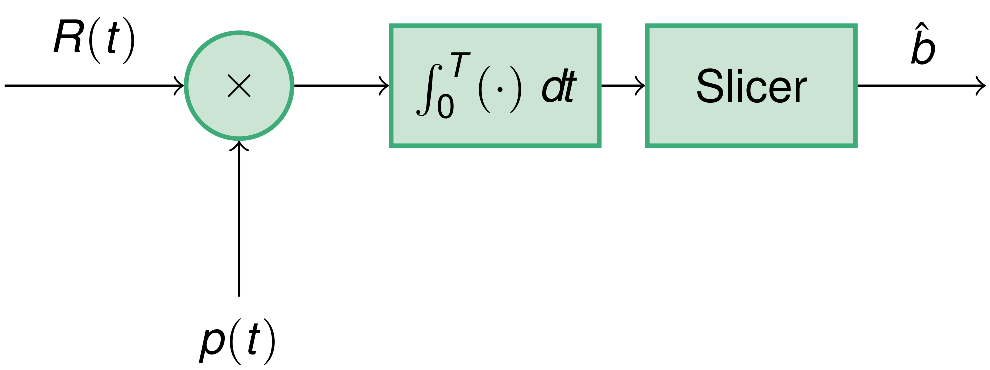
Matched Filter

- It is well known, that the optimum receiver for an AWGN channel is the matched filter receiver.
- The matched filter for a linearly modulated signal using pulse shape $p(t)$ is shown below.
  - The slicer determines which symbol is “closest” to the matched filter output.
  - Its operation depends on the symbols being used and the a priori probabilities.



Shortcomings of The Matched Filter

- While theoretically important, the matched filter has a few practical drawbacks.
  - For the structure shown above, it is assumed that only a single symbol was transmitted.
  - In the presence of channel distortion, the receiver must be matched to $p(t) * h(t)$ instead of $p(t)$.
    - **Problem:** The channel impulse response $h(t)$ is generally not known.
  - The matched filter assumes that perfect symbol synchronization has been achieved.
  - The matching operation is performed in continuous time.
    - This is difficult to accomplish with analog components.

Analog Front-end and Digital Back-end

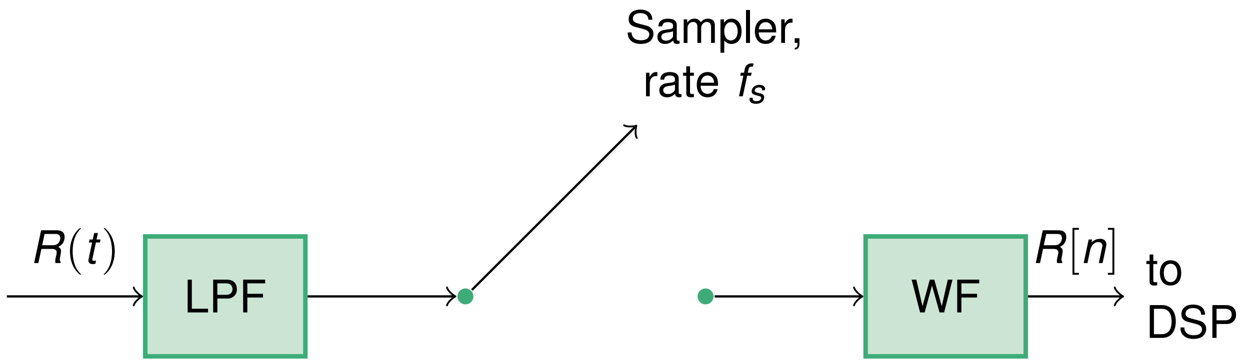
- As an alternative, modern digital receivers employ a different structure consisting of
  - an analog receiver front-end, and
  - a digital signal processing back-end.
- The analog front-end is little more than a filter and a sampler.
  - The theoretical underpinning for the analog front-end is Nyquist's sampling theorem.
  - The front-end may either work on a baseband signal or a passband signal at an intermediate frequency (IF).
- The digital back-end performs sophisticated processing, including
  - digital matched filtering,
  - equalization, and
  - synchronization.

Analog Front-end

- Several, roughly equivalent, alternatives exist for the analog front-end.
- Two common approaches for the analog front-end will be considered briefly.
- Primarily, the analog front-end is responsible for converting the continuous-time received signal $R(t)$ into a discrete-time signal $R[n]$.
  - Care must be taken with the conversion: (ideal) sampling would admit too much noise.
  - Modeling the front-end faithfully is important for accurate simulation.

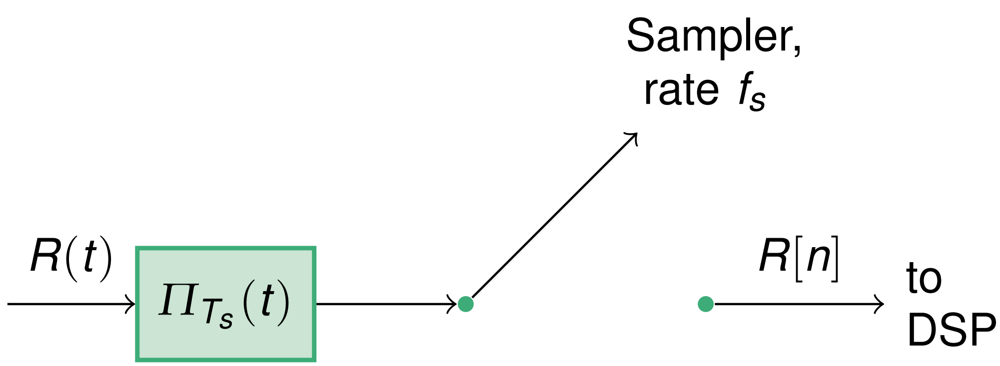
Analog Front-end: Low-pass and Whitening Filter

- The first structure contains
  - a low-pass filter (LPF) with bandwidth equal to the signal bandwidth,
  - a sampler followed by a whitening filter (WF).
    - The low-pass filter creates correlated noise,
    - the whitening filter removes this correlation.



Analog Front-end: Integrate-and-Dump

- An alternative front-end has the structure shown below.
  - Here, $\Pi_{T_s}(t)$ indicates a filter with an impulse response that is a rectangular pulse of length $T_s = 1/f_s$ and amplitude $1/T_s$.
  - The entire system is often called an integrate-and-dump sampler.
  - Most analog-to-digital converters (ADC) operate like this.
  - A whitening filter is not required since noise samples are uncorrelated.



Output from Analog Front-end

- The second of the analog front-ends is simpler conceptually and widely used in practice;
  it will be assumed for the remainder of the course.
- For simulation purposes, we need to characterize the output from the front-end.
  - To begin, assume that the received signal $R(t)$ consists of a deterministic signal $s(t)$ and (AWGN) noise $N(t)$:

    $$R(t) = s(t) + N(t).$$

  - The signal $R[n]$ is a discrete-time signal.
    - The front-end generates one sample every T_s seconds.
  - The discrete-time signal $R[n]$ also consists of signal and noise

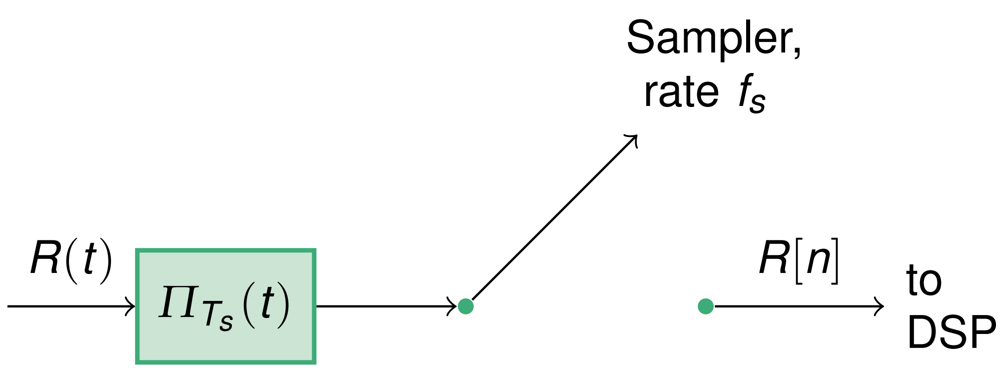
    $$R[n] = s[n] + N[n].$$

Output from Analog Front-end

- Consider the signal and noise component of the front-end output separately.
  - This can be done because the front-end is linear.
- The *n*-th sample of the signal component is given by:

$$s[n] = \frac{1}{T_s} \cdot \int_{nT_s}^{(n+1)T_s} s(t)\, dt \approx s((n+1/2)T_s).$$

  - The approximation is valid if $f_s = 1/T_s$ is much greater than the signal band-width.



Output from Analog Front-end

- The noise samples $N[n]$ at the output of the front-end:
  - are independent, complex Gaussian random variables, with
  - zero mean, and
  - variance equal to N_0/T_s.
- The variance of the noise samples is proportional to $1/T_s$.
  - **Interpretations:**
    - Noise is averaged over T_s seconds: variance decreases with length of averager.
    - Bandwidth of front-end filter is approximately $1/T_s$ and power of filtered noise is proportional to bandwidth (noise bandwidth).
- It will be convenient to express the noise variance as $N_0/T \cdot T/T_s$.
  - The factor $T/T_s = f_s T$ is the number of samples per symbol period.

System to be Simulated

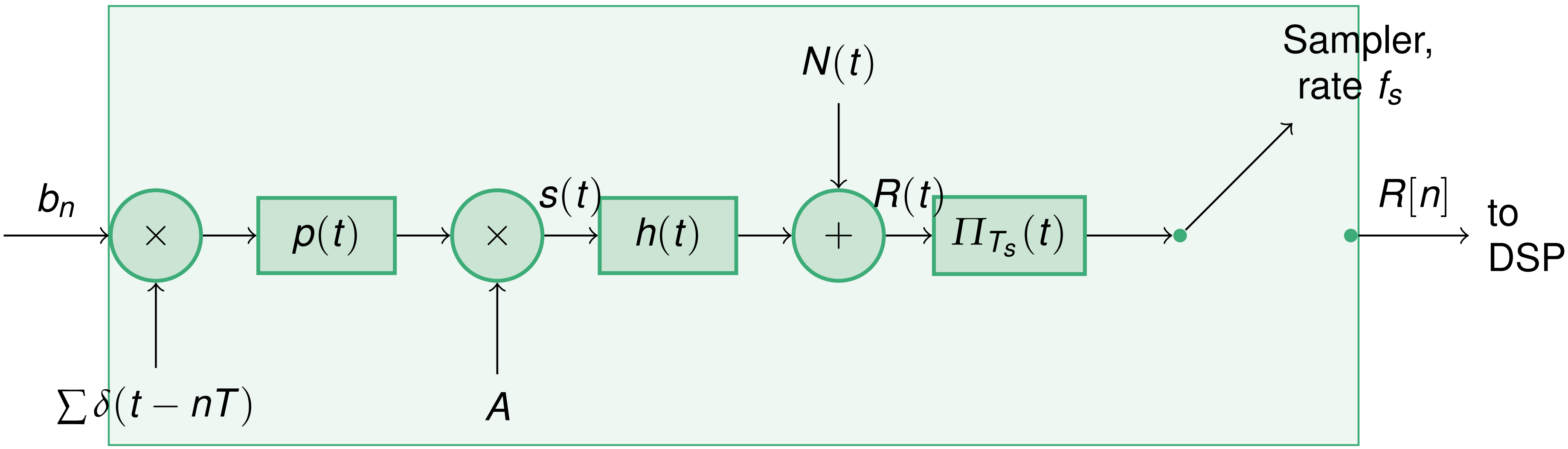


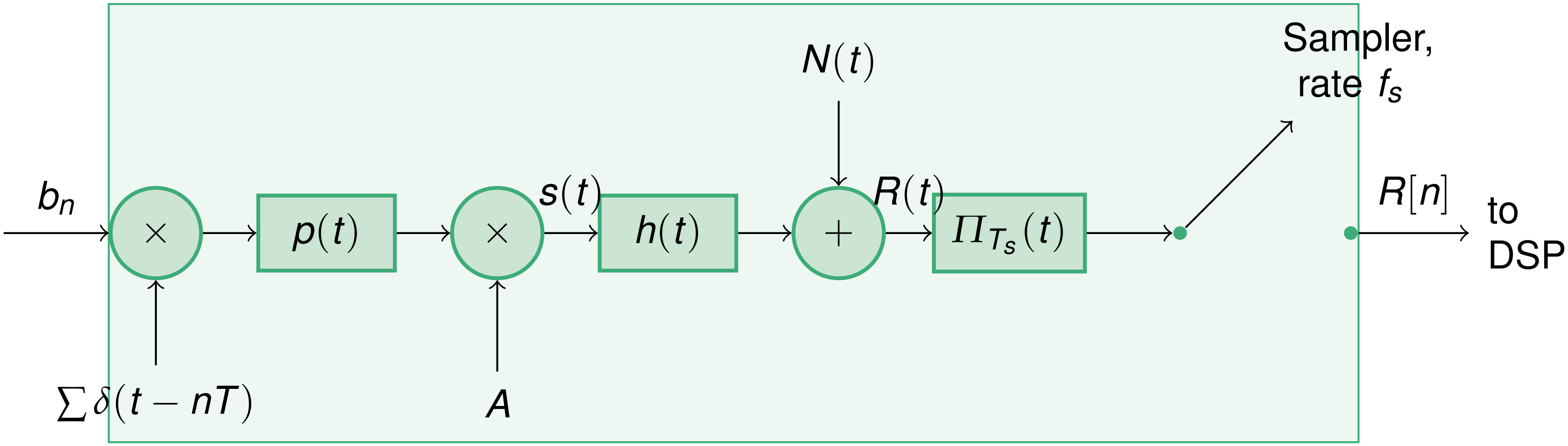
Figure: Baseband Equivalent System to be Simulated.

From Continuous to Discrete Time

- The system in the preceding diagram cannot be simulated immediately.
  - **Main problem:** Most of the signals are continuous-time signals and cannot be represented in MATLAB.
- **Possible Remedies:**
  1. Rely on Sampling Theorem and work with sampled versions of signals.
  2. Consider discrete-time equivalent system.
- The second alternative is preferred and will be pursued below.

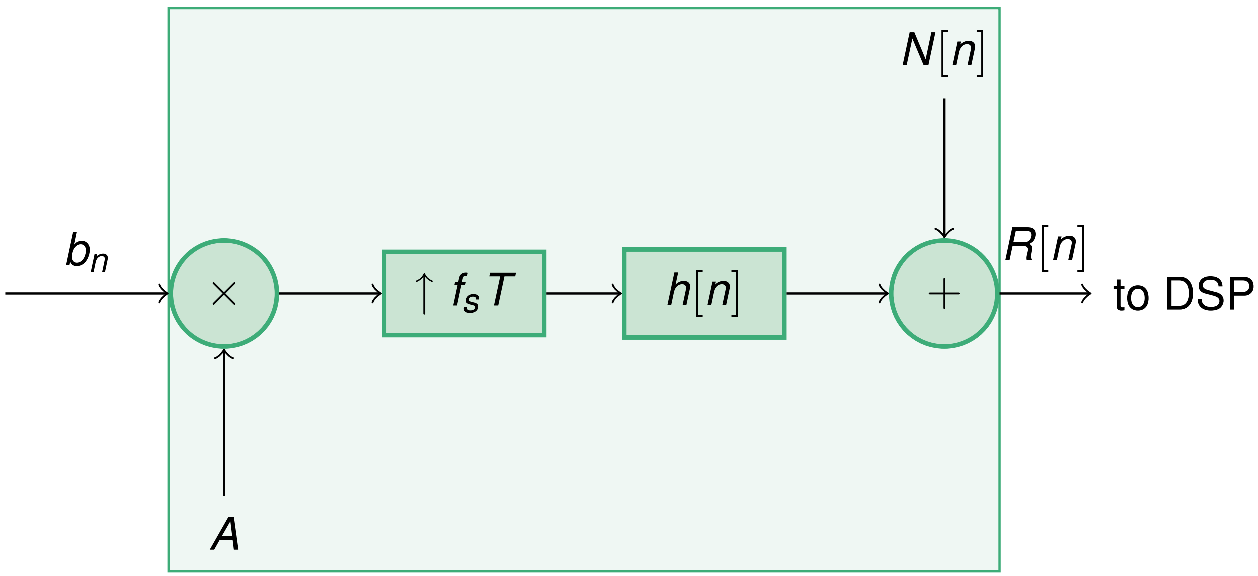
Towards the Discrete-Time Equivalent System

- The shaded portion of the system has a discrete-time input and a discrete-time output.
  - Can be considered as a discrete-time system.
  - **Minor problem:** input and output operate at different rates.



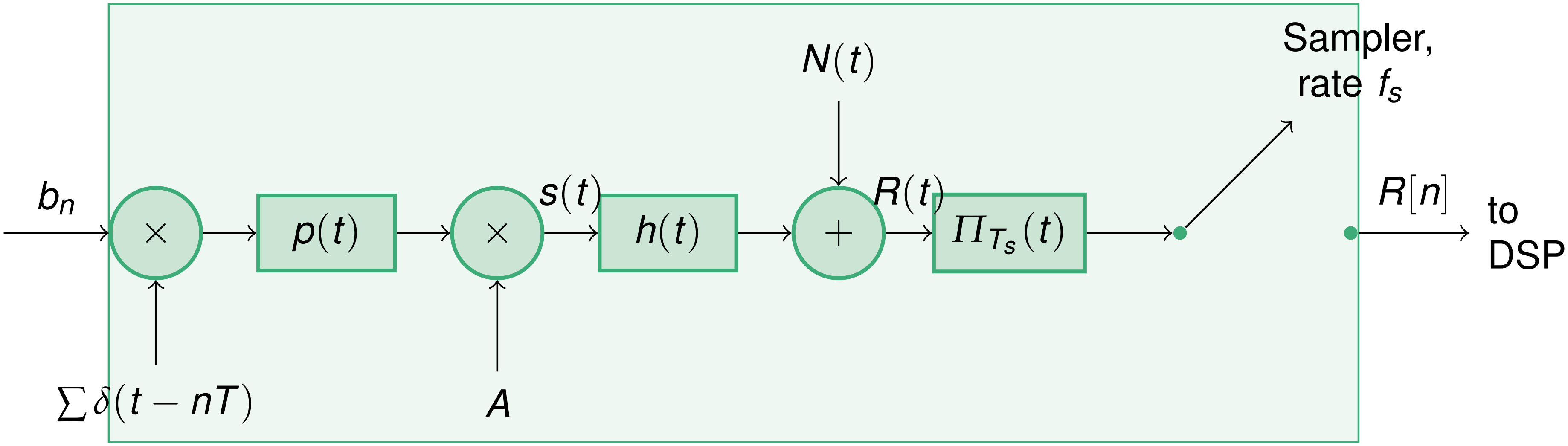
Discrete-Time Equivalent System

- The discrete-time equivalent system
  - is equivalent to the original system, and
  - contains only discrete-time signals and components.
- Input signal is up-sampled by factor $f_s T$ to make input and output rates equal.
  - Insert $f_s T - 1$ zeros between input samples.



Components of Discrete-Time Equivalent System

- **Question:** What is the relationship between the components of the original and discrete-time equivalent system?



Discrete-time Equivalent Impulse Response

- To determine the impulse response $h[n]$ of the discrete-time equivalent system:
  - Set noise signal N_t to zero,
  - set input signal b_n to unit impulse signal $\delta[n]$,
  - output signal is impulse response $h[n]$.
- Procedure yields:

$$h[n] = \frac{1}{T_s} \int_{nT_s}^{(n+1)T_s} p(t) * h(t) \, dt$$

- For high sampling rates ($f_s T \gg 1$), the impulse response is closely approximated by sampling $p(t) * h(t)$:

$$h[n] \approx p(t) * h(t)|_{(n+\frac{1}{2})T_s}$$

Discrete-time Equivalent Impulse Response

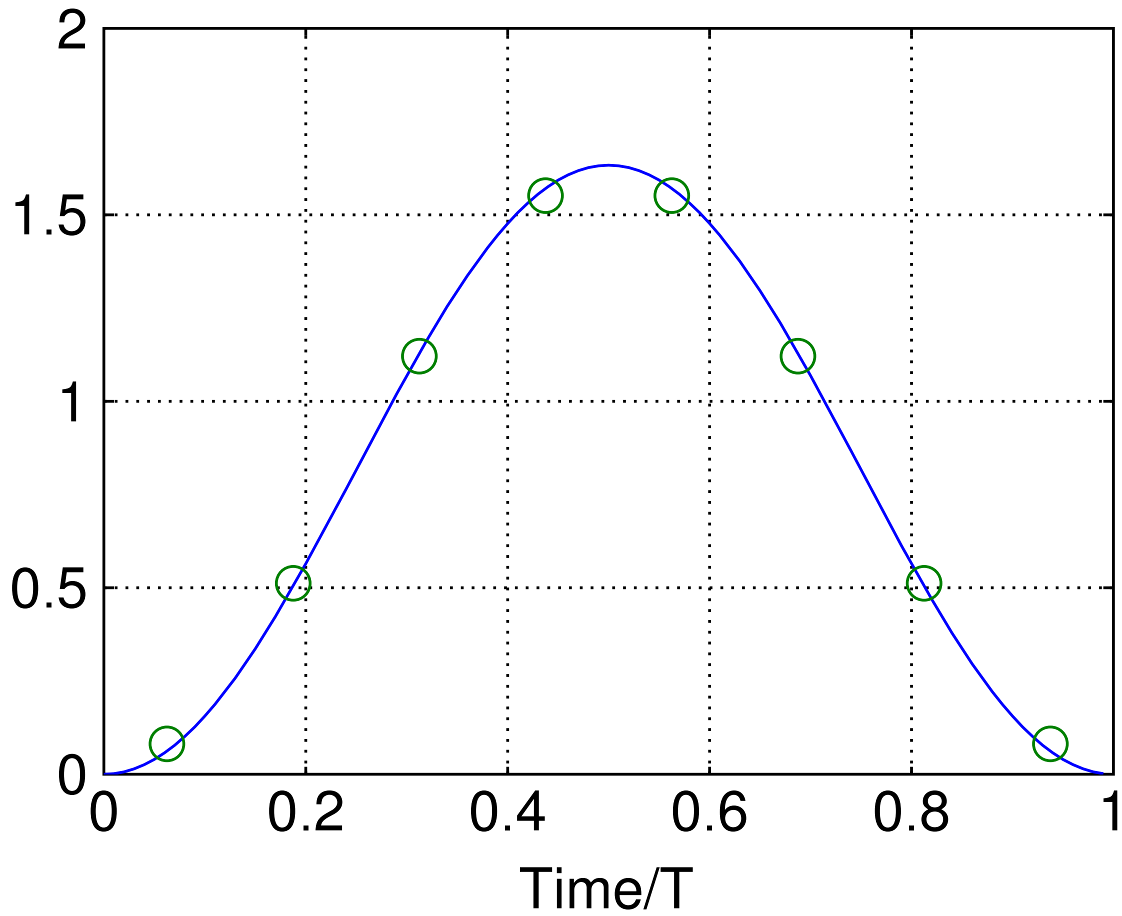


Figure: Discrete-time Equivalent Impulse Response ($f_s T = 8$)

Discrete-Time Equivalent Noise

- To determine the properties of the additive noise $N[n]$ in the discrete-time equivalent system,
  - Set input signal to zero,
  - let continuous-time noise be complex, white, Gaussian with power spectral density N_0,
  - output signal is discrete-time equivalent noise.
- Procedure yields: The noise samples $N[n]$
  - are independent, complex Gaussian random variables, with
  - zero mean, and
  - variance equal to N_0 / T_s.

Received Symbol Energy

- The last entity we will need from the continuous-time system is the received energy per symbol E_s.
  - Note that E_s is controlled by adjusting the gain A at the transmitter.
- To determine E_s,
  - Set noise $N(t)$ to zero,
  - Transmit a single symbol b_n,
  - Compute the energy of the received signal $R(t)$.
- Procedure yields:

$$E_s = \sigma_s^2 \cdot A^2 \int |p(t) * h(t)|^2 \, dt$$

  - Here, σ_s^2 denotes the variance of the source. For BPSK, $\sigma_s^2 = 1$.
  - For the system under consideration, $E_s = A^2 T$.