

A SZTAKI Rendszer és Irányításelméleti Kutatólaboratórium

NYÁRI SZAKMAI GYAKORNOKI PROGRAM

Munkavégzés helye: SZTAKI Rendszer és Irányításelméleti Kutatólaboratórium részlege

Jelentkezés határideje: 2023.07.31.

A SZTAKI-ról röviden:

A Számítástechnikai és Automatizálási Kutatóintézet (SZTAKI) az ország legnagyobb és legsikeresebb informatikai kutatóintézete. A SZTAKI az informatika, az információtechnológia, számítástudomány és rokonterületei tudományának szakmai műhelye.

A SZTAKI gyakornoki programja:

A SZTAKI kiemelt hangsúlyt fektet a jövő kutatóinak, fejlesztő mérnökeinek kinevelésére, változatos projektmunkákat, mentorokat, eszközöket kínálunk megegyezés szerinti rugalmas munkarendben. Elkötelezettek vagyunk az iránt, hogy a leendő szakemberek minél magasabb minőségű és a piaci igényekhez jobban igazodó képzést kapjanak. Több hazai egyetemmel állunk szoros kapcsolatban, hogy ismereteinket, kutatás-fejlesztési tapasztalatainkat átadjuk a hallgatóinknak. A Gyakornoki Program során a hallgatók fejleszthetik a felhívásban szereplő téma elsajátítását és a gyakorlatban való megismerését.

A fogadó részleg: Rendszer és Irányításelméleti Kutatólabor

A szakmai gyakorlat témája: csatolva a 3 témakiírás

Munkavégzés helye: SZTAKI, Kende/Lágymányosi utcai épület, Budapest, XI. kerület ,

Jelentkezés módja: Önéletrajzzal elektronikus úton: **Péni Tamás és Tóth Roland** gyakorlati témavezetők részére a peni@sztaki.hu és tothroland@sztaki.hu valamint „cc”-ben a hr@sztaki.hu e-mail címen.

1.

Kvadkopter dinamikus modelljének meghatározása gépi tanulással

Autonóm kvadkopterekkel szemben egyre inkább elvárás a nagy sebességgel történő manőverezés, a változó működési körülményekhez való gyors alkalmazkodás valamint komplex manőverek gyors és pontos végrehajtása. Ezeknek a feladatoknak a megoldása nemlineáris szabályozási módszerekkel lehetséges, amelyek megtervezéséhez szükség van a drón pontos dinamikus modelljére. Jelen nyári gyakorlat feladata egy kvadkopter nemlineáris dinamikus modelljének előállítás repülés közben gyűjtött mérési adatok felhasználásával. A feladat első lépése a nominális modell meghatározása. Ehhez a fizikai törvények alapján felírt mozgásegyenletekből indulunk ki és a cél a modellben lévő fizikai paraméterek meghatározása. A következő lépésben a nominális modellt tanuló komponenssel (neurális háló) egészítjük ki annak érdekében, hogy a modell tovább pontosítható és az esetleges, működés közben bekövetkező változásokhoz és külső zavarásokhoz adaptálható legyen.

A hallgató feladata: a nominális modell meghatározásához kapcsolódó paraméterbecslési feladat megfogalmazása és megoldása, majd a modellaugmentációhoz használt neurális háló megválasztása, az identifikációhoz szükséges mérési adatok gyűjtése, és a háló tanítása. Az identifikációt elsőként szimulált kvadkopterre kell elvégezni, majd az eljárást valós drónról gyűjtött adatokra is alkalmazni kell. Ehhez rendelkezésre áll a labor által fejlesztett szimulációs környezet, amelyben többféle kvadkopter digitális ikermodellje elérhető. A valós tesztekhez adott áll az AIMotionLab teszt aréna, melyben miniatűr Crazyflie és nagyobb méretű, egyedi tervezésű drónokkal lehet repülési tesztek végezni. Mindkét környezethez magasszintű szoftvertámogatás tartozik, amely lehetővé teszi, hogy a hallgató kizárólag a feladatához szorosan kapcsolódó funkciók fejlesztésével foglalkozzon, az egyéb, a kísérletekhez szükséges kiegészítő komponenseket felhasználásra készen megkapja.

Szükséges előismeretek: lineáris algebra és analízis ismeretek; rendszerelmélet és numerikus optimalizálás alapok; programozói képesség; **Matlab és Python** programozási nyelvek ismerete

Learning quadcopter dynamics from data

Autonomous quadcopters are increasingly expected to perform maneuvers at high speed, quickly adapt to changing operating conditions and perform complex maneuvers quickly and precisely. These challenging tasks require nonlinear control methods, for which an accurate dynamical model with precise knowledge of the physical parameters of the drone is necessary. The task of this internship project is to create a nonlinear dynamic model of a quadcopter using measurement data collected during flight. The first step is to identify a nominal model. This model is based on the motion equations derived from first principles and the goal is to determine the physical parameters the

model depend on. In the next step, the nominal model is augmented with a learning component (neural network) to make it adaptable to changes and external disturbances during operation. The student's task is to formulate and solve the parameter estimation task related to the identification of the nominal model, select the neural network used for model augmentation, collect the measurement data necessary for identification, and train the network. The identification should be performed first on a simulated quadcopter, then the procedure should be applied on real measurement data collected during real flight experiments. The laboratory has developed a simulation environment that provides digital twin models for different type quadcopters, and also we have the AIMotionLab test arena created for real, indoor flight tests with miniaturized Crazyflie and larger custom-designed drones. Both environments have high-level software support that enables the student to focus solely on developing functions closely related to his/her task while all auxiliary components needed to perform the experiments are given ready to use.

Requirements: solid background in linear algebra and mathematical analysis; basic knowledge in system theory and numerical optimization; programming skills; experience in Matlab and Python.

2.

Modell prediktív pályakövető szabályozó tervezése autonóm földi járművek számára

Autonóm járművekkel szemben egyre inkább elvárás a nagy sebességgel történő manőverezés, a változó működési körülményekhez való gyors alkalmazkodás, valamint komplex manőverek gyors és pontos végrehajtása. Ezeknek a feladatoknak a megoldása szükségessé teszi, hogy a járművet a fizikai képességeinek határán működtessük, az irányítójelek maximális tartományát kihasználjuk. Az irányítástervezést nehezíti, hogy a jármű modellje nemlineáris, esetenként a változó körülményekhez való folyamatos adaptálódás miatt valamilyen tanuló komponenssel (pl. neurális hálóval, Gauss folyamattal) kiegészített. Erre a komplex modellre és feladatra minden korlátozást betartó, optimális irányítási stratégiát leginkább modell prediktív megközelítéssel lehet tervezni. Ennek lényege, hogy minden időpillanatban, a rendszer nemlineáris modelljét felhasználva egy véges jövőbeli intervallumon prediktáljuk a rendszer viselkedését és ennek alapján, egy numerikus optimalizálási feladat megoldásával előállítjuk az irányításhoz szükséges beavatkozójelet. A szakmai gyakorlat célja a trajektóriakövetésre kidolgozott MPCC (Model Predictive Contouring Control) irányítási algoritmus adaptálása és implementációja a laborban működő, 1:10 méretarányú F1TENTH autóra.

A hallgató feladata: a jármű dinamikus modelljének megismerése, az MPCC algoritmus megértése, majd ezt követően az eljárás adaptálása a konkrét járműre. A feladatot elsőként szimulációs környezetben kell megoldani, ehhez rendelkezésre áll az F1TENTH digitális ikermodellje a labor által fejlesztett, Python nyelven programozható szimulátor programban. Ha az eljárás már működik, akkor a valós rendszerre is adaptálni kell. Ehhez adott a valós F1TENTH autó, a kontroller futtatására előkészített fedélzeti szoftverkörnyezettel. A valós implementáció fő kihívása, hogy az beavatkozójel számítást megvalósító optimalizálási feladat numerikus megoldásának bele kell férnie a 25-50 ms mintavételi időbe. Ez az implementációs kód optimalizálását és hatékony szolverprogramok (Casadi, ACADO Toolkit, stb.) integrációját igényli.

Szükséges előismeretek: lineáris algebra és analízis ismeretek; rendszerelmélet és numerikus optimalizálás alapok; programozói képesség; **Python** programozási nyelv ismerete.

Model predictive trajectory tracking control for autonomous cars

There is an increasing demand for autonomous vehicles to maneuver at high speeds, quickly adapt to changing operating conditions, and execute complex maneuvers quickly and accurately. To solve these tasks, the vehicle must be operated at the limits of its physical capabilities, utilizing the maximum range of control signals. The nonlinear vehicle model, which may require a learning component such as a neural network or Gaussian process due to continuous adaptation to changing circumstances, complicates

control design. The optimal control strategy that adheres to all constraints can be designed most effectively by using a model predictive approach. The main concept of this approach is to use the nonlinear model of the system at every sampling time instant to predict the system's behavior over a finite future time interval and produce the required control signal through numerical optimization. The objective of this internship project is to adapt and implement the Model Predictive Contouring Control (MPCC) algorithm developed specifically for trajectory tracking on the 1:10 scale F1TENTH car available in the laboratory. The student applying for this topic will be responsible for understanding the vehicle's dynamic model, becoming familiar with the MPCC algorithm, and adapting the procedure to the specific vehicle. The task must first be solved in a simulation environment, utilizing the lab's Python-programmable simulator program that includes the F1TENTH digital twin model. Once the procedure is working, it must be adapted to the real-world system. The real F1TENTH car is provided, along with on-board sensors and software prepared to run the controller. The primary challenge of the real implementation is that the optimization problem calculating the control signal must fit in the 25-50 ms sampling time period. This requires optimizing the implementation code and integrating efficient solver programs such as Casadi, ACADO Toolkit, etc.

Requirements: solid background in linear algebra and mathematical analysis; basic knowledge in system theory and numerical optimization; programming skills; experience in Matlab and Python.

3.

Navigáció dinamikus környezetben

Az autonóm drónoknak és földi járműveknek számos olyan alkalmazási területe létezik, ahol a járműnek mozgó és álló akadályokkal nehezített környezetben kell nagy sebességgel manőverezni. Néhány példa: csomagkiszállítás városi környezetben, gyártástámogató robot, felderítés épületen belül, katonai célú alkalmazások stb. Jelen szakmai gyakorlat célja hatékony (kis számítási idejű) pályatervezési eljárások kidolgozása autonóm kvadkopter számára akadályok közötti gyors navigációra. A tervezésnek alkalmasnak kell lennie nagy számú álló és előre nem pontosan ismert trajektóriájú mozgó objektumok elkerülésére. Utóbbi esetben az objektummal való ütközés kockázatát (risk) minimalizáló eljárást kell tervezni. A feladatot egyszerűsített lineáris és a teljes nemlineáris kvadkopter modell esetén is meg kell oldani, a problémát olyan (kevert egészértékű) optimalizálási problémaként kell megfogalmazni, amelyet a rendelkezésre álló megoldóprogramokkal (Gurobi, BARON, Ipopt, stb.) hatékonyan lehet kezelni. A tisztán optimalizálás alapú pályatervezés alapját képezi a későbbi gépi tanulás alapú megoldásoknak, amelyek célja a tervezési idő további csökkentése, az eljárás kiterjesztése nagy komplexitású navigációs problémák (nagy számú mozgó objektum, bonyolult navigációs tér, gyorsan változó környezet) esetére. A feladatot Python programozási nyelven kell megoldani és elsőként szimulációs környezetben kell működnie. Ehhez rendelkezésre áll a labor által fejlesztett szimulátor, amely több kvadkopter digitális ikermodelljét tartalmazza. Ha az eljárás működik, implementálni kell valós környezetben is. Ehhez adott az AIMotionLab autonóm drón tesztaréna, ahol miniatűr Crazyflie drónokkal lehet kísérleteket végezni. A rendszer minden olyan HW és SW komponenst tartalmaz, amellyel a drónok adott pálya pontos berepülésére képesek, így a hallgató feladata kizárólag a pályatervező eljárás megvalósítása.

Szükséges előismeretek: lineáris algebra és analízis ismeretek; rendszerelmélet és numerikus optimalizálás alapok; programozói képesség; **Python** programozási nyelv ismerete

Navigation in a dynamic environment

There are numerous application areas for autonomous drones and ground vehicles where the vehicle needs to maneuver at high speeds in an environment among moving and stationary obstacles. Some examples include package delivery in urban environments, production support robots, indoor building reconnaissance, military applications, etc. The goal of this professional internship is to develop efficient (low computation time) trajectory planning procedures for autonomous quadcopters for fast navigation in dynamic, cluttered environments. The design should be capable of

avoiding a large number of stationary and unpredictably-moving objects. In the latter case, a procedure that minimizes the risk of collision with the object must be developed. The task must be solved for both simplified linear and complete non-linear quadcopter models, and the problem must be formulated as a mixed integer optimization problem which can be effectively solved using available solver programs (Gurobi, BARON, Ipopt, etc.). The trajectory planning based purely on optimization forms the basis for later machine learning-based solutions, which aim to further reduce the computation time and expand the approach to complex navigation problems (large numbers of moving objects, complex navigation areas, rapidly changing environments). The task must be solved using the Python programming language and must run first in a simulation environment. The simulator developed by the laboratory, which contains multiple digital twin models of quadcopters, is available for this purpose. Once the procedure is functioning, it must be implemented in a real environment. The AIMotionLab autonomous drone test arena is provided for this purpose, where experiments can be carried out with miniature Crazyflie drones. The system includes all hardware and software components required for accurate path planning of the drones, and the student's task is solely to implement the trajectory planning procedure.

Requirements: solid background in linear algebra and analysis, fundamentals of system theory and numerical optimization, programming skills, and familiarity with the Python programming language.