

## **Abstrakt**

Práca sa zaoberá generalizáciou procesov a konfigurácie v oblasti vzdialených experimentov a online laboratórií. Obsahom je aj podrobný popis a analýza v súčasnosti existujúcich systémov a ich vlastností. Cieľom bolo vytvoriť jednotné rozhranie umožňujúce prístup, správu a ovládanie vzdialených experimentov. Je popísaný proces unifikovanej integrácie experimentu do systému, ktorý je znázornený na navrhnutej a skonštruovanej mechatronickej sústave. Po analýze už existujúcich riešení sú podrobnejšie rozoberané jej jednotlivé komponenty, ich funkcie a obmedzenia. Ovládanie experimentu je možné formou otvorenej slučky ale aj pomocou spätnej väzby cez blokovú schému simulačného prostredia Simulink. Vyzdvihnutá je najmä možnosť inicializácie parametrov schémy prostredníctvom používateľského rozhrania webovej aplikácie. V závere práce sú zhrnuté dosiahnuté výsledky a poskytnuté nápady na možné vylepšenia systému aj zariadenia.

**Kľúčové slová:** online laboratórium, vzdialený experiment, Matlab, Simulink, Arduino, senzor

## **Abstract**

Thesis deals with generalisation in the area of remote experiments and online laboratories. It includes detailed description and analysis of existing systems and their characteristics. The goal was to create a unified interface which provides access to and management of remote experiments. The thesis describes the process of unified integration of an experiment into the system, which is illustrated on a newly developed and constructed mechatronic plant. After the analysis of existing solutions the thesis describes device's individual components, their function within the device and also their limitations. Control of the experiment is provided by both open-loop and closed-loop solutions. The closed-loop one is created using Simulink block diagram. The possibility of declaring the block diagram's parameters within the web applications is also described. The thesis ends with a summary of achieved results and offers ideas for future development of both the overall system and the new device.

**Keywords:** online laboratory, remote experiment, Matlab, Simulink, Arduino, sensor

# Obsah

<b>1</b>	<b>Úvod</b>	<b>6</b>
<b>2</b>	<b>Existujúce riešenia</b>	<b>7</b>
<b>3</b>	<b>Architektúra systému</b>	<b>7</b>
<b>4</b>	<b>Komunikácia so zariadením</b>	<b>8</b>
<b>5</b>	<b>Pripojené zariadenia</b>	<b>8</b>
<b>6</b>	<b>Návrh nového zariadenia</b>	<b>10</b>
6.1	Ovládanie zariadenia . . . . .	10
6.1.1	Režim otvorenej slučky . . . . .	11
6.1.2	Integrácia simulačného prostredia . . . . .	12
6.2	Prepojenie s centrálnym serverom . . . . .	13
6.2.1	Používateľské rozhranie . . . . .	14
6.2.2	Spustenie experimentu . . . . .	15
6.2.3	Správa regulátorov . . . . .	15
6.2.4	Parametre experimentu . . . . .	16
6.3	Čítanie výsledkov . . . . .	17
<b>7</b>	<b>Potenciálne vylepšenie</b>	<b>18</b>
<b>8</b>	<b>Záver</b>	<b>19</b>
<b>9</b>	<b>Prínosy dizertačnej práce</b>	<b>20</b>
	<b>Literatúra</b>	<b>20</b>

# 1 Úvod

Vzdialené experimenty a online laboratória sa čoraz častejšie stávajú súčasťou vzdelávacieho procesu. Možnosť vykonávať experimenty na reálnych zariadeniach z pohodlia domova bez časovej limitácie ponúka študentom príležitosť rozširovať svoje vedomosti nielen z oblasti riadiacich algoritmov. Kvalitný systém online laboratória musí podporovať široké spektrum rôznych zariadení a simulačných prostredí. Aby sa tento systém mohol rozvíjať je nutné, aby integrácia nových experimentov bola jednoduchým a ľahko replikovateľným procesom.

Prenos dát medzi servermi a zariadeniami je často veľmi špecifickým procesom. Každé zariadenie funguje na inom princípe, má rôzne vstupné a výstupné parametre a je riadené iným simulačným prostredím. Online laboratórium by malo definovať najmenšie nedeliteľné procesy, ktoré slúžia na zabezpečenie chodu zariadenia a následne ich implementovať. Je tiež nutné vytvoriť rozhranie medzi simulačnými prostrediami, zariadením a webovým serverom.

Správa regulačných schém a algoritmov by mala slúžiť nielen ako úložisko súborov, ale aj ako platforma pre deklaráciu vstupných parametrov a premenných, ktoré definujú priebeh experimentu.

Celý systém by mal byť rozdelený do samostatných modulov, ktoré dokážu fungovať nezávisle jeden od druhého. Zároveň by však mali byť postavené na jednej platforme, čo zjednodušuje údržbu a rozširovanie systému.

Správne fungovanie systému je vhodné overiť vytvorením nového zariadenia, ktoré môže tiež slúžiť na výučbu a testovanie regulátorov. Takéto zariadenie musí byť schopné spracovať vstupné veličiny a na výstupe poskytovať namerané hodnoty. Zber dát senzormi musí prebiehať s takou frekvenciou, aby riadiaci algoritmus dokázal reagovať na zmeny v sústave a adekvátne nastavil úroveň akčného zásahu. Je tiež dôležité, aby namerané hodnoty boli čo možno najpresnejšie. Pri výbere komponentov na stavbu experimentu treba dbať na ich životnosť, cenovú dostupnosť a možnosť ich v prípade potreby jednoducho opraviť či vymeniť.

Nastavenie a pripojenie zariadenia ako aj celého systému by malo byť v čo najväčšej miere riešené konfiguračnými súborami, ktoré sú jednoducho čitateľné.

## 2 Existujúce riešenia

Bolo vyvinutých už veľa riešení na zastrešenie väčšieho množstva vzdialených experimentov. Systémy, ktoré slúžia len ako repositár nemôžno označiť ako online laboratórium. Tie zvyčajne obsahujú širšiu funkcionálnosť ako napríklad rezervačný systém, autorizáciu a autentifikáciu, jednotné používateľské rozhranie na prístup k experimentom.

Takéto systémy boli vyvinuté na viacerých univerzitách a v súčasnosti majú tendenciu dokonca spájať viacero univerzít, ktoré si navzájom svoje experimenty zdieľajú. Sú nimi napríklad systémy ako GOLDi labs [12], iLab [1] alebo WebLab Desuto [10, 11, 9].

I keď sa tieto online laboratória snažia o generalizáciu prístupu k zariadeniam, častokrát sú odkázané len na jeden druh softvéru (LabView, Matlab) [14, 4] alebo typu regulátora pre danú sústavu. Naším cieľom je umožniť jednoduché a unifikované prepojenie medzi online laboratóriom, reálnym zariadením a simulačným prostredím. Cieľom je vytvoriť systém umožňujúci inicializovať parametre experimentu priamo z webového prostredia portálu. Jednoduchý systém nahrávania regulačných schém umožní zadať deklarované premenné a tie používateľ už len vyplní pri spúšťaní experimentov, poprípade ponechá prednastavené hodnoty.

Jedno zariadenie by malo byť možno prepojiť a ovládať s viacerými simulačnými prostrediami. Tiež musí existovať možnosť ich spúšťania priamo bez spätnej väzby za účelom zistenia dynamiky systému.

## 3 Architektúra systému

Jednou zo základných charakteristík systému je jeho modularita. Tú je možné pozorovať hneď v dvoch úrovniach. Prvá sa prejavuje fyzickou implantáciou a topológiou systému. Každý experiment je pripojený k vlastnému serveru a všetky zdieľajú jeden centrálny webový sever, ktorý figuruje ako centrálny bod celej siete. To znamená, že odpojenie alebo porucha jedného zariadenia nijak neovplyvní chod celého systému. Ďalšou z výhod tohto prístupu je aj jednoduchá inkorporácia nových zariadení do už bežiacieho systému bez nutnosti odstávky.

Druhú úroveň predstavuje programový kód a softvérová implementácia, kde je podobná funkcionálnosť koncentrovaná do separátnych modulov, ktoré je možno ľubovoľne vypínať či zapínať. Ak by napríklad nastala chyba v jednom z týchto modulov (napr. online chat, štatistiky, rezervácie, atď.), táto sekcia sa dá zo systému dočasne odstrániť

bez akýchkoľvek následkov. Vďaka tomuto návrhu je možné vykonať relatívne rozsiahle zásahy či už do kódu alebo do samotných fyzických experimentov počas ostrej prevádzky.

## 4 Komunikácia so zariadením

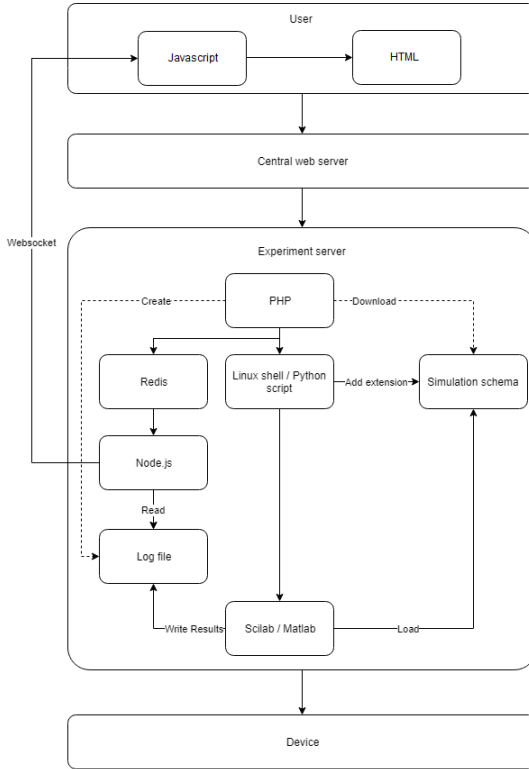
Existuje niekoľko rozhraní zabezpečujúcich plynulý prenos dát medzi všetkými komponentami systému. Proces spustenia experimentu začína odoslaním požadovaných parametrov zadaných v HTML formulári a končí vykreslením grafu z výstupných dát experimentu. To znamená, že príkaz na spustenie experimentu spolu so vstupnými parametrami prechádza z užívateľského počítača cez centrálny webový server do aplikačného servera, kde je spracovaný. Aplikačný server spustí pripojené zariadenie a popritom priebežne ukladá získané merania. Už počas priebehu merania sa dáta odosielaajú priamo na používateľov počítač, kde sa v reálnom čase zobrazujú v grafe. Po úspešnom ukončení experimentu sa všetky výstupné hodnoty odošlú na centrálny webový server, kde sa nachádza archív všetkých už vykonaných experimentov.

Väčšina komunikácie prebieha prostredníctvom HTTP metód a API rozhraní. Špeciálnu pozornosť si zaslúži prenos nameraných dát zo servera spravujúceho experiment k pripojenému používateľovi. Tento prenos obchádza centrálny webový server a odosiela dáta prostredníctvom WebSocketu [8] rovno na webový prehliadač používateľa, ktorý spustil daný experiment. Táto peer-to-peer komunikácia eliminuje zbytočné vyťaženie centrálného servera a umožňuje zobrazovanie nameraných hodnôt už počas behu experimentu.

Po načítaní príkazu na spustenie experimentu si dedikovaný server vytvorí nový dokument, do ktorého sa namerané dáta budú zapisovať. V tom istom čase beží na pozadí aj Node.js server, ktorý periodicky číta obsah súboru a cez vyššie spomínaný WebSocket jeho obsah odosiela k používateľovi (Obrázok č. 1).

## 5 Pripojené zariadenia

Existuje hneď niekoľko zariadení, ktoré boli k online laboratóriu už úspešne pripojené. Jedno z prvých bola termo-opticko-mechanická sústava [6], pozostávajúca z LED svetla, žiarovky a ventilátora. Výstupnými merateľnými veličinami sú teplota a svetelná intenzita. Ďalším zariadením je segway alebo inverzné kyvadlo na vozíku poháňané



Obr. 1: Prenos dát medzi jednotlivými technológiami a aktérmi počas priebehu experimentu

dvoma DC motormi. Je vybavené ďalším servo motorom, ktorým sa dá meniť poloha ťažiska systému. Výstupné dáta sa získavajú zo vstavaného gyroskopu a akcelerometra. Posledným pridaným systémom je towerkoptéra – vrtuľa na koľajnici, ktorá sa môže pohybovať len po jednej osi. Vstupom systému je napätie na motore a výstupom hodnota zo senzora polohy. K online laboratóriu je pripojená ešte 3D LED kocka, ktorá môže slúžiť prevažne na výuku základov algoritmickej a programovania.

## 6 Návrh nového zariadenia

Zariadenie, ktorým sa testovalo generalizované pridávanie experimentov do systému je pohybujúca sa loptička vo vertikálne orientovanom valci. Zmena jej polohy je umožnená vďaka ventilátoru umiestnenému v podstave sústavy. Tento ventilátor generuje prúd vzduchu, ktorý nadnáša loptičku. Získavanie spätnej väzby zo sústavy zabezpečuje senzor vzdialenosti. Riadenie rýchlosti otáčania motora, načítavanie dát zo senzora a komunikáciu s počítačom rieši mikrokontroler Arduino Uno [15].

Takéto zariadenie už bolo v minulosti vyvinuté na viacerých vzdelávacích inštitúciách za účelom vyrobiť nenákladnú sústavu na výuku regulačných algoritmov [2, 3, 5]. Jeho jednoduchá stavba a nízka cena komponentov sú ideálnymi vlastnosťami pri stavbe prototypu. Zariadenie je zároveň dostatočne komplexné aby sa na ňom dali demonštrovať procesy integrácie do systému.

Po konštrukčnej strane sa od iných zariadení podobného typu líši hlavne senzorom na meranie vzdialenosti. Po vykonaní extenzívnych testov s infračervenými [16] a ultrazvukovými senzormi [6] bol zvolený laserový vďaka jeho väčšej presnosti a vyššej vzorkovacej frekvencii.

Motor ventilátora je riadený PWM signálom [19, 13, 7] generovaným riadiacou jednotkou Arduino [15]. Tá od neho tiež získava aj počet otáčok vrtule za minútu.

Komunikácia s počítačom prebieha po sériovej linke. Systém v pravidelných intervaloch získava hodnoty zo senzorov a posiela vstupný parameter, ktorým je úroveň PWM signálu. Bol vytvorený nový prenosový protokol, ktorý šetrí množstvo prenesených dát a zároveň je kompatibilný s prostredím Simulink.

Samotné zariadenie prešlo viacerými iteráciami a otestovaných bolo niekoľko procesov výroby ako 3D tlač a CNC frézovanie. Tiež bolo vyskúšaných viacero druhov senzorov, ventilátorov aj loptičiek.

### 6.1 Ovládanie zariadenia

Riadiaca jednotka zabezpečuje načítavanie otáčok ventilátora a vzdialenosti loptičky od senzora. Musí byť schopná tieto údaje poslať po sériovej linke do pripojeného počítača, vedieť prečítať odpoveď a vygenerovať adekvátny PWM signál na roztočenie vrtule.

Na zjednodušenie komunikácie medzi počítačom a sústavou bol navrhnutý vylepšený spôsob prenosu dát v porovnaní s už pripojenými zariadeniami. Toto zariadenie má dve celočíselné výstupné hod-

noty (pozícia loptičky a otáčky ventilátora) a jednu tiež celočíselnú výstupnú (úroveň PWM signálu).

Dĺžka tela rámca bola stanovená na 4 bajty. V prípade prenosu hodnôt nižších než 256 sa vloží nulový bajt na miesto najvýznamnejšieho bajtu. Ak by sa prenášala aj informácia o dĺžke rámca, nutne by musel pribudnúť ešte jeden bajt, čo by v konečnom dôsledku spôsobilo väčší objem prenosu.

Z mechanických obmedzení sústavy vyplýva, že ani jedna z prenášaných veličín nikdy nepresiahne hodnotu 13200. Najvýznamnejší bajt rýchlosti otáčok môže byť maximálne 0x33 (51) a v prípade vzdialenosti je to len 0x0f (15). Hlavičku rámca môžu tvoriť dva bajty o veľkosti aspoň 0x34, nakoľko takáto sekvencia v tele rámca nastáť nemôže. Kvôli jednoduchšiemu prepojeniu so simulačným prostredím MATLAB bola zvolená sekvencia 0x7e7e, ktorá vyplýva z limitácie spôsobenej využitím Simulink blokov pre sériovú komunikáciu[18]. Ten dokáže spracovať hlavičku len v ASCII formáte, čo limituje možné hodnoty bajtov na 32 (medzera) až 126 (~) [18]. Zvyšné hodnoty kódujú netlačiteľné znaky[17].

### 6.1.1 Režim otvorenej slučky

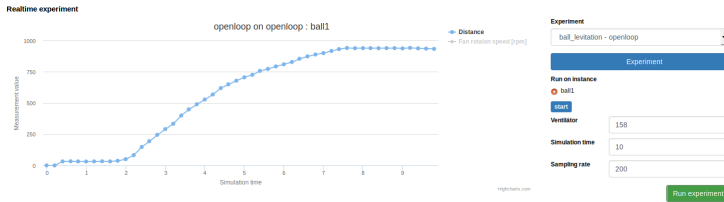
Ovládanie zariadenia bez spätnej väzby sa vykonáva len za pomoci jazyku Python. Skript na spustenie experimentu formou otvorenej slučky musí odosielať vstupné hodnoty do zariadenia, načítavať namerané dáta zo zariadenia, zapisovať výstupné údaje do súboru a nastavenú vzorkovaciu frekvenciu a dĺžku trvania experimentu.

Po odoslaní rýchlosti ventilátora sa tento údaj spracuje na riadiacej doske Arduino, ktorá odpovedá rýchlosťou točenia ventilátora a vzdialenosťou loptičky od senzora. Skript čaká, kým prejde doba rovnajúca sa dĺžke vzorkovacej periódy, načo odošle ďalšiu požiadavku a cyklus sa opakuje.

Zabezpečenie korektnej dĺžky trvania experimentu je riešené stanovením počtu dopytov, ktoré sa počas behu vykonávajú ako podiel celkovej dĺžky experimentu a vzorkovacej periódy. Po vykonaní posledného dopytu sa rýchlosť otáčok ventilátora nastaví na 0, a tým je experiment ukončený.

Získané dáta z experimentu ukazujú ako sa loptička začne vznášať a putuje až po hornú časť trubice (Obrázok č. 2).





Obr. 2: Spustenie experimentu v režime otvorenej slučky.

### 6.1.2 Integrácia simulačného prostredia

Aby bolo možné prostredie integrovať do systému, musí byť spustiteľné v operačnom systéme Linux. Skript, ktorý dané prostredie spustí, načíta potrebné parametre a súbor s blokovou schémou. Pre rôzne implementácie experimentov sa v súčasnosti používajú buď shellové skripty alebo skripty napísané v jazyku Python. Tento skript je volaný z webovej aplikácie na serveri experimentu.

Úlohou tohto skriptu je načítať názov portu, cestu k súboru s logovanými dátami a vstupné parametre experimentu. Skript buď spustí experiment priamo príkazom po sériovej linke formou otvorenej slučky alebo nepriamo cez simulačné prostredie.

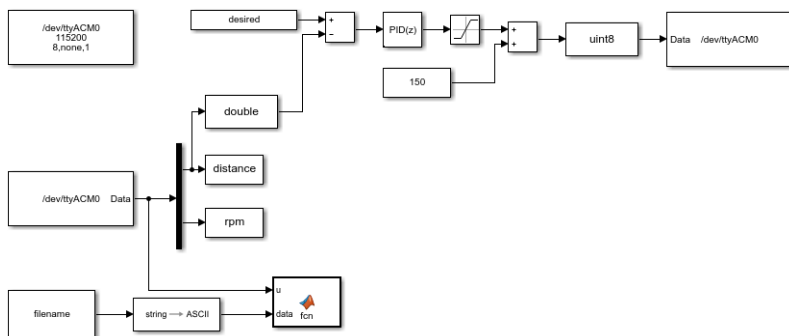
Spustenie simulačného prostredia je ďalším krokom po rozparsovaní vstupných argumentov. V prípade MATLABu sa nemusí vytvárať nová inštancia prostredia pri každom experimente, ale využíva sa tzv. shared engine, ktorý beží na pozadí. Toto riešenie značne znižuje čas odozvy serveru.

Nalinkovanie schémy a spustenie experimentu sa vykoná ako posledný krok skriptu. Skript dostane názov stiahnutého súboru schémy bez prípony, nakoľko generalizovaný centrálny webový server v tomto štádiu životného cyklu experimentu nerozlišuje medzi jednotlivými simulačnými prostrediami. Skript súbor premenuje a priradí mu korešpondujúcu príponu. Po spustení experimentu preberá kontrolu simulačné prostredie, ktoré v pravidelných intervaloch zapisuje výstupné hodnoty experimentu do definovaného súboru.

Spustenie experimentu „start“ je jedným zo štyroch príkazov, ktoré môžu byť implementované. Systém podporuje príkaz na ukončenie experimentu „stop“ a zmenu parametrov experimentu „change“. Špecifikom je nepovinný príkaz na inicializáciu „init“, ktorý obsahuje sekvenciu príkazov na prípravu zariadenia do stavu, kedy na ňom môže byť spustený samotný experiment. Každý z týchto príkazov si vy-

žaduje napísanie separátneho skriptu. Tieto skripty tvoria rozhranie medzi generalizovaným systémom online laboratória, konkrétnym zariadením vyžadujúcim špecifické ovládanie a softvérom simulačného prostredia.

Samotná bloková schéma (Obrázok č. 3) riadi komunikáciu so zariadením pripojeným k počítaču a ukladá výstupné hodnoty do definovaného súboru. Všetky zariadenia sú k systému pripojené prostredníctvom sériového portu (USB), ale neexistuje žiadna limitácia, ktorá by inú formu komunikácie zakazovala.



Obr. 3: Simulinková schéma na riadenia sústavy pomocou PID regulátora.

## 6.2 Prepojenie s centrálnym serverom

Po pridaní experimentu (Obrázok č. ??) je potrebné spustiť synchronizáciu. Tu sa centrálny server odvoláva na API rozhranie experimentu, ktoré odpovedá nastaveniami z konfiguračných súborov. Tie sa uložia do databázy na centrálnom webovom serveri. V tomto bode si už používatelia vedia vytvárať rezervácie zariadenia.

Vytvorením rezervácie a otvorením riadiaceho panelu sa dá spustiť experiment levitácie loptičky. Po spustení experimentu v móde otvorenej slučky sa vygeneruje graf reprezentujúci vzlet loptičky po vrch trubice. Na základe získanej dynamiky je možné identifikovať systém a navrhnuť riadiaci algoritmus.

Name: ball\_levitation

IP: 147.175.105.166

Port: 80

NodeJS port: 3000

Color:  Green

Deployed for:  Testing  Production

Update Server

Obr. 4: Formulár pridávania experimentu do centrálneho webového servera

### 6.2.1 Používateľské rozhranie

Systém je navrhnutý tak, aby nielen centrálny webový server, ale aj servery experimentov ponúkali používateľské rozhranie formou webovej aplikácie. Servery jednotlivých experimentov slúžia predovšetkým na ich konfiguráciu a testovanie.

**Webové rozhranie serveru experimentu** na rozdiel od centrálneho serveru neponúka možnosti rezervácie ani žiadnu používateľskú správu. Okrem sekcie pre konfiguráciu pripojených experimentov je možné daný experiment otestovať pomocou ovládacieho panelu, ktorým sa spúšťajú všetky implementované príkazy. Prístup je chránený používateľským menom a heslom. Táto aplikácia bola vytvorená primárne za účelom administrácie a rozširovania systému online laboratórií a nie ako prístupový bod pre typického používateľa.

Samotné rozhranie na spúšťanie experimentu vyzerá takmer totožne s tým na centrálnom webovom serveri, určenom pre všetkých používateľov. Najväčším rozdielom je panel zobrazujúci výstupnú chybovú hlášku pokiaľ nastane porucha v exekúcii kódu.

Okrem konfigurácie a testovania používateľské rozhranie ponúka aj históriu všetkých úspešne aj neúspešne vykonaných experimentov na zariadení, a to z dôvodu zjednodušenia procesu hľadania chýb.

Sekcia konfigurácie je najdôležitejšou funkcionalitou poskytnutou serverom experimentu. Tu používateľ zadá všetky informácie o pripojenom zariadení a priradí mu simulačné prostredia. Po vykonaní tohto nastavenia sa vygenerujú súbory obsahujúce vzorový kód, ktoré ale musí používateľ upraviť aby sa zariadenie stalo použiteľným.

**Webové rozhranie centrálneho webového servera** slúži primárne pre používateľov systému na spúšťanie experimentov. Klasický používateľ má po registrácii a prihlásení prístup k rezervačnému systému, verejnému fóru a četovacím miestnostiam, štatistikám a správe regulátorov. Používatelia s rolou administrátora môžu tiež spravovať používateľské kontá, regulačné schémy ale aj samotné experimenty.

## 6.2.2 Spustenie experimentu

Existujú dva spôsoby ako spustiť experiment. Jednou je vytvorenie rezervácie a druhou je tzv. dávkový režim. Po rezervácii konkrétneho zariadenia má používateľ istotu, že bude môcť výstupné namerané dáta experimentu vidieť v reálnom čase. Po spustení experimentu v dávkovom móde sa príkaz zaradí do fronty a spustí sa na prvom voľnom zariadení daného typu. Po zbehnutí experimentu je užívateľ notifikovaný systémom o sprístupnení novo nameraných dát v sekcii reportov.

## 6.2.3 Správa regulátorov

V rozhraní na správu regulátorov je užívateľom umožnené navrhnuť nový regulátor, ktorý neskôr môžu otestovať na zvolenom experimente. Regulátory sa dajú vytvárať pre viacero simulačných prostredí ako Matlab, Scilab, OpenModelica alebo pomocou programovacích jazykov ako Python a C.

Každý riadiaci algoritmus musí obsahovať svoju materskú schému. Táto schéma je špecifická pre každý systém a určuje, akým spôsobom má byť regulátor vytvorený. Na obrázku č. ?? je možné vidieť regulačnú schému aj s vyhradeným miestom pre algoritmus.

Niektoré materské schémy vyžadujú nahratie regulátora vo forme súboru, iné sú zadávané priamo do textového poľa. Existuje aj možnosť vytvoriť kompletnú riadiacu schému, ktorú možno spustiť bez akýchkoľvek iných vstupov. Ku vytvorenej schéme môže byť priložený aj obrázok, ktorý ju bližšie špecifikuje a pomôže študentom pri návrhu správneho algoritmu.

Každá schéma môže mať v systéme deklarované vstupné parametre. Tieto parametre si používateľ určí počas tvorby blokového diagramu v simulačnom prostredí ako je Simulink alebo Scilab. Proces deklarácie parametrov v systéme je totožný pre všetky podporované simulačné prostredia. V súbore schémy môžu tieto hodnoty zostať neinicializované, systém ich pri spúšťaní experimentu nastaví podľa

hodnôt zvolených užívateľom. Vďaka tomu je možné na jednom zariadení vytvoriť väčšie množstvo experimentov.

Vytvárať nové regulačné schémy majú možnosť len užívatelia s administrátorskými privilégiami. Administrátori tiež vedú vytvoriť verejný regulačný algoritmus. To znamená, že ho môžu otestovať všetci užívatelia systému. Tí, ktorým neboli pridelené práva administrátora, môžu požiadať o zverejnenie vlastných algoritmov, táto žiadosť však musí byť schválená administrátorom.

Dôvodom tohto dizajnu a miery zabezpečenia je ochrana zariadení pred škodlivým kódom. Vďaka materskej schéme, ktorú nemôže vytvárať hocikto, je isté, že sa na zariadení nevykoná proces, ktorý by ho mohol poškodiť. Zabráni sa tak zlomyseľným útokom, ale aj poruchám z nevedomosti učiacich sa študentov.

#### 6.2.4 Parametre experimentu

Deklarácia a definícia parametrov Pre dosiahnutie čo najvyššej miery generalizácie pri zachovaní voľnosti tvorby a integrácií nových zariadení bolo nutné definovať kategórie parametrov pre experiment. Tieto parametre charakterizujúce experimenty sa dajú rozdeliť do troch kategórií.

**Všeobecné parametre** predstavujú hodnoty, ktoré musia byť definované pre každý experiment bez ohľadu na typ zariadenia či simulačný softvér. Do tejto kategórie je možné zaradiť identifikujúce parametre, ktoré slúžia na určenie zariadenia alebo softvéru zvoleného používateľom. Jedno zariadenie môže byť kompatibilné s viacerými simulačnými prostrediami. Rovnako môže nastať situácia, kedy je do systému pripojených viac než jedno zariadenie rovnakého typu. Je teda nutné vedieť, na ktorom sa bude experimentovať a aké prostredie sa využíva.

Sú integrované priamo do kódu webovej aplikácie centrálného webového servera, nakoľko musia byť odosielané bez ohľadu na to, o aký experiment sa jedná.

**Parametre zariadenia** sú rovnaké pre všetky druhy experimentov spúšťané na jednom zariadení. Sú to napríklad poruchové veličiny. Sú uložené na serveri experimentu v konfiguračnom súbore, ktorý sa vygeneruje pri inštalácii

Tu sa okrem názvu parametra definuje aj jeho dátový typ, predvoľená hodnota, zobrazený názov poľa či povinnosť definovania hodnoty.

<input type="text" value="desired"/>	<input type="text" value="Desired value"/>	<input type="text" value="text"/>	<input type="text" value="400"/>	<input type="button" value="✘"/>
<input type="text" value="P"/>	<input type="text" value="P"/>	<input type="text" value="text"/>	<input type="text" value="0.03"/>	<input type="button" value="✘"/>
<input type="text" value="I"/>	<input type="text" value="I"/>	<input type="text" value="text"/>	<input type="text" value="0.002"/>	<input type="button" value="✘"/>
<input type="text" value="D"/>	<input type="text" value="D"/>	<input type="text" value="text"/>	<input type="text" value="0.03"/>	<input type="button" value="✘"/>
<input type="button" value="+ Add argument"/>				

Obr. 5: Deklarácia premenných schémy v prostredí online laboratória.

Tento súbor slúži aj na konfiguráciu parametrov, ktoré sa zadávajú pri spustení experimentu a tých, ktoré sa môžu meniť počas jeho behu.

**Parametre schémy** bližšie definujú už špecifický riadiaci algoritmus, ktorý sa spustí na danom zariadení. Môžu to byť akékoľvek argumenty deklarované v súbore simulačnej schémy nahranom na centrálnom webovom serveri. To sú napríklad premenné nutné pre spustenie experimentu ako sú konštanty PID regulátora (Obrázok č. 5).

Tieto premenné nie sú vždy len číselnými hodnotami. Aby používateľ nebol limitovaný snahou o dosiahnutie čím vyššej miery generalizácie, existuje aj možnosť deklarovať dátové typy premenných ako reťazce znakov alebo polia. Existujú tiež situácie, kedy takýto parameter môže nadobudnúť len hodnotu z preddefinovaného setu, ako je napríklad výber regulovanej veličiny.

Všetky tieto parametre sa ukladajú do databázy na centrálnom webovom serveri a sú priradené len k určitej schéme daného experimentu.

Dôvodom pre túto kategorizáciu je fakt, že používateľ by nemal cítiť rozdiel medzi prácou s jednotlivými zariadeniami či simulačnými prostrediami. Cieľom webovej aplikácie je prezentovať unifikované prostredie, ktoré sprostredkováva prístup k experimentom a nenúti používateľov oboznamovať sa pri každej aktivite s novými funkciami či spôsobmi ovládania.

### 6.3 Čítanie výsledkov

Po zahájení experimentu je webovým serverom spustený Node.js server, ktorý v pravidelných intervaloch číta obsah súboru, kam sú zapisované výstupy senzorov simulačným prostredím. Jeho obsah odosiela cez WebSocket [10] priamo do webového prehliadača prihláseného používateľa. Node.js server potrebuje poznať cestu k súboru a identifikátor užívateľa. Obe hodnoty sú mu poskytnuté prostredníctvom služby

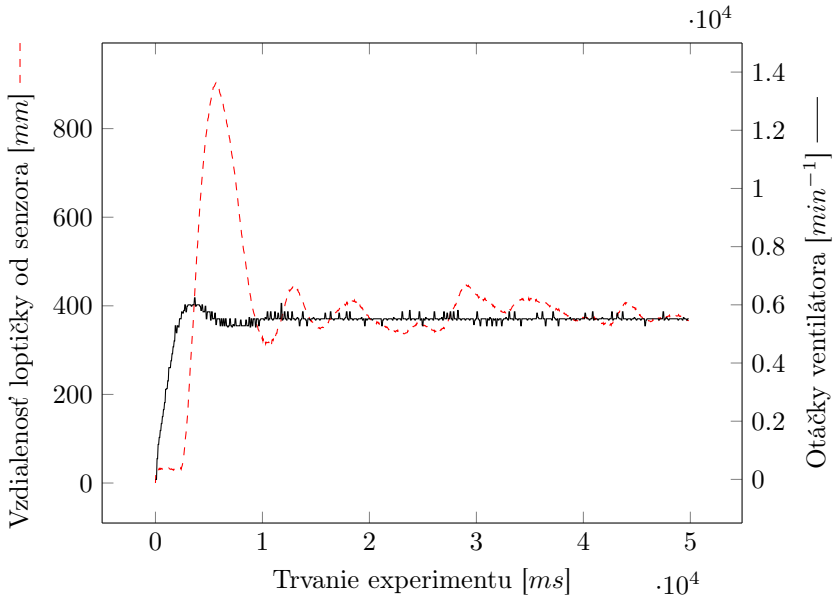
Redis. Dáta prijaté používateľom sa prezentujú formou grafu už počas behu experimentu.

Po úspešnom ukončení experimentu je celý obsah súboru odoslaný priamo na centrálny webový server, kde sa z týchto dát vygeneruje report, ktorý sa uloží do databázy. Používateľ si môže prezeráť všetky svoje predošlé experimenty a stiahnuť si namerané dáta vo forme csv súboru.

## 7 Potenciálne vylepšenie

V priebehu implementačnej fázy boli spozorované určité nedostatky systému online laboratória ako takého, ale aj novovytvoreného zariadenia. Tieto nedostatky však nebránia v používaní online prostredia.

Novovytvorené zariadenie nemá dostatočne presný ventilátor. Je možné, že má príliš vysoký výkon na to aby dokázal presne udržať loptičku v jednej polohe. Rozlíšenie jeho otáčok je primálne, čo má za následok osciláciu loptičky okolo požadovanej výšky. Čiastočne pomohlo zaťaženie loptičky.



Obr. 6: Priebeh experimentu so zaťaženou loptičkou.

V budúcnosti by bolo možné považovať o pripojení na tzv. „jednodoskový počítač“ alebo SBC. Tieto už teraz dosahujú výkonu aktuálne používaných staníc v laboratóriu, ale sú o poznanie kompaktnjšie, energeticky menej náročné a niektoré varianty na trhu v sebe majú zabudovaný Arduino chip. Hlavne pokiaľ sa toto all-in-one riešenie stane ešte cenovo dostupnejším, je možné očakávať ich výskyt aj v iných systémoch online laboratórií.

Celkový systém by mohol mať optimálnejší prenos dát po Web-Scoketoch k používateľovi. Ten totiž odosiela aj údaje prenesené v predošlých cykloch.

Súčasný stav zabezpečenia systému nie je ideálny, nakoľko každý experiment musí mať verejnú IP adresu, aby mohol byť pripojený k centrálnemu webovému serveru. Cieľom do budúcnosti je vytvoriť proxy server, ktorý by znemožnil prístup k experimentom zvonka.

Bolo by tiež vhodné vytvoriť testovaciu procedúru na overenie všetkých stavov a prechodov pri integrácii nových experimentov, aby nenastali situácie, kedy chyba v obsahu či formáte odosielaných parametrov spôsobí pád skriptu spúšťajúceho experiment. Chybové stavy by mali byť adekvátne ošetrené. Príkazy reprezentujúce prechod sústavy z jedného stavu do druhého musia byť vždy štandardizované, jednoducho implementovateľné a pokrývať celé spektrum akcií, ktoré môže zariadenie vykonávať.

## 8 Záver

Po vykonaní analýzy aktuálne vyvíjaných a používaných systémov na správu prístupu k vzdialeným experimentom boli zhodnotené ich vlastnosti. Vytvorený modulárny systém online laboratória obsahuje preferované prvky iných existujúcich riešení predovšetkým v oblasti generalizovaného prístupu k experimentom.

K systému je pripojených viacero rôznych mechatronických sústav a zariadení, na ktorých si študenti môžu prehľbovať svoje vedomosti z oblasti regulačných algoritmov či algoritmickej ako takej. Podpora viacerých simulačných prostredí dovoľuje študentom pracovať v ich preferovanom, ako aj vyskúšať a oboznámiť sa s novými.

Systém disponuje vysokou mierou škálovateľnosti vďaka zovšeobecnenému spôsobu pridávania experimentov a integrácii nových simulačných prostredí, modularite celej webovej aplikácie a využitím API rozhraní medzi centrálnym webovým serverom a servermi spravujúcimi jednotlivé experimenty. Používateľské rozhranie umožňuje



konfiguráciu experimentov, čo zabezpečuje jednoduchšiu a rýchlejšiu integráciu. Zvyšné nastavenia systému sa nachádzajú v prehľadných konfiguračných súboroch, ktorých obsah sa synchronizuje s databázou na centrálnom serveri.

Blokový diagram vytvorený v Simulinku demonštruje zovšeobecnený spôsob komunikácie cez sériovú linku bez nutnosti vytvárania špecializovaných blokov alebo s-funkcií. Všetky použité bloky diagramu je možné nájsť v základnej palete, ktorú softvér ponúka. Sústava môže byť v budúcnosti jednoducho prepojená s ďalšími simulačnými prostrediami a môže slúžiť ako pomôcka pri výučbe návrhu regulačných algoritmov.

V systéme bol kladený dôraz na možnosť vytvárania simulačných schém pre rôzne prostredia a ich spúšťanie na zariadeniach. Bola implementovaná funkcionálna prepojenia premenných medzi simulačnou schémou, systémom online laboratória a bežiacim experimentom.

Navrhnutý a skonštruovaný prototyp mechatronického zariadenia bol nakonfigurovaný tak, aby adekvátne reagoval na zmeny vstupných parametrov a hodnoty na výstupe ponúkal s dostatočnou frekvenciou a presnosťou. Bol zostavený z dostupných a nenákladných komponentov, ktoré sa dajú v prípade poruchy jednoducho nahradiť. Od ostatných experimentov podobného charakteru vyvinutých vo svete sa líši využitím laserového senzora vzdialenosti miesto infračerveného či ultrazvukového, ktoré dosahujú horšiu úroveň presnosti a nižšiu vzorkovaciu frekvenciu.

## 9 Prínosy dizertačnej práce

S rýchlym rozvojom technológií sa postupne mení aj vývoj v oblasti vzdialeného experimentovania. Postupne sa prechádza od vývoja samostatných aplikácií k vývoju komplexných systémov. V oblasti riadenia mechatronických systémov však pravdepodobne neexistuje systém, ktorý by umožňoval nielen správu užívateľov a experimentov, ale aj simulačných rozhraní a radiaciach algoritmov. Cieľom dizertačnej práce je návrh a optimalizácia modulárneho systému pre správu a riadenie vzdialených mechatronických experimentov.

## Literatúra

- [1] ALVES, G. R., GERICOTA, M. G., SILVA, J. B., AND ALVES, J. B. Large and small scale networks of remote labs: a survey.

*Advances on remote laboratories and e-learning experiences 15* (2007).

- [2] BERISCH, G., DONATH, H., HEINRICH, F., HUEBENER, I., LIPKE, T., MENDE, T., SCHRIEFER, M., SCHULTE, H., AND ZAJAC, M. Design and development of a low cost rapid control prototyping system applied to an air suspension system. *IFAC Proceedings Volumes 45*, 11 (2012), 194–199.
- [3] CHACON, J., SAENZ, J., TORRE, L., DIAZ, J., AND ESQUEMBRE, F. Design of a low-cost air levitation system for teaching control engineering. *Sensors 17*, 10 (2017), 2321.
- [4] CHACÓN, J., VARGAS, H., FARIAS, G., SÁNCHEZ, J., AND DORMIDO, S. Ejs, jil server, and labview: An architecture for rapid development of remote labs. *IEEE Transactions on Learning Technologies 8*, 4 (2015), 393–401.
- [5] CHOŁODOWICZ, E., AND ORŁOWSKI, P. Low-cost air levitation laboratory stand using matlab/simulink and arduino. *Pomiary Automatyka Robotyka 21* (2017).
- [6] CYTRON TECHNOLOGIES SDN. BHD. *Product User’s Manual – HCSR04 Ultrasonic Sensor*, 5 2013. V1.0.
- [7] DHUEY, M. J. Quiet fan speed control, Aug. 2 2005. US Patent 6,924,568.
- [8] FETTE, I. The websocket protocol.
- [9] GARCÍA-ZUBIA, J., ANGULO, I., HERNÁNDEZ, U., AND ORDUÑA, P. Plug&play remote lab for microcontrollers: Weblab-deusto-pic. In *7th European Workshop on Microelectronics Education May* (2008), pp. 28–30.
- [10] GARCIA-ZUBIA, J., DEÍPINA, D. L., AND ORDUÑA, P. Accessing weblabs from cellular phones. In *IECON 2006-32nd Annual Conference on IEEE Industrial Electronics* (2006), IEEE, pp. 3779–3781.
- [11] GARCÍA-ZUBIA, J., LÓPEZ-DE IPIÑA, D., ORDUÑA, P., HERNÁNDEZ, U., AND TRUEBA, I. Evolution of the weblab at the university of deusto. *EWME 2006* (2006).

- [12] HENKE, K., VIETZKE, T., WUTTKE, H.-D., AND OSTENDORFF, S. Goldi-grid of online lab devices ilmenau. *International Journal of Online Engineering* 12, 4 (2016).
- [13] HSIEH, H.-M. Pwm control circuit for a dc brushless fan, Aug. 24 1999. US Patent 5,942,866.
- [14] MENDES, L. A., LI, L., BAILEY, P. H., DELONG, K. R., AND DEL ALAMO, J. A. Experiment lab server architecture: A web services approach to supporting interactive labview-based remote experiments under mit's ilab shared architecture. In *2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV)* (2016), IEEE, pp. 293–305.
- [15] NAYYAR, A., AND PURI, V. A review of arduino board's, lily-pad's & arduino shields. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)* (2016), IEEE, pp. 1485–1492.
- [16] SHARP. *GP2Y0A21YK0F*, 5 2005.
- [17] TABLE, A. Ascii table, 2010.
- [18] THE MATHWORKS, I. Serial receive. <https://www.mathworks.com/help/instrument/serialreceive.html>, 7 2019.
- [19] ULBRICHT, R. 16-bit pwm resolution for arduino. <https://www.arduinosllovakia.eu/blog/2017/7/16-bitove-rozlisenie-pwm-pre-arduino?lang=en>, 7 2019.

## Publikácie autora

- RÁBEK, Matej - ŽÁKOVÁ, Katarína. Online laboratory manager for remote experiments in control. In IFAC-PapersOnLine. Vol. 50, 20th World congress on the international federation of automatic control. Toulouse, France. July 9-14, 2017 (2017), [6] s. ISSN 2405-8963. V databáze: WOS: 000423965200239. (A kategória)
- RÁBEK, Matej - ŽÁKOVÁ, Katarína. Communication in an online laboratory. In Distance learning, simulation and communication 2017 [elektronický zdroj] : International conference. Brno, Czech Republic. May 31-June 2, 2017. Brno : University of Defence, 2017, CD-ROM, S. 248-253. ISBN 978-80-7231-416-4. (B kategória)
- RÁBEK, Matej - ŽÁKOVÁ, Katarína. Simple experiment integration into modular online laboratory environment. In exp.at'17 : 4th Experiment@ international conference. Faro, Portugal. June 6-8, 2017. Danvers : IEEE, 2017, S. 264-268. ISBN 978-1-5386-0810-4. V databáze: IEEE ; WOS: 000412842600070 ; SCOPUS: 2-s2.0-85027875867. (B kategória)
- RÁBEK, Matej - ŽÁKOVÁ, Katarína. Management of control algorithms for remote experiments. In Smart industry & Smart education : 15th International conference on remote engineering and virtual instrumentation (REV 2018). Düsseldorf, Germany. March 21-23, 2018. Cham : Springer, 2019, S. 283-289. ISBN 978-3-319-95677-0. (B kategória)
- ŽÁKOVÁ, Katarína - RÁBEK, Matej. Remote control of thermo-opto-mechanical plant via Raspberry Pi. In IFAC-PapersOnLine. Vol. 51, Iss. 6 : 15th IFAC Conference on Programmable Devices and Embedded Systems (PDeS 2018). Ostrava, Czech Republic. May 23-25, 2018, s. 479-483. ISSN 2405-8963 (2017: 0.260 - SJR, Q3 - SJR Best Q). V databáze: SCOPUS: 2-s2.0-85052901096 ; WOS: 000445644900081. (B kategória)
- RÁBEK, Matej - ŽÁKOVÁ, Katarína. Ball levitation using EEG headset via bluetooth. In ICETA 2018 : 16th International Conference on Emerging eLearning Technologies and Applications. Starý Smokovec, Slovensko. November 15-16, 2018. Danvers : IEEE, 2018, S. 457-462. ISBN 978-1-5386-7912-8. V databáze:

IEEE: 8572045 ; WOS: 000457680400069 ; SCOPUS: 2-s2.0-85060316869. (B kategória)

- HUBA MIKULÁŠ - BISTÁK PAVOL - RÁBEK, Matej - ŽÁKOVÁ, Katarína. Introductory experiments for a stable plant control. In exp.at'19 : 5th Experiment@ international conference. Funchal, Portugal. June 11-14, 2019. Ešte nezaregistrované na FEI STU. (B kategória)
- RÁBEK, Matej - ŽÁKOVÁ, Katarína. Integration of new control experiments to online environment. In exp.at'19 : 5th Experiment@ international conference. Funchal, Portugal. June 11-14, 2019. Ešte nezaregistrované na FEI STU. (B kategória)
- RÁBEK, Matej - ŽÁKOVÁ, Katarína. Remote control using EEG headset via Bluetooth. In ELITECH'18 [elektronický zdroj] : 20th Conference of doctoral students. Bratislava, Slovakia. May 23, 2018. 1. ed. Bratislava : Vydavateľstvo Spektrum STU, 2018, CD-ROM, [5] p. ISBN 978-80-227-4794-3.
- MATIŠÁK, JAKUB - RÁBEK, Matej - ŽÁKOVÁ, Katarína. Use of Holographic Technology in Online Experimentation. Prijatý článok na 6th Doctoral Symposium on Recent Advances in Information Technology (DS-RAIT&#39;19), Leipzig, Germany, 1 - 4 September, 2019

## Riešené projekty

- eTalent nadácie Tatrabanky - Využitie EEG na bezdrôtovú komunikáciu a riadenie, koordinátor: Ing. Matej Rábek
- projekt mladého výskumníka - Modulárny systém pre zdieľanie vzdialene riadených experimentov, koordinátor: Ing. Matej Rábek
- KEGA - Internet vecí pre mechatronické systémy, 2017-2019, koordinátor: doc. Ing. Katarína Žáková, PhD.
- APVV - Počítačová podpora návrhu robustných nelineárnych regulátorov, 2013-2017, koordinátor: prof. Ing. Mikuláš Huba, PhD.

- APVV - Bilaterálna spolupráca - Platoon modeling and control for mixed autonomous and conventional vehicles: a laboratory experimental analysis, 2018-2020, koordinátor: prof. Ing. Mikuláš Huba, PhD.
- VEGA - Riadenie a modelovanie mechatronických systémov v emobilite, 2019-2022, koordinátor: prof. Ing. Mikuláš Huba, PhD.