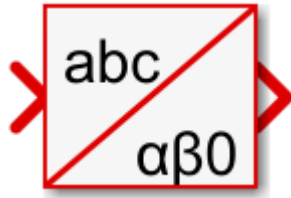
The transformation is performed using the following equation:

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

Simulink block

Signal specification

- The first input is a vector of dimension 3, containing the abc components of the three-phase signal.
- The output is a vector of dimension 3, containing the $\alpha\beta 0$ components of the three-phase signal in the stationary reference frame.



Parameters

None.

PLECS block

None. The PLECS block *Transformation 3ph->SRF* 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 *abc to αβ0* function is the following:

```
void abc2ABG(SpaceVector *fixed, const TimeDomain *physical);
```

Code language: C++ (cpp)

Parameters

- *fixed*: pointer on the *αβ0* space vector that will be updated. The *SpaceVector* structure is defined below.
- *physical*: pointer on the time domain *abc* data that will be transformed. The *TimeDomain* 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)
```