

# Comparison and Application of Multi-Rate Methods for Real-Time Simulations of Production Systems

Lars Klingel<sup>a,\*</sup>, Valentin Kamm<sup>a</sup>, Alexander Verl<sup>a</sup>,

<sup>a</sup>*Institute for Control Engineering of Machine Tools and Manufacturing Units (ISW)  
University of Stuttgart  
Seidenstraße 36, 70174 Stuttgart, Germany  
\*lars.klingel@isw.uni-stuttgart.de*

## Abstract

A distributed simulation makes it possible to couple simulation tools and lays the foundation for the usage of multicore capabilities to decrease the calculation time. In consequence the simulation is partitioned on multiple simulation tasks. If simulation tasks with different integration step sizes are used, the configuration is called multi-rate simulation. In real-time simulations the tasks are calculated parallelly, which means that fast tasks do not wait for the simulation result of slower tasks. A sequential approach where fast tasks wait for slower tasks would slow down the overall simulation and therefore tear the real-time requirements. In a real-time multi-rate approach, signal processing of the coupling signals between the tasks is required. For this signal processing, multi-rate methods are used. Easy multi-rate methods lead to stepped signals in the faster task, because the slower task does not provide a new calculated signal at each timestep of the faster task. In this work further methods are investigated in an industrial real-time simulation environment. The analysis contains continuous and discontinuous as well as energy conserving methods. It is shown how these different methods perform for various kinds of signals. The methods are compared and evaluated on signals with different characteristics, which allows a recommendation for the choice of a method in a specific simulation scenario. The application of the multi-rate methods is shown on an example virtual commissioning simulation of an industrial robot. It shows that the right choice of a multi-rate method has a big impact on the overall simulation result.

## 1. Introduction

The increasing digitization in the life cycle of production systems can be summarized under the term digital factory, which contains models, methods and tools with the aim of planning, evaluation and ongoing improvement of the real factory [1]. A method of the digital factory which can help to reduce the development time of production systems and increase their quality is virtual commissioning (VC).

VC can help to detect errors and validate the software in the engineering phase of a production system [2]. VC always contains a simulation model of the production system which interacts with the control system of the machine or plant. VC can be performed using different configurations which differ in the degree of realisation of the control system. In the early phase of the control development model-in-the-loop simulations are used, where the modeled control logic is tested. If the control logic is already available in a programming language for control systems, the code can be run on an emulated controller. The most common and realistic configuration is the hardware-in-the-loop simulation. In a hardware-in-the-loop simulation, the simulation model is connected to the real control system of the machine or plant through a fieldbus. With this structure real-time requirements arise for the simulation because the simulation must be calculated in the same cycle time as the deterministic fieldbus communication and the control system. Pritschow and Röck present an approach for

the architecture of a hardware-in-the-loop simulation tool which meets the real-time requirement by calculating the simulation on a real-time operating system on one processor core [3].

By using only one processor core on a real-time operating system, two main restrictions are imposed: It is not possible to a) calculate complex simulation models due to the limited performance and b) no other simulation models or tools can be coupled. To counter this problem a real-time co-simulation has been introduced by Scheifele [4]. In [4] a real-time capable coupling and synchronisation between simulation tasks is described. A block diagram based model can be partitioned on several simulation tasks which are calculated in a jacobi sequence. That means the tasks are calculated parallelly and only exchange signals at dedicated coupling times [5] in order to maintain the real-time capability. In Figure 1 the real-time co-simulation is shown with an example block diagram, which is partitioned on two tasks. The approach is not limited to a specific number of tasks and can be extended as needed.

In a co-simulation it often occurs that the simulation tasks are calculated with different cycle times, which is called a multi-rate simulation. This is exemplary shown in Figure 1 where the upper task is calculated with the cycle time  $h_1$  and is three times as fast as the lower task with cycle time  $h_2$ . This can be the case if a computationally intense model, like a physical simulation is integrated into the

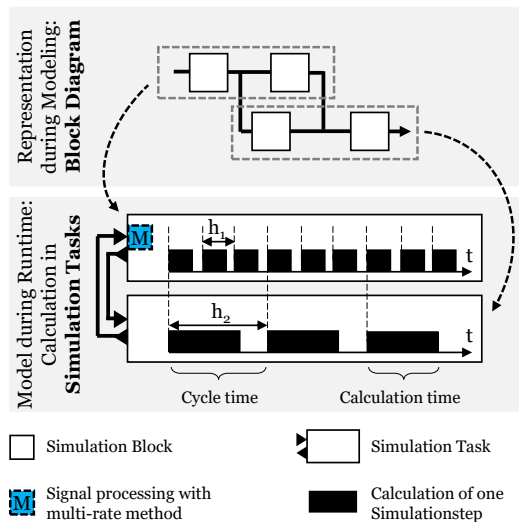


Figure 1: A block diagram based simulation model is partitioned on two simulation tasks and calculated in a jacobi sequence in real-time.

simulation model. In a VC it is also possible that the model's calculation time is too slow for a coupling with the fieldbus. In this case the fieldbus emulation can be calculated with a shorter cycle time than the rest of the model. No matter why the model is partitioned on tasks with different cycle times, a slow simulation task cannot provide a new calculated signal at each time step of a faster task. At this point multi-rate methods come into play [6]. In Figure 1 the position where multi-rate methods can be integrated into a real-time co-simulation is shown with a blue box.

In this work, the authors show how multi-rate methods integrated into a real-time industrial simulation environment can improve partitioned simulations by processing the coupling signals. This work is organized as follows: After introducing related work, a concept for integrating multi-rate methods is shown, as well as the implementation of the different methods. In the fourth section, the validation of the multi-rate methods is shown. A use case of an industrial robot simulation and the conclusion as well as an outlook on the future complete this paper.

## 2. Related Work

In this section an overview of the current state of the art concerning multi-rate methods is given. Basically, there are three approaches for the modification of coupling signals in a distributed multi-rate simulation [7]:

- Standard method: Zero-order-hold
- Application-specific methods
- General methods: Interpolation and extrapolation

Zero-order-hold is the simplest variant, in which the input of the fast system is held until the slow system provides a new value. This approach is also the standard for a real-time co-simulation. The implementation of a zero-order-hold method is simple but the method is subject to jumps in the signal course. The zero-order-hold method shall be used in the following sections as a reference for the evaluation of the investigated multi-rate methods.

Application-specific methods use behavioral models to predict the course of certain signals. In [8] a kalman

filter was used as a multi-rate method for a physics-based material flow simulation. As an example, a cuboid running on a narrowing conveyor belt is simulated. By applying the smoothing filter and boundary conditions, good results are obtained for continuous signals. An automated integration into other coupling signals is described as difficult. In general, application specific methods like a kalman filter are usually not transferable to other simulations models.

The general multi-rate methods can be divided into extrapolation and interpolation. In the case of a real-time co-simulation, due to the parallel operation of the subsystems in a jacobi sequence, no future information of the signals are available. For this reason, interpolation cannot be used. Instead, an extrapolation is necessary, where previous coupling signals are used for the prediction of the further behaviour of the signal. In [9] the hardware-in-the-loop simulation of a conveyor belt is investigated. The real control hardware is coupled to the simulation computer through a fieldbus. The simulation is divided into real-time capable models and the slower physics-based material flow model. The calculation time of a simulation step is due to the complexity of the material flow model up to 40 ms. To enable communication with the control hardware, signals must be provided for the real-time capable solver in 1 ms. The current position and velocity data of the slow model are used to extrapolate new position data for the fast solver. By using the method, the error between the reference signal and the processed signal with the multi-rate method is significantly reduced compared to the standard zero-order-hold method.

Different basic extrapolation models and their convergence behaviors have been investigated in [10]. The multi-rate methods are then successfully applied to the reduction of the simulation time in different areas of simulated vehicle dynamics. Polynomial extrapolation with and without the use of derivative values as well as continuous extrapolation are used. In [11], the overall thermal system of a bus is simulated. By partitioning it into four subsystems, a co-simulation architecture was created. A smoothed extrapolation method was used as a multi-rate method to optimize the system behavior and the simulation speed. An extension of the extrapolation methods by an error correction analogous to control engineering concepts is presented in [12]. Thereby a nearly energy-conserving multi-rate method is obtained. By simulating a driver assistance system, the effect of the developed coupling method is demonstrated.

There are multiple multi-rate methods available [6, 10–12]. Some already have been implemented into a real-time VC simulation [8, 9] but only specific simulation scenarios have been considered. Therefore it is necessary to implement and compare several general methods into a real-time simulation system which can be used for different virtual commissioning simulation scenarios. In the next chapter a concept and implementation of general multi-rate methods is shown.

## 3. Concept and Implementation

In this section the concept for the integration of the multi-rate methods into an industrial simulation environment is shown. Afterwards different multi-rate implementations are shown which can be classified into:

- Discontinuous multi-rate methods
- Continuous multi-rate methods
- Energy conserving multi-rate methods

### 3.1. Concept for the Integration of Multi-Rate Methods into Real-Time Simulation Environments

For the integration of multi-rate methods into an industrial simulation environment the approach from [9] is used. In Figure 1 this concept is shown for the example of two tasks which exchange signals. The multi-rate methods are inserted into the fast task to process signals from the slow task. In the following the step size of the slow subsystem is referred to as macro step size and the step size of the fast subsystem is referred to as micro step size. In the next subsection different multi-rate methods are shown. Those methods are implemented to be usable as black boxes which can be parameterized and used for any model.

### 3.2. Implementation of Multi-Rate Methods

In this section the implemented multi-rate methods are described and visualized on a sine model.

#### 3.2.1. Discontinuous Multi-Rate Methods

For the first set of multi-rate methods a discontinuous approach is chosen where the input is directly carried over to the output at every coupling time (macro step size). In between these communication steps an extrapolation method is executed to approximate the expected output. The most simple attempt for this is to use a polynomial. An efficient way to calculate the polynomials  $p(t)$  is Newton's method:

$$p(t) = \sum_{j=0}^{\infty} c_j N_j(t). \quad (1)$$

The Newton basis function  $N_j(t)$  is given by:

$$N_j(t) = \prod_{i=0}^{j-1} (t - t_i), \quad j = 1, \dots, n. \quad (2)$$

The initial condition is  $N_0(t) = 1$ . The coefficients  $c_j = f[t_0 \dots t_j]$  are calculated using the method of divided differences which is shown for a polynomial of second order in Table 1 in form of a Horner-scheme.

Table 1: Divided differences for a polynomial of second order.

$t_j$	$f_j$	$f[t_j, t_{j+1}]$	$f[t_j, t_{j+1}, t_{j+2}]$
$t_0$	$f[t_0]$		
$t_1$	$f[t_1]$	$\frac{f[t_1] - f[t_0]}{t_1 - t_0} = f[t_0, t_1]$	
$t_2$	$f[t_2]$	$\frac{f[t_2] - f[t_1]}{t_2 - t_1} = f[t_1, t_2]$	$\frac{f[t_1, t_2] - f[t_0, t_1]}{t_2 - t_0} = f[t_0, t_1, t_2]$

The order of the extrapolation is determined by the number of used data points  $n$  and can be adapted to each problem. Figure 2 shows the theoretical evaluation of the extrapolation using a polynomial of third order compared to the standard zero-order-hold. The blue signal is the reference signal which should be accurately predicted by the multi-rate methods. The only information, the multi-rate methods have about the reference signal is signal value on the past coupling times (shown with a dotted line).

The approximated function is  $f(t) = \sin(2\pi t)$  and the step size for the slow subsystem is 40 ms.

If additionally the derivation of the input value is known for every coupling time, the method according to Hermite can be used. Every coupling time there can be used two values for the calculation of the polynomial. Especially in the beginning of the simulation, when fewer values are known than needed for the desired order of extrapolation, the hermite variant leads to more accurate performance.

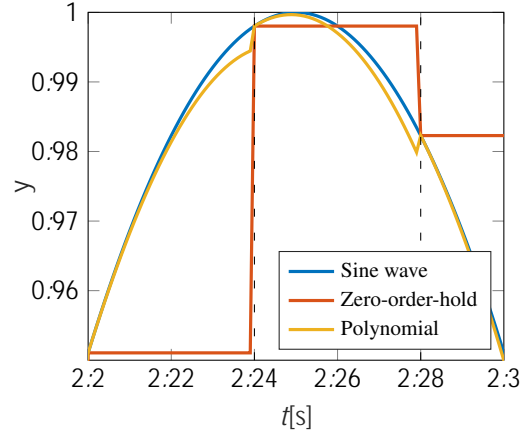


Figure 2: Polynomial of third order compared to the standard zero-order-hold for approximation of a sine. Step size for the slow subsystem is 40 ms.

#### 3.2.2. Continuous Multi-Rate Methods

As can be seen in Figure 2 the discontinuous methods result in small jumps at the coupling time. To avoid this, extrapolation methods are introduced which are continuous and differentiable. For a first approach a combination of extrapolation with interpolation is integrated which has been introduced by Knorr [10]. At every coupling time only the expected value for the next coupling time is extrapolated and saved in an array. The array is then used for interpolation to approximate the value for every step of the fast subsystem. The polynomials needed for the interpolation are calculated according to section 3.2.1. This approach is referred to as 'integrated extrapolation'.

A second approach based on Kossels work [11] is based on making the transition between the previously calculated extrapolation function  $f_p(t)$  and the new function  $f_n(t)$  calculated at the current coupling time using smoothing. The output from the multi-rate method is defined as follows:

$$f(t) = \begin{cases} g(x(t)) f_p(t) + (1 - g(x(t))) f_n(t), & t_i \leq t < t_s \\ f_n(t), & \text{else} \end{cases} \quad (3)$$

Switching from the previous to the new extrapolation polynomial happens half way to the next coupling time:

$$t_s = t_i + \frac{h_s}{2}. \quad (4)$$

Here  $t_s$  is the time of the switchover and  $t_i$  is the time stamp of the previous coupling time. The variable  $x(t)$  has to be normalized to guarantee the continuous behavior of equation (3):

$$x(t) = \frac{t - t_i}{t_s - t_i} \quad (5)$$

For the smoothing function  $g(x(t))$  then follows that  $g(x=0) = 1$  and  $g(x=1) = 0$ . Additionally to ensure differentiability:

$$g^{(i)}(x=0) = g^{(i)}(x=1) = 0, \quad i = 1, \dots, n \quad (6)$$

If  $n = 2$  meaning double differentiability the system of equations is solved to give the following smoothing function:

$$g(x) = 6x^5 + 15x^4 - 10x^3 + 1 \quad (7)$$

Using the smoothing function and the polynomials calculated at every coupling time, equation (3) is used as a multi-rate method to approximate values for each micro step. Figure 3 shows the result for both continuous methods for the same sine as before with the discontinuous approach using the same parameters. As can be seen, the jumps at the coupling times disappear at the expense of some accuracy, which means that the processed signal does not have the exact same value as the reference signal at the coupling times.

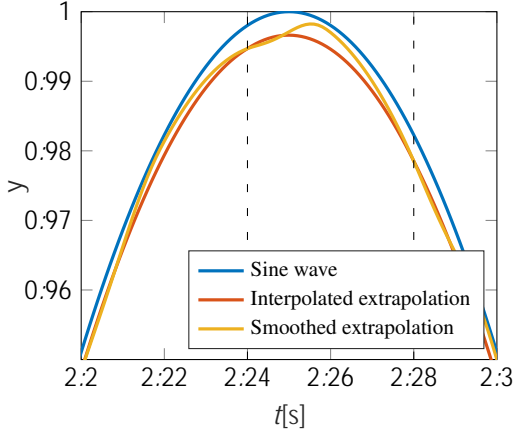


Figure 3: Comparison of the continuous multi-rate methods for approximation of the sine. Degree of extrapolation:  $n_E = 3$ ; degree of interpolation (only interpolated extrapolation):  $n = 3$ ; macro step size:  $h_s = 40$  ms.

### 3.2.3. Energy conserving Multi-Rate Methods

A closed loop approach to multi-rate methods resulting in nearly energy conserving behavior was shown by Benedikt [12]. Following his example the discontinuous as well as the continuous multi-rate methods are enhanced using control loops. The schematic overview of the approach for the discontinuous method is shown in Figure 4.

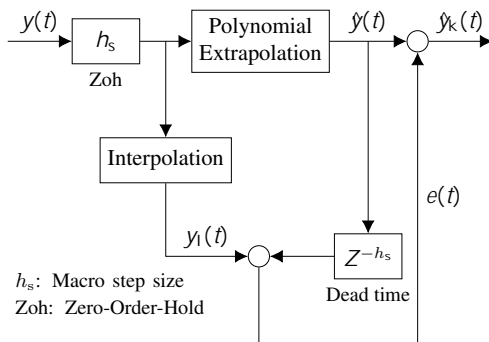


Figure 4: Schematic overview of the integrated closed loop for the extrapolation using polynomials resulting in almost energy conserving behavior.

The slow system generates new output values every  $h_s$  seconds which is shown using the zero-order-hold (Zoh). These values are used for extrapolation to generate the estimated signal  $\hat{y}(t)$  which after a dead time of the macro step size can be compared to the much more accurate interpolated signal  $y_I(t)$ :

$$e(t) = y_I(t) - \hat{y}(t) \quad (8)$$

The identified error  $e(t)$  is added to the estimated signal to generate the corrected values  $\hat{y}_k(t)$  at each micro step. If the macro step size is low enough compared to the frequency of the input signal the closed loop can lead to a significantly improved accuracy as shown in Figure 5.

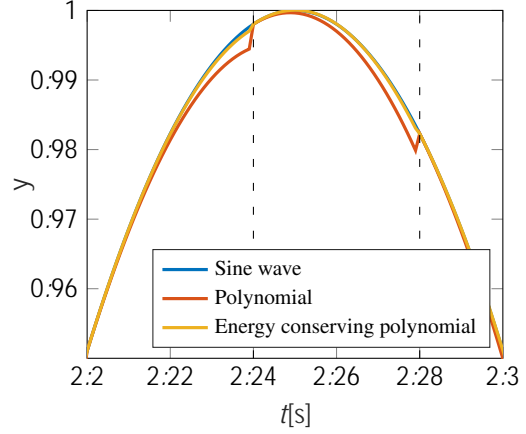


Figure 5: Approximation of the sine using polynomials with and without closed loop control.

Using the energy conserving multi-rate methods while still getting continuous behavior requires a different approach. As a basis the interpolated extrapolation method from section 3.2.2 is used. The schematic overview is shown in Figure 6.

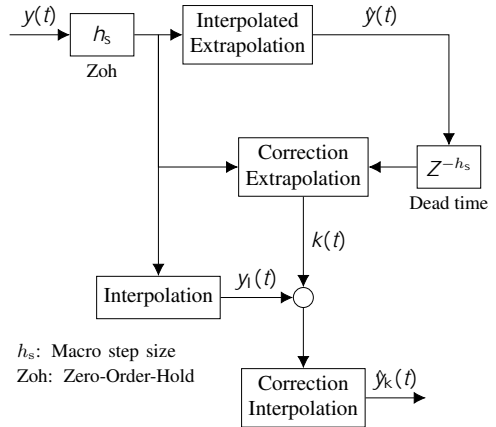


Figure 6: Schematic block diagram for the energy conserving continuous multi-rate method.

Similar to the discontinuous control loop the extrapolated value can be compared to the real value delayed by a dead time of the slow systems step size  $h_s$ . The error is added to the following estimation which is then in turn used for the continuous interpolation. In the block diagram this is marked as *correction extrapolation* resulting in the corrected signal  $k(t)$ . Although this simple approach increases the accuracy of the approximation, it does not provide energy conserving behavior. Therefore the delayed but more accurate interpolation between the actual input values is again computed to be compared to the multi-rate methods estimated interpolation. For every

macro step the individual errors added up to result in the error total  $E$ :

$$E(T_i) = \sum_{i=0}^N (y_i(t_i) - k(t_i)) \quad (9)$$

Here  $N$  is the number of micro steps:

$$N = \frac{h_s}{h_f} - 1 \quad (10)$$

$T_i$  are the coupling times at the distance of the macro step size  $h_s$  whereas  $t_i$  are the micro steps in the faster cycle time of  $h_f$ . Adding the error total  $E$  at the next macro step to the corrected signal  $k(t)$  then ensures energy conservation. To avoid losing continuity this summation is done using a quadratic distribution function  $d$  which has to be zero at the boundaries of the macro steps:

$$d(t_i) = \frac{E(T_{i-1})}{\sum_{i=0}^N (i(i-N))} \cdot i(i-N) \quad (11)$$

Here  $i$  is the counter of the micro steps which is reset after every coupling time. With this procedure, shown in the figure as *correction extrapolation*, the corrected output is obtained:

$$\hat{y}_k(t) = k(t) + v(t) \quad (12)$$

In Figure 7 the result of the correction is compared to the interpolated extrapolation without energy conserving behaviour. As can be seen, the correction leads to an improved accuracy at least for this particular example. Generally the improvement depends on the cycle time of the subsystem compared to the frequency of the input signal as explained for the discontinuous method. For the chosen example signal, the sine wave, that means that the frequency of the sine wave needs to be higher than the cycle time of the macro and micro step size, otherwise the signal may swing up. This phenomenon is based on the Nyquist-Shannon-theorem.

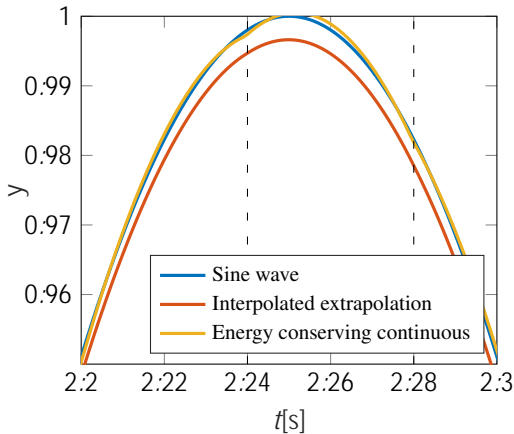


Figure 7: Approximation of the sine with and without control loops using continuous multi-rate methods.

#### 4. Validation of the Multi-Rate Methods

For the validation of the implemented multi-rate methods, three input signals are examined. The first function  $y_1(t)$  is the previously used sine wave as a periodic function. Second is an aperiodic function  $y_2(t)$  resembling a

damped harmonic oscillator with a displacement at  $t = 0$ . Third is a function  $y_3(t)$  with a point of discontinuity:

$$y_1(t) = f(t) = \sin(2\pi t) \quad (13)$$

$$y_2(t) = e^{(-\frac{3t}{2})} \cos(3t) \quad (14)$$

$$y_3(t) = \begin{cases} \frac{1}{3}(4t - t^2) & , 0 \leq t < 3 \\ 1 & , t \geq 3 \end{cases} \quad (15)$$

The three functions are shown in Figure 8.

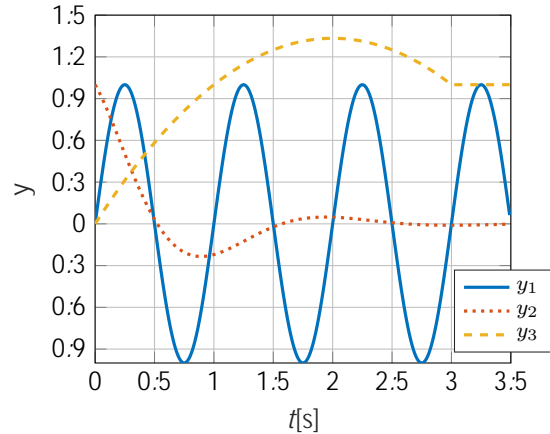


Figure 8: Input signals used to validate the multi-rate methods.

Using the validation functions in equations (13) to (15), the implemented multi-rate methods are evaluated by different criteria while using the validation functions as the input signal for the slow subsystem. The step size as well as the order of extrapolation of the multi-rate methods are varied to investigate different behaviour of the multi-rate methods. The variation of those parameters is shown in Table 2. For the validation three different combinations of step size and order of extrapolation are used, which is shown as variant A1, A2 and A3. For each variant the behavior of the multi-rate methods for the three different validation functions is recorded.

Table 2: Recordings for the theoretical validation of the multi-rate methods. The index after the dot corresponds to the respective input signal.

	A1 $f_{y_1, y_2, y_3g}$	A2 $f_{y_1, y_2, y_3g}$	A3 $f_{y_1, y_2, y_3g}$
$h_s$ (ms)	40	40	75
$n_E$	3	4	3

The first and most important criterion to validate the multi-rate methods is the achievable accuracy. For this the mean squared error is calculated:

$$MSE(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^n (y_i - \hat{y}_i)^2 \quad (16)$$

Where  $n$  is the number of considered time steps,  $y_i$  is the actual value at the time step  $i$  and  $\hat{y}_i$  is the value estimated by the multi-rate method. The results for the mean squared error of the nine recordings are listed in Table 3.

As additional criteria, continuity and differentiability are evaluated. Finally, the response time of the method is evaluated with respect to abrupt changes in the input

Table 3: Mean squared error of the multi-rate methods: Zero-Order-Hold (Zoh), discontinuous methods polynomial extrapolation (Pol), hermite extrapolation (Her), interpolated extrapolation (Int), smoothed extrapolation (Smo), energy conserving discontinuous method (Ecd) and energy conserving continuous method (Ecc).

	Zoh	Pol	Her	Int	Smo	Ecd	Ecc
<b>A1</b> $y_1$	$1.01 \cdot 10^{-2}$	$1.54 \cdot 10^{-6}$	$2.57 \cdot 10^{-6}$	$7.64 \cdot 10^{-6}$	$4.34 \cdot 10^{-6}$	$9.57 \cdot 10^{-8}$	$2.72 \cdot 10^{-7}$
<b>A1</b> $y_2$	$1.79 \cdot 10^{-4}$	$7.79 \cdot 10^{-10}$	$7.73 \cdot 10^{-9}$	$3.88 \cdot 10^{-9}$	$2.19 \cdot 10^{-9}$	$1.41 \cdot 10^{-11}$	$9.57 \cdot 10^{-11}$
<b>A1</b> $y_3$	$1.45 \cdot 10^{-4}$	$1.29 \cdot 10^{-5}$	$1.31 \cdot 10^{-5}$	$5.14 \cdot 10^{-5}$	$2.77 \cdot 10^{-5}$	$3.95 \cdot 10^{-5}$	$2.45 \cdot 10^{-4}$
<b>A2</b> $y_1$	$1.01 \cdot 10^{-2}$	$8.38 \cdot 10^{-8}$	$1.39 \cdot 10^{-5}$	$4.54 \cdot 10^{-7}$	$2.56 \cdot 10^{-7}$	$5.75 \cdot 10^{-9}$	$3.79 \cdot 10^{-8}$
<b>A2</b> $y_2$	$1.79 \cdot 10^{-4}$	$6.17 \cdot 10^{-12}$	$4.77 \cdot 10^{-8}$	$3.32 \cdot 10^{-11}$	$1.89 \cdot 10^{-11}$	$2.08 \cdot 10^{-13}$	$2.51 \cdot 10^{-9}$
<b>A2</b> $y_3$	$1.45 \cdot 10^{-4}$	$3.77 \cdot 10^{-5}$	$1.81 \cdot 10^{-5}$	$2.83 \cdot 10^{-4}$	$9.11 \cdot 10^{-5}$	$1.24 \cdot 10^{-4}$	$1.10 \cdot 10^{-3}$
<b>A3</b> $y_1$	$3.56 \cdot 10^{-2}$	$2.31 \cdot 10^{-4}$	$2.34 \cdot 10^{-5}$	$1.20 \cdot 10^{-3}$	$6.39 \cdot 10^{-4}$	$4.62 \cdot 10^{-5}$	$3.52 \cdot 10^{-4}$
<b>A3</b> $y_2$	$6.19 \cdot 10^{-4}$	$1.34 \cdot 10^{-7}$	$3.97 \cdot 10^{-8}$	$1.50 \cdot 10^{-6}$	$3.98 \cdot 10^{-7}$	$6.58 \cdot 10^{-9}$	$3.40 \cdot 10^{-6}$
<b>A3</b> $y_3$	$5.10 \cdot 10^{-4}$	$8.68 \cdot 10^{-5}$	$8.84 \cdot 10^{-5}$	$3.39 \cdot 10^{-4}$	$1.83 \cdot 10^{-4}$	$2.67 \cdot 10^{-4}$	$1.60 \cdot 10^{-3}$

signal. An example for such a change would be the point of discontinuity in  $y_3(t)$ . All the multi-rate methods as well as the standard zero-order-hold are graded using a harvey balls table. An empty circle means the method can not meet the criteria. The more filled the circle is, the better is the performance of the method for the specific criteria. It should be noted that the evaluation in Table 4 is qualitatively and should serve as a decision-making aid for the selection of a multi-rate method. All the chosen criteria depend on the parameters of extrapolation order as well as the macro step size.

Table 4: Classification of the multi-rate methods regarding the output behavior for different input signals.

	Achievable Accuracy	Continuity	Differentiability	Reaction time
<b>Zoh</b>	○	○	○	●
<b>Pol</b>	●	◐	◐	◐
<b>Her</b>	◐	◐	◐	●
<b>Int</b>	◐	●	●	◐
<b>Smo</b>	◐	●	◐	◐
<b>Ecd</b>	●	◐	◐	◐
<b>Ecc</b>	◐	●	◐	○

The table can be used to choose the proper multi-rate method for given signals in a simulation scenario. For example, the highest accuracy can be achieved using the energy conserving methods. However if the input signal tends to change abruptly another method with a better reaction time should be chosen since the energy conserving methods tend to overshoot. Additional considerations are necessary if continuity and differentiability are necessary which come at the cost of accuracy.

### 5. Application Use Case in Industrial Robotics

In this section the presented concept is applied to a real VC use case. VC simulations which contain real programmable logic controllers (PLC) and computerized numerical controllers (CNC) for motion are normally calculated in a cycle time of one or a few milliseconds on a real-time operating system [3]. In comprehensive production systems with robots, the robot tasks are

triggered and observed by the PLC. To validate the interaction of all controls in a production system, virtual robot controls are integrated into a VC simulation. Those virtual controls are also used for the programming of the robots and are normally running on a non-real-time operating system with a higher cycle time than the PLC, CNC and the virtual production system. This is why multi-rate methods come into play at this use case. As an application example an industrial robot handling task is chosen. In the simulated scene the robot is performing a handling task, which is the most common task for industrial robots [13]. The simulation setup, which is used for the application example in this work is shown in Figure 9.

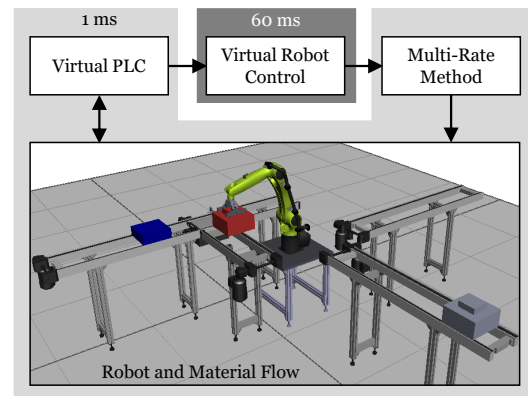


Figure 9: The example use case: A robot is performing a handling task.

For this example the robot control is running with a cycle time of 60 ms and the rest of the simulation setup is running with a cycle time of 1 ms. For the orchestration of the robot tasks a simple PLC is implemented into the simulation tool. The multi-rate methods are inserted between the virtual robot control and the simulation model of the robot. The robot itself performs a pick and place task where it interacts with parts, which are modeled physically. The robot, the material flow and the virtual PLC are modeled with the real-time simulation environment ISG-virtuos while the multi-rate methods are integrated with custom C++ blocks. As a robot control

