

# Continuous to Discrete Transfer Function Transformation Using the Euler Methods

*MATLAB Implementation*

---

Tamas Kis | [kis@stanford.edu](mailto:kis@stanford.edu)

TAMAS KIS  
<https://github.com/tamaskis>

Copyright © 2021 Tamas Kis

*Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:*

*The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.*

*THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.*



# Contents

<b>c2d_euler</b>	<b>4</b>
Syntax . . . . .	4
Description . . . . .	4
Examples . . . . .	4
Links . . . . .	5
<b>Continuous to Discrete Transformations Using the Euler Methods</b>	<b>6</b>
<b>References for Code</b>	<b>7</b>
<b>References</b>	<b>7</b>

## c2d\_euler

---

Transforms a continuous transfer function to a discrete transfer function using the forward and backward Euler methods.

### Syntax

---

```
Hz = c2d_euler(Hs,T,'forward')
Hz = c2d_euler(Hs,T,'backward')
```

### Description

---

`Hz = c2d_euler(Hs,T,'forward')` returns the discrete transfer function `Hz` obtained by applying the forward Euler (i.e. forward difference) transformation to a continuous transfer function `Hs`, where `T` is the sampling period.

`Hz = c2d_euler(Hs,T,'backward')` returns the discrete transfer function `Hz` obtained by applying the backward Euler (i.e. backward difference) transformation to a continuous transfer function `Hs`, where `T` is the sampling period.

### Examples

---

#### Example 1

Consider the following continuous transfer function:

$$H(s) = \frac{s+1}{0.1s+1}$$

Find the corresponding discrete transfer function  $H(z)$  using a sampling period of  $T = 0.25$  s. Use both the forward Euler and backward Euler transformations.

#### ■ SOLUTION

Defining  $H(s)$  as a transfer function object,

```
s = tf('s');
Hs = (s+1)/(0.1*s+1);
```

Applying the forward Euler transformation,

```
Hz_forward = c2d_euler(Hs,T,'forward')
```

This results in

```
Hz_forward =

    20 z - 15
    -----
     2 z + 3
```

```
Sample time: 0.25 seconds
Discrete-time transfer function.
```

Applying the backward Euler transformation,

```
Hz_backward = c2d_euler(Hs,T,'backward')
```

This results in

```
Hz_backward =
```

$$\frac{25z - 20}{7z - 2}$$

```
Sample time: 0.25 seconds
Discrete-time transfer function.
```

---

## Links

MATLAB<sup>®</sup> Central's File Exchange:

GitHub<sup>®</sup>:

[https://github.com/tamaskis/c2d\\_euler-MATLAB](https://github.com/tamaskis/c2d_euler-MATLAB)

## Continuous to Discrete Transformations Using the Euler Methods

---

Consider a continuous transfer function  $H(s)$  that we wish to transform to a discrete transfer function  $H(z)$  using a sampling period of  $T$  and some approximation method (i.e. some approximate transformation). Both the **forward Euler** and **backward Euler** are transformations based on evaluating  $H(s)$  at some value of  $s$  representing an *approximate* transformation between  $s$  and  $z$  [1].

$$\textbf{Forward Euler:} \quad s \leftarrow \frac{z-1}{T} \quad \therefore H(z) = H(s)|_{s=\frac{z-1}{T}}$$

$$\textbf{Backward Euler:} \quad s \leftarrow \frac{z-1}{Tz} \quad \therefore H(z) = H(s)|_{s=\frac{z-1}{Tz}}$$

## References for Code

---

**Syms to TF Conversion (`syms2tf.m`) [4]:**

- Conversion of a symbolic function object to a transfer function object.

**SYM2TF: a matlab function which converts symbolic math rationals to transfer function object [3]:**

- Conversion of a symbolic function object to a transfer function object.

**How can I convert a transfer function object from the Control System Toolbox into a symbolic object for use with the Symbolic Math Toolbox? [2]:**

- Conversion of a transfer function object to a symbolic function object.

## References

---

- [1] Gene F. Franklin, J. David Powell, and Michael Workman. *Digital Control of Dynamic Systems*. 3<sup>rd</sup>. Half Moon Bay, CA: Ellis-Kagle Press, 1998.
- [2] *How can I convert a transfer function object from the Control System Toolbox into a symbolic object for use with the Symbolic Math Toolbox?* MATLAB Answers. <https://www.mathworks.com/matlabcentral/answers/96275-how-can-i-convert-a-transfer-function-object-from-the-control-system-toolbox-into-a-symbolic-object>. (accessed: April 10, 2021).
- [3] Ichiro Maruta. *SYM2TF: a matlab function which converts symbolic math rationals to transfer function object*. GitHub Gist. <https://gist.github.com/maruta/1035254>. (accessed: April 10, 2021).
- [4] Crystal Nassouri. *Syms to TF Conversion*. MATLAB Central File Exchange. <https://www.mathworks.com/matlabcentral/fileexchange/27302-syms-to-tf-conversion>. (accessed: April 10, 2021).