

```
function dcm = LatLongAttToDCM(lat, lon)
% Convert euler angles to DCM
C_NE = [0 0 1; 0 1 0; -1 0 0];
C_NdN = [cos(lat) 0 sin(lat); 0 1 0; -sin(lat) 0 cos(lat)];
C_NNd = [1 0 0; 0 cos(lon) -sin(lon); 0 sin(lon) cos(lon)];

% Convert DCM
dcm = C_NNd * C_NdN * C_NE;




end
```

```
function dcm = eulerToDCM(yaw, pitch, roll)
% Convert euler angles to DCM angles in Radians.
C_Y = [cos(yaw) -sin(yaw) 0; sin(yaw) cos(yaw) 0; 0 0 1];
C_P = [cos(pitch) 0 sin(pitch); 0 1 0; -sin(pitch) 0 cos(pitch)];
C_R = [1 0 0; 0 cos(roll) -sin(roll); 0 sin(roll) cos(roll)];

% Convert DCM
dcm = C_R * C_P * C_Y;

end
```

Sample Times for 'translationalkinematics'

Color	Annotation	Description	Value
	Cont	Continuous	0
	FiM	Fixed in Minor Step	[0,1]
	Inf	Constant	Inf