

UNIVERZITA PARDUBICE  
FAKULTA ELEKTROTECHNIKY  
A INFORMATIKY

BAKALÁŘSKÁ PRÁCE

2025

Vojtěch Roučka

Univerzita Pardubice

Fakulta elektrotechniky a informatiky

Zařízení pro měření impulsní charakteristiky dynamických systémů

Bakalářská práce

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky  
Akademický rok: 2023/2024

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Vojtěch Roučka**  
Osobní číslo: **I21066**  
Studijní program: **B0714A150008 Automatizace**  
Téma práce: **Zařízení pro měření impulsní charakteristiky  
dynamických systémů**  
Zadávací katedra: **Katedra řízení procesů**

### Zásady pro vypracování

Cílem práce je sestavení zařízení na bázi mikro počítačového modulu a aplikace v PC, které na základě měření vstupních a výstupních signálů lineárního dynamického systému umožní vypočítat jeho impulsní charakteristiku. Mikro počítačový modul, který bude provádět vlastní měření signálů v rozsahu 0–5 V, zašle data do aplikace v PC, kde bude provedeno zpracování dat a zobrazení výsledků v grafické podobě. Výpočet bude proveden minimálně dvěma různými metodami. Může být využit software MATLAB, ale nikoliv jeho toolbox pro identifikaci systémů. Pro ověření a porovnání výsledků bude sestaven vhodný dynamický systém z elektronických součástek.

Rozsah pracovní zprávy: **cca 40 stran**  
Rozsah grafických prací:  
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

CVEJN, J. *Automatizace 1 – obsah přednášek* [online]. Univerzita Pardubice, FEI, 2020. Elektronický studijní materiál.

NOSKIEVIČ, P. *Modelování a identifikace systémů*. Ostrava: Montanex, 1999.

LÁNÍČEK, R. *Elektronika, obvody, součástky, děje*, Praha: BEN – technická literatura, 1998.

Vedoucí bakalářské práce: **doc. Ing. Jan Cvejn, Ph.D.**  
Katedra řízení procesů

Datum zadání bakalářské práce: **15. prosince 2023**

Termín odevzdání bakalářské  
práce: **16. května 2025**

**Ing. Zdeněk Němec, Ph.D.** v.r.  
děkan

V Pardubicích dne 12. ledna 2024

L.S.

**Ing. Daniel Honc, Ph.D.** v.r.  
vedoucí katedry

Prohlašuji:

Práci s názvem *Zařízení pro měření impulsní charakteristiky dynamických systémů* jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 3.5.2025

Vojtěch Roučka v.r.

## **PODĚKOVÁNÍ**

Tímto bych rád poděkoval vedoucímu práce panu doc. Ing. Janu Cvejnovi, Ph.D., za odbornou pomoc a připomínky, které mi pomohly se zpravováním práce. Zároveň děkuji mé celé rodině za podporu při studiu a vytváření práce.

## **ANOTACE**

Tato práce se zabývá sestavením zařízení pro měření impulsní charakteristiky dynamických systémů. K účelu měření je využit mikropočítač Arduino UNO Rev3 a pro zpracování naměřených dat program Matlab. Program, který zpracovává data, je napsaný v jazyku Matlab, a program pro mikropočítač Arduino je napsaný v jazyce C++.

## **KLÍČOVÁ SLOVA**

impulsní charakteristika, jednotkový skok, Arduino Uno Rev3, Matlab, C++, dynamické systémy

## **TITLE**

A device for measuring impulse response of dynamic systems

## **ANNOTATION**

This study deals with the construction of a device for measuring the impulse characteristics of dynamic systems. For measurement, the Arduino Uno Rev 3 microcomputer is used, and the Matlab program is used to process the measured data. The program that processes the data is written in Matlab and the program for the Arduino microcomputer is written in language C++.

## **KEYWORDS**

impulse characteristic, Unit step, Arudino Uno Rev3, Matlab, C++, dynamic systems

# OBSAH

SEZNAM ILUSTRACÍ .....	10
ÚVOD .....	9
1 TEORETICKÁ ČÁST .....	10
1.1 Dynamické systémy .....	10
1.2 Rovnice popisující dynamické systémy .....	10
1.3 Přechodová charakteristika .....	11
1.4 Impulsní charakteristika .....	11
1.4.1 Vztah mezi impulsní charakteristikou a obrazovým přenosem .....	12
1.5 Konvoluce v dynamických systémech .....	12
1.5.1 Matematický základ konvoluce .....	12
1.5.2 Vlastnosti konvoluce .....	13
1.6 Výpočet impulsní charakteristiky pomocí dekonvoluce .....	14
1.6.1 Vztah mezi konvolucí a dekonvolucí .....	14
1.6.2 Využití dekonvoluce .....	14
1.7 Výpočet impulsní charakteristiky pomocí diferenciací přechodové charakteristiky ... .....	15
1.7.1 Princip diferenciací přechodové charakteristiky .....	15
1.7.2 Výpočet impulsní charakteristiky pomocí dekonvoluce .....	16
1.7.3 Výhody a nevýhody oproti dekonvoluci .....	16
1.8 Arduino a jeho využití v měřicích aplikacích .....	16
1.8.1 Platforma Arduino .....	16
1.8.2 Modely Arduina .....	17
1.9 Arduino Uno Rev3 .....	17
1.9.1 Úvod k Arduino Uno Rev3 .....	17
1.9.2 Technické specifikace Arduino Uno Rev3 .....	17
1.9.3 Programování Arduina .....	18
1.10 Programování v programu MATLAB .....	19
2 Praktická část .....	21
2.1 Experimentální dynamický systém .....	21

2.1.1	Upravené schéma zapojení .....	22
2.2	Programová realizace.....	23
2.2.1	Implementace programu na Arduino .....	24
2.2.2	Implementace programu pro MATLAB .....	26
2.2.3	Měření typu „S“ .....	28
2.2.4	Měření typu „R“ – Proměnný impulsní signál.....	31
3	Výsledky měření .....	36
3.1	Výsledek pro měření typu „S“ – jednotkový skok .....	36
3.2	Výsledek pro měření typu „R“ – Proměnný vstupní signál.....	39
	ZÁVĚR .....	43
	POUŽITÁ LITERATURA .....	44
	SEZNAM PŘÍLOH.....	45

## SEZNAM ILUSTRACÍ

Obrázek 1.1: Deska Arduino Uno Rev3 – pohled shora (Arduino, 2025).....	18
Obrázek 1.2: Popis pinů desky Arduino Uno Rev3 – Pinout (Zait, 2018) .....	19
Obrázek 1.3: Uživatelské prostředí vývojového prostředí MATLAB (fotografie vlastní, 2025) .....	20
Obrázek 2.1: Originální schéma obvodu (fotografie vlastní, 2025) .....	21
Obrázek 2.2: Upravené schéma pro účely práce (fotografie vlastní, 2025) .....	23
Obrázek 2.3: Inicializace a nastavení setup (fotografie vlastní, 2025).....	24
Obrázek 2.4: Měření S (fotografie vlastní, 2025).....	25
Obrázek 2.5: Měření R (fotografie vlastní, 2025) .....	25
Obrázek 2.6: Zpracování příkazů ze sériového portu (fotografie vlastní, 2025).....	26
Obrázek 2.7: Výběr typu měření z klávesnice (S nebo R) (fotografie vlastní, 2025) .....	27
Obrázek 2.8: Inicializace sériové komunikace s Arduinem (fotografie vlastní, 2025) .....	27
Obrázek 2.9: Přijetí zprávy z Arduina a odeslání příkazu pro měření (fotografie vlastní, 2025) .....	28
Obrázek 2.10: Ukázka interaktivního zadávání parametrů v prostředí MATLAB (fotografie vlastní, 2025) .....	28
Obrázek 2.11: Výpočet parametrů modelu systému (fotografie vlastní, 2025).....	29
Obrázek 2.12: Načtení dat z Arduina pro měření typu S (fotografie vlastní, 2025).....	29
Obrázek 2.13: Převod ADC hodnot na napětí a vytvoření časového vektoru (fotografie vlastní, 2025) .....	30
Obrázek 2.14: Výpočet přechodové a impulsní charakteristiky (fotografie vlastní, 2025).....	31
Obrázek 2.15: Výpočet analytické impulsní odezvy systému (fotografie vlastní, 2025).....	31
Obrázek 2.16: Načtení dat z Arduina pro měření typu R (fotografie vlastní, 2025) .....	32
Obrázek 2.17: Přepočtení napětí, odečtení offsetu a výpočet časového vektoru (fotografie vlastní, 2025) .....	33
Obrázek 2.18: Výpočet přechodové charakteristiky odečtením počáteční ustálené hodnoty a odstraněním offsetu (fotografie vlastní, 2025).....	33
Obrázek 2.19: Aplikace exponenciálního filtru na výstupní signál a výpočet přechodové charakteristiky s filtrem (fotografie vlastní, 2025) .....	34
Obrázek 2.20: Výpočet impulsní odezvy pomocí dekonvoluce a její následné vyhlazení (fotografie vlastní, 2025) .....	35
Obrázek 3.1: Výstupní napěťová odezva systému (fotografie vlastní, 2025).....	36

Obrázek 3.2: Přechodová charakteristika systému (fotografie vlastní, 2025) .....	37
Obrázek 3.3: Analytická impulsní charakteristika systému (fotografie vlastní, 2025) .....	37
Obrázek 3.4: Impulsní charakteristika – numerická derivace (fotografie vlastní, 2025).....	38
Obrázek 3.5: Impulsní charakteristika – dekonvoluce (fotografie vlastní, 2025).....	39
Obrázek 3.6: Vstupní signál (fotografie vlastní, 2025) .....	40
Obrázek 3.7: Výstupní napěťová odezva systému (fotografie vlastní, 2025).....	40
Obrázek 3.8: Reakce na vstupní signál (fotografie vlastní, 2025).....	41
Obrázek 3.9: Impulsní charakteristika systému – dekonvoluce (fotografie vlastní, 2025) .....	41

# ÚVOD

Impulsní charakteristika dynamických systémů je vedle přechodové charakteristiky důležitým nástrojem pro analýzu jejich chování. Umožňuje popsat reakci systému na krátký a intenzivní podnět – impulz. Na rozdíl od přechodové charakteristiky, která je technicky měřitelná přímo, musí být impulsní charakteristika určena výpočtem z výstupní odezvy systému na vhodně zvolený vstupní signál omezené velikosti. Analýza impulsní odezvy poskytuje důležité informace o dynamických vlastnostech systému, jako jsou rychlost reakce, stabilita a vlastnosti přenosu. Využívá se proto v řadě technických oblastí jako v elektronice, automatizaci, zpracování signálu nebo řízení.

Tato práce má za cíl sestavit zařízení na základě mikropočítačového modulu Arduino a program v rozhraní Matlab, které umožňuje změřit odezvu připojeného dynamického systému na generovaný vstupní signál a z těchto dat vypočítat jeho impulsní charakteristiku. Dané měření bude probíhat v rozsahu 0–5 V a veškerá data budou poté poslána do programu Matlab, kde jsou následně zpracována. Jako referenční model je využit elektronický dynamický systém sestavený s využitím pasivních součástek a operačních zesilovačů.

V této práci jsou využity dvě metody a výpočtu analýz pro impulsní charakteristiky – metoda diferenciace přechodové charakteristiky a metoda dekonvoluce. První z nich je založena na měření reakce systému na jednotkovou změnu vstupního signálu a výpočtu aproximace impulsní odezvy pomocí numerické derivace. Druhá metoda, dekonvoluce, umožňuje výpočet impulsní odezvy z reakce na libovolný známý vstupní signál, a je tak vhodná i v případech složitějšího nebo proměnného vstupu.

# 1 TEORETICKÁ ČÁST

## 1.1 Dynamické systémy

V dynamickém systému hodnoty závisí výstupní veličiny nejen na aktuálních vstupech, ale také na již předešlých hodnotách vstupů a výstupů. Stav dynamického systému popisujeme jako minimální soubor veličin, jejichž okamžité hodnoty nám udělují informace o historii daného systému, která nám při známém průběhu vstupů stačí k rozpoznání budoucího vývoje dalšího systému. Jednou z významných charakteristik, která popisuje chování dynamických systémů, je impulsní charakteristika.

## 1.2 Rovnice popisující dynamické systémy

Dynamické systémy popisujeme pomocí časových diferenciálních rovnic, které vyjadřují vztah vstupního signálu s výstupem signálu a vnitřní dynamikou systému. V praxi se nejčastěji využívají lineární diferenciální rovnice s konstantními koeficienty. Dynamické systémy lze však popsat nejen diferenciálními rovnicemi, ale také pomocí přenosových funkcí. Řešení těchto rovnic může být analytické či numerické a závisí na počátečních podmínkách i typu vstupního signálu. Alternativní popis pomocí přenosu v obrazové oblasti (např. Laplaceova transformace) zjednodušuje řešení a analýzu zejména v oblasti řízení. Pomocí přenosových funkcí lze jednoduše určit odezvu na různé typy vstupů a posoudit tak vlastnosti systému.

Obecnou formu lineární diferenciální rovnice  $n$ -tého řádu lze zapsat následovně:

$$a_n \frac{d^n y(t)}{dt^n} + a_{n-1} \frac{d^{n-1} y(t)}{dt^{n-1}} + \dots + a_1 \frac{dy(t)}{dt} + a_0 y(t) = b_m \frac{d^m u(t)}{dt^m} + b_{m-1} \frac{d^{m-1} u(t)}{dt^{m-1}} + \dots + b_0 u(t) \quad (1)$$

kde:

$y(t)$  je výstupní signál,

$u(t)$  je vstupní signál,

$a_n, a_{n-1}, \dots, a_0$  jsou koeficienty výstupu signály,

$b_m, b_{m-1}, \dots, b_0$  jsou koeficienty vstupu,

$n$  je řád diferenciální rovnice. To určuje dynamiku systému.

### 1.3 Přejchodová charakteristika

Přejchodová charakteristika je označena jako  $h(t)$  a je to odezva systému na jednotkový skok při nulových počátečních podmínkách. Ten je definován tak, že je nulový v čase  $t = 0$  a náhle změní svoji hodnotu z 0 na 1, kde zůstává konstantní. Přejchodová charakteristika je užitečným nástrojem pro analýzu dynamických systémů, protože popisuje dynamiku systému během přechodu ze vstupního podnětu do ustáleného stavu. Zároveň je snadno technicky měřitelná.

Jednotkový skok je matematicky definován takto:

$$u(t) = \begin{cases} 0 & \text{pro } t < 0 \\ 1 & \text{pro } t \geq 0 \end{cases} \quad (2)$$

Přejchodová charakteristika tedy vystihuje, jak systém reaguje na skokovou změnu vstupního signálu a jeho následné ustálení. Při praktickém měření přechodové charakteristiky se obvykle vychází z ustáleného stavu systému, na který je aplikován skokový vstupní signál vhodné velikosti. Výstupní odezva systému se zaznamenává a poté dále analyzuje. Tento způsob měření umožňuje získat informace o rychlosti odezvy, přítomnosti překmitu, době ustálení a dalších dynamických vlastnostech systému. Přejchodová charakteristika tak představuje důležitý nástroj nejen pro modelování systému, ale také pro praktickou identifikaci či optimalizaci.

### 1.4 Impulsní charakteristika

Impulsní charakteristika  $g(t)$  popisuje odezvu systému na jednotkový impuls, který se nazývá také jako Diracův impuls. Platí pouze při nulových počátečních podmínkách. Diracův impuls je označován jako  $\delta(t)$  a vyjadřuje teoretický signál, který je vyjadřován jako nekonečně úzký a nabývá nekonečné hodnoty v čase  $t = 0$  a zároveň integrál je přes celý čas trvání přímo roven 1. Diracův signál tedy není fyzikálně realizovatelný.

Diracův impuls definujeme jako

$$\delta(t) = \begin{cases} 0 & \text{pro } t \neq 0 \\ \infty & \text{pro } t = 0 \end{cases} \quad (3)$$

A zároveň také platí:

$$\int_{-\infty}^{\infty} \delta(t) dt = 1 \quad (4)$$

Impulsní charakteristika  $g(t)$  vyjadřuje celý popis dynamických vlastností. Pokud známe  $g(t)$ , tak lze zjistit, jaký bude výstup daného systému pro jakýkoliv vstupní impuls.

Výstupní signál systému lze také stanovit konvolucí vstupního signálu  $u(t)$  s impulsní charakteristikou  $g(t)$ , která je popsána vztahem:

$$y(t) = (u * h)(t) = \int_{-\infty}^{\infty} u(T) \cdot g(t - T) dT \quad (5)$$

#### 1.4.1 Vztah mezi impulsní charakteristikou a obrazovým přenosem

Přechodová charakteristika  $h(t)$  úzce souvisí s impulsní charakteristikou  $g(t)$ , protože představuje integrál impulsní charakteristiky. Tento vztah lze využít nejen pro výpočet, ale také pro analýzu chování systému:

$$h(t) = \int_0^t g(\tau) d\tau. \quad (6)$$

Z tohoto vztahu lze také vyvodit, že derivace přechodové charakteristiky je rovna impulsní charakteristice:

$$g(t) = \frac{dh(t)}{dt}. \quad (7)$$

Tento vztah je převzat ze studijních materiálů [CVEJN, 2024].

### 1.5 Konvoluce v dynamických systémech

Konvoluce v dynamických systémech představuje matematický nástroj pro popis vztahu mezi vstupem a výstupem systému. U lineárních časově invariantních systémů lze výstupní signál vyjádřit jako konvoluci vstupního signálu s impulsní charakteristikou systému. Tento přístup umožňuje analyzovat chování systému vůči libovolnému vstupnímu signálu, pokud je známá jeho impulsní odezva. Konvoluce se velmi často využívá při zpracování signálů, především při návrhu a realizaci filtrů, odstranění šumu nebo v oblasti rekonstrukce signálů. Díky schopnostem popisovat dynamiku systému přesně a přehledně se jedná o klíčový nástroj v oblastech teorie systémů.

#### 1.5.1 Matematický základ konvoluce

Jak již bylo zmíněno, konvoluce je matematická operace, která kombinuje dvě funkce z níž vyjde třetí funkce vyobrazující, jak jedna funkce ovlivňuje tu druhou v čase. Tuto konvoluci dvou funkcí  $f(t)$  a  $g(t)$  definujeme vztahem:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau. \quad (8)$$

Vzorec nám popisuje, jak se funkce  $f(t)$  sestavuje s druhou funkcí  $g(t)$  pomocí posunu a celkového součtu. V dynamických systémech se využívá ke stanovení výstupního systému, přitom  $g(t)$  je impulsní odezva daného systému a její vstupní signál  $u(t)$ . Funkce  $g(t)$  se také označuje jako váhová funkce nebo konvoluční jádro. Pokud platí, že vstupní signál  $u(t) = 0$  pro  $t < 0$  a impulsní odezva  $g(t) = 0$  pro  $t < 0$ , je možné vztah (8) nahradit konvolučním vztahem (9) (Noskievič, 1999):

$$y(t) = \int_0^t u(t - \tau)g(\tau)d\tau. \quad (9)$$

### 1.5.2 Vlastnosti konvoluce

U konvoluce máme důležité vlastnosti, díky kterým je klíčovým nástrojem pro analýzu dynamických systémů. Ty nejvýznamnější jsou komutativita, distributivita a asociativita. Ty umožňují pružnou manipulaci s funkcemi při modelování systému.

#### Komutativita

Pokud je operace konvoluce komutativní, znamená to, že pro dva vstupní signály  $f$  a  $g$  platí:

$$(f * g)(t) = (g * f)(t). \quad (10)$$

Z tohoto lze usoudit, že můžeme určit jakýkoliv ze vstupních signálů jako konvoluční jádro.

#### Distributivita

Pro libovolné tři vstupní signály  $f$ ,  $g$  a  $h$  platí distributivní vlastnost konvoluce:

$$(f * (g + h))(t) = (f * g) + (f * h)(t). \quad (11)$$

#### Asociativita

Pro tři vstupní signály  $f$ ,  $g$  a  $h$  se rovná:

$$((f * g) * h)(t) = (f * (g * h))(t). \quad (12)$$

Z toho vyplývá, že množina kontinuálních konvolučních jader může být nahrazena jedním větším, který je roven výsledku jejich konvoluce. Pokud však děláme kalkulace konvoluce s více filtry nad větším počtem vstupních obrazů, je vhodné si konvoluční jádro vypočítat předem.

## 1.6 Výpočet impulsní charakteristiky pomocí dekonvoluce

Jak již bylo zmíněno dříve, impulsní charakteristika systému charakterizuje dynamickou odezvu na jednotkový impuls. V praxi je mnohdy zapotřebí určit tuto charakteristiku experimentálně na základě vstupních a výstupních signálů. Právě od toho je tu metoda dekonvoluce, díky které můžeme vypočítat impulsní odezvu systému z naměřených hodnot. Pomocí dekonvoluce tedy získáme důležité informace o vlastnostech systému, jako jsou jeho přenosové vlastnosti či jeho stabilita.

### 1.6.1 Vztah mezi konvolucí a dekonvolucí

Výpočet původního vstupního signálu  $u(t)$  z impulsní odezvy  $g(t)$  a výstupního signálu  $y(t)$  je inverzní operací ke konvoluci (viz vztah (9)). Protože v časové oblasti nelze vstupní signál jednoduše získat dělením, je nutné využít dekonvoluci, která je však výpočetně náročnější a citlivější na přítomnost šumu v datech.

### 1.6.2 Využití dekonvoluce

V praxi je dekonvoluce často problematická kvůli numerické nestabilitě a šumu. Hlavním důvodem jsou naměřená data, ve kterých se objevují zdroje rušení, které přímo ovlivňují přesnost dekonvoluce. Proto se v praxi dekonvoluce realizují ve frekvenční oblasti pomocí Fourierovy transformace, a to především z důvodu výpočetní efektivity při zpracování většího množství dat.

Fourierova transformace převádí signály z časové domény do frekvenční oblasti, kde konvoluci nahradíme násobením jednotlivých funkcí. Matematicky to vyjadřujeme jako:

$$Y(f) = X(f) * H(f) \quad (13)$$

kde:

$Y(f)$  je výstupní signál ve frekvenční oblasti,

$X(f)$  je vstupní signál ve frekvenční oblasti,

$H(f)$  je přenosová funkce ve frekvenční oblasti.

Z tohoto vzorce lze usoudit, že pokud známe obraz výstupního signálu  $Y(f)$  a přenosovou funkci  $H(f)$ , lze získat obraz vstupního signálu  $X(f)$  dělením ve frekvenční oblasti:

$$X(f) = \frac{Y(f)}{H(f)} \quad (14)$$

## 1.7 Výpočet impulsní charakteristiky pomocí diferenciacce přechodové charakteristiky

Diferenciacce přechodové charakteristiky představuje alternativní způsob výpočtu impulsní charakteristiky. Oproti dekonvoluci je méně citlivá na numerické chyby a nevyžaduje práci ve frekvenční oblasti. Její využití je však omezeno pouze na případy, kdy je vstupem systému jednotkový skokový signál, a neumožňuje tedy práci s obecným vstupním signálem.

### 1.7.1 Princip diferenciacce přechodové charakteristiky

Diferenciacce přechodové charakteristiky se vytvoří ze vztahu mezi přechodovou a impulsní odezvou systému. Přechodová charakteristika  $h(t)$  označuje odezvu systému na jednotkový skok  $u(t)$ . Mezi impulsní charakteristikou  $g(t)$  a přechodovou charakteristikou  $h(t)$  platí vztah (7). Při praktickém zpracování signálů však nepracujeme s funkcemi v čase, ale s jejich vzorkovanými hodnotami. Z toho důvodu je nutné upravit výpočty do diskrétní formy. Pro aproximaci impulsní charakteristiky z přechodové odezvy lze využít numerickou derivaci pomocí zpětné diference. Tato aproximace je dána vztahem:

$$g[k] = \frac{y[k] - y[k-1]}{\Delta t} \quad (15)$$

kde  $g[k]$  je diskrétní impulsní odezva a  $y[k]$  vzorkovaná přechodová charakteristika. Tento vztah představuje numerickou derivaci pomocí zpětného rozdílu.

### 1.7.2 Výpočet impulsní charakteristiky pomocí dekonvoluce

Analogicky, při výpočtu pomocí dekonvoluce lze impulzní odezvu určit ze známých průběhů vstupního a výstupního signálu pomocí rekurentního vztahu:

$$g[k] = \frac{y[k] - \sum_{i=1}^k g[i] * u[k-i]}{u[0]} \quad (16)$$

kde:

$u[k]$  – rekonstruovaný vstupní signál v čase  $k$ ,

$y[k]$  – naměřená vstupní odezva systému v čase  $k$ ,

$g[i]$  – impulsní odezva systému,

$g[0]$  – hodnota impulsní odezvy při  $i = 0$ .

Tento vztah vychází z vlastností diskrétní konvoluce a je vhodný pro praktickou implementaci v prostředí MATLAB.

### 1.7.3 Výhody a nevýhody oproti dekonvoluci

Diferenciace přechodové charakteristiky a dekonvoluce jsou dvě metody, jak získat impulsní odezvu systému. Každá tato metoda má své výhody a nevýhody, které závisí na podmínkách měření a způsobu zpracování signálu. Za hlavní výhodu diferenciace považujeme její jednoduchost oproti metodě s využitím dekonvoluce, a to hlavně díky tomu, že nevyžaduje inverzní operace a dělení ve frekvenční doméně. Je také méně citlivá na malé hodnoty přenosové funkce. Diferenciace však předpokládá, že vstupním signálem je jednotkový skok, což omezuje její využití pouze na specifické typy vstupních signálů.

## 1.8 Arduino a jeho využití v měřicích aplikacích

### 1.8.1 Platforma Arduino

První vývojový set Platformy Arduino vznikl v Itálii v roce 2005. Autoři vytvořili Arduino, aby bylo možné používat mikropočítačové moduly, i když jste třeba jen student. Arduino je Open Source elektronická platforma založená na prostém softwaru a hardwaru. Umožňuje jednoduchý vývoj elektronických projektů. Arduino je navrženo tak, aby bylo přístupno všem typům uživatelů. Díky jednoduchému prostředí je Arduino využíváno od naprostých začátečníků přes studenty až po zkušené inženýry. Nemalou výhodou Arduina je primitivní programovací prostředí v programu Arduino IDE. Rozsáhlá komunita s podporou a dostupnost senzorů, příslušenství a součástek je jednou z dalších výhod pro uživatele. Jelikož

mikrokontrolery od platformy Arduino využívají speciální software s jazykem, který vychází z C/C++, je schopno interagovat se všemi různými vstupy, jako jsou například senzory. Také umí ovládat výstupy, jako jsou třeba motory a LED diody. Platforma Arduino je tedy ideální všestranný nástroj pro elektronické projekty.

## **1.8.2 Modely Arduina**

Platforma Arduino nabízí velké množství produktů, které jsou rozdílné svými technickými parametry. Tím se myslí velikost, počet výstupních a vstupních pinů a možnosti připojení. Nejvíce rozšířené a nejčastěji používané jsou modely Arduino Uno, Arduino Nano, Arduino Mega a Arduino Leonardo. Každý model je určen k jiným aplikacím podle toho, jaké jsou jeho specifické vlastnosti.

## **1.9 Arduino Uno Rev3**

### **1.9.1 Úvod k Arduino Uno Rev3**

Arduino Uno Rev3 je jedním z nejrozšířenějších a nejpoužívanějších modelů z řady Arduino. Představuje všestranný a levný mikrokontroler, který je vhodný jak pro úplné začátečníky, tak pro pokročilé a náročnější uživatele. Díky své univerzálnosti a snadnosti využití je standardem ve velké komunitě uživatelů.

### **1.9.2 Technické specifikace Arduino Uno Rev3**

V mikropočítači Arduino Uno Rev3 je řízen mikrokontrolerem ATmega328P, který je pro nás díky svým vlastnostem vhodným řešením pro měřicí aplikace se zaměřením na zpracování a snímání jak analogových, tak digitálních signálů. Základním parametrem mikrokontroleru je jeho taktovací frekvence 16 MHz, díky které je zaručený dostatečný výpočetní výkon pro zpracování dat v reálném čase. Arduino Uno Rev3 má k dispozici 32 kB flash paměti, z které se 0,5 kB využívá jako bootloader. Pro další ukládání využívá 1 kB EEPROM na uchování kalibračních dat.

Klíčový prvek pro naši práci je 10bitový analogově-digitální převodník (ADC), který umožňuje snímat analogové signály v rozsahu 0 až 5 V, tedy plně odpovídá požadavkům na naše měření výstupních a vstupních signálů dynamických signálů. Arduino Uno Rev3 pro uživatele nabízí 6 analogových vstupů a 14 digitálních výstupů/vstupů. Šest z těchto digitálních pinů také podporuje PWM výstup.



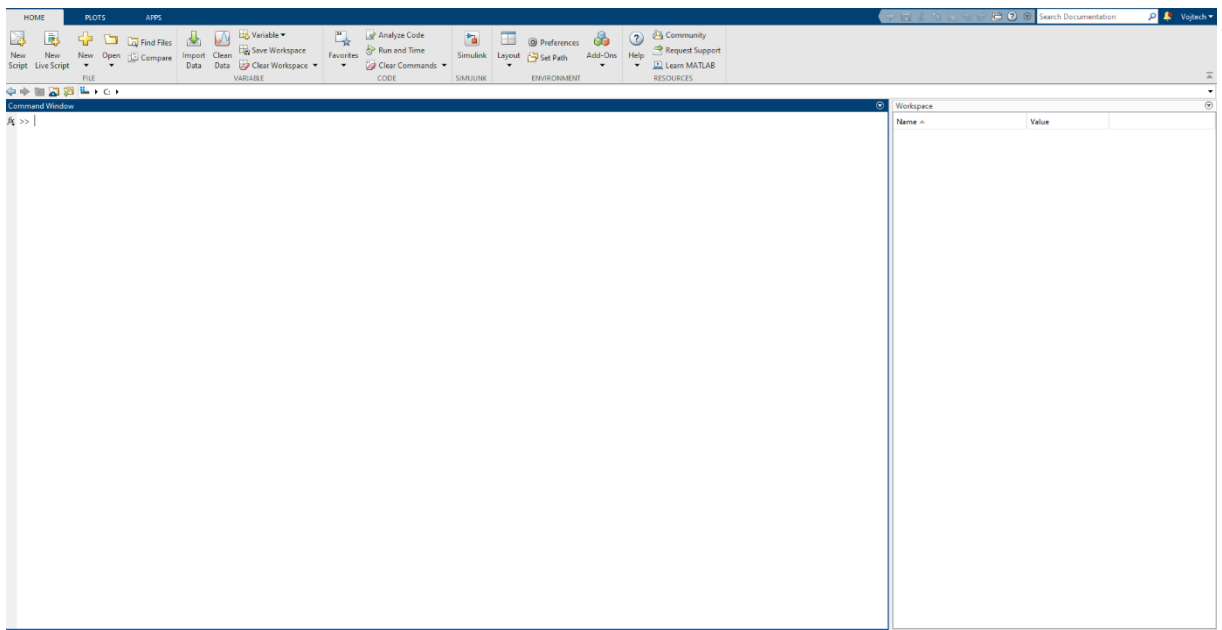
**Obrázek 1.1:** Deska Arduino Uno Rev3 – pohled shora (Arduino, 2025)

### 1.9.3 Programování Arduina

Arduino Uno Rev3 se programuje v jazyce Arduino C/C++. Vychází z klasického programovacího jazyka C++, který je však přizpůsoben specificky pro mikropočítače Arduino. K programování se využívá vývojové prostředí Arduino IDE, které uživatelům nabízí nástroje pro psaní, nahrávání a kompilování kódu přímo do mikrokontroleru. Struktura programu se skládá ze dvou částí. První z nich je `setup()`, jež proběhne pouze jednou, a to při spuštění programu. Slouží k inicializaci hardwarových periférií, jako jsou nastavení komunikačního rozhraní či konfigurace vstupních a výstupních pinů. Druhá funkce je `loop()`, je to hlavní část programu a vykonává se v nekonečné smyčce. V této části se zpracovávají data, měří signály, odesílají výsledky a jakákoliv další logika řízení měřicího procesu. Výhodou programování v Arduino prostředí je dostupnost jejich knihoven funkcí. Usnadňují práci s různými zařízeními, senzory a komunikačními protokoly jako UART, I2C nebo SPI. V naší práci je Arduino využito k měření signálů a řízení výstupů. Program nám zajišťuje sběr dat, jejich zpracování a přenos do aplikace do PC.



výpočetní algoritmy vycházející z teoretických vztahů uvedených v předchozích kapitolách. MATLAB dále slouží k vizualizaci výsledků. Pomocí dostupných grafických funkcí jsou vykresleny průběhy vstupních a výstupních signálů a také výsledná impulsní charakteristika. Tyto grafy umožňují lépe analyzovat odezvu zkoumaného systému a porovnat jednotlivé metody výpočtu. Pro základní operace s daty se využívají funkce jako plot, subplot, legend nebo xlabel, které právě slouží k tvorbě přehledných grafů. Velkou výhodou programu MATLAB je možnost ladění a testování algoritmů pomocí skriptových souborů, které dovolují strukturovat výpočty do přehledných bloků. V případě složitějších výpočtů lze funkce seskupit do vlastních definovaných funkcí, nebo využít grafické nástroje, jako je aplikace live editor. Ta umožňuje kombinovat výpočetní kód s textem a grafy v jednom dokumentu. Pro účely této práce byly v MATLABu vytvořeny vlastní skripty, které zajišťují automatické načítání dat, jejich předzpracování, výpočet impulsní charakteristiky a vizualizaci výsledků. Výsledkem je efektivní nástroj pro analýzu dynamických systémů, snadné porovnání metod a prezentaci výstupů.

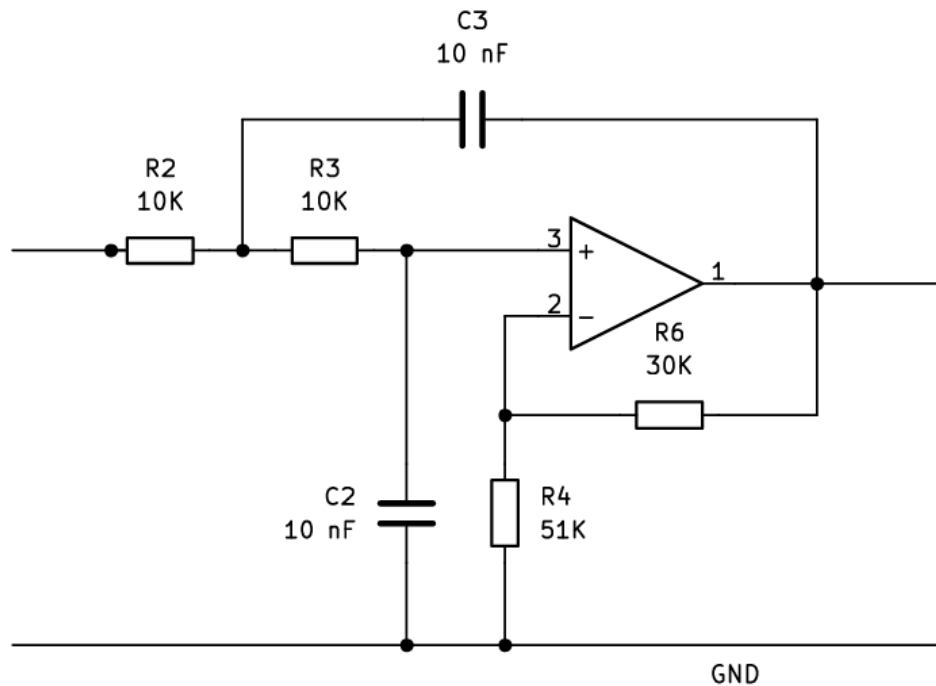


**Obrázek 1.3:** Uživatelské prostředí vývojového prostředí MATLAB (fotografie vlastní,2025)

## 2 Praktická část

### 2.1 Experimentální dynamický systém

Pro účely této práce byl realizován elektronický systém, který svým chováním odpovídá lineárnímu dynamickému systému druhého řádu. Tento experimentální systém slouží k přímému měření odezvy na různé typy vstupních signálů a k ověření metod výpočtu impulsní charakteristiky. Návrh vychází z odborné literatury, avšak byl upraven a rozšířen s ohledem na požadavky měření. Původní návrh měřicího systému vycházel ze zapojení dolní propusti druhého řádu, které bylo převzato z odborné literatury Laníček[2001]. Zapojení používá operační zesilovač zapojený jako aktivní filtr a poskytuje tak vyžadované dynamické vlastnosti systému.



**Obrázek 2.1:** Originální schéma obvodu (fotografie vlastní,2025)

Obvod realizuje chování kmitavého systému druhého řádu s přenosem:

$$G(s) = \frac{1}{(RC)^2 s^2 + 3RCs + 1} \quad (17)$$

kde zesílení  $A=1.5$  bylo zvoleno podle doporučení, viz [Láníček, 2021]. Tvar přenosu odpovídá systému druhého řádu s časovou konstantou  $T=R*C$  a tlumícím faktorem  $a=1.5$ . Výběrem hodnot  $R=10\text{ k}\Omega$ ,  $C=10\text{ nF}$  dostáváme  $RC=100\text{ }\mu\text{s}$ . Analytická impulsní charakteristika systému odpovídající tomuto přenosu je dána vztahem:

$$g(t) = \left(\frac{A}{T^2 * \omega}\right) * e^{-at} * \sin(\omega t), \quad (18)$$

kde:

$A$  je zesílení,

$T$  je časová konstanta, kde  $T=R*C$

$\omega$  je úhlová frekvence,

$a$  je tlumící faktor, kde  $a = (3-A) / T$ .

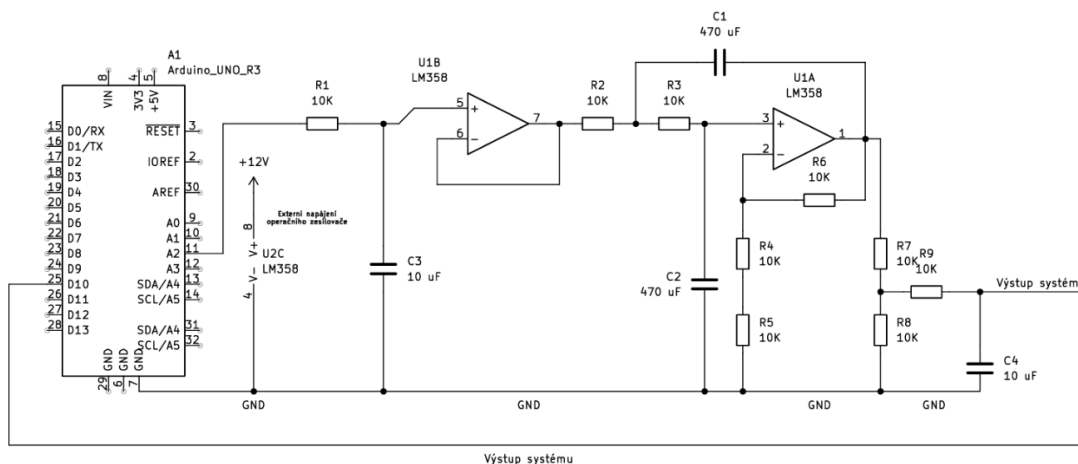
Tato rovnice byla použita při výpočtu analytické impulsní odezvy systému a odpovídá přenosu definovanému ve vztahu (17)

Vstupní signál je zpracován RC články a operačním zesilovačem. Operační zesilovač tvoří aktivní část filtru. Hodnoty rezistorů a kapacitorů určují časové konstanty, jež určují přenosové vlastnosti systému.

### 2.1.1 Upravené schéma zapojení

Jak bylo zmíněno, pro potřeby této práce bylo původní zapojení upraveno. Úpravy spočívaly v přidání nových prvků pro přizpůsobení výstupního signálu analogové-digitálnímu převodníku mikropočítače Arduino Uno Rev3.

Výsledné schéma zapojení je znázorněno na obrázku [2.2]. Měřicí systém je navržen v prostředí KiCad, které umožňuje přehlednou dokumentaci.



**Obrázek 2.2:** Upravené schéma pro účely práce (fotografie vlastní, 2025)

Na vstup měřeného systému je přiveden signál, který prochází přes odpor R1 a kondenzátor C3, toto zapojení slouží k ochraně a filtraci. Signál dále vede přes operační zesilovač U1B (LM358) zapojený jako sledovač napětí, sloužící k oddělení zdroje signálu od měřící části.

Dynamické vlastnosti systému druhého řádu jsou realizovány pomocí dvou RC článků, tvořených rezistory R2, R3 a kondenzátory C1, C2. Tyto prvky nám vytváří dolní propust. Výstupní signál je dále zpracován zesilovačem U1A (LM358) a upraven odporovým děličem R7, R8. Dělič upravuje výstupní napětí na rozsah vhodný pro analogový vstup Arduina (0–5 V). Kondenzátor C4 slouží k filtraci signálu. Operační zesilovače jsou napájeny ze zdroje +12 V.

## 2.2 Programová realizace

Kromě hardwarové části pro měření systému je důležité mít programové řešení. Programové řešení obsahuje návrh kódu pro mikropočítač Arduino Uno Rev3, který nám zajišťuje generování vstupního signálu a sběr dat, tak i následné zpracování a počítání s daty v aplikaci MATLAB. Obě části mezi sebou komunikují pomocí sériového rozhraní a tak efektivně řídí měření a analýzu signálů. V této kapitole je popsáno využití konkrétních částí programu, jejich propojení a komunikace mezi Arduinem a počítačem. Úkolem programové realizace je zajistit průběh měření a přesné vzorkování signálu, aby bylo dále možné vypočítat impulsní charakteristiku z naměřených hodnot.

## 2.2.1 Implementace programu pro Arduino

Pro měření a sběr dat byl vytvořen program pro mikropočítač Arduino Uno Rev3, jenž nám zajišťuje generování vstupního signálu a následné snímání odezvy výstupu měřeného systému. Program také zajišťuje komunikaci s PC pomocí sériového rozhraní s přenosovou rychlostí 115200 baudů.

```
#define MAX_SAMPLES 400 // Max počet vzorků

const int outputPin = 10;
const int inputPin = A1;

void measureUnitStep(unsigned long delta, unsigned int number_of_samples);
void measureSinglePulse(unsigned long delta, unsigned int number_of_samples); // Deklarace

void setup() {
  pinMode(outputPin, OUTPUT);
  pinMode(inputPin, INPUT);
  analogWrite(outputPin, 0);
  Serial.begin(115200);

  while (!Serial) {}
  Serial.println("Ready");
}
```

Obrázek 2.3: Inicializace a nastavení setup (fotografie vlastní, 2025)

Po spuštění zařízení se (viz obrázek 2.3) spustí inicializace vstupních a výstupních pinů. Následně Arduino čeká na textový příkaz vyslaný z PC, který určuje, jaký typ měření proběhne. V textovém příkazu se také nachází parametr pro časový interval mezi vzorky a celkový počet vzorků. Příkazy se posílají ve formátu:

S <delta> <samples> – měření jednotkového skoku,

R <delta> <samples> – měření proměnného impulsního signálu.

Po přijetí příkazu „S“ je na výstupní pin generovaná skoková změna signálu pomocí PWM. Arduino současně zahájí vzorkování výstupního signálu daného měřeného systému (viz obrázek 2.4). Vzorkování pracuje s rovnoměrným časovým rozestupem parametrem delta. Naměřené hodnoty se následně odesílají do PC, kde probíhá jejich následné zpracování.

```

analogWrite(outputPin, 255);
delay(100);
Serial.println("DATA_START");
unsigned long startTime = micros();
for (unsigned int i = 0; i < number_of_samples; i++) {
    unsigned long targetTime = startTime + i * delta;
    while (micros() < targetTime) {

    }
    Serial.println(analogRead(inputPin));
}

```

**Obrázek 2.4:** Měření S (fotografie vlastní, 2025)

Při použití příkazu R je generován náročnější a složitější vstupní impulsní signál, jehož hodnota se v čase mění podle předem daného průběhu. Během měření Arduino odesílá dvě hodnoty (viz obrázek 2.5). Odesílá výstupní a vstupní hodnoty PWM signálu. Díky tomu je možné analyzovat chování systému známého vstupu i výstupu.

```

int inputSignal = 50;
if (elapsedTime > 0 && elapsedTime <= 10000000) {
    inputSignal = 150;
} else if (elapsedTime > 10000000 && elapsedTime <= 20000000) {
    inputSignal = 255;
} else if (elapsedTime > 20000000) {
    inputSignal = 200;
}

```

**Obrázek 2.5:** Měření R (fotografie vlastní, 2025)

Zpracování příkazů probíhá v nekonečné smyčce loop(), kde Arduino čeká na příchozí instrukci ze sériové komunikace. Na základě typu příkazu (S nebo R) se volá odpovídající funkce (viz obrázek 2.6).

```

void loop() {
  if (Serial.available() > 0) {
    String commandLine = Serial.readStringUntil('\n');
    commandLine.trim();
    Serial.print("Received command: ");
    Serial.println(commandLine);
    char commandChar;
    unsigned long delta;
    unsigned int number_of_samples;
    if (commandLine.startsWith("S")) {
      sscanf(commandLine.c_str(), "%c %lu %u", &commandChar, &delta, &number_of_samples);
      measureUnitStep(delta, number_of_samples);
    }
    else if (commandLine.startsWith("R")) {
      sscanf(commandLine.c_str(), "%c %lu %u", &commandChar, &delta, &number_of_samples);
      measureSinglePulse(delta, number_of_samples);
    }
  }
}
}

```

**Obrázek 2.6:** Zpracování příkazů ze sériového portu (fotografie vlastní, 2025)

Naměřená data jsou odeslána v rámci bloků, které jsou označeny řetězcem DATA\_START a DATA\_END, jež slouží jako jednoduché značení datového přenosu v MATLABu. Pro zajištění jednotlivých vzorků je použita funkce micros(), která umožňuje měřit čas v mikrosekundách. PWM signál je generován funkcí analogWrite() a analogový vstup je měřen pomocí analogRead().

## 2.2.2 Implementace programu pro MATLAB

Jako nástroj pro analýzu, zpracování dat a uživatelské rozhraní byla do této práce zvolena platforma MATLAB. Tento nástroj nám umožňuje efektivní zpracování a sběr dat, grafické znázornění a výpočty impulsní charakteristiky měřeného dynamického systému. Tyto kroky jsou automatizovány pomocí vytvořeného skriptu, který je rozdělen do několika částí. První část se využívá k interaktivnímu zadávání parametrů pro měření. Po zadání parametrů MATLAB naváže sériovou komunikaci s Arduinem a je provedeno měření a sběr dat. Na závěr probíhá zpracování všech dat různými metodami. Skript je vytvořen pro dva režimy měření – jednotkový skok a proměnný impulsní signál.

### Komunikace s Arduinem a uživatelské prostředí

Na začátku skriptu v MATLABU probíhá interaktivní komunikace s uživatelem, jejímž cílem je zadání parametrů měření. Program nejprve vyžaduje zadání typu měření, tedy zda se jedná o jednotkový skok, nebo proměnný impulsní signál. Následně je uživatel vyzván k zadání hodnoty časového kroku mezi vzorky v mikrosekundách a počtu vzorků pro měření, přičemž maximální povolený počet je 400. Zadávané hodnoty jsou programem kontrolovány, aby bylo

zabráněno chybnému vstupu, a uživatel je v případě nesprávného zadání vyzván k opravě. Po úspěšném zadání všech parametrů pokračuje skript generováním výstupního signálu a sběrem dat. Ukázka části kódu, která realizuje proces interaktivního zadávání parametrů, je uvedena níže (viz obrázek 2.7).

```
valid_input = false;
while ~valid_input
    disp('Vyberte typ měření:');
    disp(' S - Jednotkový skok');
    disp(' R - Pravidelné impulsy');
    typ_mereni = input('Zadejte typ měření (S nebo R): ', 's');
    typ_mereni = upper(typ_mereni);

    if typ_mereni == "S" || typ_mereni == "R"
        valid_input = true;
    else
        disp('Neplatná hodnota! Zadejte pouze S nebo R. ');
    end
end
max_samples = 400;
```

**Obrázek 2.7:** Výběr typu měření z klávesnice (S nebo R) (fotografie vlastní, 2025)

Po zadání parametrů je navázána komunikace s Arduinem. Důležité je nastavení správného portu (např. COM4) a rychlostí přenosu (např. 115200 baud) (viz obrázek 2.8).

```
port_arduino = 'COM4'; % Nastavte správný COM port
rychlost_prenosu = 115200; % Musí odpovídat rychlosti v Arduinu

% Vytvoření sériového objektu pro komunikaci s Arduinem
objekt_arduino = serialport(port_arduino, rychlost_prenosu);
configureTerminator(objekt_arduino, "LF");
objekt_arduino.Timeout = 300;
pause(2);
```

**Obrázek 2.8:** Inicializace sériové komunikace s Arduinem (fotografie vlastní, 2025)

Arduino po připojení odesílá zprávu do MATLABu. Poté dojde odeslání příkazu s parametry měření ve formátu S/R <delta> <počet\_vzorků> (viz obrázek 2.9).

```

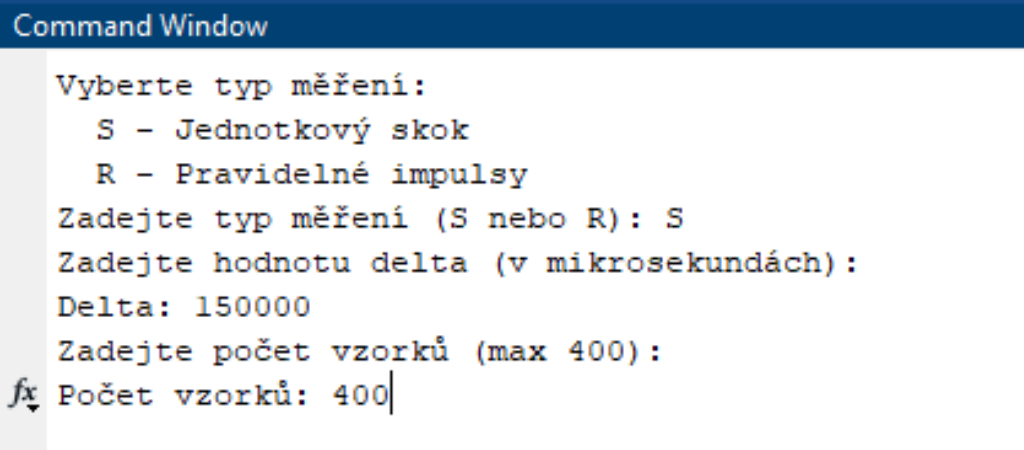
if objekt_arduino.NumBytesAvailable > 0
    initial_msg = readline(objekt_arduino);
    disp(['Arduino říká: ', initial_msg]);
end

disp('Odesílám příkaz do Arduina pro zahájení měření...');
commandString = sprintf('%s %d %d', typ_mereni, delta, number_of_samples);
writeline(objekt_arduino, commandString);

```

**Obrázek 2.9:** Přijetí zprávy z Arduina a odeslání příkazu pro měření (fotografie vlastní, 2025)

Díky této části programu zajišťuje výměnu dat mezi MATLABEM a Arduinem, což je klíčové pro synchronizaci měření. Uživatel přitom nemusí přímo pracovat s kódem, ale pouze s interaktivním prostředím v konzoli MATLABu.



```

Command Window
Vyberte typ měření:
  S - Jednotkový skok
  R - Pravidelné impulsy
Zadejte typ měření (S nebo R): S
Zadejte hodnotu delta (v mikrosekundách):
Delta: 150000
Zadejte počet vzorků (max 400):
fx Počet vzorků: 400|

```

**Obrázek 2.10:** Ukázka interaktivního zadávání parametrů v prostředí MATLAB (fotografie vlastní, 2025)

### 2.2.3 Měření typu „S“

Po přijetí odpovědi z Arduina čeká MATLAB na označení DATA\_START a následně načte naměřený počet vzorků. Měření typu S odpovídá skokové změně na vstupu. Tyto hodnoty jsou převedeny na napětí a dále využity k výpočtu přechodové charakteristiky systému. Nejdříve je nutné definovat základní parametry měřeného systému (viz obrázek 2.11). Mezi ně patří hodnota odporu a kapacity, které určují dynamické vlastnosti systému. Tyto hodnoty slouží k výpočtu časové konstanty, tlumícího faktoru a vlastní frekvence. Výpočty vycházejí z fyzikálních vztahů pro systém druhého řádu.

```

R = 10000; % Rezistor (10 kOhm)
C = 0.00047; % Kondenzátor (470 uF)
zesileni = 1.5; % Zesílení
RC_konstanta = R * C; % Časová konstanta
tlumici_faktor = (3 - zesileni) / (2 * RC_konstanta);
vlastni_frekvence = sqrt(1 / (RC_konstanta)^2 - tlumici_faktor^2);
faktor_delice = 0.5; % Faktor odporového děliče

```

**Obrázek 2.11:** Výpočet parametrů modelu systému (fotografie vlastní, 2025)

Tento úsek definuje hodnoty odporu a kapacity, které jsou využity v měřeném systému. Následně jsou vypočteny parametry, jako jsou časová konstanta systému RC, tlumicí faktor a vlastní frekvence. Parametr *faktor\_delice* pak odpovídá odporovému děliči, který je zapojen na výstupu systému. RC konstanta je časová konstanta systému, jež určuje rychlost reakce systému. Čím vyšší je hodnota RC, tím pomaleji reaguje. Tlumicí faktor určuje míru tlumení oscilací v systému. Vyšší hodnota představuje rychlejší ustálení. Tato část slouží jako příprava proměnných pro analytické výpočty, které jsou důležité pro porovnání výsledků.

Po výběru režimu „S“ MATLAB čeká na signál načítající hodnoty, které byly odeslané z mikropočítače Arduina. Tyto hodnoty znázorňují odezvu systému na jednotkový skok. Samotné čtení dat z Arduina začíná čekáním na řetězec *DATA\_START*, kterým Arduino značí počátek přenosu (viz obrázek 2. 12). Data jsou ukládána do pole *data*. Po přenosu je ukončen přijetím řetězce *DATA\_END*.

```

if typ_mereni == "S"
    % --- Čtení dat z Arduina pro měření typu S ---
    disp('Čekám na DATA_START od Arduina...');
    data_start = readline(objekt_arduino);
    while ~strcmp(strtrim(data_start), 'DATA_START')
        data_start = readline(objekt_arduino);
    end
    disp('Přijato DATA_START, začínám číst data...');

    data = zeros(1, number_of_samples);
    for i = 1:number_of_samples
        line = readline(objekt_arduino);
        data(i) = str2double(line);
    end

    data_end = readline(objekt_arduino);
    while ~contains(data_end, 'DATA_END')
        data_end = readline(objekt_arduino);
    end
    disp('Přijato DATA_END, data byla úspěšně načtena.');
```

**Obrázek 2.12:** Načtení dat z Arduina pro měření typu S (fotografie vlastní, 2025)

Po přijetí dat je potřeba je zpracovat. Zpracováním je myšleno převedení dat z 0-1023 (10bitový ADC převodník Arduina) na odpovídající hodnoty napětí. Převod dat je založen na hodnotě referenčního napětí  $V_{ref}$ , které bylo změřeno multimetrem. Dále je vytvořen časový vektor  $cas$ , který vyjadřuje časové umístění každého vzorku dle intervalu  $\delta$  (viz obrázek 2.13).

```
% Převod ADC hodnot na napětí
Vref = 4.96;
napeti = (double(data) / 1023) * Vref;

% Vytvoření časového vektoru s konstantním krokem
delta_s = delta / 1e6; % Převod mikrosekund na sekundy
cas = (0:length(napeti)-1) * delta_s;
```

**Obrázek 2.13:** Převod ADC hodnot na napětí a vytvoření časového vektoru (fotografie vlastní, 2025)

Hodnoty jsou zároveň převedeny na typ `double`, což nám umožňuje s nimi následně numericky pracovat. Tyto připravené časové a napěťové vektory slouží jako základ pro zpracování signálu. Převod napětí probíhá dle vztahu:

$$U = \frac{\text{hodnota } ADC}{1023} * V_{ref} \quad (19)$$

kde:

$U$  – počítané napětí [V],

$Hodnota\ ADC$  – naměřená hodnota z analogově-digitálního převodníku (0-1023),

$V_{ref}$  – referenční napětí.

Prvním způsobem výpočtu impulsní odezvy je derivace přechodové charakteristiky. Nejprve se ze signálu odečte ustálená hodnota a výsledná charakteristika je normalizovaná. Pro snížení šumu je na průběh uděláno vyhlazení, které je realizované pomocí klouzavého průměru (*movmean*), jenž z každé trojice sousedních bodů vytvoří jejich průměr. Samotná impulsní odezva je získána numerickou derivací pomocí funkce *diff*, která vypočítává difference sousedních hodnot, čímž aproximuje první derivaci vzhledem k času (viz obrázek 2.14).

```

vstupni_skok = max(napeti) - min(napeti);
ustalena_hodnota = mean(napeti(1:10));
prechodova_charakteristika = (napeti - ustalena_hodnota) / Vref;
vyhlazena_prechodova_char = movmean(prechodova_charakteristika, 3);
impulsni_odezva = diff(vyhlazena_prechodova_char) ./ diff(cas);
cas_impulsu = cas(2:end);
vyhlazena_impulsni_odezva = movmean(impulsni_odezva, 2);

```

**Obrázek 2.14:** Výpočet přechodové a impulsní charakteristiky (fotografie vlastní, 2025)

Tato metoda poskytuje jednoduchý numerický odhad impulsní odezvy systému a zároveň slouží ke srovnání pro analytické a matematické výpočty.

Pro ověření výsledků je možné provést porovnání s analyticky vypočítanou impulsní charakteristikou systému, jejíž tvar odpovídá očekávanému chování systému druhého řádu. Výpočet impulsní charakteristiky vychází z přenosové funkce systému, která byla uvedena v předchozí části práce (kapitola 2.1.). Tento přístup umožňuje porovnat naměřené odezvy s teoretickými hodnotami a ověřit tak správnost měření i zpracování dat (viz obrázek 2.15).

```

% Analytická impulsní charakteristika

impulsni_odezva_analyticka = faktor_delice * (zesileni / (RC_konstanta)^2 / vlastni_frekvence) ...
    * exp(-tlumici_faktor * cas) .* sin(vlastni_frekvence * cas);

```

**Obrázek 2.15:** Výpočet analytické impulsní odezvy systému (fotografie vlastní, 2025)

Poslední metodou využitou pro výpočet impulsní charakteristiky je numerická dekonvoluce. Princip numerické dekonvoluce je dán matematickým vztahem, který umožňuje rekonstruovat vstupní signál ze známé impulsní odezvy systému a naměřených dat.

Po udělení dekonvoluce na základě diskretní přechodové charakteristiky a vstupního signálu lze spočítat jednotlivé body impulsní odezvy systému. Výpočet využívá řešení obráceného konvolučního vztahu, kdy pro každý časový okamžik  $k$  je hodnota  $g_k$  spočtena dle vzorce (9), který je uveden ve vztahu (11.12) ve zdroji (Noskiewič, 1999).

## 2.2.4 Měření typu „R“ – Proměnný impulsní signál

Po přijetí příkazu pro měření typu R MATLAB počká na označení DATA\_START, kde poté následně přijímá vstupní a výstupní hodnoty ze systému.

```

elseif typ_mereni == "R"
disp('Waiting for DATA_START from Arduino...');
line = readline(objekt_arduino);
while ~strcmp(strtrim(line), 'DATA_START')
    line = readline(objekt_arduino);
end
disp('DATA_START received, starting to read data...');

input_data = [];
output_data = [];
data_section = '';

while true
    line = readline(objekt_arduino);
    line = strtrim(line);

    if strcmp(line, 'DATA_VALUES_START')
        data_section = 'data_values';
        continue;
    elseif strcmp(line, 'DATA_VALUES_END')
        data_section = '';
        continue;
    elseif strcmp(line, 'DATA_END')
        break;
    end

    if strcmp(data_section, 'data_values')

        values = strsplit(line, ',');
        if length(values) == 2
            input_data(end+1) = str2double(values{1});
            output_data(end+1) = str2double(values{2});
        end
    end
end
disp('DATA_END received, data successfully read.');
```

**Obrázek 2.16:** Načtení dat z Arduina pro měření typu R (fotografie vlastní, 2025)

Tato část programu je pro režim měření „R“ a zajišťuje příjem dat ze sériového portu. Na rozdíl od měření typu „S“, kde z Arduina přijmeme pouze výstupní signál, zde je nutné přijímat dvě datové složky – výstupní signál a vstupní signál (hodnota PWM). Matlab opět čeká na *DATA\_START*, čím Arduino udává začátek přenosu dat (viz obrázek 2.16). Následuje čtení jednotlivých řádků textu. Každý řádek obsahuje dvě hodnoty oddělené čárkou – první je vstupní PWM signál a druhý je odezva systému. Pomocí funkce *strsplit* se řádek rozdělí a jednotlivé hodnoty uloží do samostatných vektorů. Přenos je znovu ukončen značením *DATA\_END*.

```

Vref = 4.94; % Referenční napětí Arduina
V_50 = 0.9686;
napeti = (double(output_data) / 1023) * Vref;
% Přepočet vstupního napětí (signál z Arduina)
vstupni_napeti = (double(input_data) / 255) * Vref;
% Odečtení vstupního napětí odpovídající hodnotě PWM = 50
u0 = V_50; % První hodnota vstupního signálu
vstupni_napeti_odecteno = vstupni_napeti - u0;|
% Časový vektor
delta_t = delta / 1e6;
cas = (0:length(napeti)-1) * delta_t;

```

**Obrázek 2.17:** Přepočet napětí, odečtení offsetu a výpočet časového vektoru (fotografie vlastní, 2025)

Po načtení hodnot jsou hodnoty ADC převedeny na napětí. Provedeme převedení hodnot výstupního signálu z 10bitového rozsahu (0–1023) a vstupní PWM signál je převeden z rozsahu (0–255) (viz obrázek 2.17). Dále je odečtena hodnota  $u_0$  od vstupního signálu, aby odpovídala napětí při hodnotě PWM 50. Poté je proveden klouzavý průměr. Také je vytvořen časový vektor s rovnoměrným krokem  $\delta t$ , který se rovná intervalu vzorkování.

```

% Odečtení ustálené hodnoty ze začátku výstupního signálu
ustalena_hodnota = napeti(1);
prechodova_charakteristika = (napeti - ustalena_hodnota) / Vref;

% Explicitní odstranění offsetu
prechodova_charakteristika = prechodova_charakteristika - prechodova_charakteristika(1);

```

**Obrázek 2.18:** Výpočet přechodové charakteristiky odečtením počáteční ustálené hodnoty a odstranění offsetu (fotografie vlastní, 2025)

Přechodová charakteristika je získána stejným způsobem jako u měření „S“, tedy odečtením ustálené hodnoty od výstupního napětí a následným normalizováním na referenční napětí. Offset je dále odstraněn, čímž získáme čistou odezvu na změnu signálu na vstupu (viz obrázek 2.18).

```

a = 0.95; % Koeficient filtru (lze upravit pro větší vyhlazení)
prechodova_charakteristika_filtered = zeros(size(napeti)); % Inicializace

% Aplikace filtru na výstupní signál
prechodova_charakteristika_filtered(1) = 0;
for k = 2:length(napeti)
    prechodova_charakteristika_filtered(k) = a * prechodova_charakteristika_filtered(k-1)...
        + (1-a) * (filtered_napeti(k) - filtered_napeti(1));
end

prechodova_charakteristika_filtered2(1) = 0;
for k = 2:length(napeti)
    prechodova_charakteristika_filtered2(k) = a * prechodova_charakteristika_filtered2(k-1)...
        + (1-a) * prechodova_charakteristika_filtered(k-1);
end

prechodova_charakteristika_filtered=prechodova_charakteristika_filtered2;

% --- Výpočet přechodové charakteristiky s filtrem ---
ustalena_hodnota_smoothed = prechodova_charakteristika_filtered(1);
prechodova_charakteristika_smoothed = prechodova_charakteristika_filtered / Vref;
prechodova_charakteristika_smoothed = movmean(prechodova_charakteristika_smoothed, 3);

```

**Obrázek 2.19:** Aplikace exponenciálního filtru na výstupní signál a výpočet přechodové charakteristiky s filtrem (fotografie vlastní, 2025)

Před výpočtem impulsní charakteristiky bylo nutné provést filtraci změřeného výstupního signálu za účelem potlačení vysokofrekvenčního šumu a zajištění plynulejšího průběhu odezvy systému. Byla využita exponenciální filtrace s využitím vztahu:

$$y_k = a * y_{k-1} + (1 - a) * (x_k - x_1) \quad (20)$$

kde:

- $y_k$  je filtrovaný výstup v čase  $k$ ,
- $y_{k-1}$  je předchozí výstup,
- $x_k$  je aktuální vstupní hodnota napětí,
- $x_1$  je počáteční hodnota vstupu (offset),
- $a$  je koeficient filtru určující míru vyhlazení.

Filtrace byla aplikována dvakrát za sebou, aby bylo dosaženo vyššího stupně vyhlazení a účinnějšího potlačení vysokofrekvenčních složek. Hodnota parametru  $a$  byla zvolena empiricky na základě dosažené kvality výsledného signálu (viz obrázek 2.19).

Po dvojité aplikaci exponenciální filtrace byl signál ještě jednou vyhlazen pomocí klouzavého průměru (funkce *movmean* v MATLABu), kde byla použita délka okna pět vzorků. Funkce *movmean* provádí výpočet průměru z okolních pěti hodnot, což dále zajišťuje odstranění drobných fluktuací a připravuje data pro následný výpočet impulsní odezvy.

```

% --- Dekonvoluce ---
N = length(prechodova_charakteristika_smoothed);
g = zeros(1, N-1);

for k = 1:(N-1)
    suma = 0;
    for i = 0:(k-2)
        suma = suma + g(i + 1) * smoothed_vstupni_napeti(k - i + 1);
    end
    g(k) = (prechodova_charakteristika_smoothed(k) / delta_t - suma) / smoothed_vstupni_napeti(1);
end

% Vyhlazení impulsní charakteristiky
vyhlazene_g = movmean(g, 3);
cas_impulsu = cas(1:end-1);

```

**Obrázek 2.20:** Výpočet impulsní odezvy pomocí dekonvoluce a její následné vyhlazení (fotografie vlastní, 2025)

Po aplikaci filtru a úpravě přechodové charakteristiky je dalším krokem výpočet impulsní odezvy systému pomocí dekonvoluce. Cílem je získat pomocí výpočtu reálnou impulsní charakteristiku na základě znalosti vstupních a výstupních signálů. Vzhledem k tomu, že vstupní signál v tomto případě není jednotkový skok, ale obecný proměnný signál, není zde možné využít pouhou derivaci. Proto se využije zpětný výpočet podle vztahu:

$$g_k = \frac{y_k - \sum_{i=1}^{k-1} g_i \cdot u_{k-i}}{u_0} \quad (21)$$

kde:

$g_k$  je hledaná impulsní odezva v čase  $k$ ,

$y_k$  je výstupní signál systému v čase  $k$ ,

$\Delta t$  je časový krok (delta)

$g_i$  jsou již vypočítané hodnoty impulsní odezvy v předchozích časech,

$u_{k-i}$  jsou hodnoty vstupního signálu posunuté o  $i$  vzorek zpět,

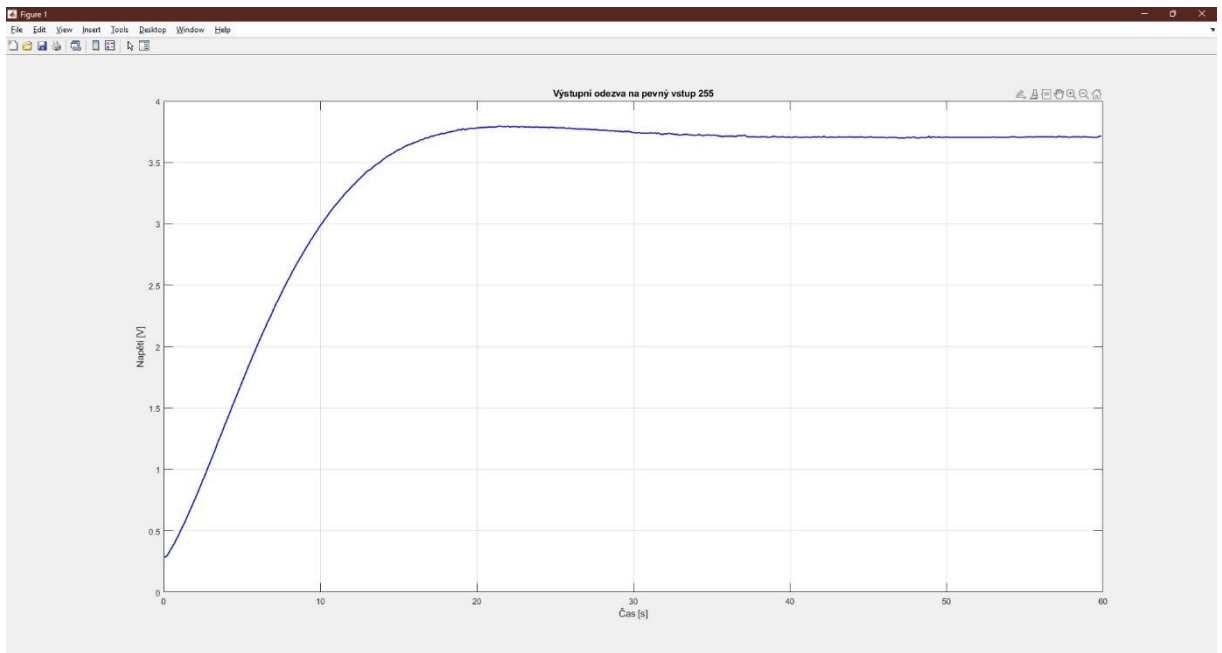
$u_0$  je počáteční hodnota vstupního signálu.

Výsledný průběh po výpočtu dekonvoluce byl vyhlazen funkcí *movmean* za účelem odstranění šumu (viz obrázek 2.20).

## 3 Výsledky měření

### 3.1 Výsledek pro měření typu „S“ – jednotkový skok

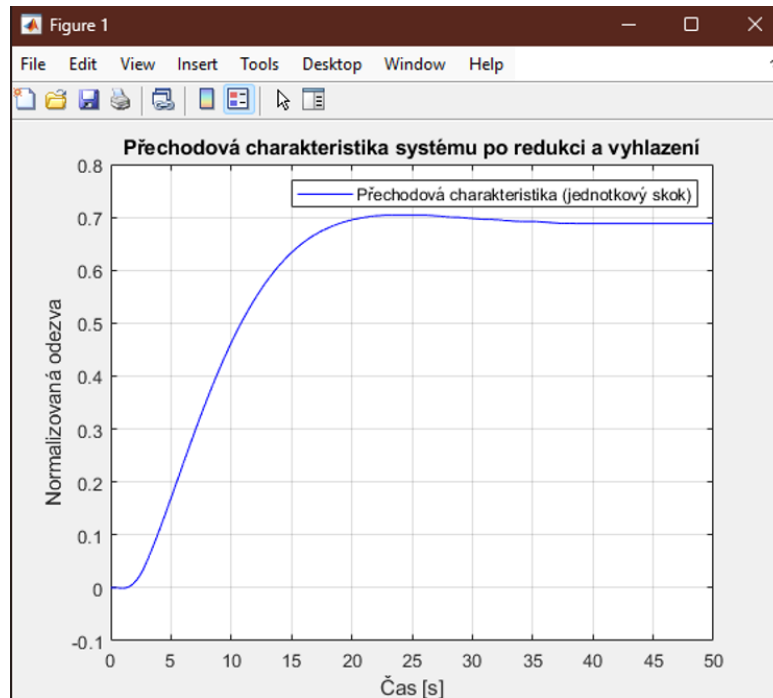
V první fázi bylo provedeno měření výstupního napětí po přivedení konstantní vstupní hodnoty, která odpovídá hodnotě PWM 255. Získaná odezva byla zaznamenána v čase a převedena na napěťové hodnoty pomocí kalibrace na referenční napětí. Následně byl vytvořený graf napětí na čase, který ukazuje dynamiku odezvy.



**Obrázek 3.1:** Výstupní napěťová odezva systému (fotografie vlastní, 2025)

Z grafu na obrázku 3.1 lze vyčíst náběh signálu na ustálenou hodnotu, charakteristický pro systém druhého řádu. Tvar odpovídá očekávané exponenciálně-logaritmické odezvě.

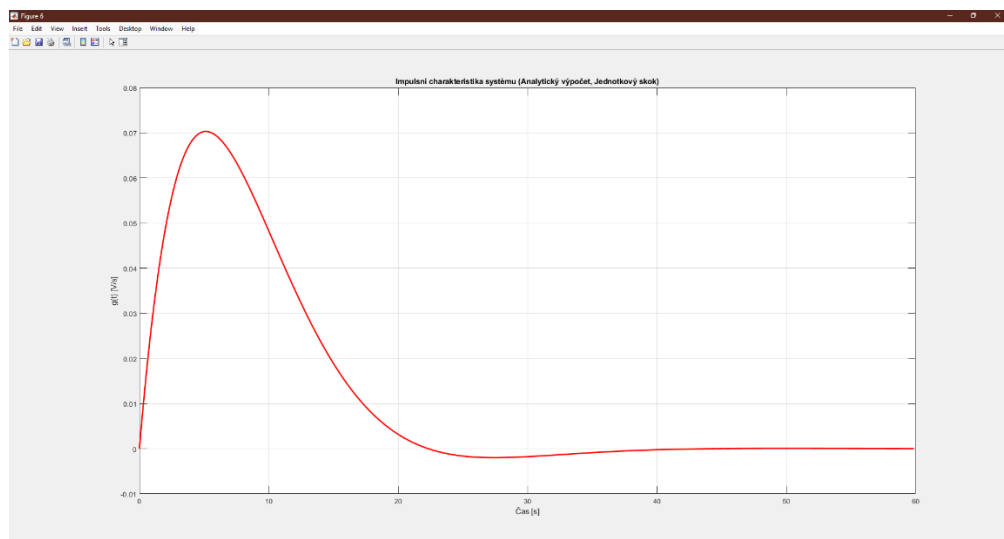
Z odezvy systému byla odečtena ustálená počáteční hodnota a výstup byl následně převeden do normalizované formy. Výsledkem je přechodová charakteristika. Ta ukazuje změnu výstupu po jednotkovém skoku.



**Obrázek 3.2:** Přechodová charakteristika systému (fotografie vlastní, 2025)

Charakteristika (obrázek 3.2) vykazuje typické chování s náběhem v počátku, následným překmitem a pozdějším ustálením výstupu.

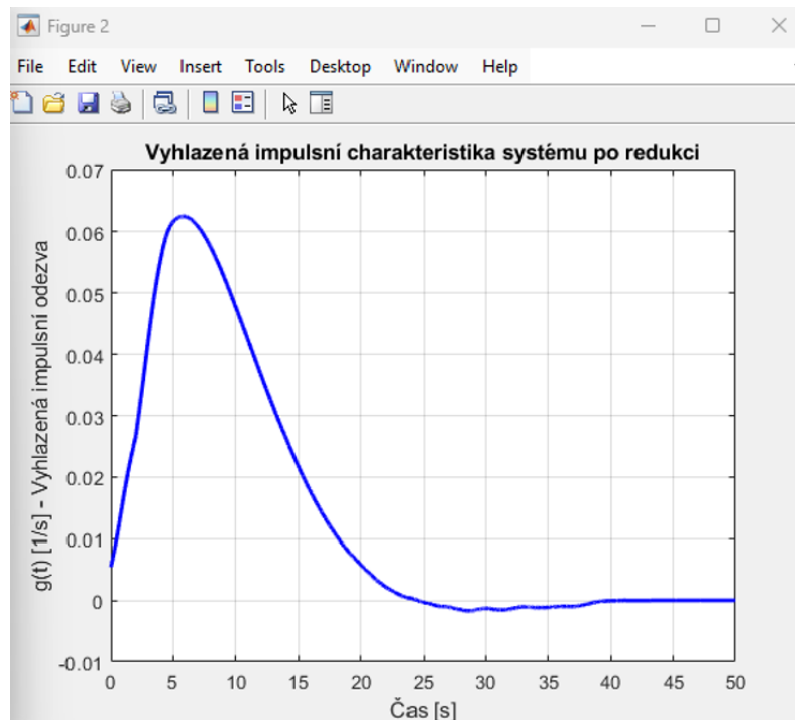
Na základě známých parametrů systému – časové konstanty, tlumicího faktoru a vlastní frekvence – byla vypočtena teoretická impulsní odezva systému druhého řádu. Výpočet je převzatý z matematického modelu harmonické odezvy s exponenciálním tlumením.



**Obrázek 3.3:** Analytická impulsní charakteristika systému (fotografie vlastní, 2025)

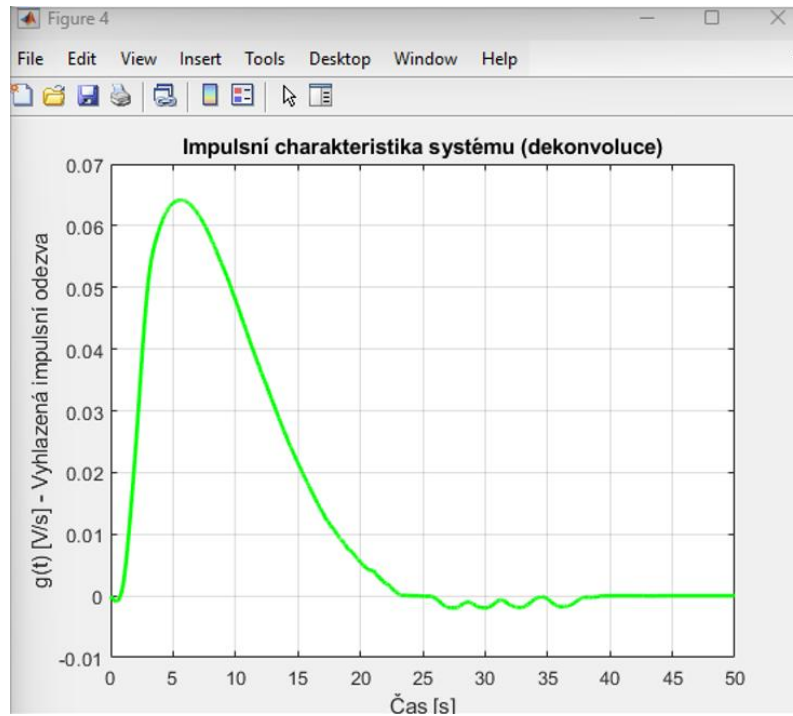
Z průběhu grafu na obrázku 3.3 vidíme klasické chování systému druhého řádu – tedy rychlý nárůst, dosažení maxima a poté pokles až k nule. Tento průběh je hladký a bez oscilací, což odpovídá ideálním podmínkám modelu bez šumu či zpoždění.

Ze změřené přechodové charakteristiky byla pomocí derivace byla vypočítána přibližná impulsní charakteristika systému. Před derivací byl signál vyhlazen klouzavým průměrem, aby se omezily oscilace způsobené šumem (viz obrázek 3.4).



**Obrázek 3.4:** Impulsní charakteristika – numerická derivace (fotografie vlastní, 2025)

Tato metoda poskytuje velmi rychlý způsob výpočtu. Avšak následný průběh může být citlivý na přítomnost šumu v datech. I přesto zachycuje tvar teoretické odezvy systému. Další způsob je dekonvoluce, která funguje ve zpětném řešení diskrétní konvoluční rovnice. Pomocí námi známého vstupního signálu a přechodové charakteristiky se krok po kroku vypočítává impulsní odezva.



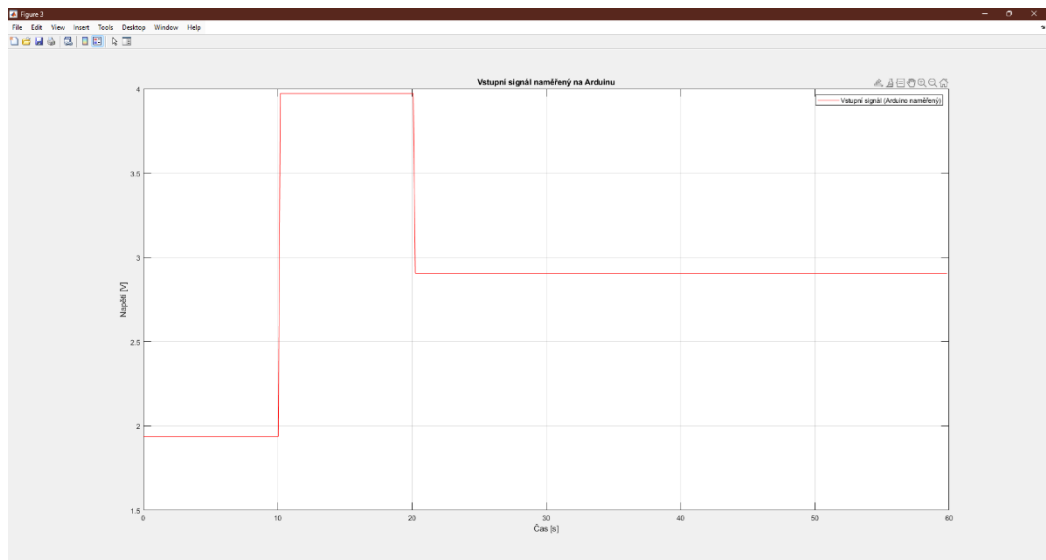
**Obrázek 3.5:** Impulsní charakteristika – dekonvoluce (fotografie vlastní, 2025)

Dekonvoluce je přesnější než numerická derivace, protože bere v potaz vliv všech předchozích vzorků. Konečný tvar impulsní odezvy velmi dobře odpovídá analytickému výpočtu. Oscilace ke konci signálu jsou pravděpodobně důsledkem numerické nestability, jak lze vidět na obrázku 3.5.

Na základě výše uvedených výsledků lze konstatovat, že tři metody (dekonvoluce, derivace, analyticky) vedou k podobným impulsním charakteristikám. Nejbližší analytickému průběhu je metoda dekonvoluce, která poskytuje nejplynulejší a nejvěrnější výstup. Je však třeba poznamenat, že při využití dekonvoluce rychle narůstá numerická chyba, a proto je tato metoda vhodná jen pro omezený počet vzorků. Derivace je vhodná pro rychlý odhad. Z výsledku lze potvrdit, že měřicí systém správně zachycuje dynamickou odezvu systému na jednotkový skok na vstupu.

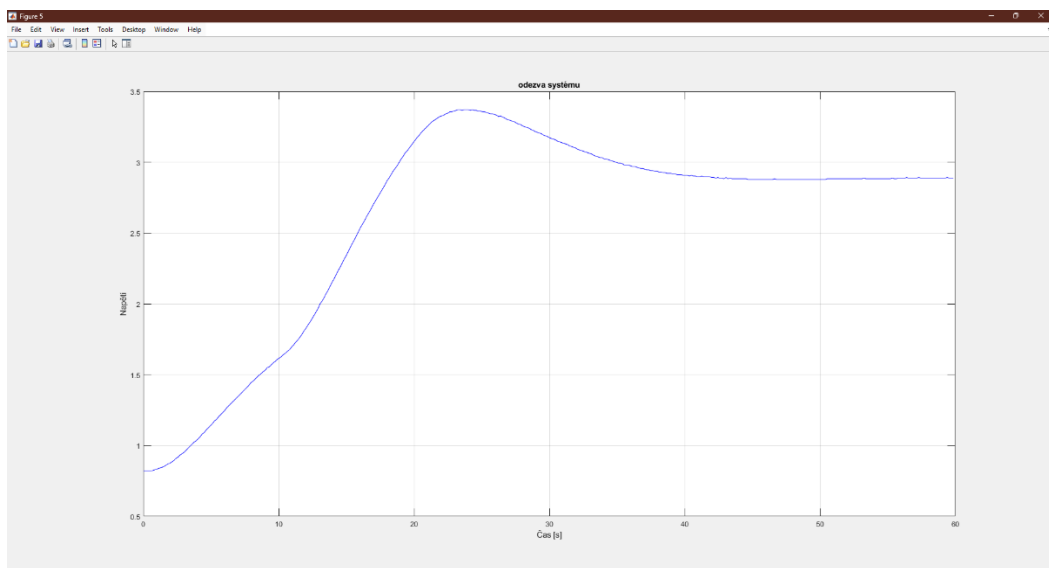
### 3.2 Výsledek pro měření typu „R“ – Proměnný vstupní signál

V této části bylo provedeno měření systému pomocí proměnného vstupního signálu. Na vstup systému byl přiveden signál s průběhem PWM:  $50 \rightarrow 150 \rightarrow 255 \rightarrow 200$ . Ke změnám dochází po časových intervalech přibližně 10 sekund. Tento tvar byl zvolen záměrně, aby bylo možné analyzovat chování daného systému nejen při náběhu ihned na vysokou hodnotu, ale i při postupném zvyšování.



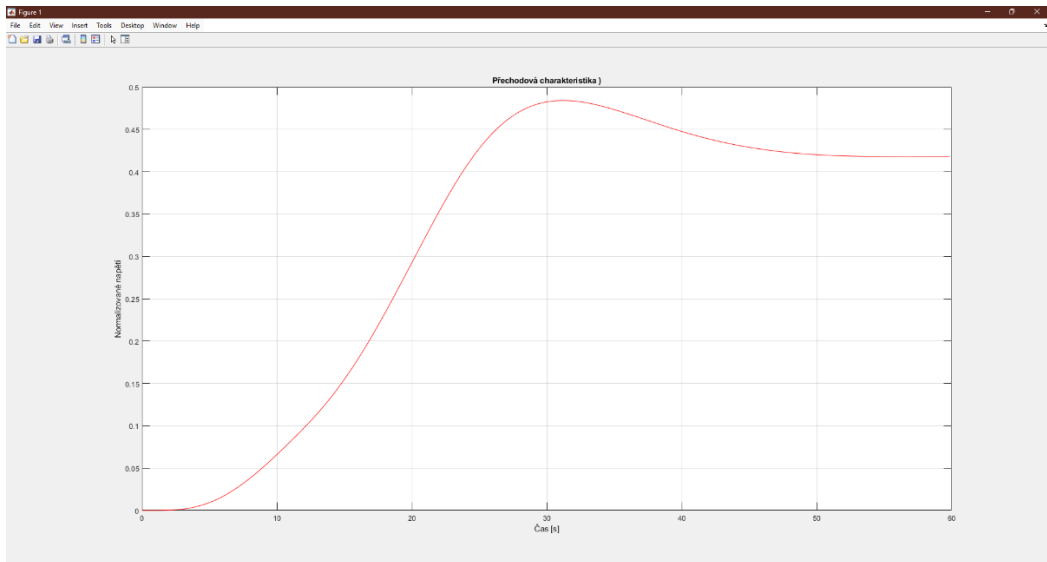
**Obrázek 3.6:** Vstupní signál (fotografie vlastní, 2025)

Z výše uvedeného grafu (viz obrázek 3.6) je vidět, že vstupní signál odpovídá čtyřúrovňovému průběhu. Na základě tohoto signálu bylo provedeno měření výstupu systému.



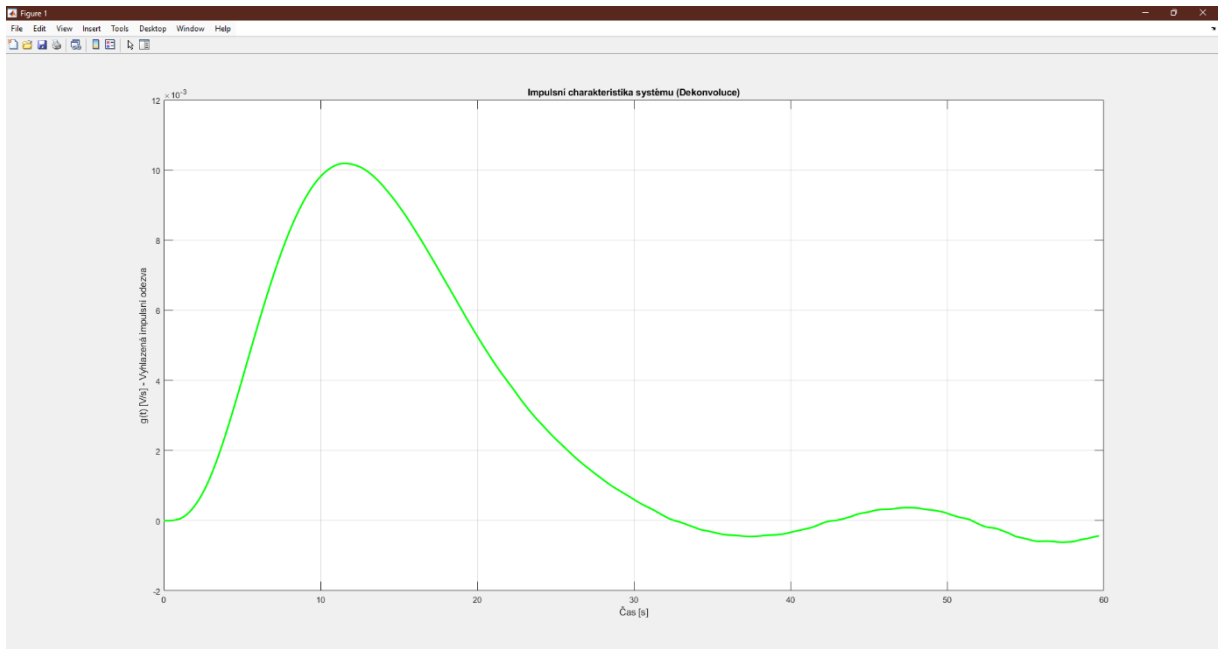
**Obrázek 3.7:** Výstupní napěťová odezva systému (fotografie vlastní, 2025)

Výstupní signál ukazuje odezvu systému na jednotlivé úrovně vstupu (viz obrázek 3.7). Lze si všimnout dynamické změny napětí odpovídající změnám na vstupu. Odečtená hodnota byla následně použita pro výpočet odezvy systému na daný vstupní signál.



**Obrázek 3.8:** Reakce na vstupní signál (fotografie vlastní, 2025)

Od výstupního signálu byla odečtena počáteční hodnota a následně byl celý signál normalizován vzhledem ke změně vstupního signálu. Normalizací signálu se zde rozumí přepočítání hodnot tak, aby výsledná odezva byla vztažena ke změně vstupního signálu, tedy dělení rozdílu napětí velikostí vstupního skoku. Signál byl vyhlazen dvojitou aplikací filtrem prvního řádu. Je patrné, že charakteristika reaguje odpovídajícím způsobem na změny vstupu (viz obrázek 3.8).



**Obrázek 3.9:** Impulsní charakteristika systému – dekonvoluce (fotografie vlastní, 2025)

Na základě již známé přechodové charakteristiky a vstupního signálu byla metodou dekonvoluce vypočtena impulsní odezva systému. Tato metoda zpětně řeší diskretní konvoluční rovnici a umožňuje tak určit jednotlivé impulsní odpovědi systému. Na obrázku 3.9 lze vidět, že impulsní odezva má tvar, který odpovídá dynamickému systému druhého řádu s mírným tlumením. Při porovnání s analytickou impulsní charakteristikou systému (viz obrázek 3.3) je patrné, že výsledek z dekonvoluce (viz obrázek 3.9) má pomalejší dynamiku. Tento rozdíl může být způsoben opakovanou filtrací, která byla aplikována za účelem potlačení šumu. Dvojití použití filtru zajišťuje vyšší míru vyhlazení, avšak zároveň zpomaluje odezvu systému. Pro srovnání by bylo vhodné ukázat výsledek s jedním filtrem nebo bez funkce `movmean()`, což by mohlo přinést přesnější průběh bez deformace amplitudy. Na začátku je patrný výrazný vrchol následovaný postupným poklesem hodnot. Je však zároveň vidět, že v pozdější části signálu dochází k výraznějším oscilacím. Tyto oscilace nejsou reálným výstupem systému, ale jsou způsobeny numerickou nestabilitou dekonvoluční metody. S narůstajícím časem roste vliv šumu. Tento jev je běžný při práci s delšími signály a výraznějšími změnami ve vstupních hodnotách.

Z výsledků měření je patrné, že měřicí systém byl schopen zaznamenat dynamickou odezvu i při použití komplikovanějšího vstupního signálu.

## ZÁVĚR

Cílem této bakalářské práce bylo navrhnout a realizovat zařízení, které umožní změřit impulsní charakteristiku lineárního dynamického systému. Pro realizaci měřicího modulu byl zvolen mikropočítač Arduino. Data byla poté zpracována v prostředí MATLAB, ve kterém byly vypočteny veškeré impulsní charakteristiky pomocí dvou metod.

Měření bylo provedeno na elektronickém systému sestaveném z pasivních elektronických součástek a operačních zesilovačů. Měřením pro skokový vstupní signál byla získána přechodová charakteristika systému, která odpovídá chování systému druhého řádu. Z této charakteristiky byla vypočtena impulsní odezva.

Dále bylo provedeno měření pro proměnný vstupní signál, které potvrdilo schopnost navrženého systému zachytit dynamiku i složitějších vstupů. Ukazuje se, že metoda dekonvoluce poskytuje plynulejší výstup, a je proto vhodnější pro detailnější analýzu, především v úvodu odezvy.

Navržený systém splňuje všechny zadané požadavky a umožňuje efektivní měření dynamických vlastností reálných systémů. Do budoucna by bylo možné rozšířit zpracování signálů o pokročilejší metody (například frekvenční doménu, použití filtrů vyšších řádů) a rovněž zlepšit uživatelské rozhraní pro přehlednější práci s naměřenými daty.

## POUŽITÁ LITERATURA

ARDUINO. Arduino UNO Rev3. *Store.Arduino.cc* [online]. © 2025 [cit. 2025-04-27]. Dostupné z: [https://store.arduino.cc/products/arduino-uno-rev3?\\_pos=3&\\_psq=M000006+OR+A000066+OR+TSX00083&\\_ss=e&\\_v=1.0](https://store.arduino.cc/products/arduino-uno-rev3?_pos=3&_psq=M000006+OR+A000066+OR+TSX00083&_ss=e&_v=1.0)

CVEJN, Jan. *Automatizace* [interní studijní materiál]. Pardubice: Univerzita Pardubice, Fakulta elektrotechniky a informatiky, 2024.

LÁNÍČEK, Robert. *Elektronika: obvody, součástky, děje*. Praha: BEN – technická literatura, 2001. ISBN 8086056252.

NOSKIEVIČ, Petr. *Modelování a identifikace systémů*. Ostrava: Montanex, 1999. ISBN 80-7225-030-2.

ZAIT, Anat. Arduino Uno Pinout: Complete Guide. In: *Circuito.io* [online]. 22. 4. 2018 [cit. 2025-04-27]. Dostupné z: <https://www.circuito.io/blog/arduino-uno-pinout/>

## **SEZNAM PŘÍLOH**

Příloha A: Zdrojový kód programu pro Arduino

Příloha B: Skript v prostředí MATLAB pro zpracování signálů

Příloha C: Zdrojové soubory navrženého plošného spoje