EECS 16B Designing Information Devices and Systems II
Fall 2020 UC Berkeley

Note 15

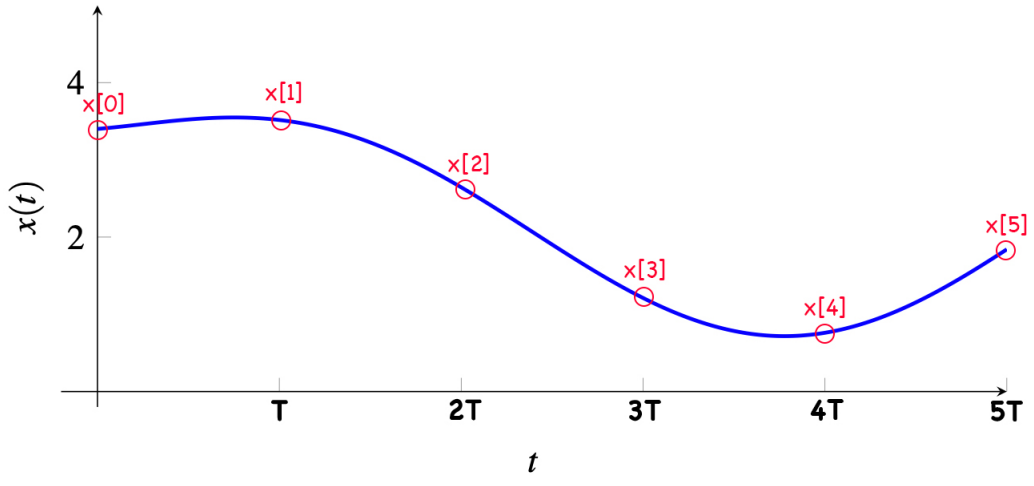
1 Overview

In our journey of understanding how to analyze and control real world sytems, we first saw how to represent a system in state-space representation. We then realized that many systems in the universe are non-linear, but we can create a linear approximation around a small neighborhood around its equilibrium point. We also saw that there were two types of systems: **Continuous-Time Systems** and **Discrete-Time Systems.** While many systems are represented using continuous-time systems, in order to control our systems, we will need a way to represent and understand this system in discrete-time. This is because computers run in discrete time.

In this note, we will learn how to take a continuous-time system and feed in piecewise constant inputs to create an equivalent discrete-time representation of the system. This process is called **discretization** and is a key technique to understanding and building digital control systems.

2 Sampling

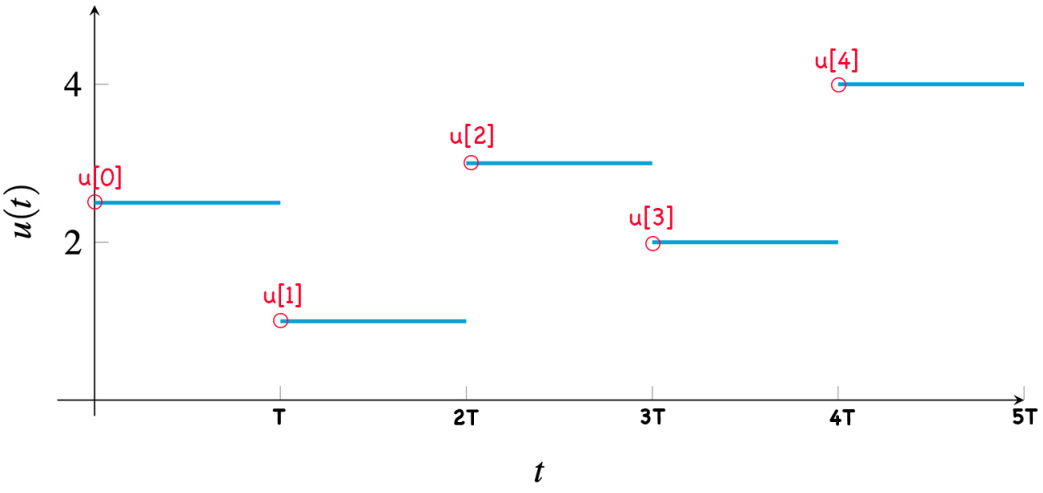
Before we dive into discretization, we first have to understand how to sample from a continuous-time system. Given a continuous function $x(t)$ that varies over time, one way to sample this function is to take a measurement at time t every T seconds.



Ideally, we would like our samples to be equally spaced and from this discrete sequence of samples: $\{x(0), x(T), x(2T), \ldots\}$ we could represent this sampled function in discrete time through the following conversion

$$x[n] = x(nT) \tag{1}$$

Now if we feed in a piecewise-constant input $u(t)$ for into our system, we see that discretizing this input gives the following intepretation:



With all of this in mind, we will now move onto discretizing our continuous time system.

3 Scalar Discretization

Suppose we had a system represented by the scalar differential equation

$$\frac{d}{dt}x(t) = \lambda(x(t) - u(t)) \tag{2}$$

If the input were constant, then recall that the solution to the differential equation is of the form

$$x(t) = Ae^{\lambda t} + B \tag{3}$$

To discretize a system, recall from the previous section that we feed in a piecewise constant input $u(t)$. This means that the input will remain constant over the region $[nT, (n+1)T)$.

Remembering that our goal is to come up with a discrete analog of the system,

$$x[n+1] = \alpha x[n] + \beta u[n] \tag{4}$$

let us take a look at a mathematical example of how this will work.

Suppose we have the a system represented by the scalar differential equation from (2). If we look at the region for $t \in [0, T)$ the input $u(t) = u[0]$. Solving the differential equation for $t \in [0, T)$, it follows that

$$x(t) = x(0)e^{\lambda t} + u[0](1 - e^{\lambda t}) \tag{5}$$

Under the hood, the function $x(t)$ is moving in the interval $[0, T)$. However, the discrete-time system only sees a snapshot of what happens at $t = 0$ and $t = T$. Plugging in $t = T$ into our solution above, we see that

$$x(T) = x[1] = e^{\lambda T}x[0] + (1 - e^{\lambda T})u[0] \tag{6}$$

Note how this resembles the equation $x[1] = \alpha x[0] + \beta u[0]$ for $\alpha = e^{\lambda T}$ and $\beta = 1 - e^{\lambda T}$!

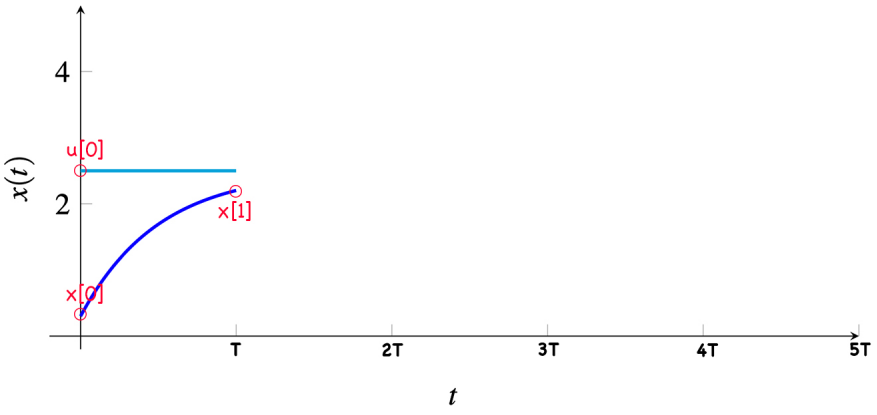


Figure 1: First step of discretization assuming $\lambda < 0$.

We will now generalize this for any $n \in \mathbb{N}$.

If we look at the window for $t \in [nT, nT+T)$, the input remains constant at $u(t) = u[n]$. The initial condition of the system is $x(nT) = x[n]$. We could then solve the differential equation to get the following solution[1]

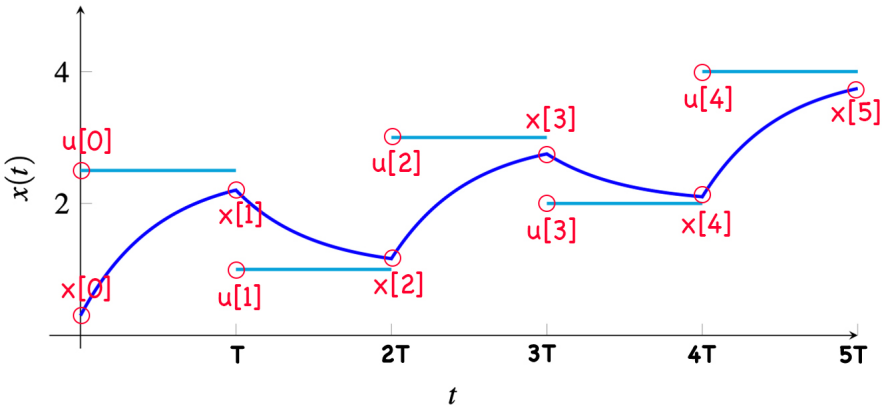
$$x(t) = e^{-\lambda nT} x[n] e^{\lambda t} + (e^{-\lambda nT}) u[n] e^{\lambda t} + u[n] \tag{7}$$

Plugging in $t = nT + T$ into our solution above, we see that

$$x(nT+T) = x[n+1] = e^{\lambda T} x[n] + (1 - e^{\lambda T}) u[n] \tag{8}$$

Which shows that our discretized system is indeed $x[n+1] = \alpha x[n] + \beta u[n]$ for $\alpha = e^{\lambda T}$ and $\beta = 1 - e^{\lambda T}$!

We show another visualization below from $n = 0$ to $n = 4$



[1] Probably easiest to do through Guess and Check.

4 Vector Discretization

We will now take everything we learned about discretizing a scalar system and apply it to the state-space system represented by

$$\frac{d}{dt}\vec{x}(t) = A\vec{x}(t) + B\vec{u}(t) \tag{9}$$

Similar to how we solved our vector differential equations, we will assume A is diagonalizable and change coordinates to the eigenbasis. This way we can discretize our individual scalar differential equations. Let V be a matrix of the eigenvectors of A. Then defining the change of coordinates $\vec{z} = V^{-1}\vec{x}$ we see that

$$\frac{d}{dt}\vec{z}(t) = \Lambda\vec{z}(t) + V^{-1}B\vec{u}(t) \tag{10}$$

A short derivation can show that the scalar differential equation

$$\frac{d}{dt}x(t) = \lambda x(t) + bu(t) \tag{11}$$

can be discretized as the following system.[2]

$$x[n+1] = \alpha x[n] + \beta u[n] \tag{12}$$

where $\alpha = e^{\lambda T}$ and $\beta = b\frac{e^{\lambda T}-1}{\lambda}$ in the case $\lambda = 0$, we can show that $\beta = bT$. We will tackle this case in a later section. In any case, given the vector differential equation in a diagonal basis, we can discretize each scalar differential equation[3]

$$\begin{aligned}
\frac{d}{dt}z_1(t) &= \lambda_1 z_1(t) + (V^{-1}B\vec{u})_1(t) \\
&\vdots \\
\frac{d}{dt}z_1(t) &= \lambda_1 z_1(t) + (V^{-1}B\vec{u})_n(t)
\end{aligned}$$

and it follows that

$$\vec{z}[n+1] = \begin{bmatrix} e^{\lambda_1 T} & & \\ & \ddots & \\ & & e^{\lambda_n T} \end{bmatrix}\vec{z}[n] + \begin{bmatrix} \frac{e^{\lambda_1 T}-1}{\lambda_1} & & \\ & \ddots & \\ & & \frac{e^{\lambda_n T}-1}{\lambda_n} \end{bmatrix} V^{-1}B\vec{u}[n] \tag{13}$$

After converting our coordinates back to the standard basis our discretized system is

$$\vec{x}[n+1] = A_d\vec{x}[n] + B_d\vec{u}[n] \tag{14}$$

where

$$A_d = V\begin{bmatrix} e^{\lambda_1 T} & & \\ & \ddots & \\ & & e^{\lambda_n T} \end{bmatrix} V^{-1} \qquad B_d = V\begin{bmatrix} \frac{e^{\lambda_1 T}-1}{\lambda_1} & & \\ & \ddots & \\ & & \frac{e^{\lambda_n T}-1}{\lambda_n} \end{bmatrix} V^{-1}B \tag{15}$$

[2] In the previous section, $b = -\lambda$

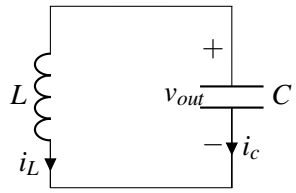
[3] the notation $(V^{-1}B\vec{u})_i$ stands for the i^{th} entry of the vector $V^{-1}B\vec{u}$

5 Examples

We'll now take a look at a couple of physical examples of discretization. We'll also take a look at an example where $\lambda = 0$.

5.1 Circuit Example

Recall the LC Tank from the Circuits Module where $L = 10\,\text{nH}$ and $C = 10\,\text{pF}$.



The differential equation representing this system was

$$\frac{d}{dt}\begin{bmatrix} v_{out} \\ i_L \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{C} \\ \frac{1}{L} & 0 \end{bmatrix}\begin{bmatrix} v_{out} \\ i_L \end{bmatrix} \tag{16}$$

Finding the eigenvectors and eigenvalues of this system, we put change coordinates into a diagonal system.

$$\frac{d}{dt}\begin{bmatrix} \widetilde{v}_{out} \\ \widetilde{i}_L \end{bmatrix} = \begin{bmatrix} j\frac{1}{\sqrt{LC}} & 0 \\ 0 & -j\frac{1}{\sqrt{LC}} \end{bmatrix}\begin{bmatrix} \widetilde{v}_{out} \\ \widetilde{i}_L \end{bmatrix},\ V = \begin{bmatrix} j\sqrt{\frac{L}{C}} & -j\sqrt{\frac{L}{C}} \\ 1 & 1 \end{bmatrix} \tag{17}$$

There is no input into this system, but we can still discretize it as

$$\begin{bmatrix} \widetilde{v}_{out}[n+1] \\ \widetilde{i}_L[n+1] \end{bmatrix} = \begin{bmatrix} e^{j\frac{1}{\sqrt{LC}}T} & 0 \\ 0 & e^{-j\frac{1}{\sqrt{LC}}T} \end{bmatrix}\begin{bmatrix} \widetilde{v}_{out}[n] \\ \widetilde{i}_L[n] \end{bmatrix} \tag{18}$$

Changing coordinates back to the standard-basis, it follows that

$$\begin{bmatrix} v_{out}[n+1] \\ i_L[n+1] \end{bmatrix} = \begin{bmatrix} 10\sqrt{10}j & -10\sqrt{10}j \\ 1 & 1 \end{bmatrix}\begin{bmatrix} e^{j\frac{1}{\sqrt{LC}}T} & 0 \\ 0 & e^{-j\frac{1}{\sqrt{LC}}T} \end{bmatrix}\begin{bmatrix} 10\sqrt{10}j & -10\sqrt{10}j \\ 1 & 1 \end{bmatrix}^{-1}\begin{bmatrix} v_{out}[n] \\ i_L[n] \end{bmatrix} \tag{19}$$

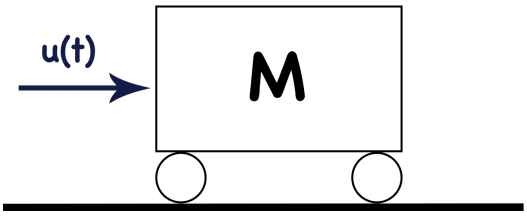
Simplifying our expression, we see that

$$\begin{bmatrix} v_{out}[n+1] \\ i_L[n+1] \end{bmatrix} = \begin{bmatrix} \cos(\omega T) & -10\sqrt{10}\sin(\omega T) \\ \frac{1}{10\sqrt{10}}\sin(\omega T) & \cos(\omega T) \end{bmatrix}\begin{bmatrix} v_{out}[n] \\ i_L[n] \end{bmatrix} \tag{20}$$

for $\omega = \frac{1}{\sqrt{LC}}$. Try plotting this result on a computer and observe how the sampling rate T affects your discretization.

5.2 Car Example

Consider a car with mass M that is pushed by an input force $u(t)$.



We can represent the system by using the states $p(t)$ and $v(t)$ representing the position and velocity of the car.

$$\frac{d}{dt}\begin{bmatrix} p(t) \\ v(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} p(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M} \end{bmatrix} \tag{21}$$

While the matrix A representing the system is non-diagonalizable, we can take the same approach as we did when solving vector differential equations and break up the system into two differential equations

$$\frac{d}{dt}p(t) = v(t) \qquad \frac{d}{dt}v(t) = \frac{u(t)}{M} \tag{22}$$

Since $p(t)$ depends on $v(t)$, we will solve for $v(t)$ first. Note how $\lambda = 0$ in this case. Since $u(t)$ is constant in the interval $[nT, nT+T)$, we use the Fundamental Theorem of Calculus [4]

$$\int_{nT}^{t} dv = \int_{nT}^{t} \frac{u[n]}{M} d\tau \tag{23}$$

$$v(t) - v(nT) = \frac{u[n]}{M} \cdot (t - nT) \implies v[n+1] = v[n] + \frac{T}{M}u[n] \tag{24}$$

Now that we've discretized $v(t)$, we can plug our solution back into the differential equation for $p(t)$.

$$\int_{nT}^{t} dp = \int_{nT}^{t} \left(v[n] + \frac{u[n]}{M}(\tau - nT) \right) d\tau \tag{25}$$

$$p((n+1)T) - p(nT) = v[n] \cdot (t - nT) + \frac{u[n]}{2M}(t^2 - (nT)^2) - nT(t - nT) \tag{26}$$

$$\implies p[n+1] = p[n] + v[n] \cdot T + \frac{T^2}{2M}u[n] \tag{27}$$

To summarize, we aggregate our results into discrete-time vector system:

$$\begin{bmatrix} p[n+1] \\ v[n+1] \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}\begin{bmatrix} p[n] \\ v[n] \end{bmatrix} + \begin{bmatrix} \frac{T^2}{2M} \\ \frac{T}{M} \end{bmatrix} \tag{28}$$

[4] Alternatively we can guess and check $v(t) = At + B$.

6 Controlling a System

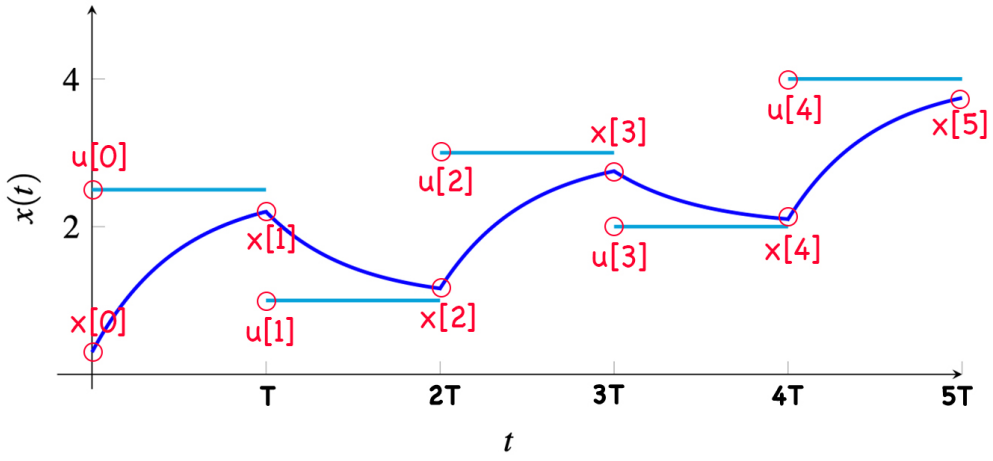
Now that we've taken a look at a couple of examples of discretization, let's see how we can control our systems using our input $u[n]$. Recall from the previous sections on discretization that the following system

$$\frac{d}{dt}x(t) = \lambda(x(t) - u(t)) \tag{29}$$

can be discretized by feeding in a piecewise-constant input and sampling at some rate T

$$x[n+1] = e^{\lambda T}x[n] + (1 - e^{\lambda T})u[n] \tag{30}$$

We show the visualization from before on how if $\lambda < 0$, the system moves toward the input.



Recall from the note on Time Constants that if $\lambda < 0$, our solution will converge to a steady state. For this specific differential equation, the steady state will be $u(t)$ and the time constant is $\tau = \left|\frac{1}{\lambda}\right|$.

Therefore, if $T \approx 5\tau$, our response $x(t)$ at time $t = nT + T$ will be within 1% of our control input $u[n]$! A further analysis of our discretized equation verifies that $x[n+1] \approx u[n]$ if λ is very large and negative.

$$x[n+1] = e^{\lambda T}x[n] + (1 - e^{\lambda T})u[n] \tag{31}$$

As a result, if we wanted to move to a new state at time $n+1$, all we would have to do is feed in $u[n] = x_d[n+1]$ where the subscript d stands for "desired."

If our system was instead represented by the differential equation

$$\frac{d}{dt}x(t) = \lambda x(t) + bu(t) \tag{32}$$

we can always scale our input $u[n] = -\frac{\lambda}{b}x_d[n+1]$. The discretized system will be

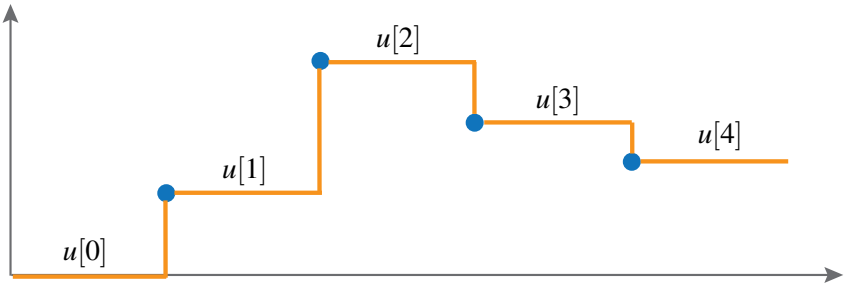
$$x[n+1] = e^{\lambda T}x[n] + b\frac{e^{\lambda T} - 1}{\lambda}u[n] \implies x_{ss}[n+1] = -\frac{b}{\lambda}u[n] = x_d[n+1] \tag{33}$$

following our desired behavior of the state tracking the input.

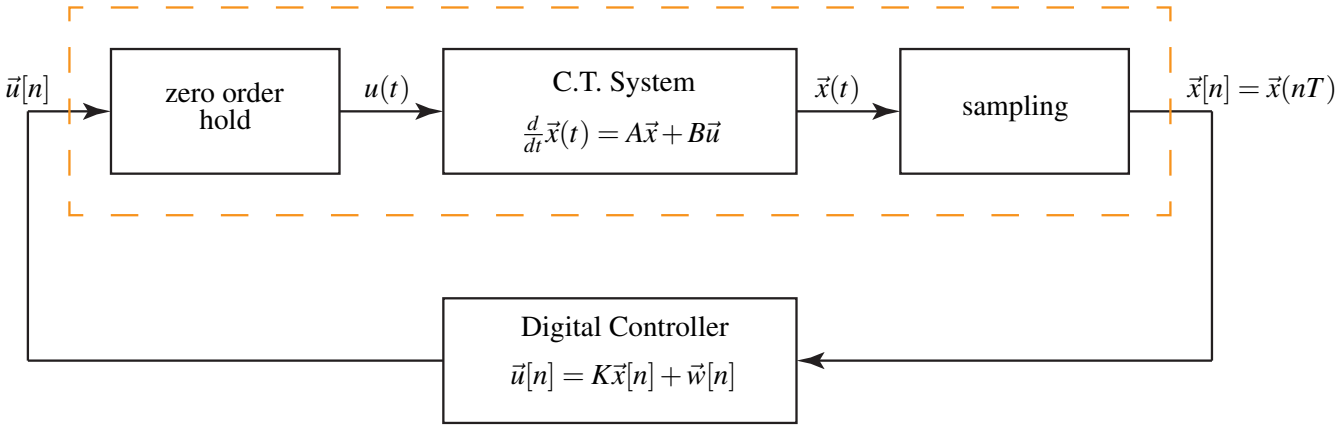
7 Digital Control Systems

In a typical application, the control algorithm for a continuous-time physical system is executed in discrete-time. Thus, the measured output must be sampled before being fed to the control algorithm.

The discrete-time control input generated by the algorithm must be interpolated into a continuous-time function, typically with a zero order hold, before being applied back to the system.



The visual below depicts the full digital control system. The discretization process is shown in the box outlined in yellow.



In the previous note, we looked at the Digital Controller section and came up with a state-feedback law to stabilize the system.

Contributors:

- Taejin Hwang.
- Murat Arcak.

