Numerical Discretization in High Accuracy MPC Applications

Tianyuan Kong
Department of Electrical and Computer Engineering, Chair of Automatic Control Engineering (LSR)
Technical University of Munich
Munich, Germany
ge23loy@tum.de

***Abstract*—This work provides a comparative review of numerical methods generally used to discretize continuous-time linear system models appearing in Model Predictive Control (MPC) problems, i.e., Runge-Kutta (RK) methods and Gauss collocation. An overview of the characteristics of different RK methods, i.e. RK1, RK2, RK4, are given and the performance of each method is evaluated in a simulation example. Furthermore, we provide a discretization method based on Gauss collocation that allows to consider piecewise polynomial input functions within MPC, akin to higher-order hold elements.**

***Index Terms*—discretization methods, model predictive control, Runge-Kutta methods, Gauss collocation, real-time optimization**

I. Introduction

MPC is a control technique that uses models of the plant in order to predict the future behaviour of the system over a prediction horizon. Usually, these models are expressed mathematically as ordinary differential equations (ODE) [1]. Applying MPC schemes to continuous time systems requires numerical discretization of the continuous time optimal control problem. Commonly used discretization approaches for MPC are based on RK methods. For the implementation in a sampled control loop, typically zero-order hold (ZOH), i.e., piecewise constant inputs between the sampling intervals are used. In applications where the continuous time model is exactly known and high accuracy control is required, a finer discretization grid or a higher order RK method can be chosen to decrease the approximation error of the discretization method, e.g., Gauss collocation [2] with appropriate higher-order hold elements.

II. Problem Statement

The goal of this work is to derive a trade-off between prediction accuracy and computational complexity within an MPC scheme for continuous time linear systems. This is investigated by varying the granularity of numerical discretization and the order of RK methods, including the order of the hold element in the sampled control loop. In addition, the effects of the discretization approaches on the computation of reachable and invariant sets are also examined. Results from this work will enable similar investigations for the sampled control of continuous time nonlinear systems by MPC schemes. In MPC, higher accuracy of the discretization leads to increased computational load as more decision variables are introduced to the optimal control problem, which can result in infeasible computation time.

III. Problem Setup

In the following we consider a continuous-time dynamical systems described by the following equation

$$\dot{x} = f(x(t), u(t)), \tag{1}$$

where $x(t) \in \mathbb{R}^m$ is the state and $u(t) \in \mathbb{R}^n$ is the control input. The MPC problem is defined as follows: determine the state $x_k \in \mathbb{R}^m$ and $u_k \in \mathbb{R}^n$ that solves the optimization problem

$$\begin{aligned} \min_{(x,u)} \sum_{k=i}^{i+N-1} & l_k(x_k, u_k) + l_N(x_{k+N}) \\ s.t. \qquad & x_i - \bar{x_i} = 0, \\ x_{k+1} - & f_k(x_k, u_k) = 0, \\ & h_k(x_k, u_k) \leq 0, \\ & \forall k \in [i, \cdots, i+N-1] \end{aligned} \tag{2}$$

where N is the control horizon, l_k is the stage cost, l_N is the terminal cost (a popular choice is the use of quadratic costs [3]). The parameter $\bar{x_i}$ is the fixed initial value, the function f_k is the discrete-time version of (1), and the function h_k describes state and control constraints. In order to solve (2) we need to find a method to discretize (1), thus allowing to solve a non-linear program (NLP) on the free variables (x,u).

Particularly in this work, we assume a continuous time LTI system with the system matrices A_{con} and B_{con}, the ODE of this system can be represented by the following equation

$$\dot{x} = A_{\text{con}}x + B_{\text{con}}u, \tag{3}$$

The discrete-time system can be expressed as

$$x_{k+1} = A_{\text{d}}x_k + B_{\text{d}}u_k, \tag{4}$$

IV. Technical Approach

In the following, we will discuss methods on how to retrieve a discretized system of the form (4) from a given contunuous-time system equation as in (3). During the analysis of different discretization methods, we firstly introduce the zero-order hold (ZOH), i.e., piecewise constant inputs u between the sampling intervals Δt, and first-order hold (FOH), i.e., piecewise linear inputs u between the sampling intervals Δt. The ZOH is illustrated in Fig. 1 and FOH is illustrated in Fig. 2

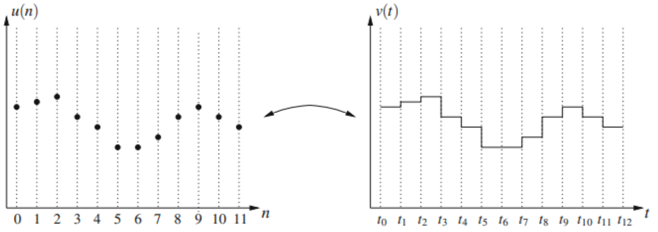


Fig. 1: Illustration of zero-order hold [4]

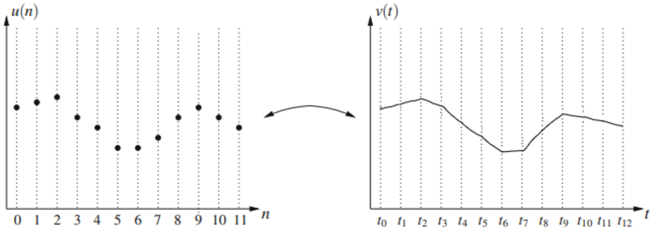


Fig. 2: Illustration of first-order hold

A. *Runge-Kutta method*

Given a continuous time LTI system in Eq. (3), the corresponding RK discretization models can be represented by the following equation:

$$x_{k+1} = A_{\mathrm{rk}} x_k + B_{\mathrm{rk}} u_k, \tag{5}$$

The family of explicit RK methods is given by

$$x_{k+1} = x_k + \sum_{i=1}^{s} b_i \kappa_i, \tag{6}$$

where

$$\begin{aligned}
\kappa_1 &= \Delta t f(t_k, x_k, u_k),\\
\kappa_2 &= \Delta t f(t_k + c_2 \Delta t, x_k + \Delta t(a_{21}\kappa_1), u_k),\\
\kappa_3 &= \Delta t f(t_k + c_3 \Delta t, x_k + \Delta t(a_{31}\kappa_1 + a_{32}\kappa_2), u_k)\\
&\ \ \vdots\\
\kappa_s &= \Delta t f(t_k + c_s \Delta t,\\
&\qquad x_k + \Delta t(a_{s1}\kappa_1 + a_{s2}\kappa_2 + \cdots + a_{s,s-1}\kappa_{s-1}), u_k).
\end{aligned} \tag{7}$$

To specify a particular method, one needs to provide the integer s (the number of stages), and the coefficients a_{ij} (for $1 \le j < i \le s$) and c_i (for $i = 2, 3, \cdots, s$). The matrix consisting of the elements a_{ij}, which is called Runge-Kutta matrix, while the b_i and c_i are known as the weights and nodes. More compactly, these parameters are written as so-called Butcher tableaus[4] of the form

$$\begin{array}{c|ccccc}
c_1 & & & & & \\
c_2 & a_{21} & & & & \\
c_3 & a_{31} & a_{32} & & & \\
\vdots & \vdots & & \ddots & & \\
c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & \\
\hline
 & b_1 & b_2 & \cdots & b_{s-1} & b_s
\end{array}$$

Table 1 shows Butcher tableaus corresponding to the Euler scheme known as RK1 (left), the Heun scheme known as RK2 (middle) and the so-called classical Runge–Kutta scheme with s = 4 stages proposed by Carl Runge and Martin Kutta in 1895 (right).

TABLE I

$$\begin{array}{c|c}
0 & \\
\hline
 & 1
\end{array}$$

TABLE II

$$\begin{array}{c|cc}
0 & & \\
1 & 1 & \\
\hline
 & \frac{1}{2} & \frac{1}{2}
\end{array}$$

TABLE III

$$\begin{array}{c|cccc}
0 & & & & \\
\frac{1}{2} & \frac{1}{2} & & & \\
\frac{1}{2} & 0 & \frac{1}{2} & & \\
1 & 0 & 0 & 1 & \\
\hline
 & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6}
\end{array}$$

We solve the MPC problem using numerical optimization methods, therefore the continuous-time dynamical system equations need to be discretized. If control input u is known continuously, then we can directly evaluate $u(t)$ at the given sampling interval of RK method. For example, when using the fourth order Runge-Kutta method with ZOH, using $x_n = x(t_0 + n\Delta t)$ and $u_n = u(t_0 + n\Delta t)$, then we obtain

$$\begin{aligned}
x_{k+1} &= x_k + \frac{1}{6}(\kappa_1 + 2\kappa_2 + 2\kappa_3 + \kappa_4),\\
\kappa_1 &= \Delta t(A_{\mathrm{con}} x_k + B_{\mathrm{con}} u_k),\\
\kappa_2 &= \Delta t(A_{\mathrm{con}}(x_k + \kappa_1/2) + B_{\mathrm{con}} u_k),\\
\kappa_3 &= \Delta t(A_{\mathrm{con}}(x_k + \kappa_2/2) + B_{\mathrm{con}} u_k),\\
\kappa_4 &= \Delta t(A_{\mathrm{con}}(x_k + \kappa_3) + B_{\mathrm{con}} u_k).
\end{aligned} \tag{8}$$

Substituting these equations into each other gives

$$\begin{aligned}
x_{k+1} = &\underbrace{\left(I + \Delta t A_{\mathrm{con}} + \frac{\Delta t^2}{2!} A_{\mathrm{con}}^2 + \frac{\Delta t^3}{3!} A_{\mathrm{con}}^3 + \frac{\Delta t^4}{4!} A_{\mathrm{con}}^4\right)}_{\mathrm{A_{rk4}}} x_k\\
&+ \underbrace{\left(\Delta t I + \frac{\Delta t^2}{2!} A_{\mathrm{con}} + \frac{\Delta t^3}{3!} A_{\mathrm{con}}^2 + \frac{\Delta t^4}{4!} A_{\mathrm{con}}^3\right) B_{\mathrm{con}}}_{\mathrm{B_{rk4}}} u_k.
\end{aligned} \tag{9}$$

Therefore, the system matrices A_{rk} and B_{rk} in (5) are:

$$\begin{aligned}
A_{\mathrm{rk4}} &= I + \Delta t A_{\mathrm{con}} + \frac{\Delta t^2}{2!} A_{\mathrm{con}}^2 + \frac{\Delta t^3}{3!} A_{\mathrm{con}}^3 + \frac{\Delta t^4}{4!} A_{\mathrm{con}}^4,\\
B_{\mathrm{rk4}} &= \left(\Delta t I + \frac{\Delta t^2}{2!} A_{\mathrm{con}} + \frac{\Delta t^3}{3!} A_{\mathrm{con}}^2 + \frac{\Delta t^4}{4!} A_{\mathrm{con}}^3\right) B_{\mathrm{con}}.
\end{aligned} \tag{10}$$

Similar to RK4 approximation, we can obtain the system matrices of RK1 and RK2 methods:

$$\begin{aligned} A_{\text{rk1}} &= I + \Delta t A_{\text{con}}, \\ B_{\text{rk1}} &= \Delta t B_{\text{con}}, \\ A_{\text{rk2}} &= I + \Delta t A_{\text{con}} + \frac{\Delta t^2}{2!} A_{\text{con}}^2, \\ B_{\text{rk2}} &= \left(\Delta t I + \frac{\Delta t^2}{2!} A_{\text{con}} \right) B_{\text{con}}. \end{aligned} \tag{11}$$

B. Gauss collocation method

Using s-stage Gauss collocation, we can get the numerical approximation of the discrete-time model. The Gauss collocation numerical solution with state stage values $x_k^{o,i}$, $i = 1, ..., s$:

$$\begin{aligned} x_{k+1} &= x_k + \Delta t \sum_{i=1}^{s} b_i^s \mathbf{f}\left(\mathbf{x}_k^{o,i}\right) + \Delta t \mathbf{g} \sum_{i=1}^{s} b_i^s u_k^i, \\ x_k^{o,i} &= x_k + \Delta t \sum_{j=1}^{s} a_{ij}^s \mathbf{f}\left(\mathbf{x}_k^{o,j}\right) + \Delta t \mathbf{g} \sum_{j=1}^{s} a_{ij}^s u_k^j, \end{aligned} \tag{12}$$

where the RK coefficients $a_{ij}^s = \int_0^{c_j} \ell_i^{s-1}(\tau) d\tau$ and $b_i^s = \int_0^1 \ell_i^{s-1}(\tau) d\tau$. In Gauss collocation method, the control input u_k can be shaped with Lagrange polynomials. The control input for $t \in [t_k, t_{k+1})$ generated via an $s-1$ order hold element using Lagrange interpolation polynomials [2]:

$$u\left(t_k + \tau \Delta t\right) = \sum_{i=1}^{s} u_k^i \ell_i^{s-1}(\tau), \tag{13}$$

where

$$\ell_i^{s-1}(\tau) = \prod_{\substack{l=1 \\ l \neq i}} \frac{\tau - c_l}{c_i - c_l}.$$

The control stage values in the collocation points (weights of Lagrange polynomials):

$$u\left(t_k^i\right) = u_k^i, \quad t_k^i = t_k + c_i \Delta t, \quad i = 1, \ldots, s$$

The collocation points on the unit interval are shown in Fig. 3, the collocation points $c_1^0 = \frac{1}{2}$ and $c_{1/2}^1 = \frac{1}{2} \mp \frac{\sqrt{3}}{6}$

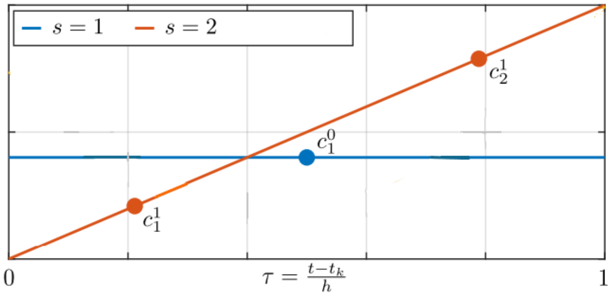


Fig. 3: Control signal shapes on the unit interval based on Gauss collocation points

Given the same continuous time LTI system in (3), the corresponding Gauss collocation discretization models can be represented by an equation of the form (4).

1) Zero-order hold control input: When $s = 1$, i.e. the zero-order hold case, the control input for $t \in [t_k, t_{k+1}]$ results in

$$u(t) = u_k$$

The exact discretization model can be represented as:

$$x_{k+1} = A_{\text{ex},1} x_k + B_{\text{ex},1} u_k, \tag{14}$$

where $A_{\text{ex},1} = e^{A_{\text{con}} \Delta t}$ and $B_{\text{ex},1} = \int_0^{\Delta t} e^{A_{\text{con}} \tau} B_{\text{con}} d\tau$.

Using (12), the Gauss collocation numerical discretization of order s = 1 yields

$$\begin{aligned} x_{k+1} &= x_k + \Delta t b_1 A_{\text{con}} x_k^1 + \Delta t b_1 B_{\text{con}} u_k^1, \\ x_k^1 &= x_k + \Delta t a_{11} A_{\text{con}} x_k^1 + \Delta t a_{11} B_{\text{con}} u_k^1. \end{aligned} \tag{15}$$

We can reformulate (15) into the explicit LTI form (4) with

$$A_{\text{d}} = I + \Delta t b_1 A_{\text{con}} (I - \Delta t a_{11} A_{\text{con}})^{-1},$$

$$B_{\text{d}} = \Delta t b_1 A_{\text{con}} (I - \Delta t a_{11} A_{\text{con}})^{-1} \Delta t a_{11} B_{\text{con}} + \Delta t b_1 B_{\text{con}}.$$

Note that all the used inverse matrices have to exist, otherwise, the derivation is not possible.

2) First-order hold control input: When $s = 2$, i.e. the first-order hold case, the control input for $t \in [t_k, t_{k+1}]$ results in

$$\begin{aligned} u(t) &= u_k^1 l^1(t) + u_k^2 l^2(t) \\ &= u_k^1 \frac{t - c_2^1}{c_1^1 - c_2^1} + u_k^2 \frac{t - c_1^1}{c_2^1 - c_1^1}. \end{aligned} \tag{16}$$

Note that in the case of FOH input, not only the Gauss collocation points should satisfy the input constraints, but also the starting and ending points of the piecewise linear function. The exact discretization model can be represented as:

$$x_{k+1} = A_{\text{ex},2} x_k + B_{\text{ex},2} \begin{bmatrix} u_k^1 \\ u_k^2 \end{bmatrix} \tag{17}$$

with

$$A_{\text{ex},2} = e^{A_{\text{con}} \Delta t},$$

$$B_{\text{ex},2} = \int_0^{\Delta t} e^{A_{\text{con}} \tau} B_{\text{con}} \left[l^1\left(\tfrac{\Delta t - \tau}{\Delta t}\right), \quad l^2\left(\tfrac{\Delta t - \tau}{\Delta t}\right) \right] d\tau.$$

The Gauss collocation numerical discretization model can be represented as:

$$\begin{aligned} x_{k+1} &= x_k + \Delta t b_1 A_{\text{con}} x_k^1 + \Delta t b_2 A_{\text{con}} x_k^2 \\ &\quad + \Delta t b_1 B_{\text{con}} u_k^1 + \Delta t b_2 B_{\text{con}} u_k^2, \\ x_k^1 &= x_k + \Delta t a_{11} A_{\text{con}} x_k^1 + \Delta t a_{12} A_{\text{con}} x_k^2 \\ &\quad + \Delta t a_{11} B_{\text{con}} u_k^1 + \Delta t a_{12} B_{\text{con}} u_k^2, \\ x_k^2 &= x_k + \Delta t a_{21} A_{\text{con}} x_k^1 + \Delta t a_{22} A_{\text{con}} x_k^2 \\ &\quad + \Delta t a_{21} B_{\text{con}} u_k^1 + \Delta t a_{22} B_{\text{con}} u_k^2. \end{aligned} \tag{18}$$

We can reformulate (18) into the explicit LTI form (4) with

$$\begin{aligned}
A_{\mathrm{d}} &= I + A_{b1} A_{1\mathrm{inv}} \left(I + A_{a12} \left(I - A_{a22} \right)^{-1} \right) \\
&\quad + A_{b2} A_{2\mathrm{inv}} \left(I + A_{a21} \left(I - A_{a11} \right)^{-1} \right), \\
A_{1\mathrm{inv}} &= \left(I - A_{a11} - A_{a12} \left(I - A_{a22} \right)^{-1} A_{a21} \right)^{-1}, \\
A_{2\mathrm{inv}} &= \left(I - A_{a21} \left(I - A_{a11} \right)^{-1} A_{a12} - A_{a22} \right)^{-1},
\end{aligned} \tag{19}$$

$$\begin{aligned}
B_{\mathrm{d}} = \Big[& A_{b1} A_{1\mathrm{inv}} \left(A_{a12} \left(I - A_{a22} \right)^{-1} B_{a21} + B_{a11} \right) + \\
& A_{b2} A_{2\mathrm{inv}} \left(A_{a21} \left(I - A_{a11} \right)^{-1} B_{a11} + B_{a21} \right) + B_{b1}, \\
& A_{b1} A_{1\mathrm{inv}} \left(A_{a12} \left(I - A_{a_{22}} \right)^{-1} B_{a_{22}} + B_{a12} \right) + \\
& A_{b2} A_{2\mathrm{inv}} \left(A_{a_{21}} \left(I - A_{a11} \right)^{-1} B_{a12} + B_{a22} \right) + B_{b2} \Big],
\end{aligned} \tag{20}$$

and

$$\begin{aligned}
A_{aij} &= \Delta t a_{ij} A_{\mathrm{con}}, \\
B_{aij} &= \Delta t a_{ij} B_{\mathrm{con}}, \\
A_{bi} &= \Delta t b_i A_{\mathrm{con}}, \\
B_{bi} &= \Delta t b_i B_{\mathrm{con}}.
\end{aligned} \tag{21}$$

Note that all the used inverse matrices have to exist, otherwise, the derivation is not possible.

V. Implementation and Evaluation

In the following, we will implement an example to compare the different-order hold RK methods and Gauss collocation method. For the RK methods, we will firstly compare the discretization accuracy of the state trajectory of the true continuous system using different RK methods, then we investigate the control invariant set and classic variant set of different RK methods. For the Gauss collocation method, we will compare the exact discretization model and the corresponding Gauss collocation model that consider piecewise polynomial input functions.

A. Runge-Kutta method

The RK discretization models can be structured by the matrices A_{rk} and B_{rk} and apply them to (5) , we will assume a piecewise constant control ZOH, so we can focus on how the state trajectory discretization is handled. We will consider three discretization models, which are RK1, RK2 and RK4 models. First, we will be driving a continuous time LTI system to the origin. We will minimize a quadratic cost, so we define the cost terms l_k and l_N

$$\begin{aligned}
l_k &= x_k^T Q x_k + u_k^T R u_k \\
l_N &= x_{k+N}^T Q_N x_{k+N}
\end{aligned} \tag{22}$$

where the matrices $Q \in \mathbb{R}^{m \times m}$, $R \in \mathbb{R}^{n \times n}$ and $Q_N \in \mathbb{R}^{m \times m}$ are symmetric definite positive. The continuous time system is given by the system matrices

$$A_{\mathrm{con}} = \begin{bmatrix} 0 & 7.5 \\ -1.5 & -0.1 \end{bmatrix}, \quad B_{\mathrm{con}} = \begin{bmatrix} 0.145 \\ 0.5 \end{bmatrix}. \tag{23}$$

The initial condition is $x(0) = [2, 2]^\top$ and the state and control input is subject to the following constraints

$$\begin{aligned}
-1 &\leq x_1 \leq 2.8 \\
-1 &\leq x_2 \leq 2.1 \\
-26 &\leq u \leq 26
\end{aligned} \tag{24}$$

The following steps investigate the closed-loop performance:

1) Given the continuous time system, apply different discretization methods (RK of different order, and also vary the sampling time/discretization grid) to retrieve different discretized models.
2) Apply an MPC iteration to retrieve an optimal (discrete time) control input. This control input is then applied to the true continuous system in a ZOH fashion, thus, constantly apply the retrieved optimal control input for the duration of one sampling time of the current discrete system. We will then retrieve the new current state.

The chosen sampling time is $\Delta t = 0.04$ seconds, the horizon length is N = 10, and the weight matrix are $Q = Q_N =$diag$([10, 1])$ and R = 1. The state trajectory of the true continuous system is shown in Fig. 4

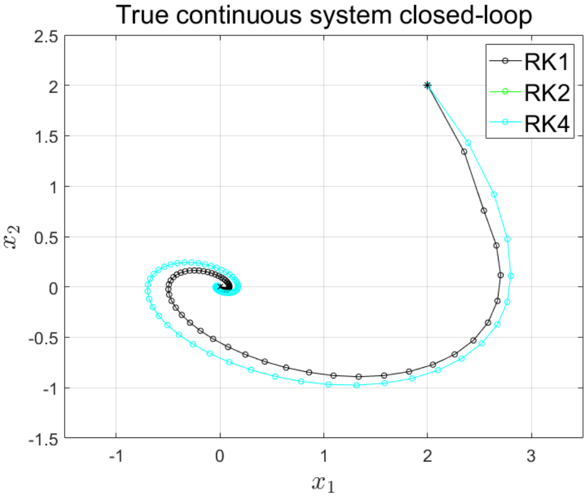


Fig. 4: State trajectory of true continuous system

In Fig. 4, the black line represents the system behavior of RK1 model, while the RK2 has a green line, and it coincides with the cyan line in the plot, which means that the approximation performance between RK2 and RK4 models at a sampling time of 0.04 seconds is extremely similar. Now assume that the sampling time $\Delta t = 0.02$ seconds, the corresponding state trajectory of the true system is shown in Fig. 5.

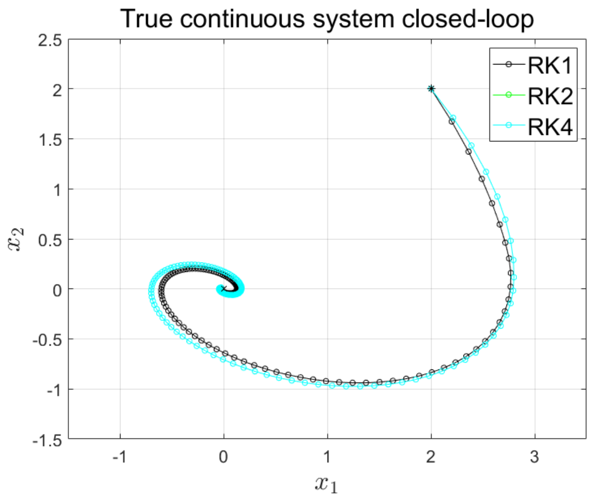


Fig. 5: State trajectory of true continuous system

In Fig. 5, it can be observed that the black line of RK1 is closer to the green line with a smaller sampling time Δt, which indicates that the shorter sampling time, the higher the accuracy.

In Fig. 6, we apply a smaller sampling time $\Delta t = 0.008$ for RK1 model, while applying a larger sampling time $\Delta t = 0.04$ for RK2 and RK4 models, the state trajectory of three models are extremely similar, which indicates decreasing the sampling time Δt of RK1 model so that it has the same accuracy as RK2 and RK4 models with larger sampling time Δt.

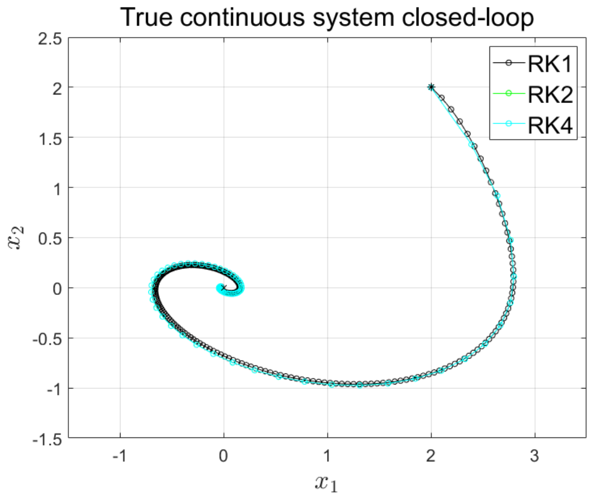


Fig. 6: State trajectory of true continuous system

We can achieve similar accuracy between RK1 model with small sampling time and RK4 model with large sampling time, however, this yields increased computational cost due to the increased number of optimization steps. Fig. 7 shows the time consumption for RK1 model with sampling time $\Delta t = 0.008$ and RK4 model with sampling time $\Delta t = 0.04$ of each MPC iteration.

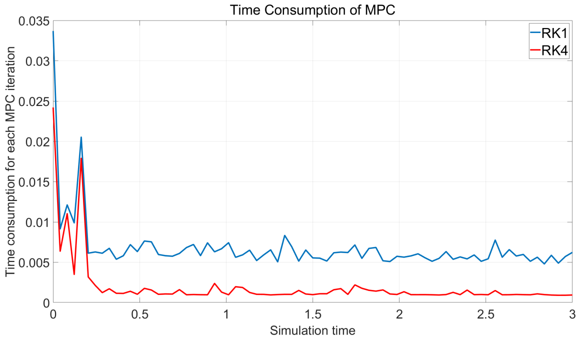


Fig. 7: Time consumption of MPC iteration

It is clear that there is a peak in computation time in the first few MPC iterations, which is due to the fact that state x_1 here violates the constraint and takes longer to compute the optimal control input u.

Next step, we investigates the open-loop performance:

1) Given the continuous time system, apply different discretization methods (RK of different order, and also vary the sampling time/discretization grid) to retrieve different discretized models.
2) Apply an MPC iteration to retrieve an optimal (discrete time) control input sequence.
3) Apply the control input sequence to the discretized model, and also transform the control input sequence to a continuous time sequence using ZOH and apply it to the true (continuous) system dynamics.

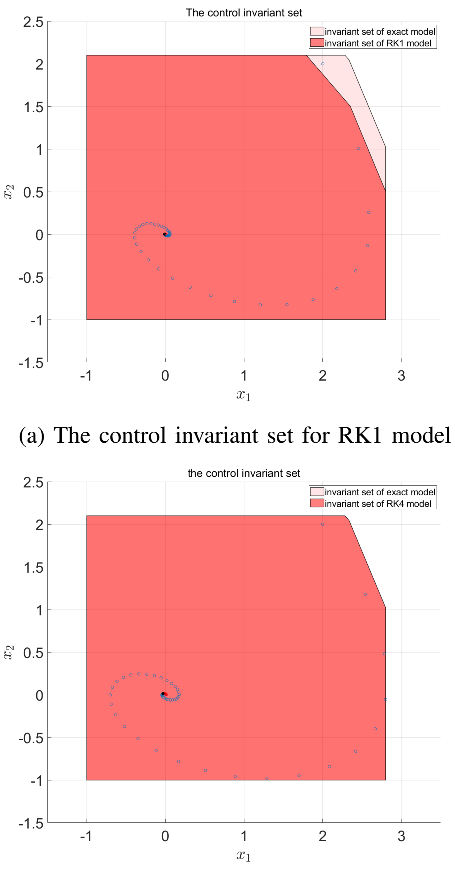
The prediction accuracy of different RK models vary from sampling time, the results shown in Table below

Error RK model Δ t	RK1	RK2	RK4
0.04	0.323	0.095	0.102
0.02	0.081	0.063	0.062

$$\epsilon = \frac{\sum_{k=1}^{N} |x_{\text{con}}(k) - x_{\text{pre}}(k)|}{N}, \tag{25}$$

where $N = T/\Delta t$, $x_{\text{con}}(k)$ is the true continuous system predicted states at step k and $x_{\text{pre}}(k)$ is the MPC predicted states at step k.

Next, we investigate the control invariant set and the invariant set for the uncontrolled system. Fig. 8a shows the control invariant set of RK1 and RK4 model with sampling time $\Delta t = 0.06$.



(a) The control invariant set for RK1 model

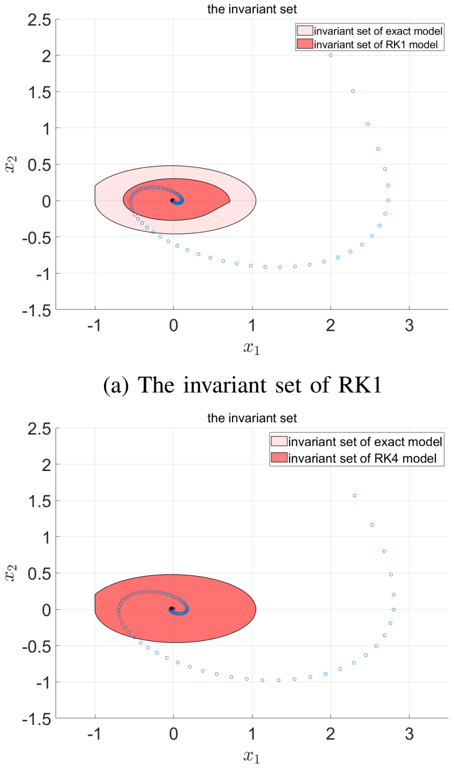
(b) The control invariant set for RK4 model

Fig. 8: The control invariant set

Compared to the control invariant set of the exact model, the initial state $x_0 = [2, 2]^\top$ is not in the control invariant set of RK1 model, thus, it is impossible to derive an feasible control input u, such that x_1 does not violate the state constraint, which indicates the inaccuracy and limitations of the RK1 model.

Fig. 8b shows the control invariant set of RK4 is the same as the one of exact model, which indicates the higher accuracy compared to RK1 model.

Fig. 9a and 9b shows the invariant set of RK1 and RK4 model with sampling time $\Delta t = 0.03$.



(a) The invariant set of RK1

(b) The invariant set of RK4

Fig. 9: The invariant set

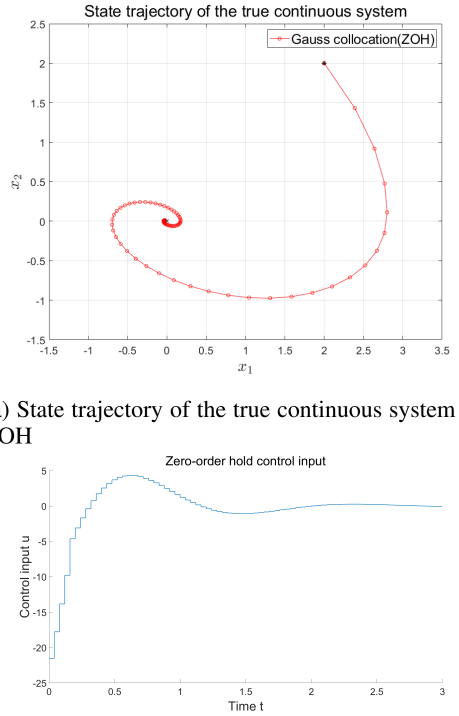
Similar to the control invariant set, the invariant set for the uncontrolled system of RK1 model is less accurate than the one of RK4 model.

B. *Gauss collocation method*

For the Gauss collocation model, the discretized system matrices A_d and B_d are different with different order hold control input u. In this example, the ZOH and FOH are implemented. The state trajectory of the Gauss collocation method is compared with the corresponding exact discretized model, whose system matrices are A_{ex} and B_{ex}.

Fig. 10a shows the state trajectory of exact model and Gauss collocation model with the sampling time $\Delta t = 0.04$ in the case of ZOH. Fig. 10b shows the corresponding ZOH control input signal.

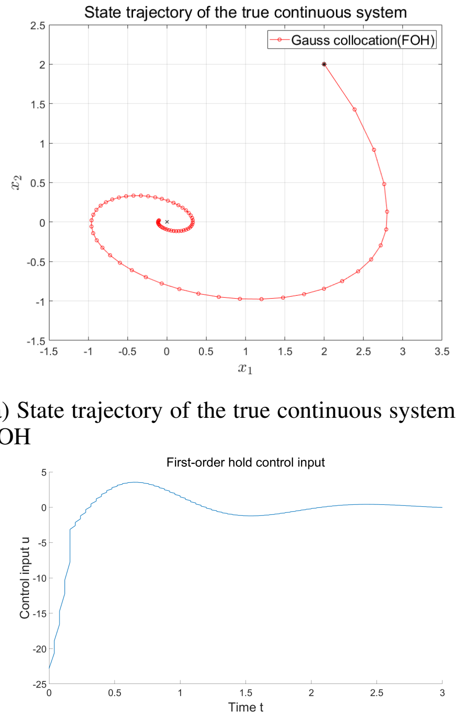
Fig. 11a shows the state trajectory of exact model and Gauss collocation model with the sampling time $\Delta t = 0.04$ in the case of FOH. Fig. 11b shows the corresponding FOH control input signal.



(a) State trajectory of the true continuous system for ZOH

(b) ZOH control input

Fig. 10: The state trajectory and ZOH control input



(a) State trajectory of the true continuous system for FOH

(b) FOH control input

Fig. 11: The state trajectory and ZOH control input

Next, we investigate the control invariant set in case of ZOH and FOH. Fig. 12 shows the control invariant set in the case of ZOH control input and FOH control input. It is obvious that the control invariant set of ZOH is a subset of the control invariant set of FOH, which indicates that the MPC solver can find piecewise linear input signals for the whole state set regarding control invariance, while in the case of piecewise constant input signals (ZOH), the control invariant set is decreased.

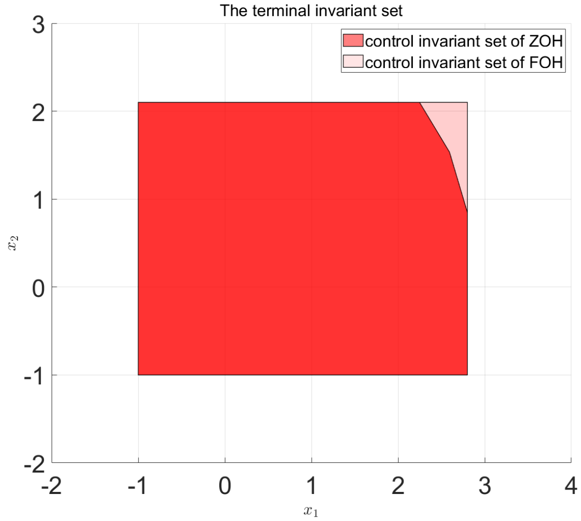


Fig. 12: The control invariant set

VI. Conclusion

In this work, we compared different discretization methods for a LTI continuous time system. Specifically, we have investigated RK methods of different order and Gauss collocation methods that allow piecewise polynomial input functions within MPC. We have found that the prediction accuracy and closed-loop performance of MPC increased with increasing order of the RK method that is used for discretizing the continuous-time system model. Our simulation example also showed that we can achieve similar accuracy between RK1 with small sampling time and RK4 with large sampling time, however, this yields increased computational cost due to the increased number of optimization variables. For the Gauss collocation method, we investigated the state trajectory for different-order hold control input. It was shown that the control invariant set of FOH control input is larger than ZOH control input, as FOH can consider more input shapes than ZOH.

For future research and further extension, we can do the same investigation on nonlinear systems, considering even higher-order hold control inputs beyond FOH, e.g., second-order hold. We also found that in Fig. 11b there are big jumps at each sampling instant, it would also be interesting to investigate how to add possible continuity constraint which would be to fix the end point of one interval with the initial point of the following interval.

References

[1] G. Sánchez, M. Murillo, L. Genzelis, N. Deniz, and L. Giovanini, "Mpc for nonlinear systems: A comparative review of discretization methods,"

in *2017 XVII Workshop on Information Processing and Control (RPIC)*, pp. 1–6, IEEE, 2017.

[2] P. Kotyczka, C. J. Martens, and L. Lefèvre, “High order discrete-time control based on gauss-legendre collocation,” *IFAC-PapersOnLine*, vol. 54, no. 19, pp. 237–242, 2021.

[3] J. M. Maciejowski, *Predictive control: with constraints*. Pearson education, 2002.

[4] L. Grüne and J. Pannek, “Nonlinear model predictive control,” in *Nonlinear model predictive control*, pp. 45–69, Springer, 2017.